A Hierarchical Framework for Physical Human-Robot Interaction

Ein hierarchisches Framework für physikalische Mensch-Roboter-Interaktion

Von der Fakultät für Maschinenbau
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation

von

M.Sc. Juan David Muñoz Osorio

2022

1. Referent: Prof. Dr.-Ing. Tobias Ortmaier
2. Referent: Prof. Dr.-Ing. habil. Eduard Reithmeier

Tag der Promotion: 12. Juli 2022

Vorwort

Die vorliegende Arbeit ist das Ergebnis von circa vier Jahren Forschung bei KUKA Deutschland GmbH in Kooperation mit dem Institut für Mechatronische Systeme der Gottfried Wilheim Leibniz Universität Hannover.

Zuallererst möchte ich meinen herzlichen Dank meinem Doktorvater, Herrn Professor Tobias Ortmaier, aussprechen, der mich in dieser Zeit sehr kompetent, sowohl fachlich als auch persönlich in der Fertigstellung der Arbeit betreute. Zudem freue ich mich sehr über die Anfertigung des Zweitgutachtens von Herrn Professor Eduard Reithmeier und die Übernahme des Prüfungsvorsitzes durch Herrn Professor Roland Scharf.

Mein besonderer Dank gilt meinem Betreuer bei Kuka, Dr. Felix Allmendinger. Sein Interesse an der Thematik und seine große Erfahrung haben es erst ermöglicht, die für diese Arbeit gesetzten Ziele zu erreichen. Für seine intensive persönliche und fachliche Unterstützung als Forscher und Entwickler bin ich sehr dankbar. Außerdem freue ich mich über die tolle kooperative Zusammenarbeit mit all meinen Kollegen aus der Konzernforschung. Dabei möchte ich vor allem Mario Fiore und Johannes Lachner für die vielen fachlichen Diskussionen danken. Mario, grazie per la collaborazione, le discussioni dettagliate e la revisione delle mie pubblicazioni. Bei dem Abteilungsleiter Dr. Rainer Bischoff bedanke ich mich für sein Vertrauen in meine Fähigkeiten und die Möglichkeit eine spannende Arbeit bei einer weltweit führenden Robotikfirma durchführen zu können.

I am very grateful to the students whom I was honored to supervise during this time: Felipe Castañeda, Manuel Martinez, Pavan Kumar, Michael Winter and Abdelrahman Abdelazim. Our discussions and learning sharing contribute to what this dissertation is today.

Finalmente me gustaría agradecer a mi familia y amigos, quienes siempre me han apoyado incondicionalmente durante mi carrera. Mis padres, Nancy Osorio y Mauricio Muñoz, me brindaron la educación y amor incondicional para alcanzar todos mis logros. Mis hermanos, Daniel, Natalia y Héctor, de quienes aprendí los valores y virtudes que hacen parte de lo que soy hoy como persona. Mis grandes amigos de la universidad y el colegio. Especialmente agradezco a Fernando Moya, Dr. Erwin López, Catalina Raymond y Felipe Arévalo por su amistad y por tomar parte de su tiempo en corregir este trabajo. Sincères remerciements à Pierre Venet pour le temps passé à la correction de ma thèse et pour son amitié.

Hannover, February 2022 *Juan David Muñoz*

Kurzfassung

Heutzutage sind Roboter mehr als Maschinen, die nur die stets gleichen Bewegungen hinter Sicherheitszäunen ausführen. Moderne Roboter müssen mehrere komplexe Aufgaben erfüllen können, während sie mit dem Menschen zusammenarbeiten. Hierzu besitzen moderne Roboter häufig zwei Haupteigenschaften: eine erhöhte Vielseitigkeit durch eine große Anzahl von Gelenken und die Fähigkeit, den Umgebungskontakt durch Drehmoment/Kraftsensoren wahrzunehmen.

Solche komplexen Roboter erfordern die Entwicklung von ausgefeilten Regelungsarchitekturen. In den letzten Jahren wurden viele Architekturen vorgeschlagen, die auf die durch die große Anzahl von Gelenken verursachte Redundanzproblematik zielen. Diese Architekturen (auch „Framework“ genannt) lösen das Redundanzproblem, indem sie verschiedene Aufgaben in einer hierarchischen, also priorisierten, Weise verarbeiten. Eine solche Priorisierung ist allerdings in Anwendungen problematisch, in denen der Roboter mehrere Einschränkungen gleichzeitig unbedingt einhalten muss, z.B. aus Sicherheitsgründen. Ein Sonderfall sind einseitige Einschränkungen wie z.B. Gelenkgrenzen oder Begrenzungen des Arbeitsraums. Um diese zu implementieren, beruhen aufgabenhierarchische Methoden auf der Aktivierung abstoßender Potentialfelder. Nachteilig ist dabei jedoch, dass diese je nach Roboterkonfiguration und Bewegungsgeschwindigkeit zu Schwingungen oder hohen Kräften des Roboters führen können.

Diese Arbeit schlägt ein Framework vor, um verschiedene Aufgaben unter gleichzeitiger Einhaltung mehreren Einschränkungen umzusetzen. Ziel dieses Frameworks ist es, redundante Roboter in kollaborativen Szenarien zu steuern. Ein besonderes Augenmerk liegt darauf, dass der Roboter auf physische Interaktion mit dem Menschen stets intuitiv und sicher reagiert: so darf der Roboter auf den Menschen keine hohen Kräfte ausüben oder mit unerwarteten Bewegungen auf äußere Kräfte reagieren. Das Framework verwendet neuartige Methoden, um Positions-, Geschwindigkeits- und Beschleunigungsgrenzen im Gelenkraum sowie im kartesischen Raum einzuhalten. Eine vergleichende Studie von weit verbreiteten Algorithmen zur Redundanzauflösung, die auf Projektoren und quadratischer Programmierung beruhen, wird durchgeführt, um die gewünschte intuitive Reaktion auf externe Kräfte herzuleiten. Die Experimente wurden an einem redundanten kollaborativen Industrieroboter durchgeführt. Die Ergebnisse zeigen, dass die vorgeschlagene Methode zur Grenzvermeidung eine stets sichere und erwartbare Reaktionen hervorruft, wenn der Mensch externe Kräfte ausübt. Dabei überschreitet der Roboter die festgelegten Grenzen nicht, während die vorgegebenen Aufgaben hierarchisch abgearbeitet werden.

Schlagwörter: Multitask-Steuerung, physische Mensch-Roboter-Interaktion, Redundanzauflösung, Grenzvermeidung.

Abstract

Robots are becoming more than machines performing repetitive tasks behind safety fences, and are expected to perform multiple complex tasks and work together with a human. For that purpose, modern robots are commonly built with two main characteristics: a large number of joints to increase their versatility and the capability to feel the environment through torque/force sensors.

Controlling such complex robots requires the development of sophisticated frameworks capable of handling multiple tasks. Various frameworks have been proposed in the last years to deal with the redundancy caused by a large number of joints. Those hierarchical frameworks prioritize the achievement of the main task with the whole robot capability, while secondary tasks are performed as well as the remaining mobility allows it. This methodology presents considerable drawbacks in applications requiring that the robot respects constraints imposed by, e.g., safety restrictions or physical limitations. One particular case is unilateral constraints imposed by, e.g., joint or workspace limits. To implement them, task hierarchical frameworks rely on the activation of repulsive potential fields when approaching the limit. The performance of the potential field depends on the configuration and speed of the robot. Additionally, speed limitation is commonly required in collaborative scenarios, but it has been insufficiently treated for torque-controlled robots.

With the aim of controlling redundant robots in collaborative scenarios, this thesis proposes a framework that handles multiple tasks under multiple constraints. The robot's reaction to physical interaction must be intuitive and safe for humans: The robot must not impose high forces on the human or react unexpectedly to external forces. The proposed framework uses novel methods to avoid exceeding position, velocity and acceleration limits in joint and Cartesian spaces.

A comparative study is carried out between different redundancy resolution solvers to contrast the diverse approaches used to solve the redundancy problem. Widely used projector-based and quadratic programming-based hierarchical solvers were experimentally analyzed when reacting to external forces. Experiments were performed using an industrial redundant collaborative robot. Results demonstrate that the proposed method to handle unilateral constraints produces a safe and expected reaction in the presence of external forces exerted by humans. The robot does not exceed the given limits, while the tasks performed are prioritized hierarchically.

Keywords: multi-task control, physical human-robot interaction, redundancy resolution, unilateral constraints.

Contents

Nomenclature

Characters and symbols that are only used in one section or in one equation are described exclusively in the text. Many of the variables in the following list appear with different subscripts, superscripts, additional symbols, and various dimensions in this thesis. The quantities are generally described here without further specification. The meaning becomes clear when the quantity or variable is introduced in the text.

General Conventions

Scalar	lowercase letter (italic): a
Vector	lowercase letter (bold and italic): $\boldsymbol{a}$
Matrix	capital letter (bold and italic): $\boldsymbol{A}$

Latin Characters

$\boldsymbol{a}$	Acceleration vector
$\boldsymbol{c}$	Vector of constraint coordinates
$\boldsymbol{e}$	Error vector
$\boldsymbol{f}$	Vector of (generalized) forces in the operational space
$\boldsymbol{g}$	Vector of (generalized) gravity forces in the joint space
k	Number of tasks
m_{t}	Dimension of task
n	Number of joints
$\boldsymbol{q}$	Vector of joint positions
$\boldsymbol{v}$	Velocity vector
$\boldsymbol{x}$	Vector of Cartesian coordinates
$\boldsymbol{D}$	Damping matrix
E	Energy
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{K}$	Stiffness matrix
$\boldsymbol{M}$	Inertia matrix
$\boldsymbol{N}$	Null space projection matrix

$\boldsymbol{U}, \boldsymbol{V}$	Matrices with singular directional components in a singular value decomposition

Greek Characters

σ	Singular values
η	Constant parameter
$\boldsymbol{\mu}$	Vector of Corioli forces in operational space
$\boldsymbol{\tau}$	Vector of (generalized) forces in joint space
$\boldsymbol{\nu}$	Vector of Coriolis torques
ζ	Damping ratio
$\boldsymbol{\Sigma}$	Matrix with singular values in a singular value decomposition

Abbreviations

aug	Augmented
cmd	Command
cur	Current
d	Desired
damp	Damping
e.g.	*Exempli gratia* (for example)
ext	External
i.e.	*id est* (that is)
inf	Infinity
int	Interaction
in	Initial
max	Maximum
min	Minimum
ns	No singular
suc	Successive
ref	Reference
tr	Trajectory
DOF	Degree(s) of freedom
DCTS	Dynamically-consistent Constrained Task Hierarchy Solver
pHRI	Physical Human-Robot Interaction
IWIS	Impedance without inertia shaping
OSC	Operational Space Control
SJS	Saturation in Joint Space
TIMC	Time Invariant Motion Controller

1 Introduction

The evolution of robots from pre-programmed to autonomous systems leads to the development of more complex hardware and software architectures. Modern robotics aims to achieve two significant goals: high versatility and high interactivity, which helps to reduce the gap between humans and robots.

The versatility of the human body comes in great part from the large number of joints. Figure 1.1 shows an example of a service task: polishing a car. The execution of the task requires at least five degrees of freedom (DOF). The trajectory of the polishing tool requires a full definition in position (three DOF) and at least two orientation coordinates[1] (2 DOF). There exists kinematic redundancy with respect to this task. The human body can be positioned in an infinite number of postures without changing the desired position and orientation of the tool. This redundancy allows the human to simultaneously perform other tasks, e.g., optimize the posture of the body for major comfort or looking around in the environment. In robotics, this multi-task control is often implemented in a hierarchical manner. Each task is executed with an order of priority. The highest priority must be given to tasks that are indispensable. For instance, joint limit avoidance or collision avoidance must be accomplished to ensure a safe task execution. The lowest priority is given to tasks that are desired but are not required to achieve the main goal of an application. For instance, use the minimum kinetic energy during the motion.

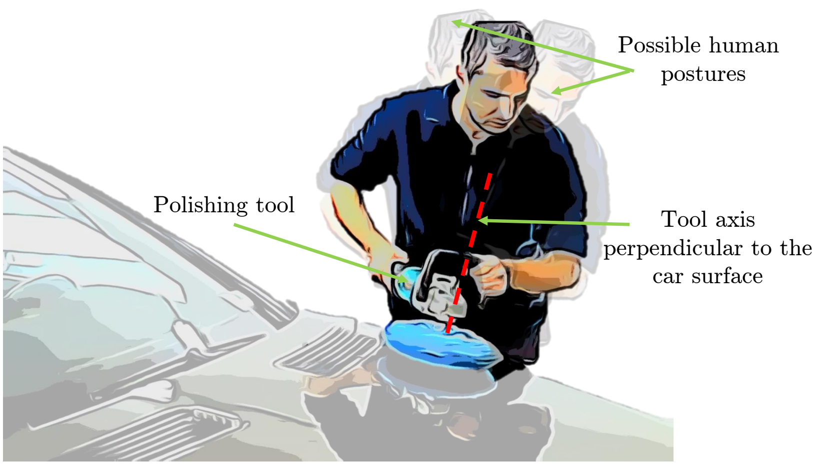


Figure 1.1: Example: human polishing a car. The human can have infinity body postures to perform the task.

[1]To execute this task, the orientation around the perpendicular axis to the surface of the car is not relevant.

Additionally, modern collaborative robots are expected to permanently interact with the human. Similar to a human co-worker, the robot is required to independently perform a task, and to be permanently ready to collaborate. For instance, a mobile manipulator may be busy picking objects from a shelf and putting them into a cart. At any time, a human may assist (by shoving the objects closer to the robot), disturb (by pushing the mobile platform aside), or abort the task and assign a new one. This interactivity implies intuitive input from the human to the robot. The robot must estimate the human intention and react accordingly. The choice of this input depends on the available sensors. Although visual gestures or voice commands may be very intuitive, industrial robots usually do not possess visual or voice sensor-based systems. Physical interaction is intuitive and force/torque sensors are usually integrated in industrial collaborative robots. Therefore, this thesis considers the force/torque as the most intuitive way to communicate with the robot.

A control framework must be employed to command the robot to achieve multiple tasks in a hierarchical manner without endangering the human, objects in the environment or itself. The framework must solve redundancies considering contact between the robot structure and the human to achieve physical interaction.

Section 1.1 presents an overview of previous works related to physical Human-Robot Interaction (pHRI) under constraints. The gaps in the research of constrained multitasking frameworks for pHRI are presented in section 1.2. Section 1.3 introduces the concept of a hierarchical framework for pHRI and the definitions that will be used along this thesis. The contributions and an overview of the thesis are presented in section 1.4.

1.1 Related work

Compliance is a key aspect to achieve a safe behavior in collaborative scenarios. The robot reacts to external forces with certain flexibility instead of being completely rigid. This is achieved in a natural way by controlling the generalized forces in the robot joints instead of positions or velocities. A common approach to perform force/torque control is to control the robot by defining a mechanical impedance law [Hog84]. Cartesian forces on a point attached to the robot structure[2] are computed between the desired point and the current point as a mass-spring-damper (see Fig. 1.2) system and mapped to desired generalized joint forces.

This force-mapping technique is formalized in the Operational Space Control Framework (OSC) [Kha87a]. The robot is controlled in the operational space[3] by forces that are mapped to generalized joint forces. OSC enables the possibility to perform multi-task control in a prioritized manner. During operation, the robot can perform a high-priority task and use the remaining mobility for

[2]Commonly, the point is the center of the robot's end effector.

[3]The space where a task is defined. Commonly the Cartesian space. For instance, a common task is reaching a desired Cartesian position to pick an object.

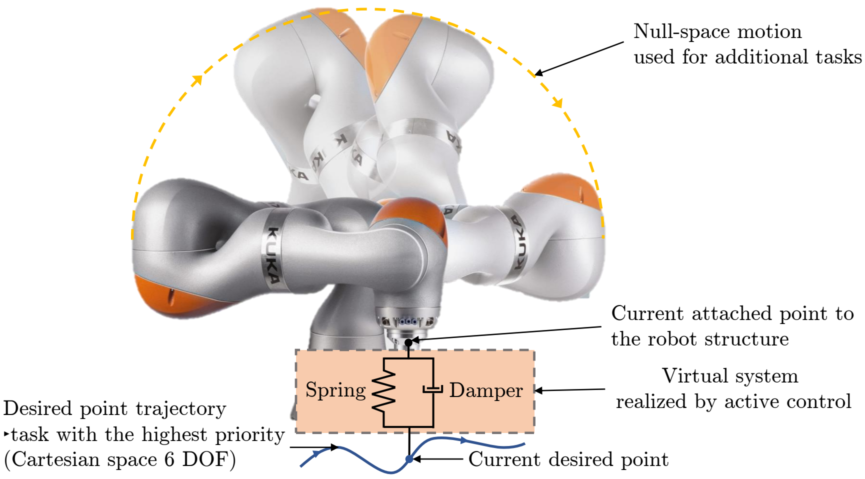


Figure 1.2: Example of a whole-body impedance control on a collaborative industrial robot with seven actuated DOF

tasks with lower priority. When no more mobility is available, the remaining tasks are sacrificed. The remaining mobility appears due to kinematic redundancy, as the robot has more degrees of freedom available what a task requires. Null-space projectors are used to exploit redundancy and create a prioritized stack of tasks for multi-task control [BHB84, HS87, NHY87]. The idea is to achieve every task in an optimal way respecting the given priority. The main drawback of the use of projectors is that they do not allow the inclusion of inequalities, which are required to achieve some of the indispensable tasks such as joint limit avoidance. The objective is not defined as an equality (e.g., end effector must achieve a desired position), instead it is defined as a range (e.g., joint must stay between a minimal and maximal value). The most frequently employed method to include these inequalities is by application of repulsive potential fields [Kha85, DASO+07, SGJG10, DWT+11]. The potential field is commonly activated in proximity to the inequality limit, producing a counteracting force that repels the robot to reach that limit. However, the activation and deactivation of the task can lead to discontinuous solutions and oscillations, as pointed out in [HLP12]. Furthermore, these techniques assign different priority to multiple indispensable tasks. For instance, the joint limit task is assigned with the highest priority, and obstacle avoidance as the second most important task.

In more recent years, other approaches have been investigated to achieve hierarchical multi-task control. [EMW14, HLM+18, QA19] formulate the redundancy problem as a quadratic programming problem. The advantages of such methods is the inclusion of inequalities directly in the redundancy solver. Activation and deactivation of tasks is not required. Multiple indispensable tasks can be handled as equality or inequality constraints having the same level of priority. The main drawback of these techniques is that stability has not been proved yet. In contrast, the stability proof of projector-based techniques has been presented in [DOAS13, DOP18].

Although several frameworks have been proposed to achieve hierarchical control, there are still problems and limitation with such techniques. These will be pointed out in the next section.

1.2 Problem statement

As mentioned above, much research has been done to create a hierarchical control framework capable of managing redundancy. However, none of them is able to properly handle diverse types of inequalities, and at the same time, to allow physical interaction. Projector-based solvers handles all indispensable tasks with different priority, although each of these tasks is equally important and they must not be prioritized among them. Quadratic programming-based solvers do not have this issue. Indispensable tasks can be formulated as constraints with equal level of importance. They all must be respected, otherwise the problem would be unsolvable. However, the robot's behavior has not been properly studied under physical interaction, when redundancy is solved with these methods. In addition, projector-based and QP-based approaches rely in a proper parametrisation of the inequalities to avoid discontinuities and oscillations.

This thesis researches and develops a control hierarchical framework to have a robotic co-worker with full physical interactivity. In the final framework, force inputs are given to the robot and the robot reacts according to it. The human can push the robot in a desired direction while experiencing a "friendly"[4] response of the robot in the proximity to a limit. The robot should be able to follow the task while respecting all constraints (or indispensable tasks) and despite any disturbances given by the human.

Three questions are treated to achieve the goal:

- How to add constraints in the stack of tasks with same priority level? (Example joint and Cartesian constraints)
- What if the human makes the task unfeasible?
- How to handle disturbances due to human interaction?

1.3 Concept of hierarchical constraints/tasks framework

To achieve the goal stated above, a framework that is able to manage task and constraints under physical human-robot interaction (pHRI) must be developed. The framework should allow for the execution of complex tasks by planning in low-dimensional, intuitive spaces while constraining the motion, instead of considering the whole configuration space. In [Die16], the author considers the

[4] In the context of this thesis, friendly refers to a not violent reaction of the robot. The robot should not impose high forces on the human.

tasks as objectives with more or less importance. Objectives are classified from minor (optimization criteria as energy efficiency or natural arm configuration) to indispensable (safety as collision avoidance). In this work, those objectives or assignments are separated in two groups: constraints and tasks. One could say that constraints are the "indispensable" objectives, while tasks are all other possible assignments that the robot may perform with an order of priority. The use of this terminology has some implications in the development of this work.

1.3.1 Constraints

To explain the concepts, envision a humanoid robot that is expected to perform some tasks. Just like humans, humanoid robots possess limitations, e.g., physical limitations such as joint limits or physical capabilities. The violation of these constraints can imply damage of robot hardware or damage to the environment. Additionally, some constraints can come from safety limitations that depend on the application. In human-robot collaborative applications, for instance, safe velocity limits may be imposed to the robot to avoid injuring the human. These are imposed virtual limitations that restrict the real capabilities of the robot, but they are an "indispensable" objective for the specific application. Virtual or physical limitations must always be respected. In the sense of the framework, these limitations are constraints and they are defined as follows:

Definition 1: *Constraints are assignments that must always be performed or held and must never be violated.*

Definition 1 implies that:

Requirement 1.1 Constraints have the highest level of priority.

Requirement 1.2 There is no prioritization among constraints unless a constraint that is not under control is in conflict with controlled constraints. This case is illustrated in section 1.3.3.

Note: In this context, joint-limit avoidance constraint is under control as the control inputs can influence the motion in the constrained space, i.e., the control inputs can influence the motion in the joint space. However, the motion of an object that is not attached to the robot structure cannot be modified by the control inputs. Therefore, a constraint to avoid that object is not under full control. For instance, if the object is moving towards the robot with a higher speed than the permitted robot speed, it will not be possible to slow down the object to keep a maximal relative speed between them.

1.3.2 Tasks

Tasks that a human is expected to perform include: grasping a cup, opening a door, sweeping the floor etc. Not performing any of these tasks or performing them poorly does not imply damages to the robot, the environment or the human. However, some tasks may be more important than others. For example, while grasping a cup, it may be desired to keep a specific posture of the torso to minimize the spent energy.

Definition 2: *Tasks are assignments that the robot should perform if possible. They should be performed according to an order of priority and respecting all constraints.*

The requirements for the tasks are then:

Requirement 2.1 Tasks must not interfere with tasks of higher priority.

Requirement 2.2 Tasks must be performed respecting all constraints.

Requirement 2.3 Tasks must be performed as good as the available DOF allows it.

1.3.3 Conflict management

Considering the aforementioned definitions and requirements, conflicts may appear between tasks and/or constraints. They should be managed in a proper manner for each specific situation. There are three different situations or cases that may happen:

Case 1.1 There exists a conflict between two constraints.

Case 1.2 There exists a conflict between two tasks.

Case 1.3 There exists a conflict between a task and a constraint.

The framework should react to these situations following the next requirements:

Requirement 3.1 If case 1.1, then the only constraints that can be respected, are the constraints that are under control. A safe strategy must be established.

Requirement 3.2 If case 1.2, then the task with higher priority prevails and the task with lower priority is sacrificed or partially fulfilled using the available mobility.

Requirement 3.3 If case 1.3, then the constraint is respected and the task is sacrificed or partially fulfilled using the available mobility.

To clarify how the framework should work, the humanoid robot shown in fig. 1.3 is taken as example. The humanoid should grasp the cup on the table with the left hand, while holding another cup with the right hand. A set of constraints and tasks is given as example:

Constraints:

JLA: Joint limit avoidance

SCA: Self collision avoidance

OCA: Obstacles collision avoidance

Tasks:

HRP: Hold initial right hand pose

HPT: Hold initial torso pose

GLA: Grasp a cup with the left hand

In the first example (Fig. 1.3(a)), the humanoid has to respect only two constraints and tries to perform three tasks. When the least important task is to grasp a cup with the left hand, the left hand does not achieve the cup because the torso pose must be held (HPT has higher priority, case 1.2). If the priorities are changed and HPT task becomes less important than GLA, the left hand still does not reach the cup, because the right hand should not be moved from its initial pose (HRP has the highest priority, fig. 1.3(b)). It becomes clear that the only way to grasp the cup on the table is giving the highest priority to the GLA task. In this case, the right hand and the torso lose their initial poses in pro of grasping the cup, fig. 1.3(c). However, there is a collision with the table while performing the left hand motion, because there is not a constraint for avoiding the obstacles yet. When the constraint OCA is included, the motion of the left hand is changed to avoid collision with the table. In this case, when the collision is about to happen, the first task (GLA) is sacrificed in favor of the constraint (case 1.3, see fig. 1.3(d)). Another example of case 1.3 would be if the joint limit of the hip were tighter. The left hand could not reach the cup because the robot had to respect its joint limits. Consider a moving obstacle (a ball) that is approaching the humanoid. The robot tries to avoid the ball by bending the hip and the neck until they reach theirs limits. In this situation, the constraints JLA and OCA are in conflict and both cannot be respected. Only the robot constraints (JLA) can be guaranteed, because the motion of the object cannot be controlled (this conflict is of course assuming motion is possible only in 2D and the hips can not bend outside of the page plane). Note that before reaching the limits, the second task HPT is in conflict with the constraint OCA. Only this task is sacrificed, while the main taks HRP is performed.

1.3.4 Response to physical interaction

Besides managing the conflicts in a proper manner, the framework must work under pHRI. In the context of this thesis, it means that the robot must move intuitively if external forces are applied to it, either by the human and/or the environment. The effect of external forces can be separated in two cases:

Case 2.1 External force is intended to violate a constraint.

Case 2.2 External force disturbs a task.

Case 2.2.1 External force has a component in the task direction.

Case 2.2.2 External force has a component in the null-space.

The following requirements are derived in order to ensure an intuitive behavior of the robot:

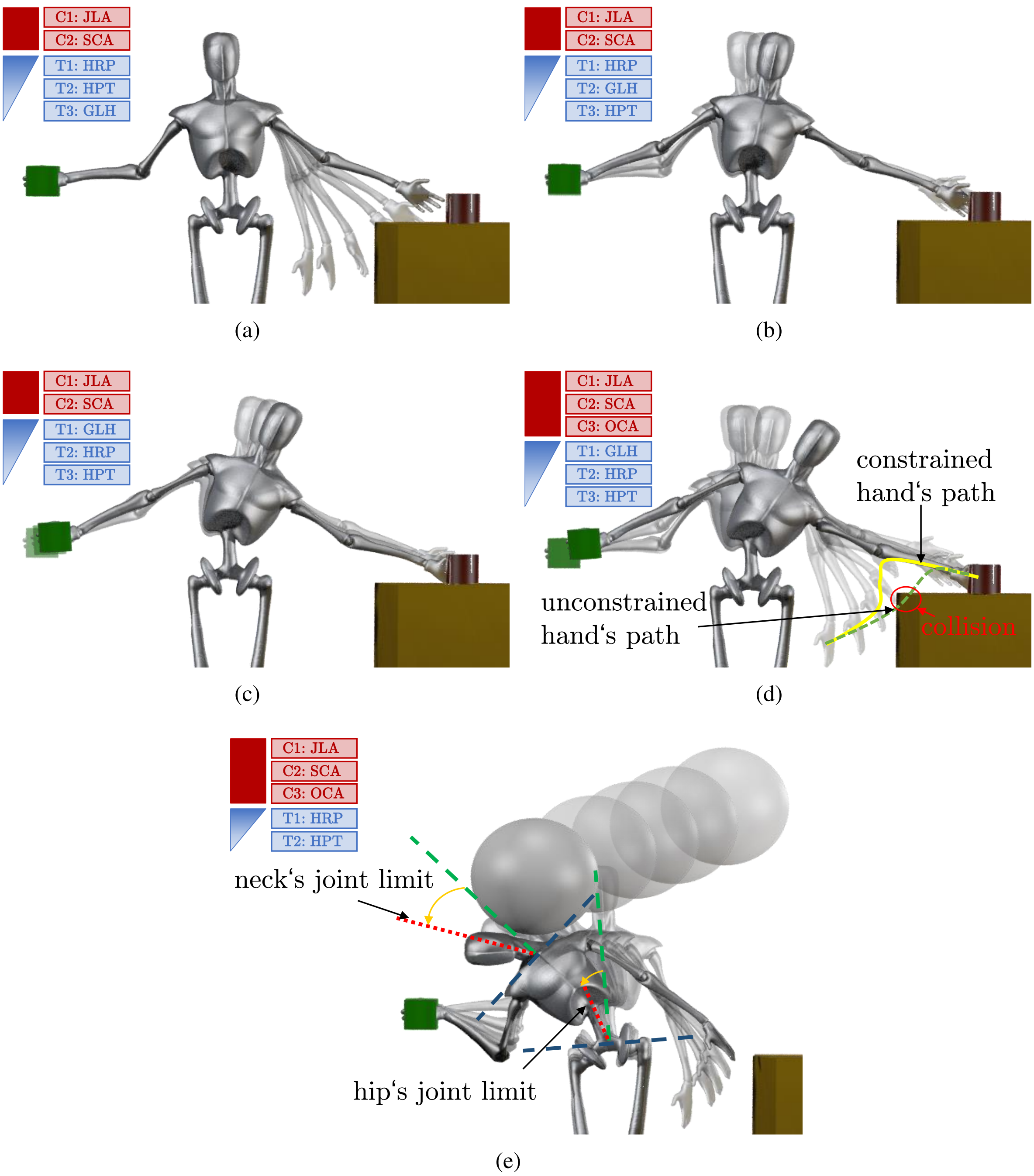


Figure 1.3: Example of a humanoid robot performing tasks in different order of priorities while respecting the constraints. Grasping a cup with the left hand (GLA) starts being the least important task, until it becomes the most important (from (a) to (d)). In (d), besides the fundamental constraints (avoiding joint limits - JLA and avoiding self collision - SCA), another constraint is added to avoid obstacles (OCA). When a moving obstacle enters in the environment a conflict between two constraints occurs (e).

Requirement 4.1 The human should not feel any high force feedback during an interaction regardless of the case.

Requirement 4.2 In case 2.2.1, the robot must behave as it is intended by a defined task dynamics.

Requirement 4.3 In case 2.2.2, the robot motion in the null-space should follow the desired dynamics. Often, the dynamics expected in the remaining null-space is a joint damper to dissipate kinetic energy. Intuitively the human expects the robot to move in the direction of the force.

1.3.5 Assumptions

Considering that the control framework must work for most collaborative robots in the market, some assumptions concerning the control are made:

Assumption 1 The robot can be controlled by forces/torques.

Assumption 2 No external sensors are used.

Assumption 3 There are no changes on the priorities of tasks.

Considering these requirements and assumptions, two of the most popular methods to implement force control are presented in section 2.1, followed by two approaches to achieve task hierarchy in sections 2.2 and 2.3.

1.4 Contributions and overview

This dissertation proposes a framework to address the problem stated in section 1.2 under the requirements given in section 1.3. The framework is validated in simulation and on a real robot in different scenarios.

The framework consists of a stack of tasks, a stack of constraints and a solver, as shown in fig. 1.4. Tasks are "assignments" that should be performed if possible. Each task is performed in a specific order of priority. Constraints are "assignments" that the robot must perform or hold always and must never be sacrificed, e.g., joint limits or obstacle avoidance. Constraints have the highest level of priority and there is no prioritization between them. Although constraints are usually defined by inequalities, they are not restricted to it. The solver solves the redundancy performing the tasks with the given hierarchy, while respecting the constraints. The output of the framework are joint generalized forces to command the robot. A robot model computes the necessary inputs for the framework from the current robot state. Kinematic and dynamic data are sent to the framework to compute the next solution and to produce the required robot motion.

The main contribution of this thesis is the handling of the stack of constraints. Chapter 3 addresses inequality constraints in joint space. The chapter proposes a novel method to have a smooth limit avoidance. The method shows numerous advantages in comparison with the commonly used potential field approach. Complex parameter setting is not required and vibrations are avoided while an intuitive physical interaction is performed. Chapter 4 extends the method to different kind of constraints in the Cartesian space. A generalized architecture is proposed where all the constraints are treated with the same level of importance. The special cases of the rotational limits and the avoidance of dynamic obstacles are included in the general framework.

Chapter 5 proposes a method to perform time invariant control. The aim of the approach is to increase physical interaction with the human. Experiments validate the method in simulation and on the real robot. Human disturbances can move the robot away from the main path or goal, while the maximal Cartesian velocity is saturated to remain in safe limits. Furthermore, the approach does not require motion planning to trace a desired path.

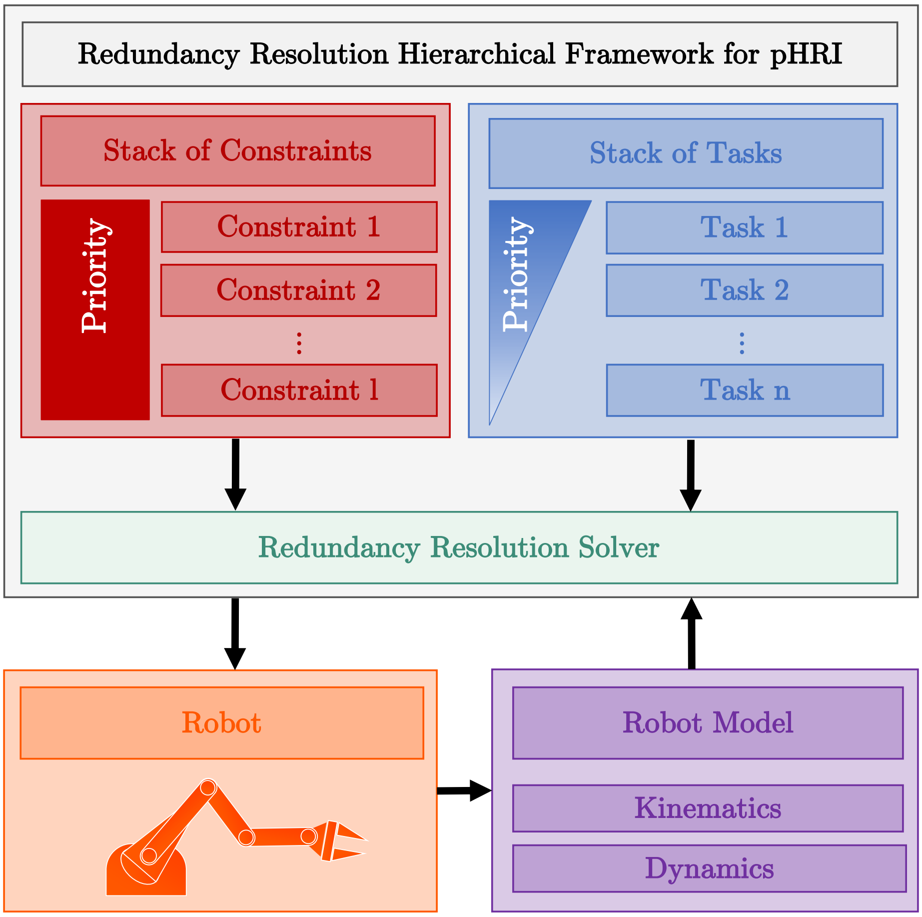


Figure 1.4: Thesis overview: A hierarchical redundancy framework that works under physical human robot interaction

Finally, the thesis analyses and compares current schemes for redundancy resolution under pHRI. Despite the advantages of quadratic programming approaches for solving redundancy (as the lack of numerical singularities), chapter 6 presents their drawback when the physical interaction is desired and interactivity is maximized. A novel method is proposed to combine quadratic programming-based with null-space projector-based techniques. The aim of this method is to have an intuitive physical interaction while including force/torque limits and avoid the use of singularity avoidance methods.

2 Fundamentals

This chapter presents the basic theoretical investigations that are used for the development of this thesis. The fundamentals of impedance control for having a compliant interaction behavior is introduced in the next section. Methods to design the dynamic behavior in the operational space are presented in section 2.1.1 and 2.1.2. Afterwards, three techniques to impose a prioritized hierarchy are briefly explained in sections 2.2, 2.3 and 2.4. Finally, the presented methods and their implementation are discussed in section 2.5.

2.1 Impedance control

The requirements presented in section 1.3 imply that the robot interacts with an unknown environment and the human operator. With that purpose, a suitable compliant behavior in Cartesian and/or joint space is required. Impedance control, introduced in [Hog84], is a well established framework to accomplish this purpose, but misses a proper tasks hierarchy. Component impedances are superimposed to achieve different tasks with no order of priority. In other words, all tasks are considered to have the same priority. They may be achieved only partially but not fully. The inclusion of tasks prioritization was treated in [Kha85] by exploiting the null-space of an operational space in which a task is defined. Khatib in [Kha85] proposes an impedance law that establishes a desired dynamic behavior for an operational point. This point can be specified in any space (the operational space, e.g., joint or Cartesian space or a subset of any of both). A secondary task can be included in a different operational space and in the control law by projecting it into the null-space of the main task. The main task is accomplished while the secondary task with lower priority is either: fully achieved, if there is enough null-space from the main task (no conflict between tasks); partially achieved, if more null-space is needed (partial conflict); or sacrificed, there is not the required null-space for the secondary task (full conflict).

In general, impedance control relies from the assumption that the dynamic relation between the operational positions, velocities and accelerations of a point, and the virtual forces acting on it is:

$$\boldsymbol{f}_{\mathrm{t}} + \boldsymbol{f}_{\mathrm{t,ext}} = \boldsymbol{\Lambda}_{\mathrm{t}}(\boldsymbol{x}_{\mathrm{t}})\ddot{\boldsymbol{x}}_{\mathrm{t}} + \boldsymbol{\mu}_{\mathrm{t}}(\boldsymbol{x}_{\mathrm{t}}, \dot{\boldsymbol{x}}_{\mathrm{t}}) + \boldsymbol{p}_{\mathrm{t}}(\boldsymbol{x}_{\mathrm{t}}), \tag{2.1}$$

where $\boldsymbol{f}_{\mathrm{t}}$ is the operational forces vector, $\boldsymbol{f}_{\mathrm{t,ext}}$ are the external forces, $\boldsymbol{\Lambda}_{\mathrm{t}}(\boldsymbol{x}_{\mathrm{t}})$ is the symmetric inertia matrix, $\boldsymbol{\mu}_{\mathrm{t}}(\boldsymbol{x}_{\mathrm{t}}, \dot{\boldsymbol{x}}_{\mathrm{t}})$ is the vector of centrifugal and Coriolis forces and $\boldsymbol{p}_{\mathrm{t}}(\boldsymbol{x}_{\mathrm{t}})$ is the vector of gravity forces, all represented in the operational space. To simplify the notation, dependency upon the operational point coordinates $\boldsymbol{x}_\mathrm{t}$ and its derivative will no longer be denoted. The subscript $[]_\mathrm{t}$ indicates the dependency on the task. The command vector of the decoupled operational point $\ddot{\boldsymbol{x}}_\mathrm{t}$ describes the desired dynamics of the task.

The dynamic terms in eq. 2.1 are related to the parameters involved in the joint space dynamic model. The dynamic equation of a robot with n-joint coordinates is:

$$\boldsymbol{\tau} + \boldsymbol{J}_\mathrm{f}^T \boldsymbol{f}_\mathrm{ext} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{\nu}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}), \tag{2.2}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the robot inertia matrix, and $\boldsymbol{\nu}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are the Coriolis- and centrifugal forces, $\boldsymbol{g}(\boldsymbol{q})$ are the gravity forces and $\boldsymbol{\tau}$ represents the actuator control generalized forces. To simplify the notation, dependency upon $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ will no longer be denoted. The Jacobian $\boldsymbol{J}_\mathrm{f}$ maps the external forces from the contact points to generalized forces in the joint space.[1] Since most of robot structures are based on revolute joints rather than prismatic joints, this thesis uses mostly the term *joint torques* instead of *generalized joint forces* without loss of generality.

If redundancy exists for a given a task, i.e., the degrees of freedom (DOF) of the robot is bigger than the number of task coordinates ($n > m_\mathrm{t}$), then there is an infinite number of elementary displacements of the redundant mechanism that could take place to perform the task. Those displacements conform the null-space of the Jacobian $\boldsymbol{J}_\mathrm{t}$. Moreover, there is an infinite number of joint torque vectors that could be applied without affecting the resulting forces at the operational point. Thus, the equation that establishes the relationship between operational forces $\boldsymbol{f}_\mathrm{t}$ and joint torques $\boldsymbol{\tau}$ is established by equation:

$$\boldsymbol{\tau} = \boldsymbol{J}_\mathrm{t}^T \boldsymbol{f}_\mathrm{t} + \boldsymbol{N}_\mathrm{t} \boldsymbol{\tau}_\mathrm{any}, \tag{2.3}$$

with:

$$\boldsymbol{N}_\mathrm{t} = \boldsymbol{I} - \boldsymbol{J}_\mathrm{t}^T \bar{\boldsymbol{J}}_\mathrm{t}^T, \tag{2.4}$$

where $\boldsymbol{N}_\mathrm{t}$ is the projection onto the null-space of task t, $\boldsymbol{\tau}_\mathrm{any}$ is an arbitrary generalized joint torque vector, which is projected onto the null-space of that task, and $\bar{\boldsymbol{J}}_\mathrm{t}^{\boldsymbol{T}}$ is the dynamically consistent generalized inverse that minimizes the robot's instantaneous kinetic energy [Kha90b]. This inverse $\bar{\boldsymbol{J}}$ is unique and is given by:

$$\bar{\boldsymbol{J}}_\mathrm{t} = \boldsymbol{M}^{-1} \boldsymbol{J}_\mathrm{t}^T \boldsymbol{\Lambda}_\mathrm{t}, \tag{2.5}$$

where $\boldsymbol{\Lambda}_\mathrm{t}$ is the inertia matrix of the robot mapped to the operational space [Kha80]:

$$\boldsymbol{\Lambda}_\mathrm{t} = (\boldsymbol{J}_\mathrm{t} \boldsymbol{M}^{-1} \boldsymbol{J}_\mathrm{t}^T)^{-1}. \tag{2.6}$$

[1] When multiple external forces are applied in different points on the robot structure or the tool, Jacobians and external forces for each of these points are stacked in $\boldsymbol{J}_\mathrm{f}$ and $\boldsymbol{f}_\mathrm{ext}$, respectively. For c_p contact points, $\boldsymbol{J}_\mathrm{f} = [\boldsymbol{J}_{\mathrm{f},1}, \ldots, \boldsymbol{J}_{\mathrm{f},c_\mathrm{p}}]^\mathrm{T}$ and $\boldsymbol{f}_\mathrm{ext} = [\boldsymbol{f}_{\mathrm{ext},1}, \ldots, \boldsymbol{f}_{\mathrm{ext},c_\mathrm{p}}]^\mathrm{T}$.

Finally, the gravity, the Coriolis and centrifugal forces in the joint space are mapped to the operational space[2]

$$\boldsymbol{\mu}_{\mathrm{t}} = \bar{\boldsymbol{J}}_{\mathrm{t}}^{T} \boldsymbol{\nu} - \Lambda_{\mathrm{t}} \dot{\boldsymbol{J}}_{\mathrm{t}} \dot{\boldsymbol{q}}, \tag{2.7}$$

$$\boldsymbol{p}_{\mathrm{t}} = \bar{\boldsymbol{J}}_{\mathrm{t}}^{T} \boldsymbol{g}. \tag{2.8}$$

The desired dynamics of the task behaves like a mass-spring-damper system and has decoupled dynamics

$$\boldsymbol{\Lambda}_{\mathrm{t,d}} \ddot{\boldsymbol{e}}_{\mathrm{t}} + \boldsymbol{D}_{\mathrm{t}}(\dot{\boldsymbol{e}}_{\mathrm{t}}) + \boldsymbol{K}_{\mathrm{t}}(\boldsymbol{e}_{\mathrm{t}}) = \boldsymbol{f}_{\mathrm{ext}}, \tag{2.9}$$

where $\boldsymbol{K}_{\mathrm{t}} \in \mathbb{R}^{m_{\mathrm{t}} \times m_{\mathrm{t}}}$ and $\boldsymbol{D}_{\mathrm{t}} \in \mathbb{R}^{m_{\mathrm{t}} \times m_{\mathrm{t}}}$ are the positive definite matrices that define the stiffness and damping, and $\boldsymbol{\Lambda}_{\mathrm{t,d}} \in \mathbb{R}^{m_{\mathrm{t}} \times m_{\mathrm{t}}}$ denotes the desired inertia matrix. The vector $\boldsymbol{e}_{\mathrm{t}} = \boldsymbol{x}_{\mathrm{t}} - \boldsymbol{x}_{\mathrm{t,d}}$ is the error between the current and the desired position.[3] The desired dynamics takes the form:

$$\ddot{\boldsymbol{x}}_{\mathrm{t}} = \ddot{\boldsymbol{x}}_{\mathrm{t,d}} - \boldsymbol{\Lambda}_{\mathrm{t,d}}^{-1} (\boldsymbol{D}_{\mathrm{t}} \dot{\boldsymbol{e}}_{\mathrm{t}} + \boldsymbol{K}_{\mathrm{t}} \boldsymbol{e}_{\mathrm{t}} - \boldsymbol{f}_{\mathrm{ext}}). \tag{2.10}$$

$\ddot{\boldsymbol{x}}_{\mathrm{t}}$ is the operational acceleration that leads to the desired dynamic behavior in eq. 2.10. Replacing this acceleration in eq. 2.1 and then $\boldsymbol{f}_{\mathrm{t}}$ in eq. 2.3, the command torque becomes

$$\boldsymbol{\tau} = \boldsymbol{J}_{\mathrm{t}}^{T} (\boldsymbol{\Lambda}_{\mathrm{t}} \ddot{\boldsymbol{x}}_{\mathrm{t,d}} - \boldsymbol{\Lambda}_{\mathrm{t}} \boldsymbol{\Lambda}_{\mathrm{t,d}}^{-1} (\boldsymbol{D}_{\mathrm{t}} \dot{\boldsymbol{e}}_{\mathrm{t}} + \boldsymbol{K}_{\mathrm{t}} \boldsymbol{e}_{\mathrm{t}} - \boldsymbol{f}_{\mathrm{ext}}) - \boldsymbol{f}_{\mathrm{ext}} + \boldsymbol{\mu}_{\mathrm{t}}) + \boldsymbol{N}_{\mathrm{t}} \boldsymbol{\tau}_{\mathrm{any}} + \boldsymbol{g}, \tag{2.11}$$

where the gravity compensation is added directly at joint level to avoid unnecessary computations.[4]

[Hog84] introduces an empirical rule for designing the impedance behavior. The manipulator impedance should be proportional to the environmental admittance. If the use-case implies to interact with a rigid environment or objects, the manipulator should have low impedance to accommodate to the environment; if the environment is flexible, the manipulator may have high impedance to impose motion upon the environment.

The desired inertia matrix $\boldsymbol{\Lambda}_{\mathrm{t,d}}$ must be chosen in order to be a positive symmetric definite matrix. The choice of $\boldsymbol{\Lambda}_{\mathrm{t,d}}$ completes the desired behavior of the robot, when external forces disturbs the operational point. In this thesis, this choice is reduced to two cases: an identity matrix to emulate a unit mass system [Kha87a] and the inertia matrix of the robot expressed in the operational point [ASOFH03]. Each of these cases provides benefits and limitations, each with different methods to design the stiffness and damping matrices. The next sections 2.1.1 and 2.1.2 explain in detail these differences.

[2] Derivation of these equations can be found in [Kha87b, Kha90a].

[3] The desired position is a virtual equilibrium position, which should only be reached in case external forces do not disturb this point.

[4] If the lowest priority task is defined in joint space, the computation of $\boldsymbol{p}_{\mathrm{t}}$ for each task is unnecessary. The proof can be found in appendix A.1.

2.1.1 Operational Space Control (OSC)

By choosing the desired inertia matrix as an identity matrix $\boldsymbol{\Lambda}_{\text{t,d}} = \boldsymbol{I}$, the operational point can be seen as a single unit mass moving in a m_{t} dimensional space [Kha87a]. The command joint torque becomes

$$\boldsymbol{\tau} = \boldsymbol{J}_{\text{t}}^{T} \underbrace{\left(\boldsymbol{\Lambda}_{\text{t}} \underbrace{\left(\ddot{\boldsymbol{x}}_{\text{t,d}} - \boldsymbol{D}_{\text{t}}\dot{\boldsymbol{e}}_{\text{t}} - \boldsymbol{K}_{\text{t}}\boldsymbol{e}_{\text{t}} + \boldsymbol{f}_{\text{ext}}\right)}_{\ddot{\boldsymbol{x}}_{\text{t}}} - \boldsymbol{f}_{\text{ext}} + \boldsymbol{\mu}_{\text{t}}\right)}_{\boldsymbol{f}_{\text{t}}} + \boldsymbol{N}_{\text{t}}\boldsymbol{\tau}_{\text{any}} + \boldsymbol{g}. \tag{2.12}$$

To achieve the desired dynamic behavior of 2.10, the external forces $\boldsymbol{f}_{\text{ext}}$ need to be measured. However, this measurement is typically ignored in the OSC due to the need of an accurate dynamic model of the robot and high performance force/torque sensors.

The design of the stiffness and damping matrices depends on the expected dynamics and it is detailed below.

Design of dynamic behavior with decoupled dynamics

In OSC, the desired inertia matrix $\boldsymbol{\Lambda}_{\text{t,d}}$ is constant and its eigenvectors coincide with those of $\boldsymbol{K}_{\text{t}}$ (assuming a definite positive diagonal stiffness matrix). The design of the damping matrix can be reduced to choosing damping coefficients for m_{t} second order systems. Each pair of spring-damper coefficients describes the dynamics of the system along one of the eigenvectors. Each system can be describe as:

$$\frac{f_i}{\lambda_{i,i}} = \ddot{x}_i + \frac{d_{i,i}}{\lambda_{i,i}}\dot{x}_i + \frac{k_{i,i}}{\lambda_{i,i}}x_i, \tag{2.13}$$

where the subindex $i = 1..m_{\text{t}}$ indicates the coefficient of each matrix or vector along the eigenvector. The scalar $\lambda_{i,i}$ is the virtual mass of the system. For such systems, $\frac{k_{i,i}}{\lambda_{i,i}} = \omega_n^2$ is the undamped natural frequency and $\frac{d_{i,i}}{\lambda_{i,i}} = 2\zeta\omega_n$, where ζ is the damping ratio. A common choice of the damping coefficient is $d_{i,i} = 2\lambda_{i,i}\sqrt{k_{i,i}/\lambda_{i,i}}$ to have a critically damped system ($\zeta = 1.0$).

2.1.2 Impedance control without inertia shaping

A well-known approach to achieve a mass-spring-damper dynamics and to avoid the measurement of external forces is to not shape the inertia. The desired inertia matrix is set as the current inertia matrix of the robot expressed in the operational point [ASOFH03, Ott08, Die16]. With this choice $\boldsymbol{\Lambda}_{\text{t,d}} = \boldsymbol{\Lambda}_{\text{t}}$, the command joint torque becomes

$$\boldsymbol{\tau} = \boldsymbol{J}_{\text{t}}^{T} \underbrace{\left(\boldsymbol{\Lambda}_{\text{t}}\ddot{\boldsymbol{x}}_{\text{t,d}} - \boldsymbol{D}_{\text{t}}\dot{\boldsymbol{e}}_{\text{t}} - \boldsymbol{K}_{\text{t}}\boldsymbol{e}_{\text{t}} + \boldsymbol{\mu}_{\text{t}}\right)}_{\boldsymbol{f}_{\text{t}}} + \boldsymbol{N}_{\text{t}}\boldsymbol{\tau}_{\text{any}} + \boldsymbol{g}. \tag{2.14}$$

The desired acceleration $\ddot{\boldsymbol{x}}_{\mathrm{t,d}}$ is usually neglected to avoid computation of the inertia matrix in the operational space[5] $\boldsymbol{\Lambda}_{\mathrm{t}}$. If not neglected, a desired acceleration profile decreases the error for trajectory tracing.[6] However, many use cases of impedance control (e.g. insertion of pieces, sweeping the floor, polishing a table, hand guidance) usually do not require high tracking accuracy.

The design of the stiffness and damping matrix is not as straightforward as in the OSC. Due to the variability of the desired inertia matrix, the choice of these matrices has to be properly treated.

Design of dynamic behavior with decoupled dynamics

Having the desired inertia matrix variable implies that a constant damping matrix is not a good choice. In many applications, users need a well defined behavior in every Cartesian direction. Therefore, $\boldsymbol{D}_{\mathrm{t}}$ has to be chosen as a function of $\boldsymbol{\Lambda}_{\mathrm{t}}$.

Two methods to design the damping for a given symmetric positive definite stiffness matrix are presented in [ASOFH03]. This thesis considers only the *double diagonalization* design. This approach uses the generalize eigenvalue problem to design a damping matrix that depends on the eigenvectors of the inertia matrix.

Given that $\boldsymbol{\Lambda}_{\mathrm{t}}$ is symmetric positive definite and $\boldsymbol{K}_{\mathrm{t}}$ is defined to be so, too; a non-singular matrix $\boldsymbol{Q}_{\mathrm{t}} \in \mathbb{R}^{m_{\mathrm{t}} \times m_{\mathrm{t}}}$ can be found such that $\boldsymbol{\Lambda}_{\mathrm{t}} = \boldsymbol{Q}_{\mathrm{t}} \boldsymbol{Q}_{\mathrm{t}}^T$ and $\boldsymbol{K}_{\mathrm{t}} = \boldsymbol{Q}_{\mathrm{t}} \boldsymbol{K}_{\mathrm{t,0}} \boldsymbol{Q}_{\mathrm{t}}^T$, where $\boldsymbol{K}_{\mathrm{t,0}} \in \mathbb{R}^{m_{\mathrm{t}} \times m_{\mathrm{t}}}$ is a positive definite diagonal matrix. Thus, the matrices $\boldsymbol{\Lambda}_{\mathrm{t}}$ and $\boldsymbol{K}_{\mathrm{t}}$ can be diagonalized simultaneously and the system in eq. 2.9 can be rewritten as

$$\boldsymbol{Q}_{\mathrm{t}} \boldsymbol{Q}_{\mathrm{t}}^T \ddot{\boldsymbol{x}}_{\mathrm{t}} + \boldsymbol{D}_{\mathrm{t}} \dot{\boldsymbol{e}}_{\mathrm{t}} + \boldsymbol{Q}_{\mathrm{t}} \boldsymbol{K}_{\mathrm{t,0}} \boldsymbol{Q}_{\mathrm{t}}^T \boldsymbol{e}_{\mathrm{t}} = \boldsymbol{f}_{\mathrm{ext}}. \tag{2.15}$$

Diagonal elements of $\boldsymbol{K}_{\mathrm{t,0}}$ are the generalized eigenvalue of $\boldsymbol{K}_{\mathrm{t}}$ with respect to $\boldsymbol{\Lambda}_{\mathrm{t}}$. The damping matrix can be then computed as

$$\boldsymbol{D}_{\mathrm{t}} = 2 \boldsymbol{Q}_{\mathrm{t}} \boldsymbol{D}_{\xi} \boldsymbol{K}_{\mathrm{t,0}}^{1/2} \boldsymbol{Q}_{\mathrm{t}}^T, \tag{2.16}$$

where $\boldsymbol{D}_{\xi} \in \mathbb{R}^{m_{\mathrm{t}} \times m_{\mathrm{t}}}$ is a diagonal matrix with diagonal terms ξ_i ($0 \leq \xi_i \leq 1$, 0 for undamped behavior and 1 for real eigenvalues). For a better understanding, the reader is referred to [ASOFH03] and [Ott08].

Some applications require more than one point following a dynamic behavior and/or having a prioritization between the coordinates. In any case, a task hierarchy has to be implemented. In the following sections, three methods are presented to have a prioritized task scheme.

[5]This computation needs a singularity avoidance approach due to the computation of the inverse of $(\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T)$.
[6]The robot must follow a trajectory with a desired position, velocity and acceleration in each time step.

2.1.3 Summary

In general for force/torque control, eq. 2.3 becomes

$$\boldsymbol{\tau}_{\text{cmd}} = \boldsymbol{J}_{\text{t}}^{T}\boldsymbol{f}_{\text{t}} + \boldsymbol{N}_{\text{t}}\boldsymbol{\tau}_{\text{any}} + \boldsymbol{g}, \tag{2.17}$$

where $\boldsymbol{\tau}_{\text{cmd}}$ are the generalized forces to command the robot. The force vector $\boldsymbol{f}_{\text{t}}$ is defined differently if the inertia in the operational space is shaped or not. For the two special cases:

- Inertia shaping as a diagonal unit mass (OSC with $\boldsymbol{\Lambda}_{\text{t,d}} = \boldsymbol{I}$), and ignoring the external forces. The force becomes $\boldsymbol{f}_{\text{t}} = \boldsymbol{\Lambda}_{\text{t}}\ddot{\boldsymbol{x}}_{\text{t}}$. The term $\ddot{\boldsymbol{x}}_{\text{t}}$ is a desired acceleration that defines a dynamic behavior, as given in section 2.1.1. This term may not be confused with the desired acceleration $\ddot{\boldsymbol{x}}_{\text{t,d}}$ that can be computed from a trajectory planning.
- Inertia is not shaped (IWIS with $\boldsymbol{\Lambda}_{\text{t,d}} = \boldsymbol{\Lambda}_{\text{t}}$), $\boldsymbol{f}_{\text{t}}$ is a force that defines the dynamic of the system as defined in section 2.1.2.

It is worth to mention that although the procedure for the translational stiffness is straightforward, the design of a rotational spring is more complex and it depends on the representation of the orientation [ZF00]. [Nat03] presents the consequences of using different orientation coordinates in Cartesian controllers. The use of quaternions is a typical choice given that they are a singularity-free representation of the rotation [Ott08, ORH11, HRO16].

2.2 Task hierarchy based on null-space projectors

Having multiple tasks to be solved by a robot with many DOF necessarily requires a redundancy resolution. The idea is to solve an optimization problem for a hierarchical arrangement of them. The execution of the highest-priority task employs the whole capability of the robotic system. The second-priority task is performed without disturbing the execution of the highest-priority task. The execution of a third-priority task does not interfere with the first and the second level tasks. In other words, a task does not disturb the execution of any other higher-priority one.

[DMB93] presents one approach frequently used to solve redundancy. The approach employs pseudo-inverses and projection matrices. The so-called null-space projections developed in [Lie77, HS87, NHY87, Kha87b] are standards tools today in kinematic control. In torque control, the torque input is processed by the null-space projector $\boldsymbol{N}$ related to all task with higher priority. The resulting joint torque commands the robot to execute the task as well as possible without disturbing any higher-priority task.

Consider a manipulator with n DOF that must perform k tasks with priorities $i = 1..k$ and dimensions m_i, where 1 indicates the highest priority and k the lowest priority. Each task is defined by two components, a desired acceleration $\ddot{\boldsymbol{x}}_i \in \mathbb{R}^{m_i \times 1}$ or force vector $\boldsymbol{f}_i \in \mathbb{R}^{m_i \times 1}$, and a Jacobian $\boldsymbol{J}_{\boldsymbol{i}} \in \mathbb{R}^{m_i \times n}$ that does the mapping from joint velocities to task velocities

$$\dot{\boldsymbol{x}}_i = \boldsymbol{J}_i(q)\dot{\boldsymbol{q}} \tag{2.18}$$

and, with its derivative, defines the mapping from joint accelerations to tasks accelerations

$$\ddot{\boldsymbol{x}}_i = \boldsymbol{J}_i(q)\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_i(q)\dot{\boldsymbol{q}}. \tag{2.19}$$

An overview of the null-space projectors is presented in [DOAS15a]. This thesis only treats dynamically consistent projectors: augmented and successive, where the weighting matrix is the inertia matrix. This matrix is considered as the only valid weighting matrix to keep a dynamic behavior, as shown by [Kha80].

In the following, $\boldsymbol{J}_i$ is assumed to have full rank, hence, is not singular. Dealing with singular situations is treated in section 2.2.3.

2.2.1 Soft hierarchy using successive projectors

In the successive null-space projection, a task joint torque $\boldsymbol{\tau}_2 \in \mathbb{R}^n$ (with second priority level) is projected into the null-space of the main task

$$\boldsymbol{\tau}_2^{\mathrm{P}} = \boldsymbol{N}_1 \boldsymbol{\tau}_2, \tag{2.20}$$

where $\boldsymbol{\tau}_2^{\mathrm{P}}$ is the projected joint torque that does not interfere with the main task. When there are more than two tasks, the hierarchy is given by:

$$\boldsymbol{\tau}_{\mathrm{cmd}} = \boldsymbol{\tau}_1 + \boldsymbol{N}_1(\boldsymbol{\tau}_2 + \boldsymbol{N}_2(\boldsymbol{\tau}_3 + \ldots \boldsymbol{N}_{k-1}\boldsymbol{\tau}_k))). \tag{2.21}$$

The joint torque of the last task $\boldsymbol{\tau}_k$ is projected into a successive multiplication of null-space projectors $\boldsymbol{N}_1^T \boldsymbol{N}_2^T \ldots \boldsymbol{N}_{k-1}^T$. This hierarchy can be recursively written as proposed by [DOAS15a]:

$$\boldsymbol{\tau}_i^{\mathrm{P}} = \boldsymbol{N}_i^{\mathrm{suc}} \boldsymbol{\tau}_i, \tag{2.22}$$

where $\boldsymbol{N}_2^{T,\mathrm{suc}}$ is the successive null-space projector and is obtained by:

$$\boldsymbol{N}_i^{\mathrm{suc}} = \boldsymbol{N}_{i-1}^{\mathrm{suc}}(\boldsymbol{I} - \boldsymbol{J}_{i-1}^T \bar{\boldsymbol{J}}_{i-1}^T), \tag{2.23}$$

with $\boldsymbol{N}_1^{\mathrm{suc}} = \boldsymbol{I}$.

The command joint torque is computed by adding up the main task torque and all projected joint torques:

$$\boldsymbol{\tau}_{\text{cmd}} = \sum_{i=1}^{k} \boldsymbol{\tau}_i^{\text{P}} + \boldsymbol{g}. \tag{2.24}$$

Note that $\boldsymbol{N}^{\text{suc}}$ is not a projector in the strict math sense because it is not idempotent in general (i.e. $\boldsymbol{N}^{\text{suc}}\boldsymbol{N}^{\text{suc}} \neq \boldsymbol{N}^{\text{suc}}$). In robotics, though, it is common to use the term projector in this context. The successive projector does not decouple the tasks completely, i.e. a low priority task could affect all higher priority tasks. The only task that would not be affected is the highest one, as the projector $\boldsymbol{N}_2$ is the only idempotent projector.[7] Therefore, the hierarchy using successive projections is considered as a soft hierarchy. The main advantage of using these projectors is the lack of numerical singularities due to conflicting tasks.

2.2.2 Strict hierarchy using augmented projectors

The augmented projection is computed based on an augmented Jacobian. This projector is idempotent, hence mathematically correct. This hierarchy is then considered as a strict hierarchy. The projected joint torque is computed as follows:

$$\boldsymbol{\tau}_i^{\text{P}} = \boldsymbol{N}_i^{\text{aug}} \boldsymbol{\tau}_i, \tag{2.25}$$

with:

$$\boldsymbol{N}_i^{\text{aug}} = \boldsymbol{I} - \boldsymbol{J}_{i-1}^{\text{aug,T}} \bar{\boldsymbol{J}}_{i-1}^{\text{aug,T}}. \tag{2.26}$$

The main difference with the successive projectors is that the augmented Jacobian matrix $\boldsymbol{J}_{i-1}^{\text{aug}}$ considers all higher priority Jacobian matrices:

$$\boldsymbol{J}_{i-1}^{\text{aug}} = \begin{bmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \\ \vdots \\ \boldsymbol{J}_{i-1} \end{bmatrix}. \tag{2.27}$$

The final control joint torque is obtained with eq. 2.24 by using eq. 2.25 instead of eq. 2.22. To reduce the numerical effort of computing directly eq. 2.27, recursive algorithms are applied [SK05]:

$$\boldsymbol{N}_1^{\text{aug}} = \boldsymbol{I}, \tag{2.28}$$

$$\hat{\boldsymbol{J}}_i = \boldsymbol{J}_i \boldsymbol{N}_i^{\text{aug},T}, \tag{2.29}$$

$$\boldsymbol{N}_i^{\text{aug},T} = \boldsymbol{N}_{i-1}^{\text{aug},T}(\boldsymbol{I} - \hat{\boldsymbol{J}}_{i-1}^T \bar{\hat{\boldsymbol{J}}}_{i-1}^T). \tag{2.30}$$

[7] $\boldsymbol{N}_2^{\text{suc}} = \boldsymbol{I} - \boldsymbol{J}_1^T \bar{\boldsymbol{J}}_1^T = \boldsymbol{N}_2^{\text{suc}} \boldsymbol{N}_2^{\text{suc}}$. Instead, for the third-priority level: $\boldsymbol{N}_3^{\text{suc}} = \boldsymbol{N}_2^{\text{suc}}(\boldsymbol{I} - \boldsymbol{J}_2^T \bar{\boldsymbol{J}}_2^T) \neq \boldsymbol{N}_3^{\text{suc}} \boldsymbol{N}_3^{\text{suc}}$.

The Jacobian $\hat{\boldsymbol{J}}_i$ describes the Jacobian matrix of level i projected onto the null-space of all higher priority tasks.

In addition, to compute the optimal solution of task i, the effect of the task $i-1$ on the space of task i should be extracted from the computation of the force $\boldsymbol{f}_i$ [SK05]. Otherwise, the joint torques computed to produce the desired dynamic behavior in the space of task $i-1$ can produce forces in the space of the task i. Subtracting such forces in the computation of the desired forces for task i, the joint torque vector of task i becomes:

$$\boldsymbol{\tau}_i = \boldsymbol{J}_i^T(\boldsymbol{f}_i - \boldsymbol{\Lambda}_i \boldsymbol{J}_i \boldsymbol{M}^{-1}(\boldsymbol{\tau}_{i-1} - \boldsymbol{\nu})). \tag{2.31}$$

The augmented projector enforces uncoupling of all involved tasks. However, algorithmic singularities arise when tasks on different levels are in conflict with each other [DOAS15a]. This thesis implements an approach based on [HP13] to avoid these singularities.

2.2.3 Singularity avoidance

Singularity robust techniques such as damped-least-square methods [DW95] can be used to avoid singularities. However, this complicates the hierarchy design. In [KSPW04], singularities are avoided using the eigenvalue decomposition of the inverse of the inertia matrix $\hat{\boldsymbol{\Lambda}}_i^{-1} = \hat{\boldsymbol{J}}_i \boldsymbol{M}^{-1} \hat{\boldsymbol{J}}_i^T$ and cancel out the directions with eigenvalues equal to zero. However, high values of $\hat{\boldsymbol{\Lambda}}_i$ can be computed not only when the eigenvalues are exactly zero. High values in the computation of the inertia matrix $\hat{\boldsymbol{\Lambda}}_i^{-1}$ appear also when eigenvalues are close to zero or near to the singularity. Moreover, oscillations can be produced by canceling out the singular directions in one instant of time, as shown in [HP13]. Consider

$$\hat{\boldsymbol{\Lambda}}_i^{-1} = [\boldsymbol{U}_{\text{ns}}\, \boldsymbol{U}_{\text{s}}] \begin{bmatrix} \boldsymbol{\Sigma}_{\text{ns}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\text{s}} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_{\text{ns}}^T \\ \boldsymbol{U}_{\text{s}}^T \end{bmatrix}, \tag{2.32}$$

where $\boldsymbol{\Sigma}_{\text{ns}}$ is a $r \times r$ (r as the rank of $\hat{\boldsymbol{J}}_i$) matrix of the so-considered non singular eigenvalues (over a defined threshold, σ_{max}[8]), $\boldsymbol{U}_{\text{ns}}$ is a $m_{\text{t}} \times r$ matrix with columns that represent direction vectors related to the eigenvalues of $\boldsymbol{\Sigma}_{\text{ns}}$, $\boldsymbol{\Sigma}_{\text{s}}$ is a $(m_{\text{t}} - r) \times (m_{\text{t}} - r)$ matrix of the so-considered singular eigenvalues (under a threshold σ_{max}) and $\boldsymbol{U}_{\text{s}}$ is a $l \times (m_{\text{t}} - r)$ matrix with columns corresponding to the direction of the singular eigenvalues. The matrix $\hat{\boldsymbol{\Lambda}}_i$ can be separated in two matrices:

$$\hat{\boldsymbol{\Lambda}}_i = \hat{\boldsymbol{\Lambda}}_{\text{ns},i} + \hat{\boldsymbol{\Lambda}}_{\text{s},i}, \tag{2.33}$$

[8]Higher values than σ_{max} do not produce high values in the computation of $\hat{\boldsymbol{\Lambda}}_i$. The choice of this value is done empirically.

where $\hat{\mathbf{\Lambda}}_{\text{ns},i}$ and $\hat{\mathbf{\Lambda}}_{\text{s},i}$ are the inertia matrices in the operational space that consider only the non-singular directions, and the singular directions, respectively:

$$\hat{\mathbf{\Lambda}}_{\text{ns},i} = \boldsymbol{U}_{\text{ns}}\mathbf{\Sigma}_{\text{ns}}^{-1}\boldsymbol{U}_{\text{ns}}^T, \tag{2.34}$$

$$\hat{\mathbf{\Lambda}}_{s,i} = \boldsymbol{U}_{\text{s}}\mathbf{\Sigma}_{\text{s}}^{-1}\boldsymbol{U}_{\text{s}}^T. \tag{2.35}$$

A solution to avoid singularities could be the use of $\hat{\mathbf{\Lambda}}_{\text{ns},i}$ in the computation of the task. Nevertheless, this choice does not solve the problem of oscillations. Therefore, as recommended in [HP13], a smooth transition between $\hat{\mathbf{\Lambda}}_i$ and $\hat{\mathbf{\Lambda}}_{\text{ns},i}$ should be implemented. Thus, eq. 2.35 becomes:

$$\hat{\mathbf{\Lambda}}_{\text{s},i} = \boldsymbol{U}_{\text{s}}\boldsymbol{H}(\sigma)\mathbf{\Sigma}_{\text{s}}^{-1}\boldsymbol{U}_{\text{s}}^T, \tag{2.36}$$

where $\boldsymbol{H}(\sigma)$ is a $(m_{\text{t}} - r) \times (m_{\text{t}} - r)$ diagonal matrix with diagonal values $h_i, i = 1...m_{\text{t}} - r$ changing smoothly from a 1 to 0 depending on the corresponding singular eigenvalue, σ.

$$h_i = \begin{cases} 1.0, & \text{if } |\sigma| = \sigma_{\text{max}} \\ h(\sigma), & \text{if } \sigma_{\text{min}} < |\sigma| < \sigma_{\text{max}} \\ 0, & \text{if } |\sigma| < \sigma_{\text{min}} \end{cases} \tag{2.37}$$

where

$$h(\sigma) = \frac{1}{2} + \frac{1}{2}sin\left(\frac{\pi}{(\sigma_{\text{max}} - \sigma_{\text{min}})}(|\sigma| - \sigma_{\text{min}}) - \frac{\pi}{2}\right). \tag{2.38}$$

σ_{max} and σ_{min} are defined by the user. σ_{min} should be high enough to not produce high torques in the computation of the task. The definition of these two parameters depend on the task and the robot. This problem will not be discussed in this thesis. Using eq. 2.33 with eq. 2.34 and 2.36, high values during the computation of the inverse of $\hat{\mathbf{\Lambda}}_i^{-1}$ are avoided. Singularities can also be avoided by using a different prioritized task scheme based on quadratic programming.

2.3 Task hierarchy based on Quadratic Programming (QP)

The redundancy can be formulated as an optimization problem and solved via dedicated solvers. The evolution of hardware and new fast algorithms to solve quadratic problems facilitates its implementation in robotics [DSBS09, KLW11, EMW14, HLM+18, QA19]. Tasks are formulated as objective functions that are minimized depending on relative importance. For a single task, the dynamic control can be formulated as a least square problem in terms of joint accelerations and torques [DPNMN15]

$$\min_{\ddot{\boldsymbol{q}},\boldsymbol{\tau}} \quad \|\boldsymbol{J}_{\mathrm{t}}\ddot{\boldsymbol{q}} - (\ddot{\boldsymbol{x}}_{\mathrm{t}} - \dot{\boldsymbol{J}}_{\mathrm{t}}\dot{\boldsymbol{q}})\|^2 + w\|\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_{\mathrm{d}}\|^2 \tag{2.39a}$$

$$\text{subject to} \quad \boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\nu} + \boldsymbol{g}. \tag{2.39b}$$

The constraint 2.39b keeps the dynamic consistency between the joint torques $\boldsymbol{\tau} \in \mathbb{R}^{n\times 1}$ and the joint accelerations $\ddot{\boldsymbol{q}} \in \mathbb{R}^{n\times 1}$. The term w is the weight of a second task that regularizes the accelerations in the null-space to get a unique solution. This task is usually defined to keep a desired joint configuration. However, in this thesis, it is desired that external forces (e.g., exerted by the human) generate null-space motions which can reconfigure the robot to a more comfortable one, e.g., far from joint limits. Therefore, this task is defined as a joint space damper $\ddot{\boldsymbol{q}}_d = -\boldsymbol{D}\dot{\boldsymbol{q}}$, where $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is the positive-definite damping matrix. The choice of this regularization term is deeply discussed in chapter 6.

The objective function in eq. 2.39a is sufficient to compute the command torques or accelerations while minimizing the operational task error. Nevertheless, to achieve the desired dynamic behavior established in eq. 2.1, the external forces must be considered. If the desired acceleration $\ddot{\boldsymbol{x}}_{\mathrm{t}}$ is computed as in eq. 2.2, the quadratic problem becomes

$$\min_{\ddot{\boldsymbol{q}},\boldsymbol{\tau}} \quad \|\boldsymbol{J}_{\mathrm{t}}\ddot{\boldsymbol{q}} - (\ddot{\boldsymbol{x}}_{\mathrm{t}} - \dot{\boldsymbol{J}}_{\mathrm{t}}\dot{\boldsymbol{q}} - \boldsymbol{\Lambda}_{\mathrm{t}}^{-1}\boldsymbol{f}_{\mathrm{ext}})\|^2 + w\|\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_{\mathrm{d}}\|^2 \tag{2.40a}$$

$$\text{subject to} \quad \boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\nu} + \boldsymbol{g}. \tag{2.40b}$$

Equation 2.40 is the general form of the impedance control of eq. 2.11 using quadratic formulation. Any of the methods in section 2.1.1 or 2.1.2 can be implemented by choosing the respective desired inertia matrix $\boldsymbol{\Lambda}_{\mathrm{t,d}}$. The result of not shaping the inertia ($\boldsymbol{\Lambda}_{\mathrm{t,d}} = \boldsymbol{\Lambda}_{\mathrm{t}}$) computes the same result as the Cartesian impedance control based on QP proposed in [LTP16] and [HLM+18]. Pure force control can be also implemented by setting the desired acceleration according to a desired force, $\ddot{\boldsymbol{x}}_{\mathrm{t}} = \boldsymbol{\Lambda}_{\mathrm{t}}^{-1}\boldsymbol{f}_{\mathrm{d}}$ as stated in [LTP16]. Note that no inverse of the Jacobian must be computed despite the inclusion of the external forces in the operational space. Additionally, the external forces can be expressed in the joint space by replacing $\boldsymbol{\Lambda}_{\mathrm{t}}^{-1}\boldsymbol{f}_{\mathrm{ext}}$ with $\boldsymbol{J}\boldsymbol{M}_{\mathrm{t}}^{-1}\boldsymbol{\tau}_{\mathrm{ext}}$. This brings the advantage of avoiding the measure of external forces in the space of each task.

Including more tasks can be achieved softly by adding more minimization functions with different weights as proposed in [CMAL07, LME+11, SPB11, BK11], or strictly by solving multiple QP problems as proposed in [HLM+18].

2.3.1 Soft hierarchy using weights

In a soft hierarchy scheme, every task is assigned with a weighted objective and then added in to be minimized together as a single quadratic problem. The weight of each task defines its importance, however, there is no guarantee that a low weighted task will not intervene with a higher one. The minimization function becomes

$$\min_{\ddot{\boldsymbol{q}},\boldsymbol{\tau}} \quad \sum_{i=1}^{k} w_i \|\boldsymbol{J}_i\ddot{\boldsymbol{q}} - (\ddot{\boldsymbol{x}}_i - \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}} - \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\tau}_{\text{ext}})\|^2 \tag{2.41a}$$

$$\text{subject to} \quad \boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\nu} + \boldsymbol{g}, \tag{2.41b}$$

with the weight $w_i < w_{i-1}$. The higher is the ratio between two weights, the stricter is the hierarchy. For a very high ratio, though, numerical errors can be produced in the solution. This approach is known in the literature as the *Weighted Sum Method*. An overview and drawbacks are presented in [DD97] and [Yan14]. To have a strict hierarchy, multiples QP problems can be solved as it is shown in the following section. The least important task must be a joint space task to regularize the acceleration in the null-space and get a unique solution. For instance, the second task given in eq. 2.40a becomes the last task k in this multitask formulation. In that case, $\boldsymbol{J}_k = \boldsymbol{I}$, and $\ddot{\boldsymbol{x}}_k$ becomes $\ddot{\boldsymbol{q}}_{\text{d}} = -\boldsymbol{D}\dot{\boldsymbol{q}}$.

2.3.2 Strict hierarchy solving multiple QP problems

In the strict hierarchy scheme, the number of QP problems depends on the number of tasks involved. For k tasks, k-QP problems are solved, which increases the computation time. Every task is described by a main objective function subject to constraints and then included in the next task as an equality constraint. This scheme brings the advantage of not choosing any weights. The optimization problem becomes:

$$\min_{\ddot{\boldsymbol{q}}_i,\boldsymbol{\tau}_i} \quad \|\boldsymbol{J}_i\ddot{\boldsymbol{q}}_i - (\ddot{\boldsymbol{x}}_i - \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}} - \boldsymbol{J}\boldsymbol{M}_{\text{t}}^{-1}\boldsymbol{\tau}_{\text{ext}})\|^2 \tag{2.42a}$$

$$\text{subject to} \quad \boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\nu} + \boldsymbol{g} \tag{2.42b}$$

$$\boldsymbol{J}_{i-1}\ddot{\boldsymbol{q}}_{i-1} = \boldsymbol{J}_{i-1}\ddot{\boldsymbol{q}}_i \tag{2.42c}$$

$$\vdots \tag{2.42d}$$

$$\boldsymbol{J}_1\ddot{\boldsymbol{q}}_1 = \boldsymbol{J}_1\ddot{\boldsymbol{q}}_i. \tag{2.42e}$$

The least important task must be a joint space task to regularize the acceleration in the null-space and get a unique solution as previously explained in section 2.3.1.

2.4 Mixed task hierarchy

A new approach that mixes the two hierarchies schemes has been proposed in [LTP16]. The Generalized Hierarchical Control (GHC) describes the tasks as a minimization problem, and solves the hierarchy using projectors in the constraints of the problem.

$$\min_{\ddot{\boldsymbol{q}}^{\text{aug}},\boldsymbol{\tau}_i} \quad \sum_{i=1}^{k} \|\boldsymbol{J}_i\ddot{\boldsymbol{q}}_i - (\ddot{\boldsymbol{x}}_i - \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}} - \boldsymbol{J}\boldsymbol{M}_{\text{t}}^{-1}\boldsymbol{\tau}_{\text{ext}})\|^2 \tag{2.43a}$$

$$\text{subject to} \quad \boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\nu} + \boldsymbol{g} \tag{2.43b}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{N}\ddot{\boldsymbol{q}}^{\text{aug}} = \sum_{i=1}^{k} \boldsymbol{N}_{\text{i}}\ddot{\boldsymbol{q}}_i, \tag{2.43c}$$

with $\ddot{\boldsymbol{q}}^{\text{aug}} = \begin{bmatrix} \ddot{\boldsymbol{q}}_1 \\ \vdots \\ \ddot{\boldsymbol{q}}_k \end{bmatrix}$ and $\boldsymbol{N} = [\boldsymbol{N}_1 \dots \boldsymbol{N}_k]$.

This formulation gives the prioritization via null-space projectors. More specifically, the GHC approach in [LTP16] employs statically consistent projectors[9] with a priority matrix to allow change of priorities or insertion and deletion of tasks.[10] The approach is improved in [DS19] to use dynamically consistent projectors. For this thesis, the mixed hierarchy uses only dynamically consistent projectors. They can be computed successively as in eq. 2.23 or augmented as in eq. 2.26. Successive projectors lead to soft hierarchy, while augmented projectors lead to a strict hierarchy. The advantages of this method are: the possibility to use the priority matrix, and the minimization of the kinetic energy in the projection by using dynamic consistent projectors, as shown in [Kha87b]. The drawback is high computational effort required to solve the problem. The computation of the projectors and the use of high-dimensional matrices in the minimization problem makes the method slower than all the previously presented schemes.

2.5 Discussion

This chapter presented two special cases of force control. While the OSC uses inertia shaping, making the operational point behave as a unit mass point, the IWIS uses the natural inertia of the robot avoiding the measurements of external forces. The OSC can be seen as a controller at acceleration level if external forces are ignored, while IWIS controls directly the force.

[9] The statically consistent projectors use a different weighting matrix than the mass matrix. A detailed explanation is provided in [DOAS15b].

[10] A complete explanation of how to compute the priority matrix and how it is included in the projector can be found in [LTP16] and [DS19].

To have a hierarchical stack of tasks, three methods were presented: projector-based, QP-based and a mix of both. Each of them manages the hierarchy in two different ways: soft and strict. Table 2.1 summarizes the advantages and disadvantages of the architectures. The chosen type of hierarchy is directly related to the requirements specified in section 1.3.2 and 1.3.3. A soft hierarchy does not guarantee decoupling of tasks, i.e., task with low priority could interfere with higher-priority tasks, violating requirement 2.1. A strict hierarchy guarantees the decoupling of tasks, fulfilling this requirement. However, to have a strict hierarchy, a higher computational effort is required. The QP-based strict hierarchy solves as many QP problems as tasks. The projector-based and the mixed strict hierarchy require the computation of the augmented Jacobian and its inverse for each task. This inverse not only increases the computation time, but also requires a singularity avoidance strategy in case of complete task conflicts.[11] These singularities arise when the augmented Jacobian $\boldsymbol{J}_{\mathrm{t}}^{\mathrm{aug}}$ loses rank. It is worth to mention that the projector-based soft hierarchy requires a singular avoidance strategy, but only for singularities encountered in a specific task[12] and not when a conflict between tasks occurs.

The implementation of the stack of constraints varies from the one of the stack of tasks. On the one hand, constraints can be included easily in QP-based techniques. On the other hand, the projector-based scheme does not allow this inclusion straightforward. One can argue that to include constraints in the projector-based techniques, they can be defined as tasks. Then, if a conflict occurs between them, a flag could be triggered to show that the problem is unfeasible. This is how QP solvers are implemented. This approach works only for equality constraints. Inequality constraints can not be included in the stack of constraints for projector-based approach in the same way. As stated in the introduction, the inclusion of inequalities requires: smooth activation/deactivation and a proper parametrisation of the dynamic behavior (specially to guarantee the fulfillment of requirement 4.1).

Besides, there are constraints at different kinematic or dynamic levels. For instance, due to safety measures, not only position but also velocity and acceleration limits must be guaranteed.[13] Furthermore, different than tasks, constraints can be formulated as inequalities. The next chapter proposes a method to include inequalities at different levels in projector-based approaches. The method handles the challenges addressed before: avoiding the need of parametrisation, and the need of a smooth task activation/deactivation, while preserving a smooth physical interaction with the human.

Despite the clear advantages of the QP-based over the projector-based approaches, projector-based techniques are still being studied. The main reason is that stability of these techniques has been proven (using IWIS in [DOAS13], and more recently using OSC for the regulation case in

[11] If there is not null-space left in which the task with lower priority could be partially executed, the lower priority task has to be dropped [Die16].

[12] The task Jacobian $\boldsymbol{J}_{\mathrm{t}}$ loses rank.

[13] This is also the case for tasks. A task can be specified to track a velocity and not only a position profile.

[DOP18]). A stability analysis for the QP-based approach has not been conducted. Besides, these approaches have not been properly analyzed under pHRI. Chapter 6 offers a comparison of the approaches under pHRI. Simulations demonstrate that QP-based approaches shown in section 2.3 are not well suited for pHRI.

Table 2.1: Comparison of task hierarchy solvers

Architecture		**Null-space Projectors based**	**Mixed architecture (GHC)**	**Qp based**
Advantages	Soft Hierarchy	Fast computation compared to strict hierarchy		
		No numerical singularities due to task conflicts		
	Strict Hierarchy	Fulfillment of requirement 2.1. Full decoupling between tasks		
	Independently on the hierarchy type	Stability has been proved	Easy inclusion of inequality constraints	
				No need of a singularity avoidance approach
Disadvantages	Soft Hierarchy	Requirement 2.1 is not completely fulfilled. Interference between tasks		
				Need to choose the correct weight value for each task
	Strict Hierarchy	High computational effort in comparison with soft hierarchy		
		Numerical singularities appear due to task conflicts		
	Independently on the hierarchy type	No direct inclusion of inequality constraints	Stability proof has not been conducted. Computation time strongly depends on the used solver and the optimization problem formulation	

3 Unilateral constraints in joint space

The use of force/torque control with kinematic redundancy brings many challenges especially in industrial applications. One of the biggest issues is how to deal with motion constraints in joint space. For instance in human-robot collaboration: how can a controller deal with the fact that the robot is pushed close to a joint limit while it is performing a task? In other words, how can a controller ensure that a task (e.g., pick-and-place) is constantly being pursued, while the user can interact with the compliant robot and joint limits are avoided? One intuitive answer is to implement two tasks: one being the avoidance of joint limits, and the other one being pick-and-place. This control scheme can be achieved by stack-of-task techniques [SS91, MKK09, SK05, SK06, ODAS15]. However, three major challenges remain:

- smooth vs. hard: for torque-controlled robots, tasks have to be formulated in terms of forces (or joint torques). The control programmer must specify a force function (called potential field) that reliably pushes a joint[1] away from its limit. It is very difficult, though, to correctly specify the strength of this potential field: the required repulsion torques strongly depend on the robot state (joint configuration and velocity), attached tools that alter the robot dynamics, and the magnitude of the force applied by the user.
- smooth task activation: the joint limit avoidance task must receive top priority. Nevertheless, it must not continuously produce a torque but can only be switched on when the joint comes close to its limit. Switching the top-priority task on and off leads stack-of-task techniques to discontinuous solutions and oscillations.
- multi-level constraints: hardware joints do not only have limits to the joint position, but also to the joint velocity and acceleration. If these different limits are implemented in a stack of task, one of them may be sacrificed for another one.

The first two challenges are particularly important for torque-controlled robots. If they are not properly dealt with, the robot will either react too weak (joint limit not avoided, robot shuts down), too hard (high torque peaks), or with jerky behavior and oscillations that prevent the task from being completed.

Techniques to achieve smoother task activation, e.g., by a continuous activation function, are described in [MKK09, LMP12, HLP12, HP13]. However, their functionality depends strongly on chosen parameter values that describe when and how fast a task is switched on or off. The problem

[1] Note that if the limits are given in the Cartesian space, the potential field must push a point of interest away from its limit. See chapter 4.

of parameter-dependency is not solved but just shifted to another set of values. More recently, [LTP16, SC16] proposed approaches for smooth activation/deactivation and changing of priority of tasks. The approach in [LTP16] was extended to the dynamic case in [DS19] to avoid possible unstable robot motion. The approaches proved to work stable with different parametrisation of the sigmoid function. Though, they did not overcome the first and third challenge and were tested only in simulation.

Other techniques take a different approach, e.g. quadratic programming (QP) [EMW14, HLM$^+$18, QA19]. This technique has the advantage of including inequalities as hard constraints without the need of activation or deactivation techniques. [FDK12, FDK15, FL13] link QP to the stack-of-task approach: joint position, velocity, and acceleration limits are implemented as hard constraints and the cost function is replaced by a stack of tasks. While this procedure overcomes all of the three challenges listed above, its limitation is that it can only be applied to velocity-controlled robots, while collaborative robots are usually torque-controlled. Furthermore, none of the above mentioned methods was proven to work during physical human-robot interaction.

This chapter demonstrates the drawback of using potential fields to avoid limits, and proposes two novel methods to include joint limit avoidance. The first method proposes a variable potential field that reduces the kinetic energy in the border of the limit. The idea is to reach the limit with zero speed to avoid oscillations. The second method extends the method proposed in [FDK12] for torque controlled robots. The repulsive force is computed based on the robot's state avoiding the setting of parameters.

For simplicity, the inclusion of inequalities is evaluated using the soft hierarchy based on successive projectors (see section 2.2.1). Assume the result of a task hierarchy computation from the stack of tasks results in a torque command $\boldsymbol{\tau}_{\text{stack}}$. With this torque, the state of the robot (joint positions $\boldsymbol{q}$ and velocities $\dot{\boldsymbol{q}}$) and the dynamic data ($\boldsymbol{M}$, $\boldsymbol{g}$ and $\boldsymbol{\nu}$), it is possible to compute a torque $\boldsymbol{\tau}_{\text{cmd}}$ that considers the joint limit avoidance. Figure 3.1 shows a block representation of this process. The methods previously mentioned to avoid joint limits are different approaches to compute the block "Include Constraints".

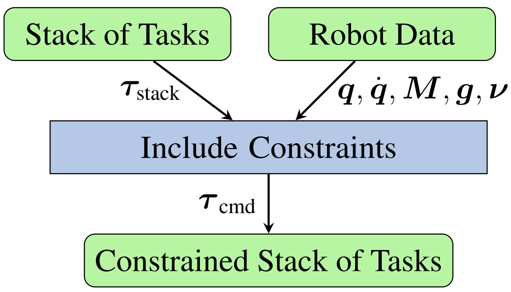


Figure 3.1: Addition of constraints in the stack of tasks

Note that in the first two levels of the priority-hierarchy, successive or augmented projectors are idempotent (see section 2.2). Therefore, including a new task in the highest priority to accomplish the inequalities is not influenced by lower priority tasks, no matter which kind of projectors are used. Large parts of this chapter have already been publish in [MFA18] and [MAF19b]

3.1 Potential field approach

A common implemented approach to avoid joint limits is the creation of a potential field in the proximity to the limit. Given the joint coordinate q_j with its respective upper $\bar{q}_j$ and lower $\underline{q}_j$ physical limit, where the sub-index $j = 1..n$ indicates the individual joint. Barriers of potential are created at each of these limits $q_j = \underline{q}_j$ and $q_j = \bar{q}_j$ using the potential field approach and the FIRAS function, as defined in [Kha85]. Let $\underline{\rho}_{j(0)}$ and $\bar{\rho}_{j(0)}$ be the distance limits of the potential field influence and the distances $\underline{\rho}_j$ and $\bar{\rho}_j$ be defined by:

$$\begin{aligned} \underline{\rho}_j &= q_j - \underline{q}_j, \\ \bar{\rho}_j &= \bar{q}_j - q_j. \end{aligned} \tag{3.1}$$

A critical zone is defined between the joint limits ($\underline{q}_j$ and $\bar{q}_j$) and the distance limit of the potential field influence ($\underline{\rho}_{j(0)}$ and $\bar{\rho}_{j(0)}$). The corresponding joint torques $\gamma_{\underline{q}_j}$ and $\gamma_{\bar{q}_j}$ are:

$$\gamma_{\underline{q}_j} = \begin{cases} \eta \left(\frac{1}{\underline{\rho}_j} - \frac{1}{\underline{\rho}_{j(0)}} \right) \frac{1}{\underline{\rho}_j^2}, & \text{if } \underline{\rho}_j \leq \underline{\rho}_{j(0)} \\ 0, & \text{if } \underline{\rho}_j > \underline{\rho}_{j(0)} \end{cases} \tag{3.2}$$

and

$$\gamma_{\bar{q}_j} = \begin{cases} -\eta \left(\frac{1}{\bar{\rho}_j} - \frac{1}{\bar{\rho}_{j(0)}} \right) \frac{1}{\bar{\rho}_j^2}, & \text{if } \bar{\rho}_i \leq \bar{\rho}_{j(0)} \\ 0, & \text{if } \bar{\rho}_j > \bar{\rho}_{j(0)} \end{cases}, \tag{3.3}$$

where η is a constant parameter. An example of the function is plotted in fig. 3.2 where the influence of the gain η is shown. The joint limit $\underline{q}_j$ is set to zero in this example. A higher gain implies a faster increment of the torque when the joint is reaching its limit.

The general joint torque that pulls the joint out of the critical zone, $\boldsymbol{f}_{\text{FIRAS}}$, is given by:

$$\boldsymbol{f}_{\text{FIRAS}} = \boldsymbol{\gamma}_{\bar{\boldsymbol{q}}} + \boldsymbol{\gamma}_{\underline{\boldsymbol{q}}}. \tag{3.4}$$

The "Include Constraints" block shown in fig. 3.3 starts by checking the current position of the joints relative to the barriers of potential. If any joint is in the critical zone, $\boldsymbol{f}_{\text{FIRAS}}$ is computed according to eq. 3.2, 3.3 and 3.4.

The Jacobian of the joint-limit avoidance-task $\boldsymbol{J}_{\text{lim}}$ is the mapping between the joint space and a space of the joints under potential field influence. The columns of the Jacobian represent the joints

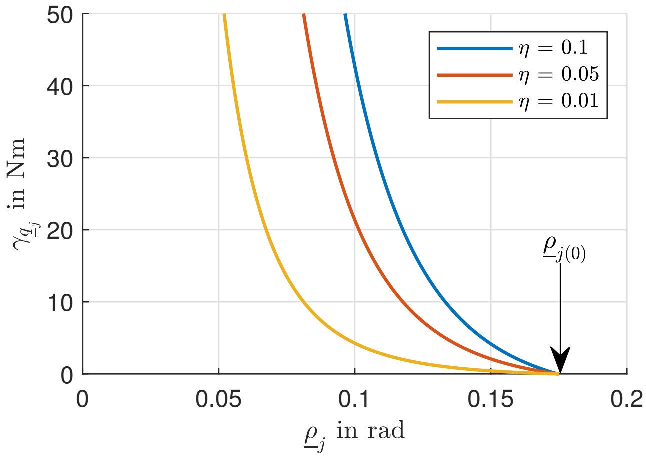


Figure 3.2: Example of the influence of η in the FIRAS function

of the robot, and the number of rows is given by the number of joints that are pulled by the force $\boldsymbol{f}_{\text{FIRAS}}$. For instance, if the joint q_2 is achieving its position limit for a 7 DOF robot, a repulsive torque is built to prevent the joint reaching its limit. Thus, the Jacobian is defined as:

$$\boldsymbol{J}_{\text{lim}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.5}$$

The column index value 1 indicates for which joint the force is mapped. If the torque has to be mapped for two joints, e.g., q_7 and q_2 are coming to their limits, the Jacobian becomes:

$$\boldsymbol{J}_{\text{lim}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.6}$$

The joint torques $\boldsymbol{\tau}_{\text{lim}}$ and the null-space projector $\boldsymbol{N}_{\text{lim}}$ are then computed following the force control equations in section 2.1 with $\boldsymbol{J}_{\text{t}} = \boldsymbol{J}_{\text{lim}}$.

Finally, $\boldsymbol{\tau}_{\text{stack}}$ is added to the torque that keeps the joints within their bounds $\boldsymbol{\tau}_{\text{lim}}$ to produce the constrained torque vector $\boldsymbol{\tau}_{\text{cmd}} = \boldsymbol{\tau}_{\text{lim}} + \boldsymbol{N}_{\text{lim}}\boldsymbol{\tau}_{\text{stack}}$. The projector $\boldsymbol{N}_{\text{lim}}$ guarantees that $\boldsymbol{\tau}_{\text{stack}}$ does not interfere with $\boldsymbol{\tau}_{\text{lim}}$. Note that if the joints are outside of the critical zone, the projector $\boldsymbol{N}_{\text{lim}}$ is the identity matrix and the forces $\boldsymbol{\tau}_{\text{lim}}$ are zero. In that case, $\boldsymbol{\tau}_{\text{cmd}}$ is equal to $\boldsymbol{\tau}_{\text{stack}}$.

However, the behavior of the robot under the influence of the potential field strongly depends on the setting of the gain η. Undesired oscillations can be produced by the sudden insertion of the task, when the potential field is activated. Such oscillations can make the robot lose the task, and in rare cases, they could damage the robot hardware. Therefore, an approach to avoid these oscillations is proposed in the next section.

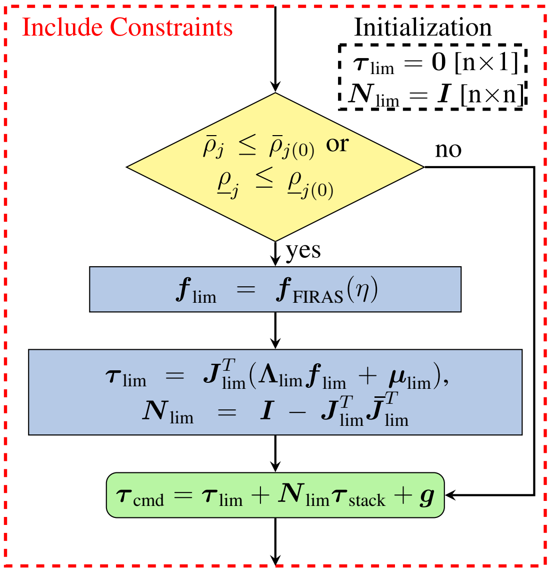


Figure 3.3: Flowchart of the Include Constraints block using the potential field approach proposed by [Kha85]

3.2 Potential field approach with variable field force

A potential field with a variable gain can help to reduce oscillations. The idea is to compute the parameter in such way that the kinetic energy is reduced to reach the outer limit of the critical zone with zero speed.

A second potential field is created with variable η that reduces the kinetic energy E_{kin} to zero in the outer limit of FIRAS potential field ($\underline{q}_j + \underline{\rho}_{j(0)}$ or $\bar{q}_j + \bar{\rho}_{j(0)}$). Parting from the principle that the change of the kinetic energy of an object when it moves from one point to another is equal to the work produced by all forces on it [Tom20], the kinetic energy can be computed as the integral of the work produced by the force $\boldsymbol{f}_{\text{FIRAS}}$:

$$\frac{1}{2}\Lambda_j \dot{\rho}_j^2 = -\int_{\rho_j}^{\tilde{\rho}_j} f_{\text{FIRAS}}(\rho_j) d\rho_j, \tag{3.7}$$

$$E_{\text{kin}}(\rho_j) = \int_{\rho_j}^{\tilde{\rho}_j} f_{\text{FIRAS}}(\rho_j) d\rho_j = -\frac{1}{2}\eta\left(\frac{1}{\rho_j} - \frac{1}{\bar{\rho}_{j(0)}}\right)^2 \Bigg|_{\rho_j=\bar{\rho}_{j(0)}}^{\tilde{\rho}_j} = -\frac{1}{2}\eta\left(\frac{1}{\tilde{\rho}_j} - \frac{1}{\bar{\rho}_{j(0)}}\right)^2, \tag{3.8}$$

from eq. 3.7:

$$\frac{1}{2}\Lambda_j \dot{\rho}_j^2 = \frac{1}{2}\eta\left(\frac{1}{\tilde{\rho}_j} - \frac{1}{\bar{\rho}_{j(0)}}\right)^2, \tag{3.9}$$

which means:

$$\eta = \frac{\Lambda_j \dot{\rho}_j^2}{\left(\frac{1}{\tilde{\rho}_j} - \frac{1}{\bar{\rho}_{j(0)}}\right)^2}, \tag{3.10}$$

where $\tilde{\rho}_j = \bar{q}_j + \bar{\rho}_{j(0)} - q_j$. The value $\tilde{\rho}_{j(0)}$ is a new threshold that defines a damping zone (see fig. 3.4). The mass inertia matrix of the joint in this zone Λ_j is computed according to eq. 2.6.

The potential field in the damping zone is active only if the joint goes towards its limit. In this way, if other tasks produce a force to move the joint out of the damping zone, the joint moves to accomplish the task. The torque $\tau_{\text{lim},j}$ reduces the velocity of joint j, so that the outer limit of the critical zone is reached with zero speed. If the joint enters the critical zone due to external forces, a FIRAS potential field is activated producing a counteracting force as in classical FIRAS.

The "Include Constraints" block is modified to have two different zones of potential influence: critical zone and damping zone (see fig. 3.4). The modification of the block is presented in fig. 3.5. If the joint is in the critical zone, the algorithm works as in fig. 3.3. If the joint position is in the damping zone and the velocity is in direction to the joint limit (the joint is moving towards its limit), $\boldsymbol{f}_{\text{FIRAS}}$ is computed with a variable gain η that depends on eq. 3.10. Otherwise, $\boldsymbol{\tau}_{\text{cmd}}$ will be equal to the output of the stack of tasks $\boldsymbol{\tau}_{\text{cmd}} = \boldsymbol{\tau}_{\text{stack}}$.

The FIRAS potential field in the critical zone requires a suitable choice of η. However, the influence of a proper setting is reduced, because the kinetic energy is reduced before the influence of this potential field. Thus, a high value η can be set to ensure that the joint does not go into this zone.

A novel approach for computing the force $\boldsymbol{f}_{\text{lim}}$ is proposed in the next section. The force ensures that the joints stay within their limits, by saturating the joint accelerations.

3.3 Saturation in Joint Space method - SJS

A novel and efficient approach to handle joint constraints concerning the differential kinematics problem for redundant manipulators was presented in [FDK15]. The Saturation in the Null-Space (SNS) algorithm successively saturates the joints that would exceed their limits and it discards these joints for the computation of the main task. The method was first defined in the velocity level and later moved to acceleration level. In this section, the method is modified to be implemented at the generalized force control level within the Operational Space Formulation.

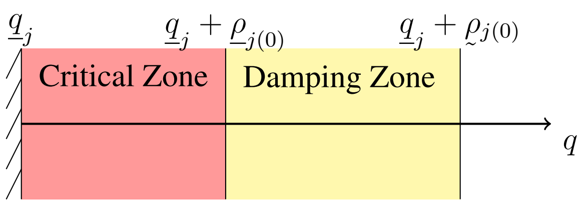


Figure 3.4: Definition of zones under a potential field influence

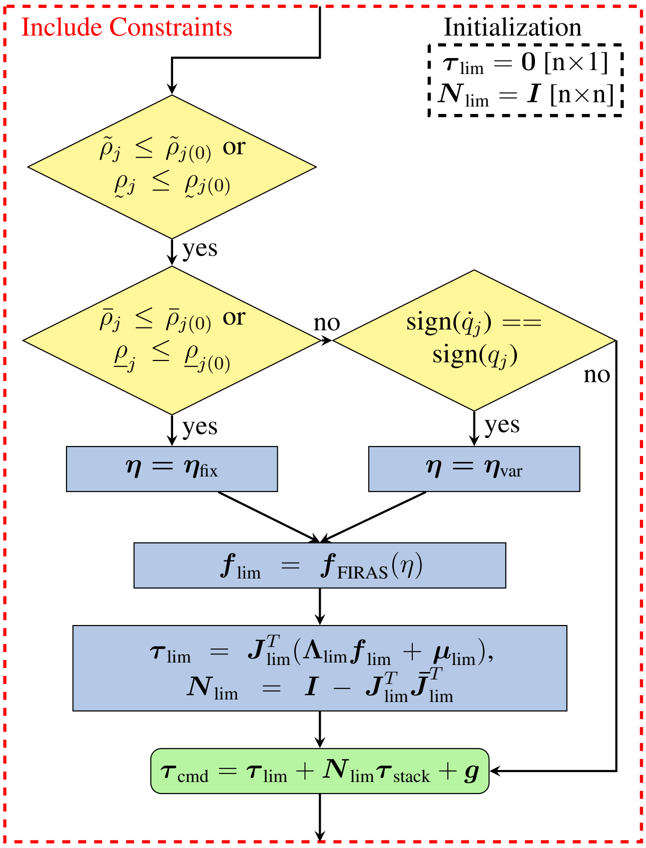


Figure 3.5: Flowchart of the Include Constraints block using the potential field approach with variable η

Consider the following limits on the joint ranges, joint velocities and joint accelerations:

$$\begin{aligned} \boldsymbol{q}_{\text{min}} &\leq \boldsymbol{q} \leq \boldsymbol{q}_{\text{max}}, \\ \boldsymbol{v}_{\text{min}} &\leq \dot{\boldsymbol{q}} \leq \boldsymbol{v}_{\text{max}}, \\ \boldsymbol{a}_{\text{min}} &\leq \ddot{\boldsymbol{q}} \leq \boldsymbol{a}_{\text{max}}. \end{aligned} \tag{3.11}$$

The joint position limits $\boldsymbol{q}_{\text{min}}$ and $\boldsymbol{q}_{\text{max}}$ are virtual limits that should be set in a safe distance to the physical joint limits of the robot. Otherwise, external forces could drive the joints into its physical joint limits. The safe distance could be given by the critical zone defined in section 3.1. Thus, $\boldsymbol{q}_{\text{max}} = \bar{\boldsymbol{q}} - \bar{\rho}_{j(0)}$ and $\boldsymbol{q}_{\text{min}} = \underline{\boldsymbol{q}} + \underline{\rho}_{j(0)}$, where $\bar{\boldsymbol{q}}$ and $\underline{\boldsymbol{q}}$ are the physical joint limits. The acceleration limits can be given from pure kinematic reasoning as $\boldsymbol{a}_{\text{min}} = -\boldsymbol{a}_{\text{max}}$. Nevertheless, the need of a numeric differentiation would lead to a very noisy result. Alternatively, the authors in [FDK15, FDK12] suggest to compute them from the dynamic model and the maximum torques $\boldsymbol{\tau}_{\text{max}}$. However, this computation does not guarantee limitation of the joint torques, unless all joint torques are saturated at once (unreal situation) or there is no coupling between joints i.e. mass matrix is a diagonal matrix (rare case in robotics). An explanatory example of this issue can be found in appendix A.2.

Consider a robot with n joints performing a single task $\boldsymbol{f}_\mathrm{t}$ with dimension m. In the Operational Space framework, the command vector of the task is defined as a desired operational space acceleration $\ddot{x}_\mathrm{t}$. If the task is defined in the joint space, the command vector can be set to a desired joint space acceleration $\ddot{\boldsymbol{q}}_\mathrm{sat}$ that breaks the joint motion to saturate the position, velocity and acceleration to their limits.

In a discrete implementation, the command joint torques $\boldsymbol{\tau}_\mathrm{stack}$ are kept constant during the time step h at the computed value $\boldsymbol{\tau}_{\mathrm{stack},h}$. That torque vector $\boldsymbol{\tau}_{\mathrm{stack},h}$ causes a constant joint acceleration $\ddot{\boldsymbol{q}}_h = \ddot{\boldsymbol{q}}(t_h)$ for a sample time δt. Assuming that at $t_\mathrm{h} = h\delta t$ the current joint position $\boldsymbol{q} = \boldsymbol{q}_h$ and velocity $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_h$ are both feasible. The next joint position and velocity can be estimated as:

$$\dot{\boldsymbol{q}}_{h+1} \simeq \dot{\boldsymbol{q}}_h + \ddot{\boldsymbol{q}}\delta t,\ \boldsymbol{q}_{h+1} \simeq \boldsymbol{q}_h + \dot{\boldsymbol{q}}_h\delta t + \frac{1}{2}\ddot{\boldsymbol{q}}\delta t^2. \tag{3.12}$$

If the acceleration generated by the command torques keeps the position and the velocity within their limits, the next constraints will be given by the inequalities:

$$\frac{\boldsymbol{v}_\mathrm{min} - \dot{\boldsymbol{q}}_\mathrm{h}}{\delta t} \leq \ddot{\boldsymbol{q}} \leq \frac{\boldsymbol{v}_\mathrm{max} - \dot{\boldsymbol{q}}_h}{\delta t}, \tag{3.13}$$

and

$$\frac{2(\boldsymbol{q}_\mathrm{min} - \boldsymbol{q}_h - \dot{\boldsymbol{q}}_h\delta t)}{\delta t^2} \leq \ddot{\boldsymbol{q}} \leq \frac{2(\boldsymbol{q}_\mathrm{max} - \boldsymbol{q}_h - \dot{\boldsymbol{q}}_h\delta t)}{\delta t^2}. \tag{3.14}$$

Considering these constraints and those given by the inequalities in eq. 3.11, a box constraint for the acceleration $\ddot{\boldsymbol{q}}$ at time t_h is obtained (see also [FDK12]).

$$\ddot{\boldsymbol{q}}_\mathrm{min}(t_h) \leq \ddot{\boldsymbol{q}} \leq \ddot{\boldsymbol{q}}_\mathrm{max}(t_h). \tag{3.15}$$

However, this implementation implies a high reduction of the velocity in one instant of time. If the point of interest reaches its joint position limit while having the maximal velocity, a high deceleration will be required. Therefore, the velocity limits $\boldsymbol{v}_\mathrm{max}$ and $\boldsymbol{v}_\mathrm{min}$ in eq. 3.13 are replaced by the variable limits $\dot{\boldsymbol{q}}_\mathrm{max}$ and $\dot{\boldsymbol{q}}_\mathrm{min}$, which reduce the maximal allowed velocity in the proximity to the position limits given the maximal acceleration.[2] This change also considers the controller in discrete time, as shown in [RPB+12] and [Del18].

$$\dot{\boldsymbol{q}}_\mathrm{max} = \min(\boldsymbol{v}_\mathrm{max}, \sqrt{2\boldsymbol{a}_\mathrm{max}(\boldsymbol{q}_\mathrm{max} - \boldsymbol{q}_h)}) \tag{3.16}$$

$$\dot{\boldsymbol{q}}_\mathrm{min} = \max(\boldsymbol{v}_\mathrm{min}, -\sqrt{2\boldsymbol{a}_\mathrm{max}(\boldsymbol{q}_h - \boldsymbol{q}_\mathrm{min})}) \tag{3.17}$$

Figure 3.6 shows an example plot of the smooth functions in eq. 3.16 and 3.17. For instance,

[2] Note that the dot in $\dot{\boldsymbol{q}}_\mathrm{max}$ and $\dot{\boldsymbol{q}}_\mathrm{min}$ denotes in this context a change of the variable over time, and it does not follow the Newton's notation for derivatives (as they are not time derivatives of $\boldsymbol{q}_\mathrm{max}$ and $\boldsymbol{q}_\mathrm{min}$).

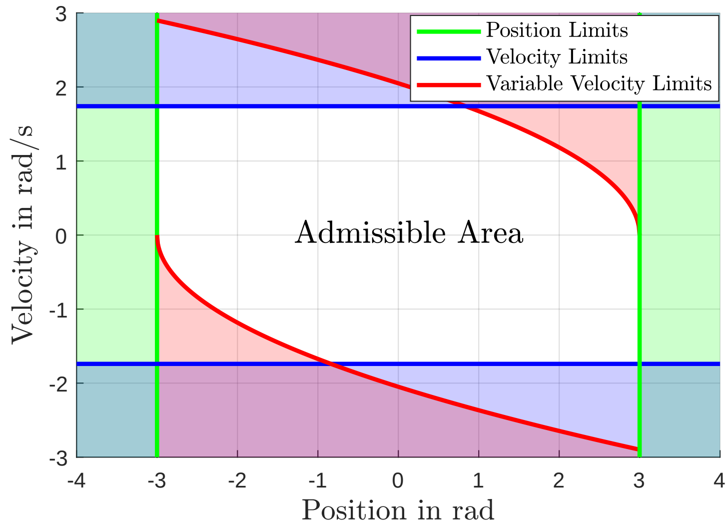


Figure 3.6: Shaping of velocity limits example. $q_{\text{max}} = -q_{\text{min}} = 3$ rad, $v_{\text{max}} = -v_{\text{min}} = 1.74$ rad/s, $a_{\text{max}} = -a_{\text{min}} = 1.74$ m/s^2

when the point of interest is coming to its maximal position limit (plotted in green), the maximal velocity gets smoothly reduced by the square root function plotted in red. When the position limit is reached, the maximal velocity allowed is zero. In the other areas, the velocity gets saturated to its maximal or minimal limit (plotted in blue). What remains is an admissible area, in which the point of interest can move with certain velocity.

However, if unknown external forces are applied to the robot (as expected in pHRI), the limits could be exceeded. If this is the case, the root of negative numbers must be computed in eq. 3.16 or 3.17. To avoid this computational inconsistency: if there is such a violation, $\dot{\boldsymbol{q}}_{\text{max}} = \boldsymbol{v}_{\text{max}}$ and $\dot{\boldsymbol{q}}_{\text{min}} = \boldsymbol{v}_{\text{min}}$.

3.3.1 Consideration of external forces

All elements to implement an approach similar to the SNS algorithm in [FDK12] are available. Nevertheless, one of the requirements is that the framework must work under pHRI. Hence, the external forces have to be considered. A previously proposed task hierarchy framework in [LTP16], the generalized hierarchical control, includes the external force term in the main dynamic equation as a constraint and performs force control. That implementation does not allow for interaction. The cancellation of external forces by adding them to the command vector produces zero acceleration, i.e., no motion if the robot is in a static state (an illustrative example is provided in appendix A.3). Another approach to consider external forces and allowing for interaction is proposed in the following paragraphs.

The goal is to keep the joint acceleration between its limits, i.e., eq. 3.15 must always hold. Inverse dynamics can be used to estimate the joint accelerations $\ddot{\boldsymbol{q}}_{\text{e}}$ that are produced by the command joint torques

$$\ddot{\boldsymbol{q}}_{\mathrm{e}} = \boldsymbol{M}^{-1}(\boldsymbol{\tau}_{\mathrm{cmd}} + \boldsymbol{\tau}_{\mathrm{ext}} - \boldsymbol{g} - \boldsymbol{\nu}). \tag{3.18}$$

These joint accelerations must be compared with their limits to enforce a saturation if necessary. If a joint acceleration value $\ddot{q}_{\mathrm{e},j}$ exceeds its bound $\ddot{q}_{\mathrm{max},j}$ or $\ddot{q}_{\mathrm{min},j}$, a joint torque $\boldsymbol{\tau}_{\mathrm{lim}}$ must be computed to generate $\ddot{q}_{\mathrm{max},j}$ or $\ddot{q}_{\mathrm{min},j}$ instead of $\ddot{q}_{\mathrm{e},j}$. A task force is created to produce the desired saturated acceleration vector $\ddot{q}_{\mathrm{sat,j}} = \ddot{q}_{\mathrm{max},j}$ if $\ddot{q}_{\mathrm{e},j} > \ddot{q}_{\mathrm{e},j}$ or $\ddot{q}_{\mathrm{sat,j}} = \ddot{q}_{\mathrm{min},j}$ if $\ddot{q}_{\mathrm{e},j} < \ddot{q}_{\mathrm{e},j}$. The desired dynamic behavior in eq. 2.10 is the saturated joint acceleration. Thus, $\ddot{\boldsymbol{x}}_{\mathrm{t}} = \ddot{\boldsymbol{x}}_{\mathrm{d,lim}} = \ddot{\boldsymbol{q}}_{\mathrm{sat}}$. The command torque that produces $\ddot{\boldsymbol{q}}_{\mathrm{sat}}$ can be computed by using the forward dynamics in the operational space in eq. 2.1

$$\boldsymbol{\tau}_{\mathrm{lim}} = \boldsymbol{J}_{\mathrm{lim}}^{T}(\boldsymbol{\Lambda}_{\mathrm{lim}}\ddot{\boldsymbol{x}}_{\mathrm{d,lim}} + \boldsymbol{\mu}_{\mathrm{lim}} - \boldsymbol{f}_{\mathrm{ext,lim}}). \tag{3.19}$$

3.3.2 The SJS algorithm

The "Include Constraints" block is modified, as shown in fig. 3.7. The null-space projection $\boldsymbol{N}_{\mathrm{lim}}$, the command torques $\boldsymbol{\tau}_{\mathrm{lim}}$ and the saturated acceleration $\ddot{\boldsymbol{q}}_{\mathrm{sat}}$ are initialized, assuming that there is no saturation in the first step. That initialization equalizes the auxiliary torques $\boldsymbol{\tau}_{\mathrm{aux}}$ to $\boldsymbol{\tau}_{\mathrm{stack}}$. Then, the accelerations produced by the auxiliary torques and the external torques, as well as the limits for the acceleration, are computed. If no limit is violated, the output is $\boldsymbol{\tau}_{\mathrm{cmd}} = \boldsymbol{\tau}_{\mathrm{aux}}$. If any joint acceleration violates the limits, it is saturated to its maximal or minimal value. Then, the desired operational acceleration $\ddot{\boldsymbol{x}}_{\mathrm{d,lim}}$ is equalized to the saturated acceleration. Thus, a task to avoid the limits is created. The torques $\boldsymbol{\tau}_{\mathrm{lim}}$ and null-space projector $\boldsymbol{N}_{\mathrm{lim}}$ are computed. A new torque vector $\boldsymbol{\tau}_{\mathrm{aux}}$ is computed, with $\boldsymbol{\tau}_{\mathrm{lim}}$ that ensures saturation of the critical joint and $\boldsymbol{\tau}_{\mathrm{stack}}$ which is projected into the null-space of the task Jacobian $\boldsymbol{J}_{\mathrm{lim}}$. The inclusion of the external forces in the saturated joint space $\boldsymbol{f}_{\mathrm{ext,lim}}$ balances the external forces, which are inserted by the human or the environment. The process repeats to ensure that the new solution is within the limits. The Jacobian $\boldsymbol{J}_{\mathrm{lim}}$ is computed as in section 3.1. In this case, the value one in the Jacobian indicates the saturated joint.

In the next section, experiments are performed to compare the performance of the two new presented methods in comparison to the classical potential field method.

3.4 Experiments and results

The approaches (potential field - FIRAS, variable potential field - FIRAS with η variable and saturation in joint pace - SJS) were tested in simulation and in real environment using the KUKA LBR iiwa. This collaborative robot has seven rotational joints ($n = 7$) and internal force/torque sensors for each joint. Table 3.1 shows the limits for the experiments. A sample time $\delta t = 5\mathrm{ms}$ is

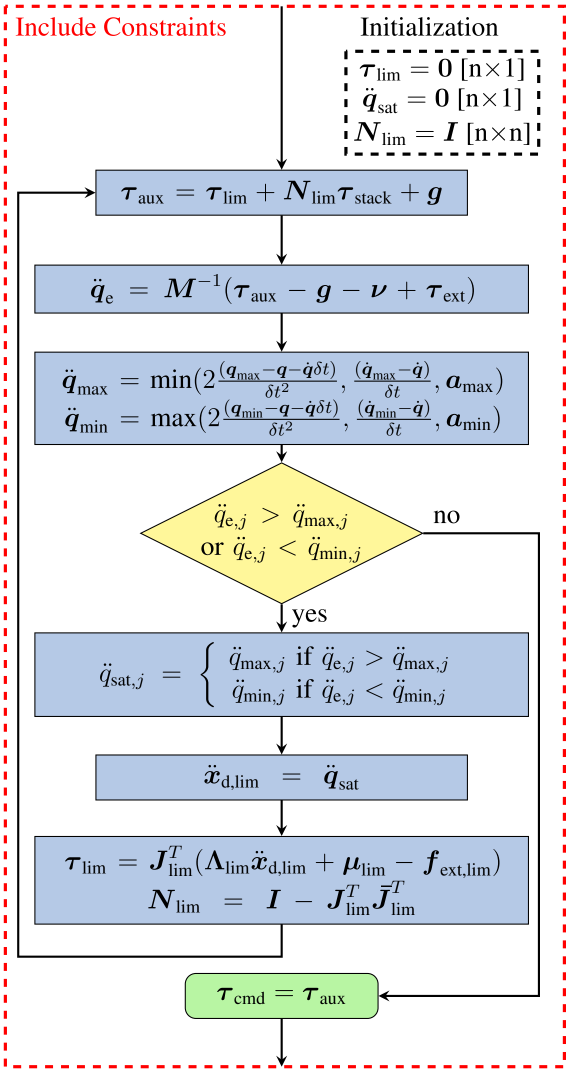


Figure 3.7: Flowchart of Include Constraints block using Saturation in Joint Space (SJS) method

used for simulation and on the real robot with the KUKA Fast Robot Interface (FRI [SSB10]) to control the robot.

3.4.1 Simulation results

In the first simulation, the primary task is to follow a circular trajectory with the end effector. The stack of tasks is chosen to have two tasks. The primary task of dimension $m_1 = 3$ has a linear dynamic behavior as specified in section 2.1.1. The stiffness matrix $\boldsymbol{K}_1 \in \mathbb{R}^{3\times3}$ and the damping matrix $\boldsymbol{D}_1 \in \mathbb{R}^{3\times3}$ are set to $400\boldsymbol{I}$ kg/s^2 and $40\boldsymbol{I}$ kg/s to have a critically damped unit-mass system, where $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the identity matrix. The error $\boldsymbol{e}_1$ is defined in x, y and z coordinates. The vectors $\boldsymbol{x}_{\text{d},1}$ and $\dot{\boldsymbol{x}}_{\text{d},1}$ are the desired positions and velocities given by a trajectory planner to

Table 3.1: Joint limits of the KUKA LBR iiwa

Limits \ Joint	q_1	q_2	q_3	q_4	q_5	q_6	q_7
$\bar{\boldsymbol{q}}$ in deg	170	120	170	120	170	120	175
$\underline{\boldsymbol{q}}$ in deg	-170	-120	-170	-120	-170	-120	-175
$\boldsymbol{v}_{\text{max}}$ in deg/s	100	110	100	130	130	180	180
$\boldsymbol{v}_{\text{min}}$ in deg/s	-100	-110	-100	-130	-130	-180	-180
$\boldsymbol{a}_{\text{max}}$ in deg/s^2	300	300	300	300	300	300	300
$\boldsymbol{a}_{\text{min}}$ in deg/s^2	-300	-300	-300	-300	-300	-300	-300

perform a circular motion.

The three methods show a similar resulting motion of the end effector. Therefore, the result is presented just once for the classic FIRAS in fig. 3.8. The secondary task is specified in the joint space to damp the joint velocities reducing the kinetic energy, $\ddot{\boldsymbol{x}}_{\text{damp}} = -\boldsymbol{D}_2\dot{\boldsymbol{q}}$, where $\boldsymbol{D}_2 \in \mathbb{R}^{n\times n}$ is a damping matrix set to $40\boldsymbol{I}$ Nms/rad. The final stack of tasks structure is:

$$\begin{aligned} \boldsymbol{\tau}_{\text{stack}} &= \boldsymbol{J}_1^T \boldsymbol{f}_1 + \boldsymbol{N}_1 \boldsymbol{\tau}_{\text{damp}}, \\ \boldsymbol{\tau}_{\text{damp}} &= \boldsymbol{J}_{\text{damp}}^T \boldsymbol{f}_{\text{damp}} = \boldsymbol{J}_{\text{damp}}^T \boldsymbol{M} \ddot{\boldsymbol{x}}_{\text{damp}}. \end{aligned} \tag{3.20}$$

The resulting joint positions and torques are presented for the three methods in fig. 3.9. This figure compares the three presented methods to avoid joint limits. During the motion, the 5-th and the 7-th joint reach the critical zone. Note that the joints stop moving towards their limits and the Cartesian trajectory is followed making use of the remaining joints.

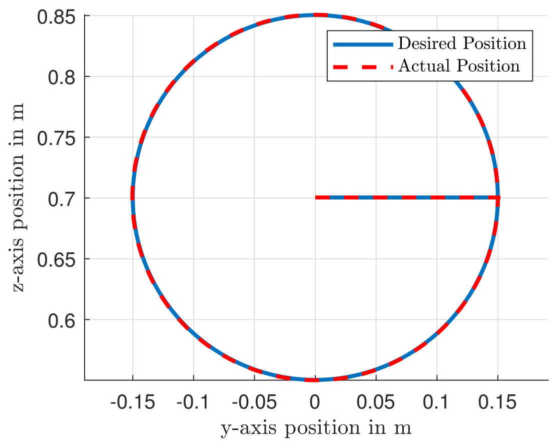


Figure 3.8: Cartesian position of end effector in simulation

Classical FIRAS presents undesired oscillations and high peaks in the torques, when the task joint

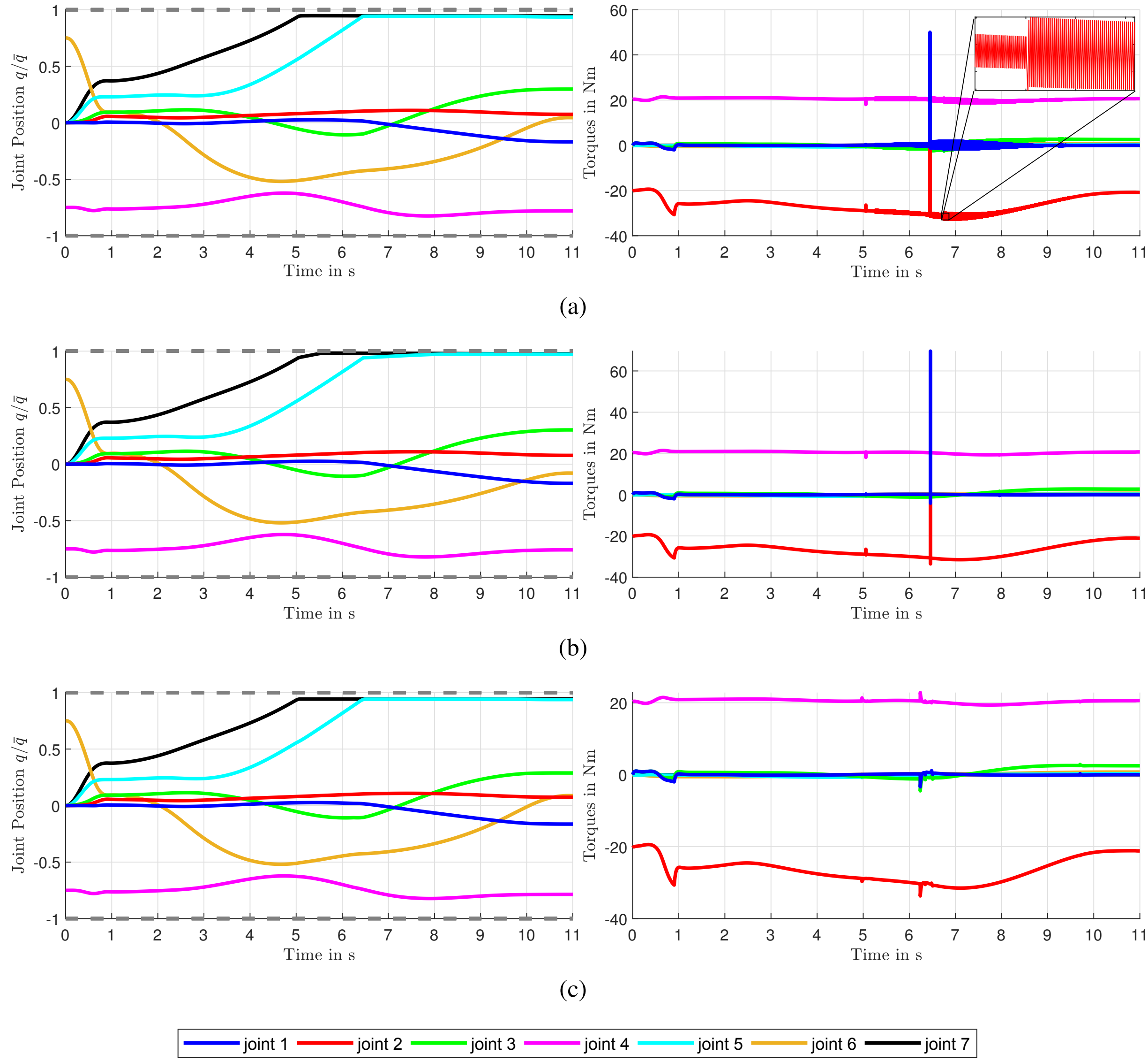


Figure 3.9: Results of tracking a circle in simulation. (a) FIRAS classic, (b) FIRAS with variable η and (c) SJS. The plots in the left side show the normalized joint positions. The limits are represented by the gray dashed lines. The plots in the right side show the joint torques.

limit avoidance is activated ($\boldsymbol{\tau}_{\text{lim}} \neq \mathbf{0}$ and $\boldsymbol{N}_{\text{lim}} \neq \boldsymbol{I}$), see fig. 3.9(a). Approximately at time = 5 s, when the 7-th joint enters the critical zone, the task limit avoidance is activated and oscillations in all joints are generated. When the 5-th joint enters the zone, bigger oscillations and high peaks are produced in the torques. Results are different for the FIRAS with variable η approach. By the addition of a damping, the joint velocity is reduced and the force is activated only when the joint moves toward the limit. As a result, the oscillations disappear (see fig. 3.9(b)). However, the high peak of fig. 3.9(b) is still present in the solution due to the high required acceleration to respect the position limits in an instant of time. The best results are presented using the SJS approach. This approach generates a smooth behaviour in the proximity of the joint limits. Oscillations do not appear and the peaks are greatly reduced (see fig. 3.9(c)). In the next section, the SJS algorithm is tested on the real robot.

3.4.2 Experimental results

The same trajectory and task structure were implemented on the real robot only for the SJS approach given its good performance in simulation. The motion of the end effector is shown in fig. 3.10, while the resulting joint positions and torques are presented in fig. 3.11.

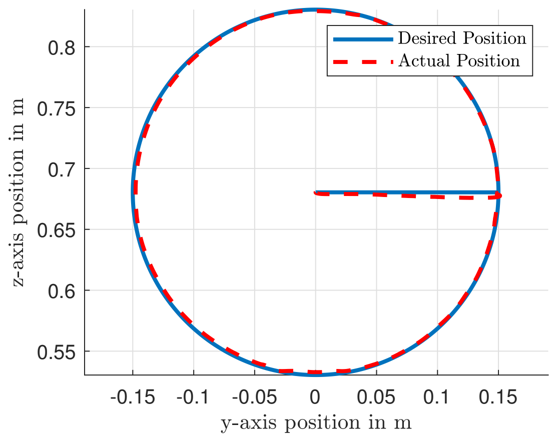


Figure 3.10: Cartesian position of the end effector for the real robot

The behaviour of the robot is different as in simulation. The difference in the joint motions depend most likely on the accuracy of the dynamic model and joint friction. However, different motions in the joint space can lead to the same motion in Cartesian space due to the redundancy of the system. For instance, the 7-th joint does not move as much as in simulation, but the Cartesian motion is similar to the simulated motion (see fig. 3.10).

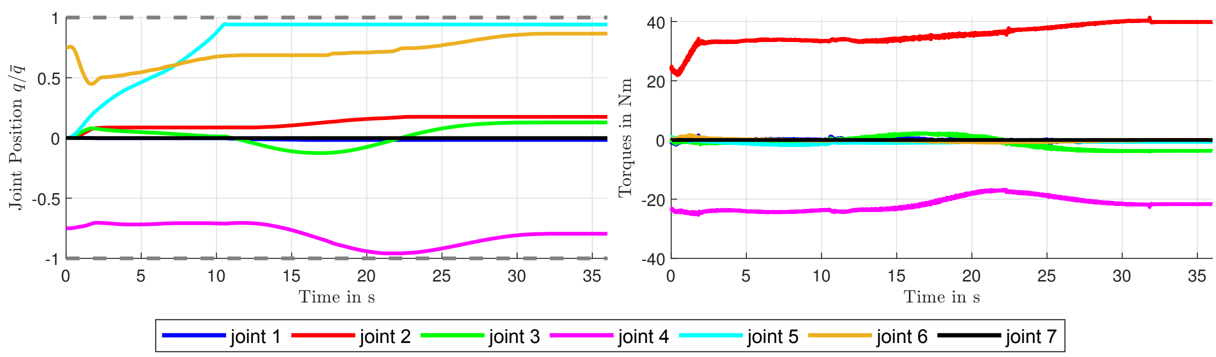


Figure 3.11: Results of tracking a circle on a real robot using SJS approach

The good performance of the SJS method is corroborated with the results on the real robot. Fig. 3.10 shows how the primary task is fulfilled (following the desired Cartesian trajectory) while two joints are saturated to their maximum position value, as it can be seen in fig. 3.11. Furthermore, the joint torques present a smooth behavior. Neither high peaks nor oscillations appear.

The second big advantage of the SJS method is the inclusion of limits at a velocity level. A new experiment was carried out to evaluate the velocity saturation. The circular trajectory is chosen again, but the radius of the circle is increased to 35 cm. The trajectory should be driven at a speed of 0.314 rad/s (i.e., the end-effector travels the whole circle in 20 s). The velocity limits of joint 2 and 3 were intentionally reduced to force a saturation at velocity level (see table 3.2).

Figure 3.12 shows the results of the experiment. Joints 2, 4 and 6 reached their position limits but never exceeded them. Note that once a saturation is not required anymore, the saturated joints are instantly excluded from the joint limit avoidance task. They are free to move away from their limit (see for example the motion of joint 2). The velocity of joint 2 was saturated to 10 deg/s and then reduced to 0 when the joint reached its position limit. Around second 13, the velocity of joint 3 was saturated to its velocity limit. The "oscillations" of the velocity measures shown in fig. 3.12 come from noise and not from the control algorithm (see e.g. joint 3 is noisy even far from saturation).

A final experiment was carried out to test the robustness and good performance of the algorithm, while the human interacts with the robot. The intention of the human is to drive the joints to their upper and lower limits.

Fig. 3.13 shows the robustness of the algorithm when the human applies external forces to the robot. Almost every joint was driven to its limit. Sometimes, even more than two joints were saturated at the same time. Even when the human applies external forces in the direction to the joint limit (see fig. 3.13(c)), the robot remains in the admissible area respecting also velocity limits.

It is worth to mention that a direct measurement of the joint accelerations is not possible with the implemented hardware. The joint acceleration given in fig. 3.13(b) was computed based on the measured torques and the dynamic model, which brings many inaccuracies. The noise observed in fig. 3.13(a) and 3.13(b) depends on the noise of the system and should not be confused with oscillations. The external torque plotted in 3.13(b) demonstrate the application of external forces of the human.

Table 3.2: Joint velocity limits used for the second experiment on the real robot

Limits \ Joint	q_1	q_2	q_3	q_4	q_5	q_6	q_7
$\boldsymbol{v}_{\max}$ in deg/s	100	10	15	130	130	180	180
$\boldsymbol{v}_{\min}$ in deg/s	-100	-10	-15	-130	-130	-180	-180

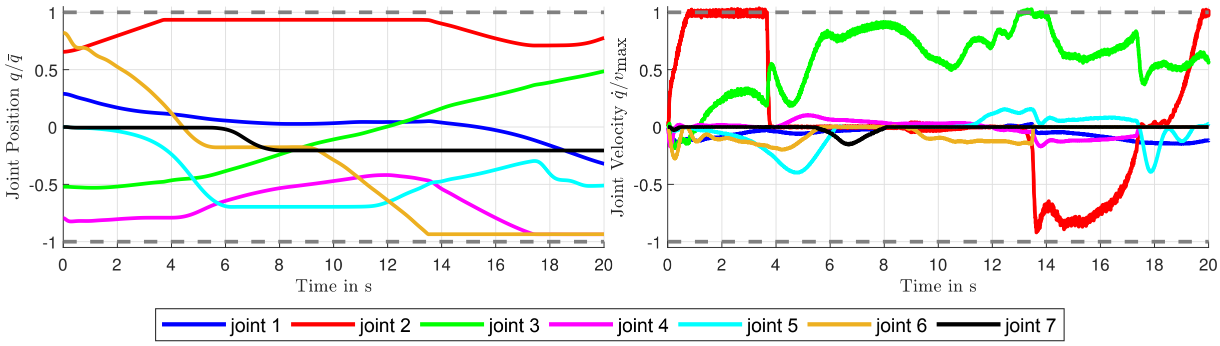
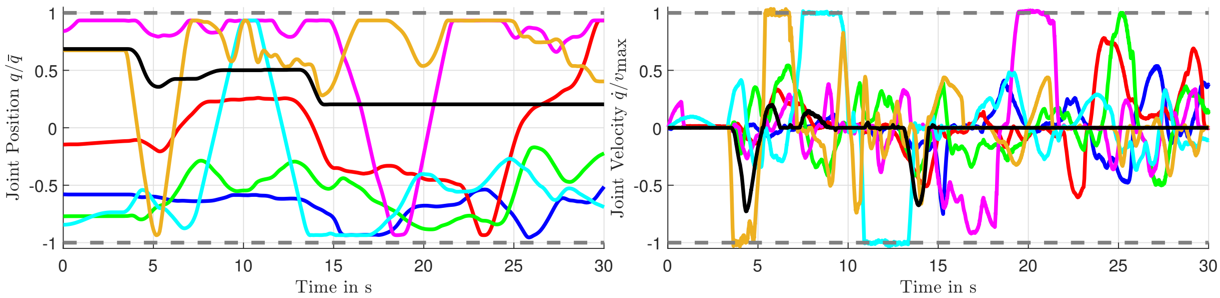
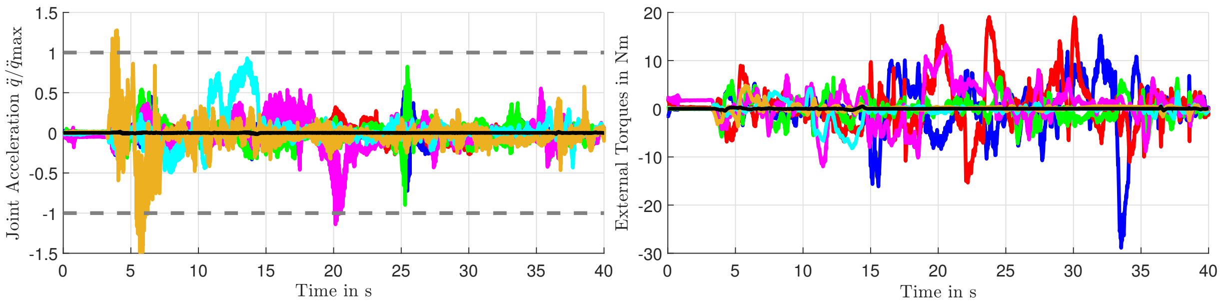


Figure 3.12: Joint position and velocities while the end effector is following a circular trajectory with reduced virtual velocity limits for joint 2 and 3



(a)



(b)

(c)

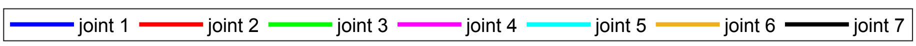


Figure 3.13: Results while human is physically interacting with the robot. (a) Joint positions and velocities. (b) Joint acceleration and torques. (c) Snapshots of the performed experiment.

3.5 Discussion

This chapter provides an overview of the classical method and proposes two new methods to perform joint limit avoidance in force-controlled robots. The first method is a modification of the known potential field method, that uses the FIRAS force proposed by Khatib [Kha85]. The proposed method reduces the kinetic energy in the proximity to the joint position limit, avoiding oscillations due to the activation and deactivation of the task. Results demonstrate that oscillations are eliminated. Additionally, the repercussion of choosing a proper parameter for the FIRAS force is highly reduced due to an automatic computation of η. Despite the improvement of this method, limits at acceleration and velocity level are not considered. Furthermore, simulations showed that high peaks could appear, which are undesired when performing pHRI. The second proposed method is entitled as the saturation in joint space algorithm (SJS). This approach successfully fulfills the two requirements given in section 1.3.1 for constraints at different levels: joint position, velocity and acceleration limits are respected with the highest level of priority; without any prioritization between them. Moreover, the repulsive force is computed based on the current state of the robot. Thus, different to the potential field-based methods, setting of gains is not required. Results showed that the behavior of the robot using this method has an intuitive and "friendly[3]" behavior in the proximity to the joint limits. This result fulfills requirement 4.1 to have the framework working under pHRI. Plots show, however, that there are small violations of the velocity limits. This phenomenon can be explained by inaccuracies of the dynamic model. Errors in the estimation of external forces, gravity and inertia terms lead to wrong estimation of the forces needed to correctly saturate the joints. Despite these inaccuracies, the velocity limits were exceeded less than one percentage. These results are encouraging considering that the human applied high external forces to try to exceed the limits.

Finally, fig. 3.13(b) shows that accelerations limits were exceeded for joint 4 and 6. The results can, though, not be considered as bad performance of the algorithm. As the signal in this plot is only estimated, this result can not conclude that the accelerations were not correctly saturated. One can conclude, however, that if joint limits and velocity limits are respected is because the torque vector $\boldsymbol{\tau}_{\text{lim}}$ saturated the joint accelerations to $\ddot{\boldsymbol{q}}_{\text{sat}}$.

The next chapter extends the SJS approach to handle unilateral constraints in different spaces. The inclusion of unilateral constraints with this method pretends to fulfill the requirements given in section 1.3.

[3]Friendly in this context means that the human does not experience high forces against him, or vibrations of the robot.

4 Unilateral constraints in different spaces

The Saturation in Joint Space (SJS) approach presented in section 3.3.2 includes joint space constraints at different levels in the stack of tasks. This chapter extends this approach to include unilateral constraints in different spaces. Besides translational limits (virtual walls), special cases of unilateral constraints in Cartesian space are considered: obstacle collision avoidance and rotational limits.

Collision avoidance has been treated typically as a motion planning problem. The aim is to obtain a path for the robot, in which the robot does not collide with any object in the environment [Mor80, Bro83, LP90]. From this perspective, the robot motion is limited to avoid collision in fix environments, if the path is planned off-line before executing the robot motion. As the aim of this thesis is an on-line reactive controller, an approach to avoid collision of the robot in a cluttered and evolving environment for interactive operations is necessary. Although in pHRI, intentional contact is required, there are cases, where a contactless obstacle avoidance is desired. For instance, collision of the robot with the user's head is critical and must be avoided. Contact with delicate objects in the environment is also not allowed to avoid damages. Typically, a repulsive force is created based on a potential field in the proximity to the object to expel the robot structure away from it [Kha85, DF12]. Disadvantages of potential field approaches were discussed in section 1.1 and illustrated in chapter 3. [FKLK12] proposes a different approach to avoid dynamic obstacles (such as humans). The velocity of the moving obstacle is estimated and included in the algorithm to produce a smooth robot behavior when the robot is approaching the object. The control, though, is at velocity level. Section 4.4 includes a collision avoidance method in the stack of constraints considering the advantages of the SJS approach.

The inclusion of rotational limits increases exploitation of redundancy, which brings more solutions to the redundancy resolution problem. Not all tasks require 6 Cartesian coordinates to be fulfilled, and not all coordinates require high accuracy. In some cases, a high tolerance in some coordinates is still acceptable to accomplish the task. For instance, a welding task does not require 3 DOF for orientation. The rotation of the welding-gun around the electrode is not relevant to accomplish the task [HB05]. Moreover, the orientation around the other two axes may not need to be accurate but it could have some degrees of tolerance. These admissible tolerances can be given in the programming side by the user or computed by a planning software. The control receives the tolerances as inequality constraints that must be held. Different to the joint limitation explained in chapter 3, these inequalities do not come only from the physical limitation of the robot. They come also from relaxing the task to increase redundancy.[1]

Limitation of rotational coordinates received few attention in the robotics community. Intuitively, it looks like the solution is analog to translational coordinates, however, orientation coordinates are not decoupled. [MLOH03] handles the problem of rotational limits to some extend. Virtual fixtures are built creating a forbidden region for a surgical tool. However, the rotation coordinates are represented in Euler angles. This representation is known for having singularity problems. [KLT06, SRG$^+$16] formulate motion primitives to simplify the task programming. Rotational limits are indirectly managed by limiting the motion of these primitives. However, [MLOH03, KLT06] are implemented at velocity level, and [SRG$^+$16] is used only for off-line planning. Recently, [QA19] defined vector field inequalities and uses the advantages of double quaternions to limit rotational coordinates. The method is also implemented for a torque-controlled humanoid. However, it lacks of inclusion of angular velocity limitation, the definition of the repulsive force is still done as a potential field and it was not proven to work under pHRI.

Next section generalizes the inclusion of unilateral constraints, as shown in section 3.3.2, to any space. Sections 4.2 and 4.3 include these constraints in the hierarchy based on null-space projectors and in the hierarchy based on quadratic programming, respectively. Section 4.4 implements the unilateral constraints for performing collision avoidance. Section 4.5 extends the framework proposed in this chapter to limit rotational coordinates. Angular velocity is also limited thanks to the shaping of limits. Finally, the whole framework is evaluated under pHRI to demonstrate a smooth behavior of the robot motion.

4.1 Generalization of unilateral constraints

Consider the following limits of dimension l in a limited space[2] for a frame of interest that can be located along the robot structure or the tool (e.g. the end effector):

$$\begin{aligned}\boldsymbol{c}_{\text{min}} &\leq \boldsymbol{c} \leq \boldsymbol{c}_{\text{max}},\\ \boldsymbol{v}_{\text{min}} &\leq \dot{\boldsymbol{c}} \leq \boldsymbol{v}_{\text{max}},\\ \boldsymbol{a}_{\text{min}} &\leq \ddot{\boldsymbol{c}} \leq \boldsymbol{a}_{\text{max}},\end{aligned} \tag{4.1}$$

where $\boldsymbol{c} \in \mathbb{R}^l = FK(\boldsymbol{q})$, $\dot{\boldsymbol{c}}$ and $\ddot{\boldsymbol{c}}$ are the pose, velocity and acceleration of the limited directions. $FK(\boldsymbol{q})$ is a forward kinematics function that computes the pose of the frame of interest in the limited space. For physical interaction, the pose limits $\boldsymbol{c}_{\text{min}}$ and $\boldsymbol{c}_{\text{max}}$ are virtual limits that should be set in a safe distance to the physical limit. Otherwise, high external forces could drive the point of interest into its physical limit. The velocity and acceleration limits ($\boldsymbol{v}_{\text{min}}$, $\boldsymbol{v}_{\text{max}}$, $\boldsymbol{a}_{\text{min}}$ and

[1]This applies also for inequalities in the translational coordinates.
[2]Space in which the constraints are included. For instance, the Cartesian space or a subspace of it.

$\boldsymbol{a}_{\text{max}}$) are given by the user according to hardware limitations, application use-case and/or safety requirements.

For the sake of simplicity, in the following, assume a point of interest instead of a frame, so that the limitation of rotation is ignored for now. The extension to rotational coordinates is given in section 4.5. Consider that the command torque vector $\boldsymbol{\tau}_{\text{stack}}$ produces a constant acceleration in the limited space $\ddot{\boldsymbol{c}}_{\text{h}} = \ddot{\boldsymbol{c}}(t_{\text{h}})$ for a sample time δt. Similar to SJS, assuming that at $t_{\text{h}} = h\delta t$, the current position $\boldsymbol{c} = \boldsymbol{c}_{\text{h}}$ and velocity $\dot{\boldsymbol{c}} = \dot{\boldsymbol{c}}_{\text{h}}$ are both feasible. The next position in the Cartesian space can be estimated by:

$$\dot{\boldsymbol{c}}_{\text{h+1}} \simeq \dot{\boldsymbol{c}}_{\text{h}} + \ddot{\boldsymbol{c}}\delta t,\ \boldsymbol{c}_{\text{h+1}} \simeq \boldsymbol{c}_{\text{h}} + \dot{\boldsymbol{c}}_{\text{h}}\delta t + \frac{1}{2}\ddot{\boldsymbol{c}}\delta t^2, \tag{4.2}$$

with

$$\ddot{\boldsymbol{c}}_{\text{h}} = \boldsymbol{J}_c\ddot{\boldsymbol{q}}_{\text{h}} + \dot{\boldsymbol{J}}_c\dot{\boldsymbol{q}}_{\text{h}}, \tag{4.3}$$

where $\boldsymbol{J}_c \in \mathbb{R}^{l\times n}$ relates the velocity on the limited space to the joint velocities ($\dot{\boldsymbol{c}} = \boldsymbol{J}_{\text{c}}\dot{\boldsymbol{q}}$). When constraints in different spaces are considered, the Jacobian matrices of each space are stacked in $\boldsymbol{J}_{\text{c}} = [\boldsymbol{J}_{\text{c,1}}\ \boldsymbol{J}_{\text{c,2}}\ \ldots\ \boldsymbol{J}_{\text{c},n_c}]^{\text{T}}$, where n_c is the number of constraints in different spaces. If $\boldsymbol{J}_{\text{c}}$ does not have full rank, it means there exists a conflict between constraints, and not all constraints can be respected. The problem does not have a solution for these cases and a strategy must be defined according to safety and/or application requirements. This thesis considers stopping the robot as the safest reaction, as stated in section 1.3.

Next step is to shape the limits in a generalized form. If the acceleration produced by the command joint torque has to keep the position and velocity within their limits:

$$\frac{\boldsymbol{v}_{\text{min}} - \dot{\boldsymbol{c}}_{\text{h}}}{\delta t} \leq \ddot{\boldsymbol{c}} \leq \frac{\boldsymbol{v}_{\text{max}} - \dot{\boldsymbol{c}}_{\text{h}}}{\delta t}, \tag{4.4}$$

$$\frac{2(\boldsymbol{d}_{\text{min}} - \dot{\boldsymbol{c}}_{\text{h}}\delta t)}{\delta t^2} \leq \ddot{\boldsymbol{c}} \leq \frac{2(\boldsymbol{d}_{\text{max}} - \dot{\boldsymbol{c}}_{\text{h}}\delta t)}{\delta t^2}, \tag{4.5}$$

where d_{max} and d_{min} are the distances to the upper and lower limits, respectively, and are given by:

$$\boldsymbol{d}_{\text{max}} = \boldsymbol{c}_{\text{max}} - \boldsymbol{c}_{\text{h}}, \tag{4.6}$$

$$\boldsymbol{d}_{\text{min}} = \boldsymbol{c}_{\text{min}} - \boldsymbol{c}_{\text{h}}. \tag{4.7}$$

As shown in section 3.3.2, to keep a smooth behavior in the velocity near to the velocity limits, $\boldsymbol{v}_{max}$ and $\boldsymbol{v}_{min}$ are variable and depend on the distance to the limit

$$\dot{\boldsymbol{c}}_{\text{max}} = \min(\boldsymbol{v}_{\text{max}}, \sqrt{2\boldsymbol{a}_{\text{max}}(\boldsymbol{d}_{\text{max}})}), \tag{4.8}$$

$$\dot{\boldsymbol{c}}_{\text{min}} = \max(\boldsymbol{v}_{\text{min}}, -\sqrt{2\boldsymbol{a}_{\text{max}}(-\boldsymbol{d}_{\text{min}})}). \tag{4.9}$$

Considering the inequalities, a box of constraints for the acceleration $\boldsymbol{c}_{\text{max}}$ and $\boldsymbol{c}_{\text{min}}$ at time t_{h} is obtained

$$\ddot{\boldsymbol{c}}_{\text{min}}(t_{\text{h}}) \leq \ddot{\boldsymbol{c}} \leq \ddot{\boldsymbol{c}}_{\text{max}}(t_{\text{h}}), \tag{4.10}$$

where

$$\ddot{\boldsymbol{c}}_{\text{min}} = \max(2\frac{(\boldsymbol{d}_{\text{min}} - \dot{\boldsymbol{c}}\delta t)}{\delta t^2}, \frac{(\dot{\boldsymbol{c}}_{\text{min}} - \dot{\boldsymbol{c}})}{\delta t}, \boldsymbol{a}_{\text{min}}), \tag{4.11}$$

$$\ddot{\boldsymbol{c}}_{\text{max}} = \min(2\frac{(\boldsymbol{d}_{\text{max}} - \dot{\boldsymbol{c}}\delta t)}{\delta t^2}, \frac{(\dot{\boldsymbol{c}}_{\text{max}} - \dot{\boldsymbol{c}})}{\delta t}, \boldsymbol{a}_{\text{max}}). \tag{4.12}$$

The next section explains how to include the constraints into the task hierarchy.

4.2 Inclusion of unilateral constraints in the task hierarchy based on null-space projectors

After computing the maximal accelerations, a task for keeping the motion within the limits must be created and included in the stack of tasks. This section shows the development of the algorithm to include the constraints in the stack of tasks using augmented projections.

Consider algorithm 1 based on null-space projectors to compute the command torque. The method receives the stack of tasks and computes the joint torque with the help of the dynamic model of the robot. Each task has two components: the command vector $\ddot{\boldsymbol{x}}_i$ and the Jacobian $\boldsymbol{J}_i$. Note that for IWIS, $\ddot{\boldsymbol{x}}_{\text{t}}$ is equal to $\boldsymbol{\Lambda}_{\text{t}}^{-1}\boldsymbol{f}_{\text{t}}$.

The process shown in fig. 3.7 is modified to include the inequalities in different spaces and at different levels (see fig. 4.1). First, the torque of the stack of tasks is computed according to algorithm 1. Then, the joint accelerations produced by those torques are calculated from the dynamic model. Next, joint accelerations are mapped to the limited space.

The box of constraints is computed and the maximal and minimal accelerations are set. If none of the limits are exceeded, the command torque is equal to the torque of the stack of tasks. If any acceleration exceeds its limits, that acceleration is saturated to its maximal or minimal value. Then, the new task is created with a command vector $\ddot{\boldsymbol{x}}_{\text{t}}$ equal to the saturated acceleration $\ddot{\boldsymbol{c}}_{\text{sat}}$. The Jacobian $\boldsymbol{J}_{\text{sat}}$ maps the joint space to the subspace of the saturated space.

With this new information, the stack of tasks is computed again according to algorithm 1. The saturation task gets the highest priority $i = 1$, while all the other tasks decrease their level of priority. To ensure that all the joints remain in their limits, the process starts again.

Algorithm 1: Computation of the command torque using augmented, dynamically consistent projections

```
function getTorques(task_1, task_2, ···, task_k)
    for i = 1 to k do
        J_i = task_i.J, ẍ_i = task_i.ẍ_t
        if i == 1 then
            N_i^aug = I
        else
            N_i^aug = N_{i-1}^aug (I − Ĵ_{i-1}^T \hat{\bar{J}}_{i-1}^T)
        end
        Ĵ_i = J_i N_i^{aug,T}, Λ̂_i = Λ̂_{ns,i} + Λ̂_{s,i}
        \hat{\bar{J}}_i = M^{-1} Ĵ_i^T Λ̂_i
        μ̂_i = \hat{\bar{J}}_i^T ν − Λ̂_i \dot{Ĵ}_i q̇
        f̂_i = Λ̂_i(ẍ_i − J_i(M^{-1}(τ_{i-1} − ν))) − f_{ext,i} + μ̂_i
        τ_i = J_i^T f̂_i
        τ_i^P = N_i^aug τ_i
    end
    τ_stack = Σ_{i=1}^k τ_i^P + g
    return τ_stack ;
end
```

4.3 Inclusion of unilateral constraints in the task hierarchy based on Quadratic Programming

The advantage of Quadratic Programming (QP) is the easy inclusion of linear inequalities in the QP problem. Thanks to the unification of constraints at different levels, it is possible to include the constraints also in different spaces in the QP problem [MAAZ20, MA20]. The QP problem becomes:

$$\min_{\ddot{\boldsymbol{q}}_i, \boldsymbol{\tau}_i} \quad \|\boldsymbol{J}_i\ddot{\boldsymbol{q}}_i - (\ddot{\boldsymbol{x}}_{\mathrm{d},i} - \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}} - \boldsymbol{\Lambda}_{\mathrm{t}}^{-1}\boldsymbol{f}_{\mathrm{ext}})\|^2 \tag{4.13a}$$

$$\text{subject to} \quad \boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\nu} + \boldsymbol{g} \tag{4.13b}$$

$$\boldsymbol{\tau}_{\mathrm{min}} \leq \boldsymbol{\tau_i} \leq \boldsymbol{\tau}_{\mathrm{max}} \tag{4.13c}$$

$$\ddot{\boldsymbol{c}}_{\mathrm{min}} - \boldsymbol{J}_{\mathrm{c}}\boldsymbol{M}^{-1}\boldsymbol{\tau}_{\mathrm{ext}} \leq \boldsymbol{J}_{\mathrm{c}}\ddot{\boldsymbol{q}}_i + \dot{\boldsymbol{J}}_{\mathrm{c}}\dot{\boldsymbol{q}} \leq \ddot{\boldsymbol{c}}_{\mathrm{max}} - \boldsymbol{J}_{\mathrm{c}}\boldsymbol{M}^{-1}\boldsymbol{\tau}_{\mathrm{ext}} \tag{4.13d}$$

$$\boldsymbol{J}_{i-1}\ddot{\boldsymbol{q}}_{i-1} = \boldsymbol{J}_{i-1}\ddot{\boldsymbol{q}}_i \tag{4.13e}$$

$$\vdots \tag{4.13f}$$

$$\boldsymbol{J}_1\ddot{\boldsymbol{q}}_1 = \boldsymbol{J}_1\ddot{\boldsymbol{q}}_i. \tag{4.13g}$$

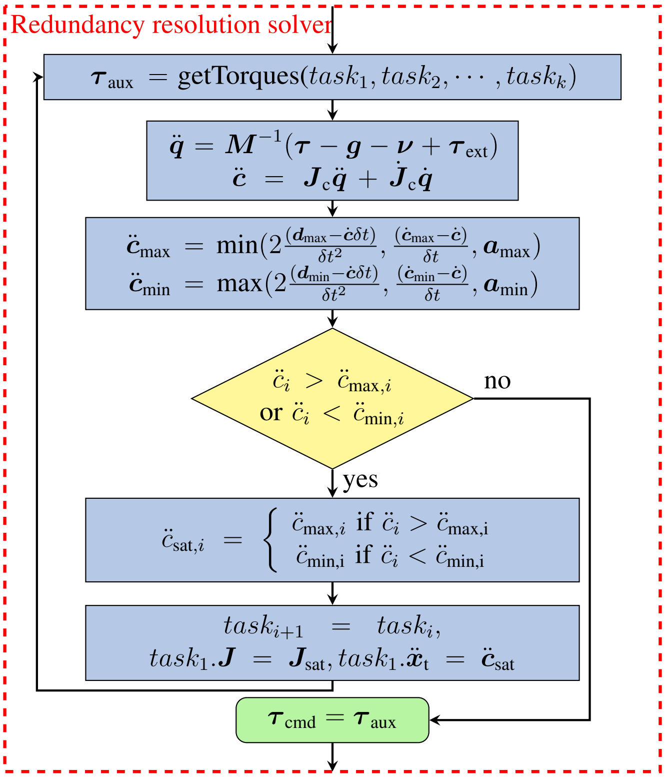


Figure 4.1: Redundancy resolution solver based on augmented null-space projectors

Limiting the Cartesian space in translational coordinates can be done straightforwardly. The only required additional inputs (besides the dynamic model and the current state of the robot) are: the Jacobian that relates the joint velocities to the Cartesian velocities of a point of interest that can be placed on robot structure or the tool, and the current position of that point.[3] However, avoiding obstacles needs a small modification of the algorithm.

4.4 Obstacle avoidance

Let the closest points between the robot and the obstacle be the critical points. Distance calculation algorithms provide the location of the critical points based on 3D models of the robot and the obstacle [Wel13]. Assuming the critical point on the robot $\boldsymbol{x}_{\text{c,r}}$ and the critical point on the obstacle $\boldsymbol{x}_{\text{c,o}}$ are known and are expressed in the same frame as the Jacobian, it is possible to compute the direction u_c between these two points

$$\boldsymbol{u}_{\text{c}} = \frac{\boldsymbol{x}_{\text{c,o}} - \boldsymbol{x}_{\text{c,r}}}{\|\boldsymbol{x}_{\text{c,o}} - \boldsymbol{x}_{\text{c,r}}\|_2}. \tag{4.14}$$

[3]With the help of a forward kinematics function, the position of any point on the robot structure or the tool can be computed based on the measured joint positions of the robot.

With this vector, the Jacobian $\boldsymbol{J}_{\mathrm{p}} \in \mathbb{R}^{3\times n}$ that relates the translational velocities of the critical point $\boldsymbol{x}_{\mathrm{c,r}}$ to the joint velocities is aligned with the critical direction

$$\boldsymbol{J}_{\mathrm{c}} = \boldsymbol{u}_{\mathrm{c}}^{T} \boldsymbol{J}_{\mathrm{p}}. \tag{4.15}$$

The minimal distance between the robot and the obstacle is given by:

$$d_{\mathrm{o}} = \|\boldsymbol{x}_{\mathrm{c,o}} - \boldsymbol{x}_{\mathrm{c,r}}\|_2. \tag{4.16}$$

If the robot must keep a safe distance d_{s} to the object, then:

$$d_{\mathrm{max}} = d_{\mathrm{o}} - d_{\mathrm{s}}. \tag{4.17}$$

The current velocity of the critical point is given by:

$$\dot{\boldsymbol{c}} = \boldsymbol{J}_{\mathrm{c}} \dot{\boldsymbol{q}}. \tag{4.18}$$

As the goal is to avoid the object and not to keep close to it, $\ddot{\boldsymbol{c}}_{\mathrm{min}}$ is set to be -inf.

However, if the object is moving, the velocity of the object (or more accurately of the point $\boldsymbol{x}_{\mathrm{c,o}}$) must be taken into account. For this case, the velocity $\dot{\boldsymbol{c}}$ is the relative velocity between the critical point on the robot and the one on the obstacle

$$\dot{\boldsymbol{c}} = \boldsymbol{J}_{\mathrm{c}} \dot{\boldsymbol{q}} - \mathrm{u}_{\mathrm{c}}^{T} \dot{\boldsymbol{x}}_{\mathrm{c,o}}. \tag{4.19}$$

Besides obstacle avoidance, a special case of limitations is to have limits on rotational coordinates. This strongly depends on the application and the definition of the constraints. In the next section, a way to perform this limitation is provided.

4.5 Limitation for rotational coordinates

Limiting one rotational coordinate is not straightforward and heavily depends on the used representation. On the one hand, Euler angles do not have a unique representation and suffer from singularity problems. On the other hand, quaternions or rotation matrices are clear defined representations of a rotation. Additionally, different to translational coordinates, rotational coordinates are coupled. Any representation needs a full set definition to be a meaningful representation of a rotation. Euler angles require three angles of rotation, unit quaternions require 4 values and rotation matrices must be (3×3) orthonormal matrices with determinant 1. This section proposes a suitable geometric representation for limiting rotational coordinates. The use of rotation matrices leads to a better visual understanding.

Assume, a rotation matrix $\boldsymbol{R}_{\text{cur}}$ that represents the rotation of a current coordinate frame $\boldsymbol{\mathcal{F}}_{\text{cur}}$ attached to the robot's kinematic chain, and a rotation matrix $\boldsymbol{R}_{\text{ref}}$ that represents the rotation of a reference frame $\boldsymbol{\mathcal{F}}_{\text{ref}}$. The rotation matrices $\boldsymbol{R}$ are defined by three unit vectors $[\boldsymbol{u}_{\text{x}}, \boldsymbol{u}_{\text{y}}, \boldsymbol{u}_{\text{z}}]$. Limiting one rotational coordinate means that a unit direction vector $\boldsymbol{u}_{\text{i,cur}}$ of $\boldsymbol{R}_{\text{cur}}$ may not be separated by an angle greater than θ_{max} from the plane P_{ref} formed by the two unit vectors ($\boldsymbol{u}_{\text{i,ref}}$ and $\boldsymbol{u}_{\text{j,ref}}$) of $\boldsymbol{R}_{\text{ref}}$. The unit vector $\boldsymbol{u}_{\text{k,ref}}$ is then the normal vector to the plane P_{ref}. The sub-index $[]_{\text{j}}$ indicates the axis in which the rotation is limited, $[]_{\text{i}}$ and $[]_{\text{k}}$ are the non-limited rotational axes. E.g., to limit the rotation around $\boldsymbol{u}_{\text{x}}$, $\boldsymbol{u}_{\text{j}} = \boldsymbol{u}_{\text{x}}$, $\boldsymbol{u}_{\text{i}} = \boldsymbol{u}_{\text{z}}$ and $\boldsymbol{u}_{\text{k}} = \boldsymbol{u}_{\text{y}}$ (see fig. 4.2 left).

The projection of $\boldsymbol{u}_{\text{i,cur}}$ on P_{ref} is defined by:

$$\boldsymbol{p}_{\text{cur}} = \boldsymbol{u}_{\text{k,ref}} \times (\boldsymbol{u}_{\text{i,cur}} \times \boldsymbol{u}_{\text{k,ref}}), \tag{4.20}$$

where $\boldsymbol{p}_{\text{cur}}$ is the projected vector. The angular distance between $\boldsymbol{u}_{\text{i,cur}}$ and the plane P_{ref} is given by the angle

$$\theta = \operatorname{atan2}(\|\boldsymbol{u}_{\text{i,cur}} \times \boldsymbol{p}_{\text{cur}}\|_2, \boldsymbol{u}_{\text{i,cur}} \cdot \boldsymbol{p}_{\text{cur}}) \tag{4.21}$$

The maximal distance for the limits is

$$d_{\text{max}} = \theta_{\text{max}} - \theta \tag{4.22}$$

Let $\boldsymbol{J}_{\text{o}} \in \mathbb{R}^{3\times n}$ be the Jacobian that relates the angular velocities of frame $\boldsymbol{\mathcal{F}}_{\text{cur}}$ to the joint velocities. The Jacobian $\boldsymbol{J}_{\theta}$ that relates the angular velocities in the direction of θ with the joint velocities can be computed by aligning $\boldsymbol{J}_{\text{o}}$ in that direction. The unit vector $\boldsymbol{u}_{\theta}$ (or axis of rotation) that controls this direction, is given by:

$$\boldsymbol{u}_{\theta} = \frac{\boldsymbol{p}_{\text{cur}} \times \boldsymbol{u}_{\text{i,cur}}}{\|\boldsymbol{p}_{\text{cur}} \times \boldsymbol{u}_{\text{i,cur}}\|_2}. \tag{4.23}$$

Thus, the Jacobian of the constraint is:

$$\boldsymbol{J}_{\text{c}} = \boldsymbol{J}_{\theta} = \boldsymbol{u}_{\theta}^{T} \boldsymbol{J}_{\text{o}} \tag{4.24}$$

To limit two coordinates, it is possible to stack different Jacobians $\boldsymbol{J}_{\theta}$ in $\boldsymbol{J}_{\text{c}}$ with a consistent vector of distances $\boldsymbol{d}_{\text{max}}$. Nevertheless, a more robust definition is the limitation of the angle between the vectors $\boldsymbol{u}_{\text{i,cur}}$ and $\boldsymbol{u}_{\text{i,ref}}$. That means limiting the rotation around the vectors $\boldsymbol{u}_{\text{j,ref}}$ and $\boldsymbol{u}_{\text{k,ref}}$. Graphically, it means that $\boldsymbol{u}_{\text{i,curr}}$ can not be outside of a cone with an angle between its central axis and its surface equal to θ_{max} (see fig. 4.2 right).[4] The distance between $\boldsymbol{u}_{\text{i,cur}}$ and $\boldsymbol{u}_{\text{i,ref}}$ is given by eq. 4.22, where:

$$\theta = \operatorname{atan2}(\|\boldsymbol{u}_{\text{i,cur}} \times \boldsymbol{u}_{\text{i,ref}}\|_2, \boldsymbol{u}_{\text{i,cur}} \cdot \boldsymbol{u}_{\text{i,ref}}) \tag{4.25}$$

[4]When $\theta_{\text{max}} > 90°$ the cone is inverted and $\boldsymbol{u}_{\text{i,curr}}$ must remain outside of it.

The Jacobian $\boldsymbol{J}_{\mathrm{c}}$ is computed by eq. 4.24, where:

$$\boldsymbol{u}_\theta = \frac{\boldsymbol{u}_{\mathrm{i,ref}} \times \boldsymbol{u}_{\mathrm{i,cur}}}{\|\boldsymbol{u}_{\mathrm{i,ref}} \times \boldsymbol{u}_{\mathrm{i,cur}}\|_2} \tag{4.26}$$

To limit the three rotational coordinates, Jacobians $\boldsymbol{J}_\theta$ and distances $\boldsymbol{d}_{\mathrm{max}}$ can be stacked.

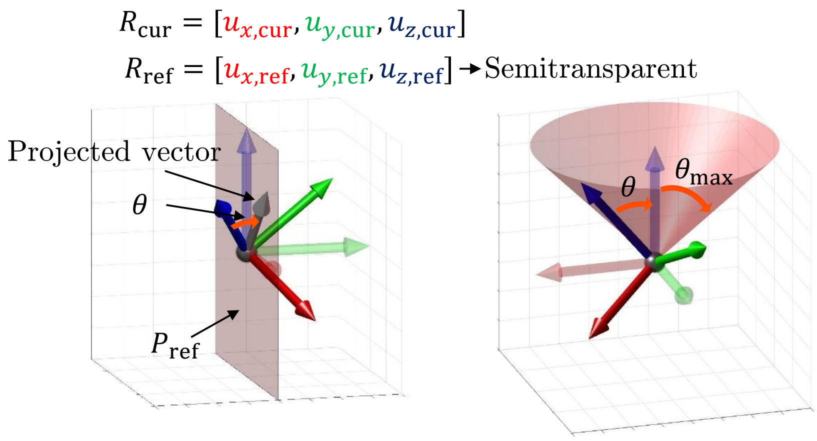


Figure 4.2: Example of limitation of rotational coordinates represented graphically. Left figure represents the limitation of one coordinate. Right figure illustrates the limitation of two coordinates. The red semitransparent cone is the surface limit that the vector $\boldsymbol{u}_{z,\mathrm{cur}}$ can not exceed

4.6 Simulations and experiments

This section performs a variety of experiments to evaluate the performance of the unilateral constraints in different spaces. As in the last chapter, the tests are run on the industrial collaborative robot, KUKA LBR iiwa with seven rotational joints ($n = 7$). First, the translational constraints are tested (section 4.6.1), followed by the rotational ones (section 4.6.2). Finally, the constraints are used for dynamically obstacle avoidance (section 4.6.3).

4.6.1 Translational Cartesian constraints

In all the experiments, the end effector has to follow a circular trajectory given in Cartesian coordinates. The radius of the circle is 35 cm and the velocity is 0.314 rad/s. The initial joint configuration of the robot is $\boldsymbol{q} = [50, 80, -90, -97, 0, 100, 0]^T$ deg. A desired orientation is not required to have more degrees of redundancy. Thus, the primary task is described in 3 Cartesian position coordinates $[x, y, z]$ ($m_1 = 3$) and it has a linear dynamic behaviour as described in section 2.1.1. $\boldsymbol{K}_1 \in \mathbb{R}^{3\times3}$ and $\boldsymbol{D}_1 \in \mathbb{R}^{3\times3}$ are the stiffness and damping gains set to $800\boldsymbol{I}$ kg/s^2 and

$2\sqrt{800}\boldsymbol{I}$ kg/s to have a critically damped unit-mass system. The vectors $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ represent the current Cartesian position and velocity of the end-effector in x, y and z coordinates given by the forward kinematics. The desired positions and velocities $\boldsymbol{x}_{\mathrm{d}}$ and $\dot{\boldsymbol{x}}_{\mathrm{d}}$ are given by a trajectory planner to perform a circular motion. The Jacobian of the primary task $\boldsymbol{J}_1$ maps the velocities in three Cartesian coordinates $[x, y, z]$ to the joint space. A secondary task is specified in the joint space to reduce the kinetic energy, with an acceleration vector $\ddot{\boldsymbol{x}}_2 = -\boldsymbol{D}_2\dot{\boldsymbol{q}}$. The diagonal damping matrix $\boldsymbol{D}_2$ has diagonal elements set to 10 Nms/rad. For this task, the Jacobian is the identity matrix $\boldsymbol{J}_2 = \boldsymbol{I}$. The torques to command the robot and achieve the tasks are computed with algorithm 1. The implementation on the real robot was done at a frequency of 1 kHz ($\delta t = 1$ ms). The results presented in this section were previously published in [MAF$^+$19a].

Avoidance of joint and Cartesian constraints

In the first experiment, joint and Cartesian limits are avoided following the process shown in fig. 4.1. The acceleration vectors $\ddot{\boldsymbol{c}}_{\mathrm{max}}$ and $\ddot{\boldsymbol{c}}_{\mathrm{min}}$ are computed based on both: joint and Cartesian unilateral constraints. The Cartesian space part is based on a one dimensional Cartesian limit for the robot's elbow in the vertical coordinate (coordinate z of the world coordinate system). The Jacobian matrix $\boldsymbol{J}_{\mathrm{c}}$ is in this case composed by two Jacobians: Jacobian of joint space $\boldsymbol{J}_{\mathrm{c},1} = \boldsymbol{I} \in \mathbb{R}^{n\times n}$ and the Jacobian $\boldsymbol{J}_{\mathrm{c},2} \in \mathbb{R}^{1\times n}$ that relates the Cartesian velocity of the elbow z coordinate to the joint velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^{n\times 1}$. The limits are shown in table 4.1.

Figure 4.3 shows the results of saturating the coordinate z of the elbow in Cartesian space. In the first four seconds, the velocity of the elbow got saturated to its lower limit -0.06 m/s. The elbow continued moving with this constant velocity until joint 2 reached its upper limit (see time interval ①). During this time interval, the velocity of the elbow was reduced to zero, because the motion of the elbow in the z-direction depended only on joint 2. In time interval ②, the elbow started moving in the positive direction, joint 2 moves away from its limit, while joint 6 reached its limit. During interval ③, the elbow reached its upper position limit at 0.5 m. In that moment, the reduction of the velocity to zero can be observed in the velocity profile. Note how joint 4 reached also its position limit, while joint 6 was still being saturated.

The desired and actual trajectories of the end effector, $\boldsymbol{x}_{\mathrm{EE,d}}$ and $\boldsymbol{x}_{\mathrm{EE,cur}}$, are shown together with the elbow motion $\boldsymbol{x}_{\mathrm{elb,cur}}$, in fig. 4.4. Two cases are illustrated: motion saturating the joints, and motion saturating joints and elbow. The Cartesian limit of the elbow is represented as a virtual roof. When the elbow reached the roof, it continued moving in other directions (x and y) but not in $+z$. The tracking of the trajectory did not get affected by the saturation of the elbow.

Note that the experiment above was performed using a strict hierarchy with augmented projectors. Unlike a soft hierarchy as in chapter 3, the strict hierarchy allows less interference of the damping task in the main task (tracking the circle). This directly reduces the Euclidean error. To demonstrate this effect, the experiment was repeated using the SJS process (as shown in fig. 3.7) only having

Table 4.1: Limits set for the first experiment

Coordinate	$c_{\max}$	$c_{\min}$	$v_{\max}$	$v_{\min}$	$a_{\max}$	$a_{\min}$
Rotational	deg	deg	deg/s	deg/s	deg/s^2	deg/s^2
q_1	165	-165	100	-100	300	-300
q_2	115	-115	120	-120	300	-300
q_3	165	-115	110	-110	300	-300
q_4	115	-165	130	-130	300	-300
q_5	165	-165	130	-130	300	-300
q_6	115	-115	180	-180	300	-300
q_7	170	-170	180	-180	300	-300
Translational	m	m	m/s	m/s	m/s^2	m/s^2
z_{elb}	0.5	-inf	0.2	-0.06	0.2	-0.2

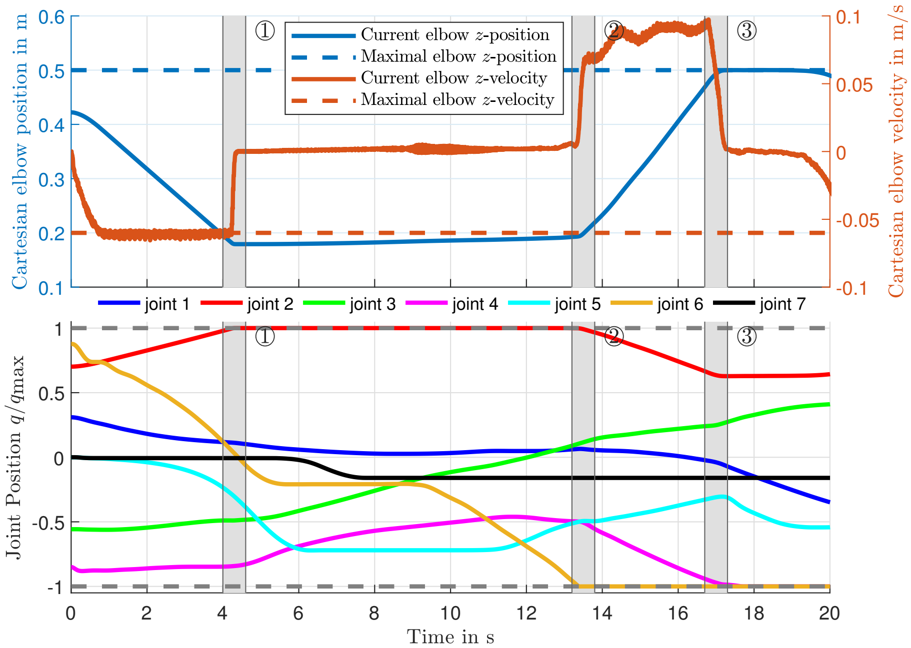


Figure 4.3: Cartesian position and velocity of the elbow in z coordinate (top) and Joint positions (bottom). Limits are shown in dashed lines. The gray zones represent three time intervals of interest.

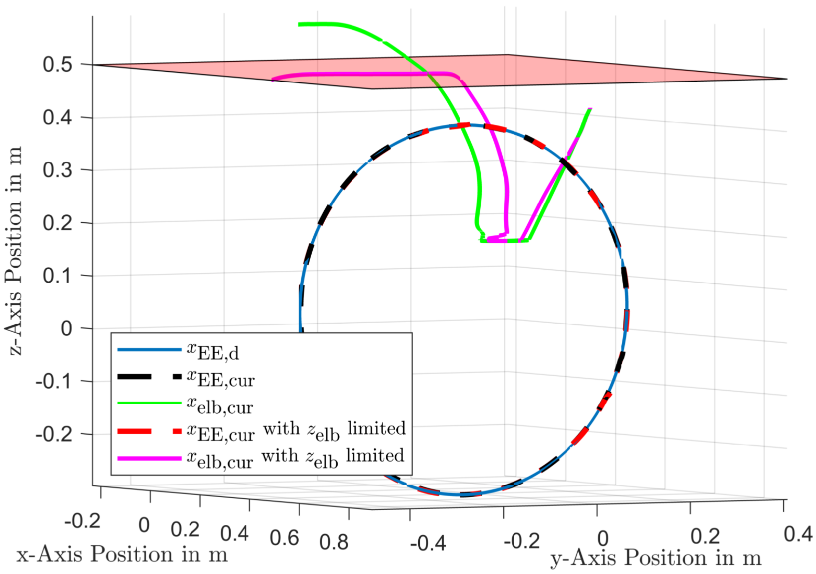


Figure 4.4: End effector and elbow Cartesian current position while following the desired trajectory, with and without saturation of the elbow in Cartesian space. The Cartesian limit is represented by the semi-transparent plane.

joint constraints. This also demonstrates that the inclusion of the constraint for the elbow did not affect the main task (see fig. 4.5).

Avoidance of joint and Cartesian constraints during physical Human-Robot Interaction (pHRI)

The performance of the algorithm was evaluated during pHRI. The robot had to follow the same circular trajectory while the human applied external forces. Additionally, a conflict situation between the task and one unilateral constraint was tested by including the Cartesian limits in the y-coordinate of the end effector (see table 4.2) besides the ones in table 4.1. To respect the Cartesian limit, the circular trajectory cannot be followed in the y-coordinate.

Table 4.2: Limits of the y-Cartesian coordinate added to the limits in table 4.1

Coordinate	$c_{\max}$	$c_{\min}$	$v_{\max}$	$v_{\min}$	$a_{\max}$	$a_{\min}$
Translational	m	m	m/s	m/s	m/s^2	m/s^2
y_{EE}	0.2	-inf	0.3	-0.3	0.2	-0.2

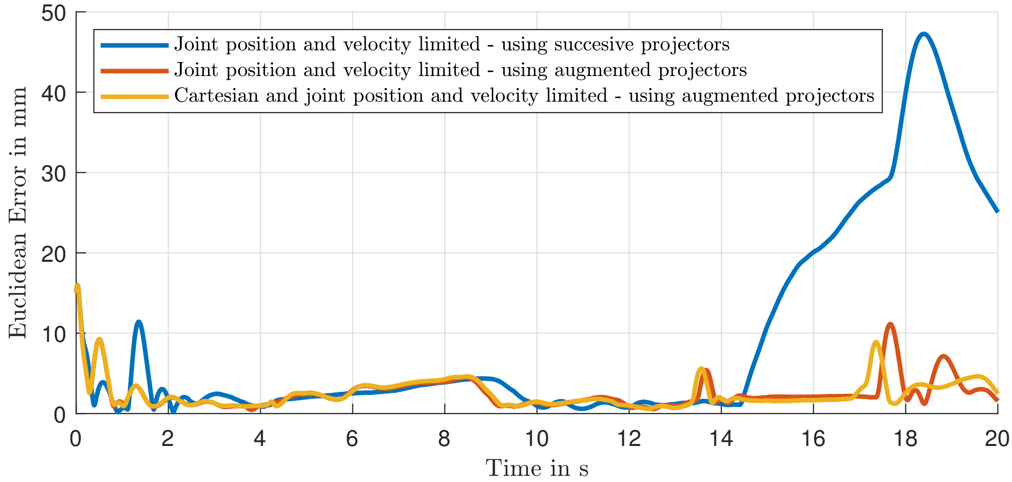


Figure 4.5: Euclidean error of the end effector position while following the circular trajectory. The use of successive projectors (as in chapter 3) results in a soft hierarchy, while using augmented projectors results in a strict hierarchy

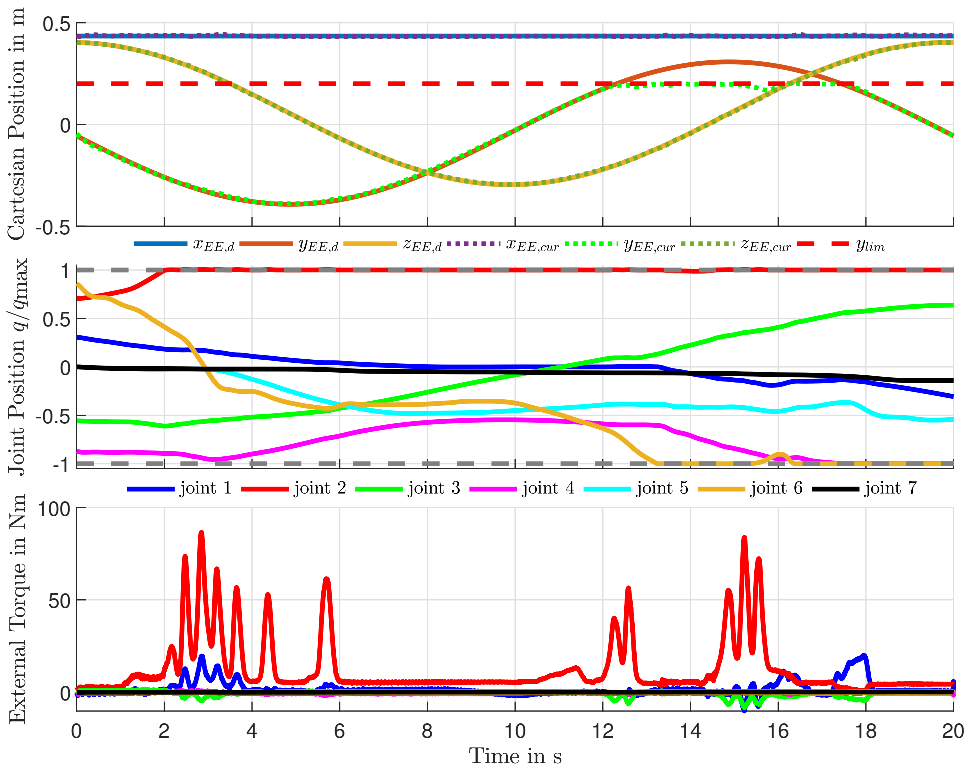


Figure 4.6: Cartesian and Joint Positions during pHRI and with a conflict between the Cartesian limit and the Cartesian position task

Figure 4.6 shows how the end effector was able to follow the desired trajectory, while the human inserted external forces (see fig. 4.6 bottom). When the end effector reached its limit in the coordinate y, the desired trajectory in this coordinate was sacrificed in favour of the limit (as intended). This solution of the conflict was given thanks to the implementation of the singularity avoidance approach presented in section 2.2.3. Without it, the intertia matrix $\hat{\mathbf{\Lambda}}$ can not be properly computed using eq. 2.6, because the Jacobian $\hat{\boldsymbol{J}}$ becomes singular[5] due to the conflict between the active constraint and the task.

To test the performance of the approach at high speed, the first experiment was repeated with a higher velocity of the circular trajectory (2.51 rad/s). Table 4.1 shows the limits used for these experiment. After ten seconds, a human interferes applying external forces. Figure 4.7 proves the tracking performance and the limits saturation achieved using the algorithm. At this speed, the interaction of the human is limited for safety reasons. Note that even though the task is not perfectly followed due to the human intervention, the limits are always respected. However, these can not be called hard limits, because a torque controlled robot cannot strictly enforce limits in the presence of high inertia forces.

So far, the approach was tested using the task hierarchy based on null-space projectors. The following experiments (section 4.6.2 and 4.6.3) were done using the QP based scheme to prove the independence of the constraints definition and the task hierarchy solver. Simulations are run in MATLAB/Simulink using the qpOASES library [FKP$^+$14] as QP solver. The control code was compiled to run in the real time Sunrise Controller of KUKA for experiments 1 and 2. Both, simulations and real experiments were run at a control cycle of 2 ms.

4.6.2 Rotational Cartesian constraints

For testing the limitation of rotational coordinates as shown in section 4.5, two experiments are designed: the robot must grasp a cup (experiment 1) and place it on a table (experiment 2). To perform these two tasks, typically a programmer would define 2 frames (one to grasp the cup $\mathcal{F}_{\mathrm{g}}$ and one to place it on the table $\mathcal{F}_{\mathrm{p}}$). If the tasks are performed separately, an initial frame $\mathcal{F}_{\mathrm{in}}$ is needed and a trajectory is planned with fully defined orientation and position for each point. This classic programming approach is illustrated in fig. 4.8. From the figure, it can be seen that there is no need to have a fully defined frame nor a fully defined trajectory.

Two independent motions are required:

- Motion 1 (M1): from $\mathcal{F}_{\mathrm{in},1}$ to $\mathcal{F}_{\mathrm{g}}$ (experiment 1)
- Motion 2 (M2): from $\mathcal{F}_{\mathrm{in},2}$ to $\mathcal{F}_{\mathrm{p}}$ (experiment 2)

To grasp the cup, it is not necessary to go exactly to $\mathcal{F}_{\mathrm{g}}$. The z direction and the orientation around $\boldsymbol{z}$ can vary and the robot could still grasp the cup (see fig.4.9(a)). Therefore, the task is defined only in 4 coordinates (2 for position $[x, y]$ and 2 for orientation, orientation: around $\boldsymbol{x}$ and $\boldsymbol{y}$ $[\theta_{\mathrm{x}}, \theta_{\mathrm{y}}]$). A

[5] Rows of the Jacobian become linearly dependent.

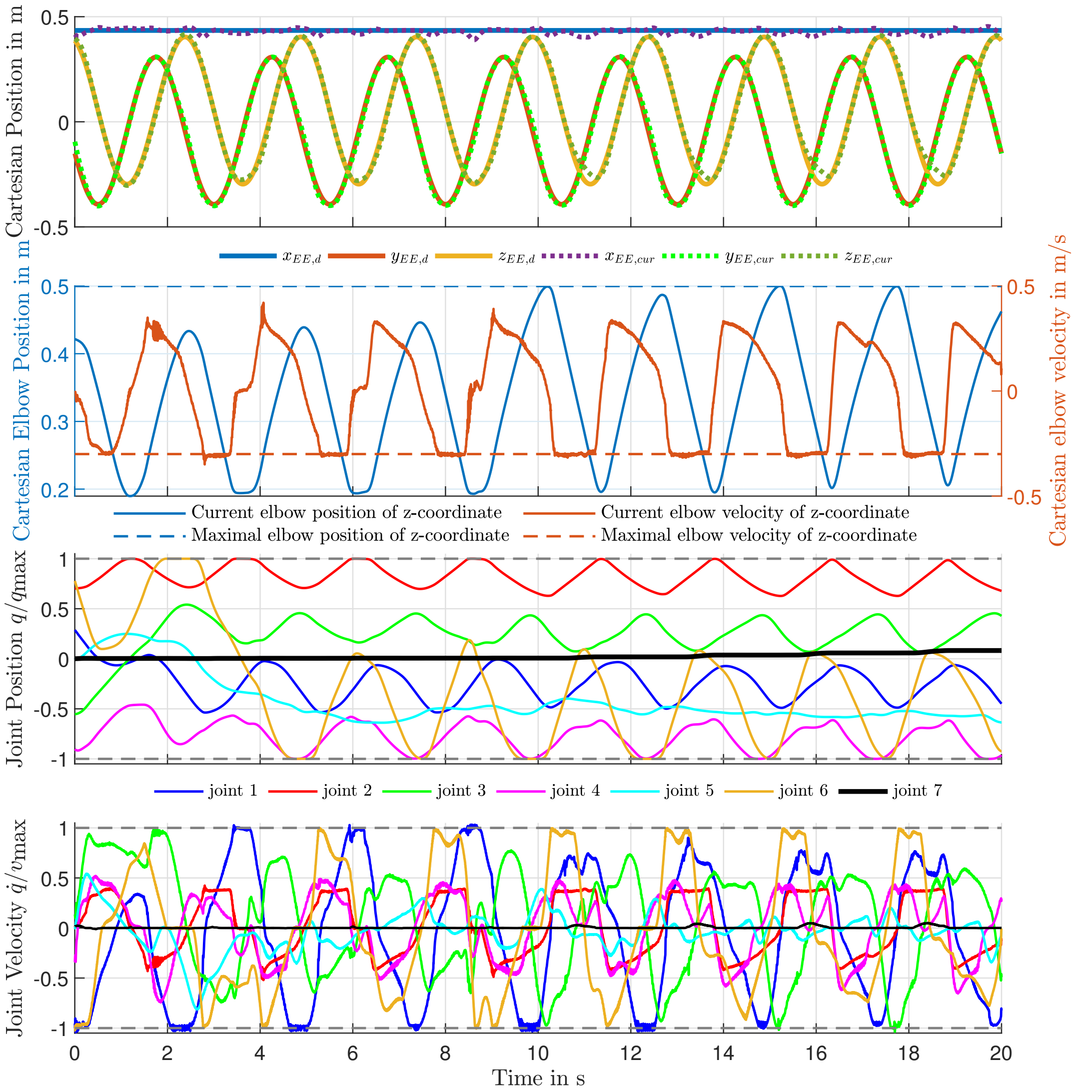


Figure 4.7: Plots of second experiment with increased speed of circle trajectory to 2.5133 rad/s. The dashed gray lines represent the joint space limits

secondary task is added in z-direction. It is expected that the motion in this direction is scarified if required, while the first task will be perfectly executed. However, this direction and the orientation around it must be limited to have a successful grasping: $z_{\text{min}} \leq z \leq z_{\text{max}}$ and $\theta_{\text{z,min}} \leq \theta_z \leq \theta_{\text{z,max}}$.

On the second experiment. Placing the cup only requires the z-location of $\mathcal{F}_{\text{p}}$ (distance to the table must be zero). Orientations θ_{x} and θ_{y} are limited, e.g., to not spill the content of the cup. Positions x and y must be limited to place the cup in a specific region on the table represented by the blue zone in fig. 4.8. The limitation of x and y comes from 4 planes that converge to the desired zone of the table. The limitation of the orientation is illustrated in fig. 4.9(b), and the planes that limit the motion are shown in fig. 4.11(a) and 4.11(b). This setup replicates the experiment in [QA19].

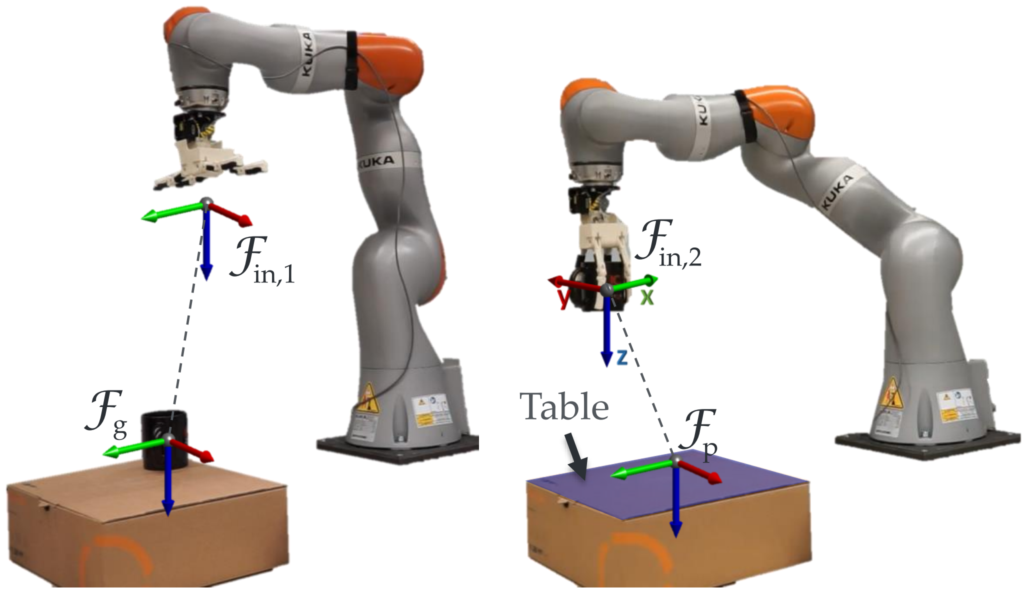
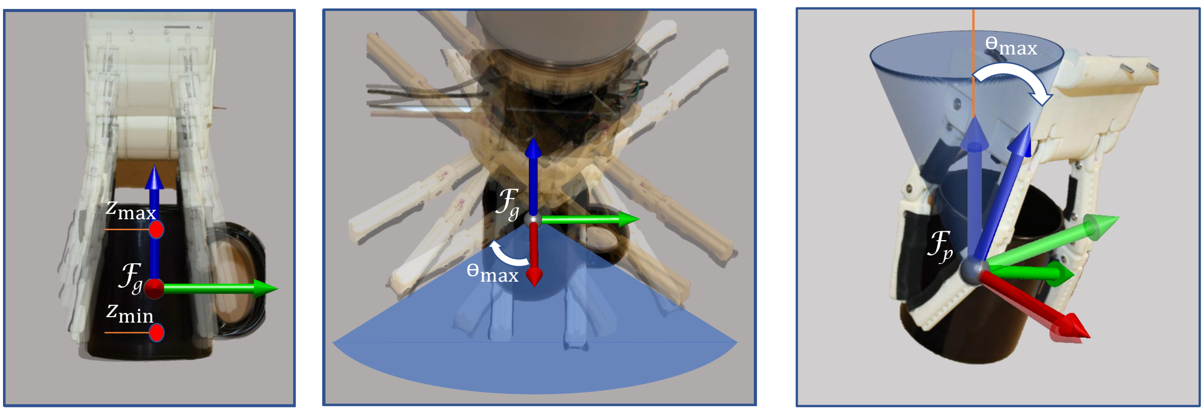


Figure 4.8: Visualization of frames for grasping (left) and placing (right) the cup. The blue zone represents the placing zone.



(a) (b)

Figure 4.9: (a) Grasping example. The frame can move along axis z (blue arrow) between z_{min} and z_{max} The red arrow (y-axis) must be inside the blue zone given by θ_{max}. (b) Placing example: the blue vector of $\mathcal{F}_{\text{p}}$ must be inside of the blue transparent cone. The transparent frame is $\mathcal{F}_{\text{ref}}$.

Experimental setup

As only final frames are defined (in 6 or less coordinates), there is no desired trajectory. The desired acceleration is calculated as follows:

$$\ddot{\boldsymbol{x}}_{\mathrm{d}} = -\boldsymbol{D}(\dot{\boldsymbol{x}}_{\mathrm{c}} - v\dot{\boldsymbol{x}}_{\mathrm{d}}), \tag{4.27}$$

where $\dot{\boldsymbol{x}}_{\mathrm{d}} = \boldsymbol{K}^{-1}\boldsymbol{D}(\boldsymbol{e})$ is the desired velocity calculated with error $\boldsymbol{e}$ between the current $\boldsymbol{x}_c$ and the desired pose $\boldsymbol{x}_{\mathrm{d}}$;[6] $\boldsymbol{K}$ and $\boldsymbol{D}$ are the stiffness and damping parameters. The scaling velocity factor v is calculated as follows:

$$v = \min(1, \frac{s_{\max}}{\|\dot{\boldsymbol{x}}_{\mathrm{d}}\|_2}). \tag{4.28}$$

The magnitude of the vector $\dot{\boldsymbol{x}}_{\mathrm{c}}$ is bound to the maximum speed $s_{\max} = 0.4$ m/s and points towards the desired point $\boldsymbol{x}_{\mathrm{d}}$. The joint limits are the limits in table 4.1, except for joint 6. The limits of that joint plus the limits in the Cartesian space are shown in table 4.3. The velocity of joint 6 was changed to force a velocity saturation of that joint. The coordinates $x_{\mathrm{p},i}$ with $i = 1..4$ represent the coordinate direction towards each of the planes.

Simulations were performed to compare the results with the implementation on the real robot, as detailed below.

Simulation results

Experiment 1 and 2 are simulated using Matlab and Simulink. Ten random frames $\mathcal{F}_{\mathrm{in}}$ give the starting joint configuration of the robot. For experiment 1, two planes for every motion are defined to converge towards 2 cm vertically (coordinate z) from $\mathcal{F}_{\mathrm{g}}$. Figure 4.10(a) shows all final configurations of the robot and the traveled Cartesian position by the end effector. In all

Table 4.3: Limits set for grasping and placing the cup experiments

Coordinate	$\boldsymbol{c}_{\max}$	$\boldsymbol{c}_{\min}$	$v_{\max}$	$v_{\min}$	$a_{\max}$	$a_{\min}$
Rotational	deg	deg	deg/s	deg/s	deg/s^2	deg/s^2
q_6	115	-115	15	-15	300	-300
θ	25	-inf	20	-20	20	-20
Translational	m	m	m/s	m/s	m/s^2	m/s^2
$x_{\mathrm{p},1}$	d_{o}- 0.04	-inf	0.1	-0.1	0.2	-0.2
$x_{\mathrm{p},2}$	d_{o}- 0.04	-inf	0.1	-0.1	0.2	-0.2
$x_{\mathrm{p},3}$	d_{o}- 0.04	-inf	0.1	-0.1	0.2	-0.2
$x_{\mathrm{p},4}$	d_{o}- 0.04	-inf	0.1	-0.1	0.2	-0.2

[6] The error of the rotational part is computed using quaternions.

cases, the motion converges to a pose that satisfies the constraints, where the cup can be grasped. The differences of the motions and the final robot configurations depend on the initial robot configuration. The velocity directed towards the plane is also limited (see Fig. 4.10(b)). Figure 4.10(c) shows the saturation of the orientation in 8 of 10 cases. One of the ten cases is chosen (plotted in green) and joint position and velocities are plotted. The joint positions of joint 4 and 6 are limited but the robot continues moving to the objective.

Figure 4.11 shows the simulation results of experiment 2. The rotation and the angular velocity are saturated to their maximal values for some of the tests. Joint position 4 as well as velocity of axis 6 are saturated for the case of study. Note that in all tests the controlled point slides on the planes and the distance in z-coordinate is minimized. Thus, the robot is able to place the cup on the specific zone (represented by the blue square) without tilting the cup more than 25 deg from a vertical line (this angle is measured as showed in fig. 4.9(b) between the two blue axes). Velocities directed towards the planes are not shown for sake of space, but results where similar than for experiment 1.

Experimental results

Experiment 2 is repeated on the real robot with the same parameters and starting configurations. Results in fig. 4.12 show the performance of the control on the real robot. Figure 4.12(a) and 4.12(b) illustrate the robot motion performed from recorded data. It can be observed that the robot always places the cup in the desired zone of the table. Nevertheless, the motion to achieve the goal is different than the one done in simulation. As the dynamic model is not perfect and no desired trajectory must be followed, the algorithm computes the end effector motion to minimize the distance to the table depending on the constraints. As expected, some of the limits (especially the rotational limit) are exceeded (see fig. 4.12(c)). The over peaks of 2 deg observed in the plots are (proportionally) considerably higher than the limit violation in translational coordinates or in joint space. This excess is given because the force estimation in the rotational coordinates of the end effector is very sensitive to inaccuracies of the dynamic model (at least for the kinematic and mechanical structure of the LBR iiwa).

4.6.3 Dynamic obstacle avoidance

The dynamic obstacle avoidance is evaluated on two LBR iiwa robots. The obstacle avoidance constraint is implemented only in robot 1 (right robot in fig. 4.13). Robot 2 (left robot in fig. 4.13) emulates a dynamic obstacle. Both robots must perform a task simultaneously. Their end effectors must reach a common target point midway between the two robots and come back to the initial Cartesian pose. The robots start moving from the same initial configuration $q = [0, \frac{\pi}{9}, 0, \frac{-\pi}{2}, 0, \frac{\pi}{2}, 0]^T$ rad, but their bases are separated by 70 cm in the y-axis. It is expected that when both robots start moving towards the target point, robot 1 should slow down and wait for

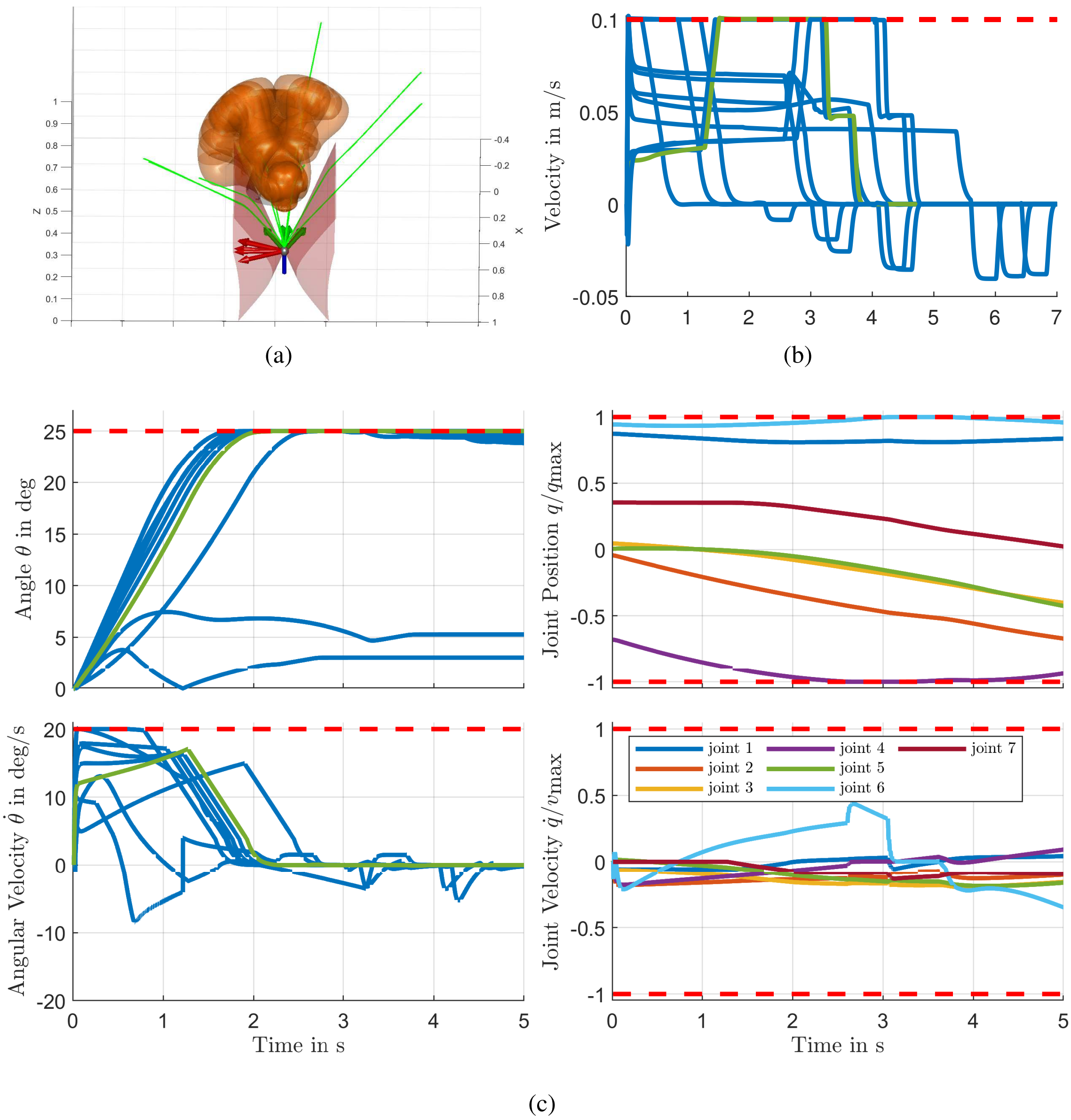


Figure 4.10: Simulation results of grasping a cup. (a) traveled position by the desired point in green from initial position to target frame. (b) velocities from the point of interest to the shortest point on the planes. (c) angular position and velocity of the point of interest (left). Joint positions and velocities of one case of study (right)

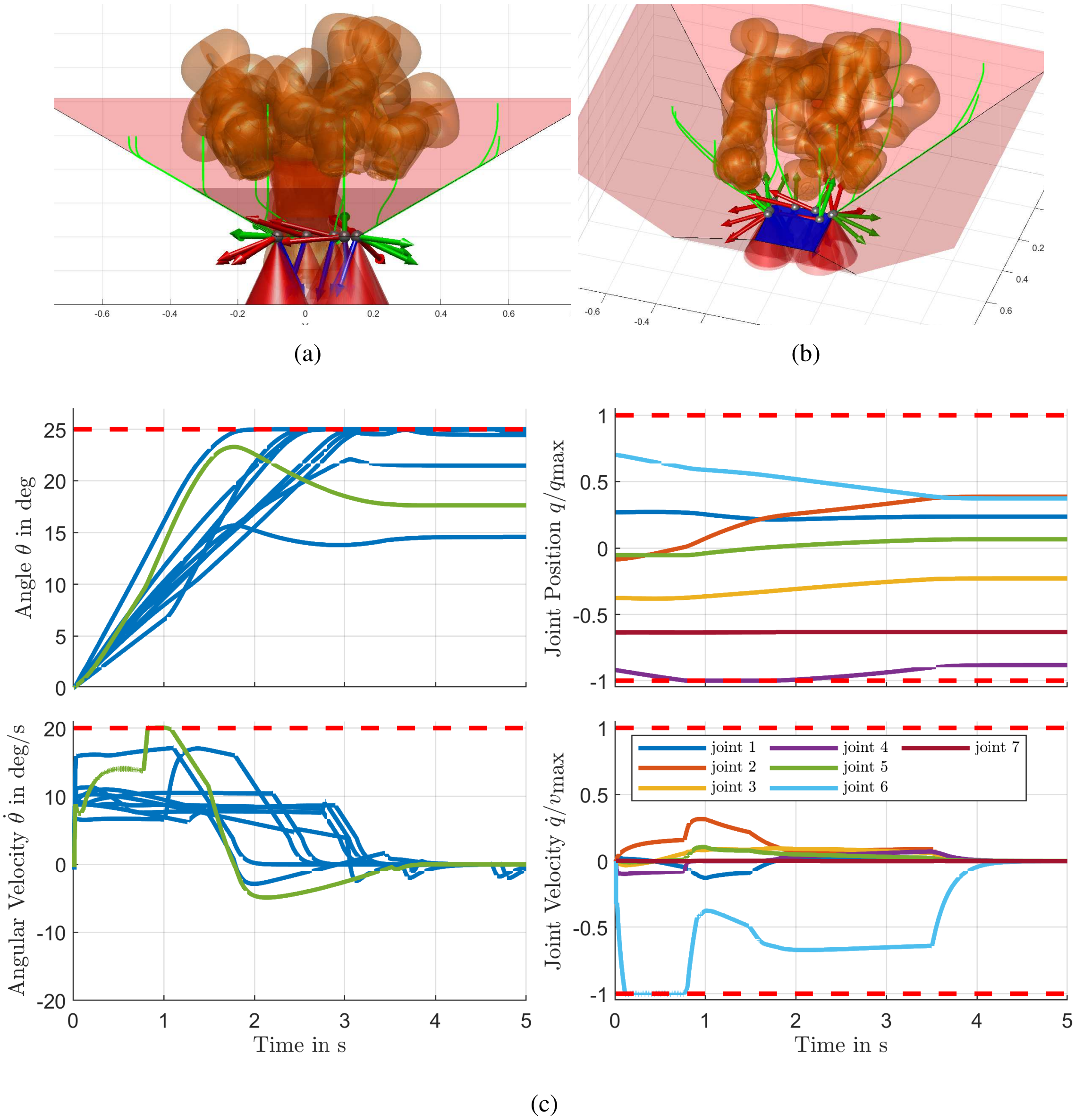


Figure 4.11: Simulation results of placing the cup. (a) and (b) travelled position by the desired point in green from initial position to target table zone (blue area). (c) angular position and velocity of the point of interest (left). Joint positions and velocities of one case of study (right)

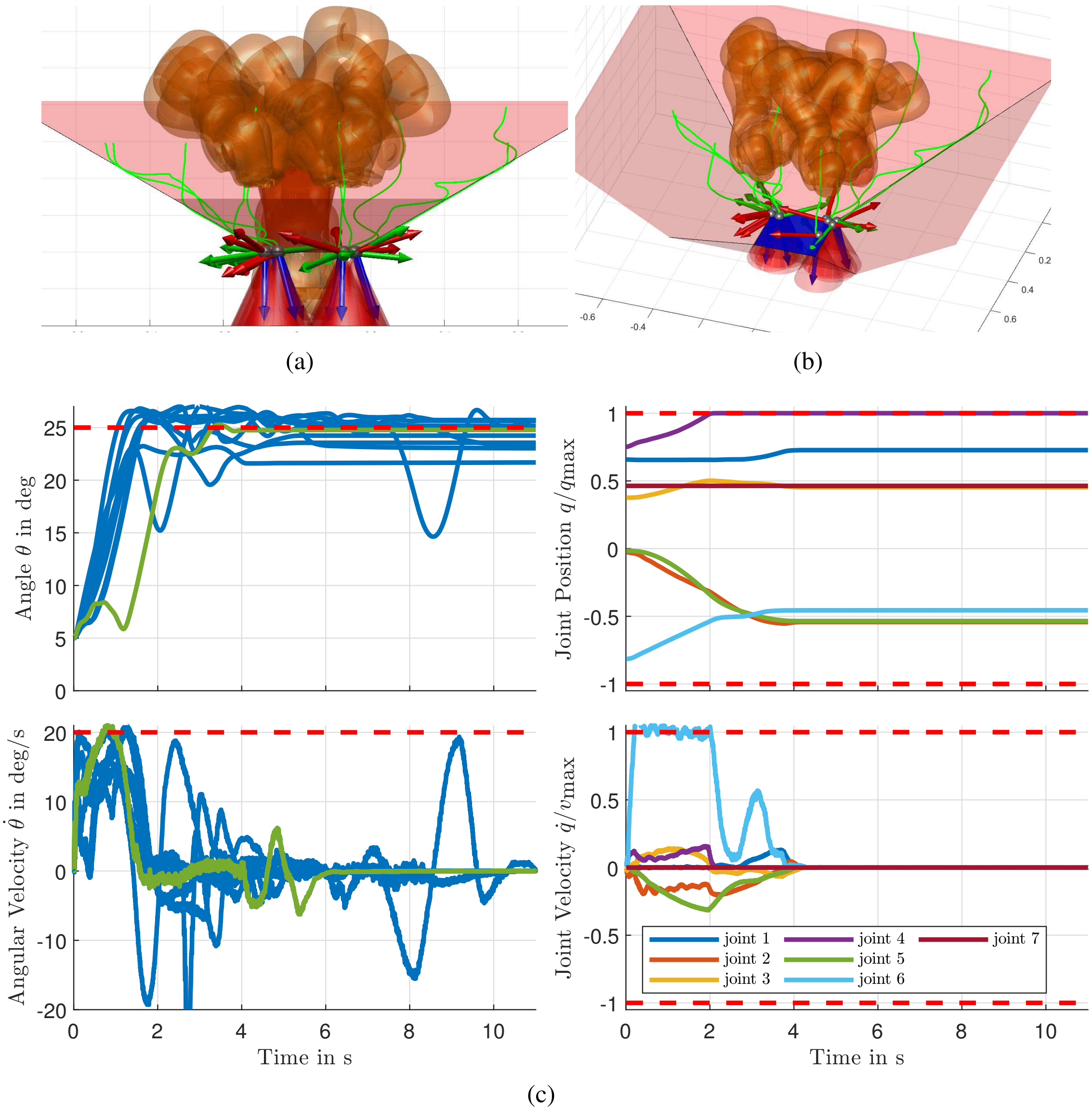


Figure 4.12: Results on a the real robot - Placing the cup. (a) and (b) traveled position by the desired point in green from initial position to target table zone (blue area). (c) angular position and velocity of the point of interest (left). Joint positions and velocities of one case of study (right)

robot 2 to reach the point and move back. Robot 1 slows down as it estimates the velocity of the moving obstacle. The experiment was performed using the FRI (Fast Robotic Interface [SSB10]) of KUKA connected to a distance calculator module using a virtual interface.

To test the approach with pHRI, external forces are applied between $t = 0.5$ s and $t = 4.9$ s, while both robots are moving towards the common target point. Fig. 4.14 bottom shows the external torques plotted with and without interaction. Without interaction, the estimated external torques are not exactly zero due to inaccuracies in the dynamic model.

Fig. 4.14 top shows the shortest distance between the different robot links and the obstacle. As the robots are approaching each other, the distance decreases smoothly until it reaches d_s = 0.04 m for link 5. Links 6 and 7 do not reach this minimum distance. With interaction, the shortest distance for all links crosses the limits. The lowest value for links 5, 6 and 7 were 0.0304 m, 0.0321 m and 0.0308 m respectively.

Fig. 4.14 center shows the relative velocity for the closest points for links 5, 6 and 7 plotted over time. At the start, both robots are coming together, the relative velocity saturates at 0.4 m/s as robot 1 slows down. The relative velocity is reduced as robot 2 stops. Then, it decreases again as robot 2 goes back to its initial position and robot 1 moves to the target point. However, with interaction, robot 1 is pushed towards the second robot, which causes the increase above the maximum relative velocity. The change of velocity that occurs between t = 1.5 s and t = 2.5 s is the consequence of pushing the main robot in the direction of the second robot and releasing it. A deeper analysis of the obstacle avoidance using QP-based hierarchy can be found in [Abd20].

4.7 Discussion

This chapter generalizes the definition of constraints in different spaces at different levels. The constraints were extended to handle Cartesian rotational coordinates and dynamic obstacles. The resulting inequalities can be easily included in any of the task hierarchy solvers: projector-based in section 2.2 and qp-based in section 2.3.

Having unilateral constraints demonstrated to exploit the redundancy better than having a fully defined goal pose. Even without off-line path planning, the robot was able to perform tasks like placing a cup on a table starting from different configurations. This was only possible by increasing the redundancy (i.e. reducing the DOF of the controlled coordinates) and limiting other coordinates to a maximal tolerable "error".[7] Results showed that the constraints were respected while the task was fulfilled as good as possible. Limits were never exceeded by more than 1% in position, 2 deg in orientation and 2% in velocity during pHRI. In a real-life application, a user could set the programmed limits below the real hard limits. More important for physical interaction is the smooth

[7] Error here means the difference from the current value of a limited coordinate to the middle value between the max and min value. In this sense, the task is defined to be in a range, where the task can be accomplished.

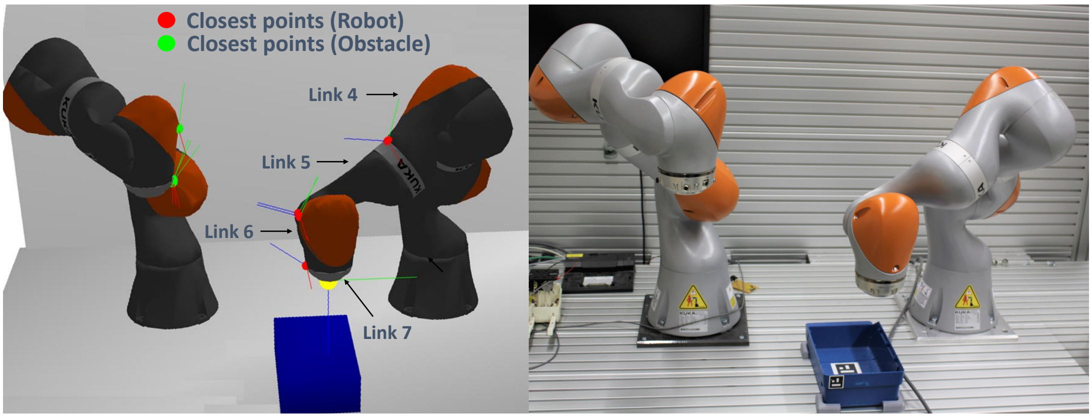


Figure 4.13: Experimental setup for the dynamic obstacle avoidance experiment. Virtual environment for distance calculation (left). Real setup (right)

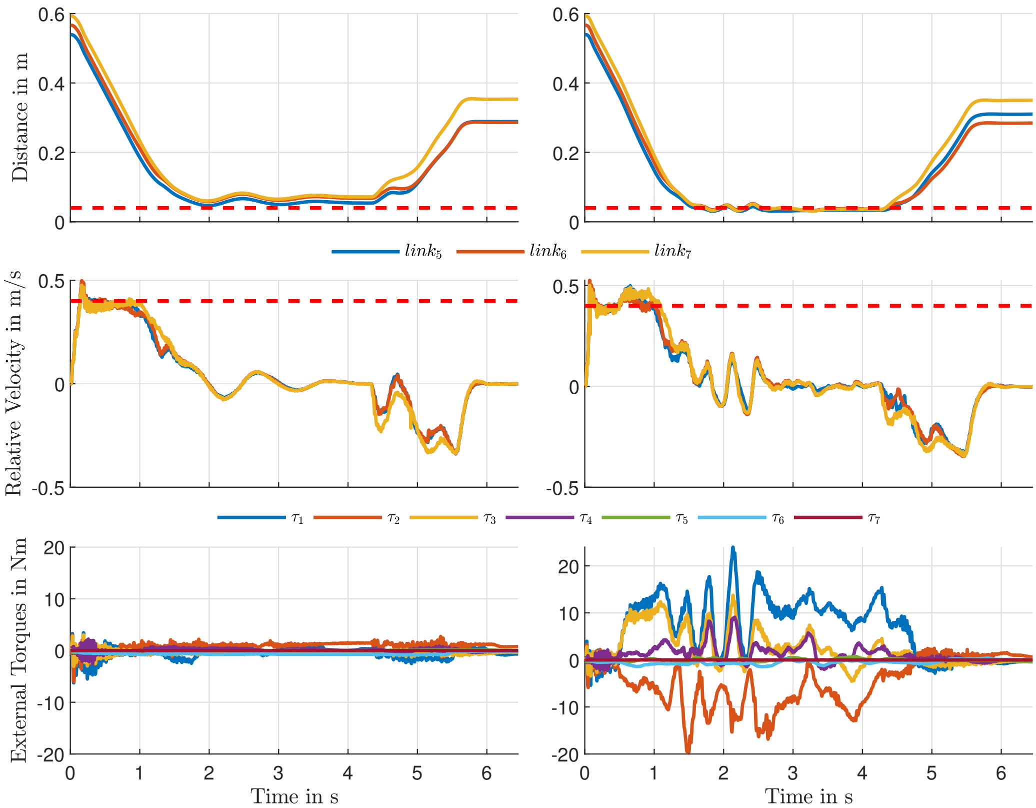


Figure 4.14: Results of the real robot of the dynamic obstacle avoidance experiment. Distance from closest point on first robot to second robot (top). Relative velocity between the two points (middle). External torques (bottom)

behaviour produced by this approach. The advantage of the proposed method compared to classic methods was demonstrated in chapter 3. As the majority of the approaches rest upon repulsive potential fields to avoid limits [Kha85, KSPW04, SK05, DWASH12, LVYK13], the repulsive force highly depends on the robot configuration and robot speed. Using the approach proposed in this chapter, the robot presents a smooth behavior without the need of empirically setting parameters to define the repulsive force. This is a remarkable aspect given that the approach was tested under low and high speed motions.

Regarding obstacle avoidance, the robot must avoid unforeseen changes in the environment. For instance, let the obstacle be a second robot that is being controlled by a human using a teleoperation system. The motions of the obstacle would change unexpectedly. Therefore, pre-planning the robot motion of the first robot is not possible. The performance of the method was evaluated using two LBR robots. One of the robots must avoid the other one, while it tried to perform a single task. The obstacle avoidance was successfully performed even when the human inserted external forces to intentionally produce a collision.

The inclusion of velocity information of the obstacle, improves the reaction of the robot to avoid collisions. The robot reacts and changes its motion to approach the object (robot 2 for our experiments) with low velocity. However, estimation of the obstacle velocity is required. The use of a second robot as a obstacle was intentionally chosen. The encoders of both robots facilitate the estimation of the relative velocity between the them. If the obstacle were a human or any moving object without internal sensors, the velocity should be estimated using any kind of external sensor like a vision system [CDCK06, CY14]. In that case, the control must be extended to deal with the low bandwidth of cameras. Other solutions have been proposed to avoid this problem, as estimating the velocity from the variation of a repulsive vector [FKLK12]. In any case, the algorithm presented here must be analyzed and probably extended to deal with such problems. For instance, a poor estimation of the velocity can lead to oscillations when the velocity saturation occurs.

The two strict hierarchy schemes presented good performance even under pHRI. However, a comparison using the two schemes was not deeply done. The difference between the schemes and its performance under pHRI is discussed in chapter 6. In the next chapter, the definition of the task is discussed and a method to allow a smooth pHRI during task execution is presented.

5 Task definition to maximize pHRI

The previous chapters treated the inclusion of constraints in the framework using different solvers. The main requirements of the framework stated in section 1.3 were accomplished. Constraints have the highest priority, while tasks are performed in a hierarchical manner. However, the common definition of a task (see section 2.1.1 and 2.1.2) does not fulfill requirements for any human robot collaboration application. In some cases, the robot control requires more flexibility. For instance, an impedance controller with constant stiffness is not the best strategy for robotic assisted neurorehabilitation. A better approach is to adapt the impedance parameters according to the force exerted by the patient. A review of controllers used for this kind of application is offered in [PCRBJ16].

Different definitions of the task have been proposed in the past to achieve more flexibility and embrace more application fields: pure force control, hybrid force-position control [RC81, Kha95, DLM92], variable impedance [DR18, GKB11, KKB14, KZK17, KB14] and safety awareness control [RCT$^+$18]. This chapter seeks to enhance the pHRI in human-robot collaboration tasks. The human must not feel in danger due to fast unexpected robot motions. In other words, the robot must behave "friendly" to the human, while being safe. Different approaches have been proposed to have a robot controller with this aim. The authors in [HASH09] present an evaluation overview of safety in human robot interaction. Various aspects of the most significant injury mechanisms are covered in order to quantify the potential injury-risk emanating from manipulators. The experiments demonstrated that no physical collision detection and reaction mechanism is fast enough to reduce the force of fast impacts for rigid manipulators. The impact velocity has the biggest influence in the injury severity. Furthermore, collision detection methods [DAHH06, HADH08] use the estimation of external forces based on joint torque sensors, which implies a dependency on the accuracy of the dynamic model of the robot. Therefore, to prevent injuries to the human if an impact with the robot occurs, the robot velocity must be saturated according to the standards and regulations. For collaborative robotics, the safety requirements are presented in the ISO/TS 15066 of 2016 [ISO16].

Another key aspect in safety for human robot collaboration applications is the avoidance of clamping between the human and the robot, or the human, the robot and external objects. Approaches, like switching the mode to gravity compensation mode after collision occurs [ASOH07] or variable stiffness based on energy limitation methods [RCT$^+$18], allow the human to get out of a clamping situation. However, these methods rely on choosing the proper energy amount for performing a task, which is not intuitive for a human.

Impedance control with fixed stiffness and damping parameters can be used to solve redundancy and to keep contact with an unknown object surface. Nevertheless, the controller cannot react efficiently against the nature of unpredicted perturbations. The controller performance is not always suitable when the environment is dynamically changing. For this reason, adaptive controllers have been recently highly researched with the aim of achieving more convenient results. For instance, [DR18] shows a comparison between fixed and time-dependent variable impedance controllers. The variable impedance controller displayed to be more accurate than a controller with low constant stiffness, but more compliant than a stiff impedance controller. However, a time-dependent description of the path implies a changing reference point in time. If the robot is stopped by an object or a human, the reference point moves further with time. This leads to a high spring force between the current and the reference point and leads to an increased potential energy of the spring.

Another technique used to determine the impedance parameters is the implementation of machine learning [KB14, GKB11]. External force/torque sensors enable the controller to learn the impedance parameters based on measurements of the external forces. Although these methods could enable the dynamic physical interaction with a human, the model requires accurate feedback information about external forces and the learning is done for every specific trajectory. Besides, with unpredictable human interaction, there is not guarantee that the controller is going to be correctly tuned. Therefore, a stable behavior can not be ensured. [KKB14] develops an unified motion and impedance controller. This approach is contrary to classical impedance control, where the feedback and planning of the motion is done by a different controller. A set of target points define a path, but only the final target must be accurately reached. The other targets are attractions points that change the robot motion towards the final target point. The approach was improved in [KZK17] to achieve better path tracking by learning potential functions of each target.

This chapter proposes a new time-invariant approach based on the granted patent [Muñ19]. Physical interaction with the human is maximized by allowing the human to intervene at any time while the robot end effector is tracing a path. Results on simulation and on a real robot are shown to evaluate the approach. The experiments on the real robot are carried out with and without physical interaction to show different aspects of the controller. Main parts of this chapter have been published in [MCAZ19].

The MURAB (MRI and Ultrasound Robotic Assisted Biopsy) Project [MUR] is taken as use-case. The aim of the project is to revolutionize the way cancer screening and muscle diseases are researched for patients. The idea is to automatize the ultrasound-scanning process of the breast by means of a robotic system. The light-weight robot KUKA LWR iiwa with a mounted ultrasound sensor to its flange performs the motion to acquire 2D-ultrasound images. The 2D-ultrasound images can be converted to a 3D-volume to be matched with a MRI (Magnetic Resonance Imaging) image. This matching allows to have the high resolution of the MRI in an on-line ultrasound-image. After the lesion is located in the matched image, the robot is positioned to guide the biopsy needle with an extra 3-DOF tool attached to the robot flange. The needle insertion and biopsy are performed by a doctor with on-line information feedback of the matched ultrasound-MRI image. During the ultrasound-scanning phase, the doctor (non-technical person) is expected to work together with the robot. The robot not only should behave safe and be compliant, but it is also expected to fully perform its task despite of human disturbances. The use of learning-based approaches like in [KZK17] is unsuitable for this use-case. The scanning path changes from patient to patient, which requires to learn parameters of the path for each patient. Such approach would increase the time of the MURAB clinical-workflow.

5.1 Time Invariant Motion Controller (TIMC)

The objective of the time invariant motion controller is to define a path to be followed in an unspecified time. Therefore, motion-planning is not required, i.e., there is not a predefined velocity or acceleration profile. The path consists of a series of points or "waypoints" to be reached. Each waypoint acts as an attraction point. A final waypoint of interest must be defined to indicate the end of the point.

Additionally, the controller should consider pyhsical Human-Robot Interaction. Hence, the human operator must have an intuitive comprehension of the robot movements during interaction. For instance, if the operator blocks the trajectory, the robot should neither hit the person nor accelerate uncontrollably to other directions, but it should wait for the operator to move or avoid the obstacle in an intuitive manner. A safe and intuitive robot motion has direct relation to the energy flow of the controller. The next section analyses this aspect.

5.1.1 Analysis of the controller from the energetic perspective

The design of the task as a spring damper system can be described as:

$$\boldsymbol{f} = \left(\frac{\partial E_{\text{p}}(x)}{\partial \boldsymbol{e}_{\text{t}}}\right)^T - \boldsymbol{D}\dot{\boldsymbol{x}}\text{t}. \tag{5.1}$$

The spring part is a gradient of potential $E_{\text{p}}(x)$. Consider only the translational part without loss of generality. The quantity $\boldsymbol{e}_{\text{t}}$ describes the task-space error $\boldsymbol{e}_{\text{t}} = \boldsymbol{x}_{\text{d}} - \boldsymbol{x}$.[1] Such potential field is commonly defined as $E_{\text{p}}(x) = 1/2\boldsymbol{e}_{\text{t}}^T\boldsymbol{K}\boldsymbol{e}_{\text{t}}$. Hence, the total energy of the system is the sum of the kinetic energy of the mass and the potential energy of the spring

$$E_{\text{T}} = E_{\text{kin}} + E_{\text{p}}(x) = \frac{1}{2}\dot{\boldsymbol{x}}^T\boldsymbol{\Lambda}_{\text{t}}\dot{\boldsymbol{x}} + \frac{1}{2}\boldsymbol{e}_{\text{t}}^{\boldsymbol{T}}\boldsymbol{K}\boldsymbol{e}_{\text{t}}. \tag{5.2}$$

[1] The extension to the rotational coordinates can be made by a proper definition of the rotational error, e.g., using quaternions.

The longer the distance from the current to the desired point (i.e., higher error $\boldsymbol{e}_\mathrm{t}$), the longer is the application of the force $\boldsymbol{f}$. This implies a longer time accelerating the system, which leads to high velocities, and so to an increment of the kinetic energy. Additionally, the higher error $\boldsymbol{e}_\mathrm{t}$ causes an increment of the potential energy.

The stiffness of the impedance controller should be computed such that the total energy does not increase with the task-space error. Limiting the maximum desired velocity of the end effector inherently saturates the kinetic energy. If this velocity is saturated by varying the stiffness of the spring, the potential energy will be automatically saturated. In [Kha87a], a rearrangement of the impedance behaviour in eq. 5.1 leads to a saturation of the velocity magnitude to a maximal speed:

$$\dot{\boldsymbol{x}}_\mathrm{d} = \boldsymbol{D}^{-1}\boldsymbol{K}\boldsymbol{e}_\mathrm{t}, \tag{5.3}$$

$$\boldsymbol{f} = -\boldsymbol{D}(\dot{\boldsymbol{x}} - v\dot{\boldsymbol{x}}_\mathrm{d}), \tag{5.4}$$

where v is a scale factor to limit the velocity magnitude to a desired maximum speed s_max and it is defined by:

$$v = \min(1, \frac{s_\mathrm{max}}{\|\dot{\boldsymbol{x}}_\mathrm{d}\|_2}). \tag{5.5}$$

Replacing eq. 5.3 in eq.5.4. The force can be written as an impedance behavior

$$\boldsymbol{f} = -\boldsymbol{D}(\dot{\boldsymbol{x}} - v\boldsymbol{D}^{-1}\boldsymbol{K}\boldsymbol{e}_\mathrm{t}) = v\boldsymbol{K}\boldsymbol{e}_\mathrm{t} - \boldsymbol{D}\dot{\boldsymbol{x}}, \tag{5.6}$$

where the stiffness $\boldsymbol{K}$ is scaled by the factor v when the magnitude of the desired velocity $\dot{\boldsymbol{x}}_\mathrm{d}$ exceeds the maximal speed s_max.

The energy behaves as in eq. 5.2 as long as the magnitude of the desired velocity $\dot{\boldsymbol{x}}_\mathrm{d}$ is smaller than the desired maximum speed s_max, which is when

$$\|\boldsymbol{e}_\mathrm{t}\|_2 < s_\mathrm{max}\boldsymbol{u}^T\boldsymbol{D}\boldsymbol{K}^{-1}\boldsymbol{u}, \tag{5.7}$$

where $\boldsymbol{u}$ is the direction vector of the error $\boldsymbol{u} = \boldsymbol{e}_\mathrm{t}/\sqrt{\|\boldsymbol{e}_\mathrm{t}\|_2}$.

In the saturation case, when the stiffness is scaled, the potential energy can be computed as:

$$E_\mathrm{p}(x) = \frac{1}{2}\boldsymbol{e}_\mathrm{t}^T v\boldsymbol{K}\boldsymbol{e}_\mathrm{t}. \tag{5.8}$$

Thus, if the error increases and the condition 5.7 does not hold, then the stiffness will be scaled together with the potential energy. The kinetic energy is also saturated because the magnitude of $\dot{\boldsymbol{x}}$ should not be bigger than s_max.

Figure 5.1 shows an example of the difference between the energy dissipation using constant stiffness and variable stiffness. The stiffness varies according to a maximum speed $s_\mathrm{max} = 0.1$ m/s. The initial error in both cases is 0.1 m. The maximum energy was reduced significantly by changing the stiffness of the spring.

Based on eq. 5.6, a time invariant description of a path can be created. The control law enables the motion generation between the current state and a desired one. The robot end effector is just able to reach the desired waypoint, but it does not follow a path. This implementation makes the control unsuitable for some applications.

As the goal is to have an adaptable control method, the concept of a path as a set of waypoints is defined. The waypoints are placed in certain order, so that they describe the desired the path.

5.1.2 Controller modes

The impedance variability of the control method is divided in three different state-dependent modes.

Interaction Mode 1: This mode allows an easy interaction with the robot, i.e., the robot does not react with a high force to an external perturbation. The human can easily manipulate the robot. In this mode, the robot does not move quickly, as a prevention measure for possible collisions, $s_{\text{max}} = s_{\text{int}}$.

Transition Mode 2: Due to the substantial change of the velocity and force of the robot between modes, a smooth transition must be implemented. This mode controls the increase or decrease of the kinetic energy. $s_{\text{max}} = S(r)$, where $S(r)$ is a smooth function transition between the interaction speed s_{int} and the trajectory speed s_{tr}, defined as:

$$S(r) = s_{\text{tr}} + (s_{\text{int}} + s_{\text{tr}})(0.5 + 0.5\sin(\pi\frac{(r - r_{\text{i}})}{(r_{\text{e}} - r_{\text{i}})} - 0.5\pi)), \tag{5.9}$$

where r is the distance from the waypoint to the current position, r_{i} and r_{e} are operational radii, which are defined in section 5.1.3.

Trajectory Mode 3: When the robot is close to the proposed path, the stiffness and damping change to generate a faster and more precise motion during the path tracking. This modification increases the rigidity of the robot. The robot is more difficult to manipulate, but the parameters

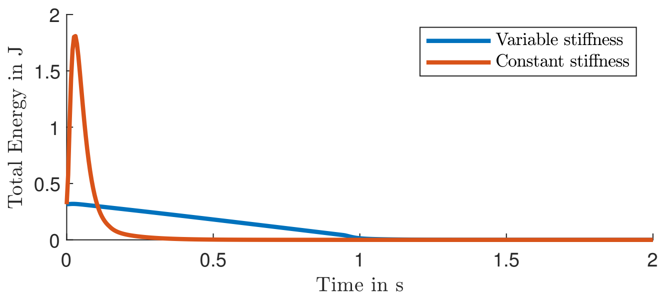


Figure 5.1: Energy dissipation for an impedance behavior with constant and variable stiffness.

are set to allow human-robot collaboration and to ensure safety to the operator. Besides, a high force (e.g., applied by the operator) can move the end effector outside of this mode, and of the path. $s_{\text{max}} = s_{\text{tr}} > s_{\text{int}}$.

Figure 5.2 shows an example in 2D of the different modes. The orange star represents the active waypoint, and the gray ones the inactive waypoints. The trajectory mode 3 corresponds to the light orange section. The smooth transition mode 2 is shown in green. The interaction mode 1 is active when the robot is not on any of the other modes.

Each mode has its own impedance parameters and maximum speed value. A radius defines the active zone of the mode. The user can set each radius to adapt the controller to the necessities and conditions of the application.

5.1.3 Operational radius

For switching modes, as fig. 5.2 illustrates, two distances to the waypoint are defined: the inner radius (r_{i}) and the outer (r_{e}) radius.

The inner radius defines the trajectory zone in which the robot has less compliance but can follow the path more efficiently. The outer radius delimits the interaction zone and its distance to the inner radius sets the length of the transition mode. A third distance is defined to deactivate the current waypoint and activate the next one following the order of the path: the disappearance radius (r_{d}). If the error is less than this radius, the next waypoint is activated to be the current attraction point. Thus, the path is defined as a sequence of waypoints that are traveled independently of time. The disappearance radius also assures that the path will be traced with a maximum Euclidean error, otherwise the tracking stops (the next waypoint is not activated).

5.1.4 Force computation for path tracking

Equations 5.3 - 5.5 compute the force for a single active waypoint. However, this computation is not suitable in case of path tracking. When the distance to the current active waypoint decreases, the desired velocity $\dot{\boldsymbol{x}}_{\text{d}}$ is reduced instead of keeping a constant velocity along the path. To keep a constant desired velocity, the error is not the difference between the current position and the desired one, but it must be the distance to the last waypoint following the path. The direction should be kept towards to the active current waypoint. Therefore, following eq. 5.3:

$$\dot{\boldsymbol{x}}_{\text{des}} = \boldsymbol{D}^{-1}\boldsymbol{K}l\left(\frac{\boldsymbol{x}_w - \boldsymbol{x}}{\|\boldsymbol{x}_w - \boldsymbol{x}\|_2}\right), \tag{5.10}$$

with:

$$l = \|\boldsymbol{x}_w - \boldsymbol{x}\|_2 + \sum_{i=w}^{n_{\text{w}}-1} \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|_2 \,, \tag{5.11}$$

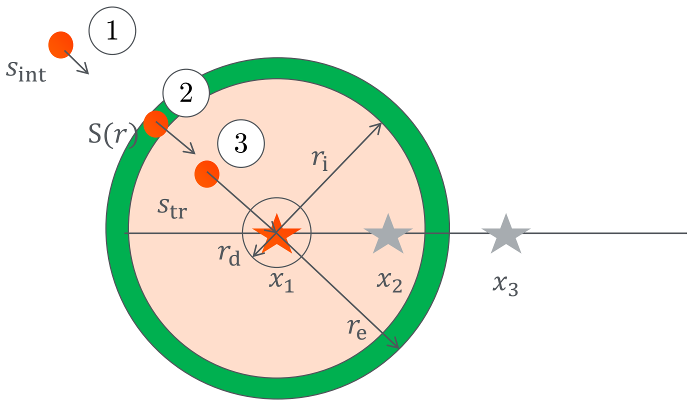


Figure 5.2: Path example in 2D. The orange circle represents the operational point in three different states, while being in the three different modes (interaction 1, transition 2 and trajectory 3). The maximum speed increases while change between modes.

where l is the sum of the length of the path plus the distance from the current position to the current active waypoint (which has the sub-index $[]_w$). The path is defined by waypoints to be traveled from $\boldsymbol{x}_1$ to $\boldsymbol{x}_{n_{\mathrm{w}}}$, where n_{w} is the number of waypoints in the path. Fig. 5.2 shows an example path that goes from the left to the right direction.

5.2 Simulation and experiments

The method is proven under the requirements of the MURAB project. The path of the scanning phase is generated by projecting a desired trajectory on the breast surface. A spiral trajectory around the breast keeps contact of the ultrasound sensor with the breast. The ultrasound should be perpendicular to the surface and to the direction of the trajectory. This example application is appropriate to test the algorithm performance, because the trajectory points must not be traveled in a certain amount of time. With classical methods, a velocity profile should be planned between all points. New points should be interpolated depending on the cycle time of control. Furthermore, this method allows the doctor to intervene during scanning, e.g. to put gel on the tip of the ultrasound sensor without the need of manually changing the control mode of the robot.

The parameters of the controller are shown in table 5.1. To avoid joint limits and have full interactivity, joint constraints follow the approach explained in chapter 4.

5.2.1 Simulation results

The method was tested initially in a simulation environment in Matlab. Three different 3D-models of breasts were used to test the performance and the robustness of the algorithm. Model 1 was modeled in a 3D software. This model is considered an ideal case. Due to the geometric definition

Table 5.1: Parameters of the controller used in the experiments. $\boldsymbol{I}$ is the identity matrix

Parameter	Value	Parameter	Value
r_{d}	0.012 m	s_{int}	0.048 m/s
r_{i}	0.02 m	s_{tr}	0.08 m/s
r_{e}	0.03 m	n_{w}	400
$\boldsymbol{K}$	400 $\boldsymbol{I}$ kg/s^2	$\boldsymbol{D}$	40$\boldsymbol{I}$ kg/s

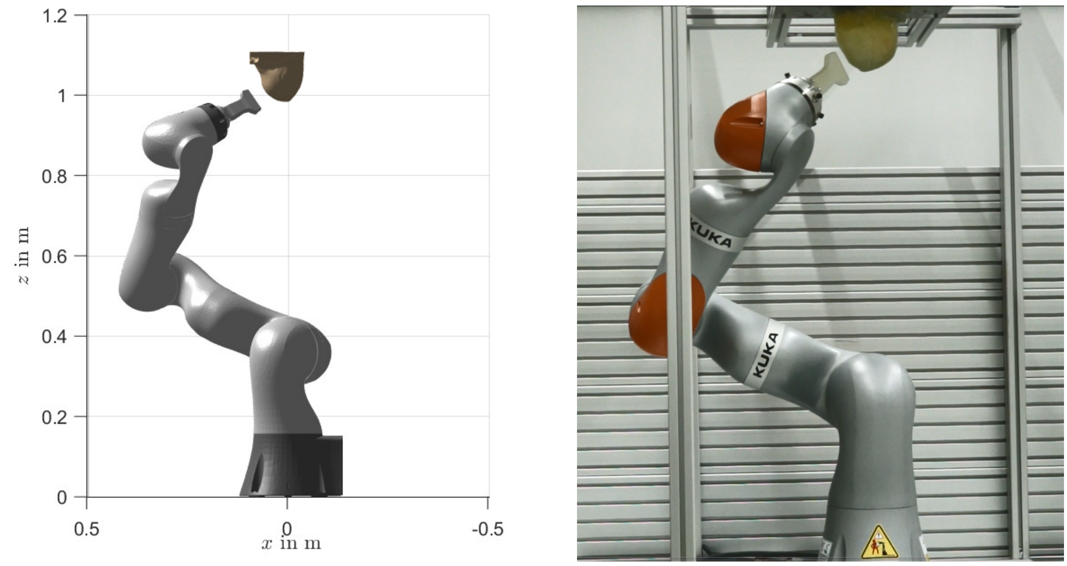
of the model, the projection of the trajectory on the model's surface does not contain discontinuities in the orientation change or in the distance between the trajectory points. Model 2 is a 3D model, which is taken from real MRI data of a patient's breast. The noise of the model was reduced to have a smoother shape using the Laplacian smoothing filter [Her76, SCOL$^+$04] offered by the open source system for editing and processing triangular meshes MeshLab [Mes]. The model in this case is considered as a real case. Model 3 is a 3D model of a puppet used for students' training. The model was recorded with the rc-visard 65 monochrome 3D camera of Roboception [Rob] using the volumetric-3D mapping algorithm described in [SSC14]. This model tests the robustness of the algorithm, when the projection has discontinuities and the orientation can suddenly change due to noise in the model and a poorly modeled surface. The resolution of the models is shown in Table 5.2. Note that a higher number of faces and vertices does not necessarily imply a smother surface. In fact, the higher the number of faces, the more irregular surface areas can be originated.[2]

The breast models are located in prone position centered to a high of about 110 mm from the robot base. Figure 5.3(a) shows the initial setting for the simulation. Table 5.3 presents the simulation results. The three different models are shown with the desired spiral path projected in red on the surface of the breast model. Note that model 3 is noisier and it has blank spaces due to a poor recording of the 3D camera system. The second row shows the desired and the traveled actual path by the robot. The green spheres represent the limit of the area of the transition mode. Outside the green zone, the robot is in interactive mode. The robot starts in this zone and moves at a speed s_{int} to the first waypoint, then it continues following the path. For the three breasts, the path was followed with less than the minimal accepted error, which is the disappearance radius r_{d}. The magnitude of the end effector velocity gets saturated at s_{int} until the end effector reaches the transition radius r_{e}. Then, the speed increases to s_{tr} with a smooth change defined by $S(r)$ as given by eq. 5.9. When

Table 5.2: Resolution information of the 3D models

	Faces	Vertices
Model 1	19434	9719
Model 2	174655	87633
Model 3	212595	132448

[2]This statement depends on the approach to obtain the 3D model and the filtering used in the post-processing. In these experiments, the high resolution was intentionally used to get more irregular surfaces.



(a) (b)

Figure 5.3: Initial configuration of the robot and location of the breast. (a) Simulation, (b) real robot.

the final waypoint of the path is reached, the velocity smoothly decreases following the impedance law of eq. 5.1. The last row shows the velocity profile and the energy progress. The potential energy decreases linearly, as the robot approaches the final waypoint. For model 2, more potential energy is required because the length of the path is longer. The system behaves stable even when the path is poorly prescribed (points have different distances and orientation changes abruptly) for model 3. The velocity suddenly reduces at some points. This reduction is caused by abrupt changes of direction of the path. These changes cause also a non-continuous progress of the kinetic energy. However, the kinetic energy remains below the same limit for the three cases (it is never higher than 0.15 J). The kinetic energy was measured in joint space, i.e., using the joint velocities and the inertia matrix of the robot (see eq. 5.12). This computation considers velocities in the null-space of the operational space and computes the kinetic energy of the whole system as:

$$E_{\text{kin}} = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M} \dot{\boldsymbol{q}}. \tag{5.12}$$

5.2.2 Experimental results

Two experiments were carried out with the real robot. A scanning of the model 2 was performed using a plastic printed dummy ultrasound sensor. The first experiment does not involve interaction of the human. In the second experiment, the human pulls the ultrasound sensor in three different instants. The implementation of the algorithm ran in real-time at a control frequency of 1 kHz.

Table 5.3: Simulation results using three different 3D breast models. The first row shows the spiral trajectory in blue, and its projection on the model's surface in red. The second row shows the path traveled by the ultrasound sensor tip. The green semi-transparent volume delimits the transition mode. The last row shows the speed profile and the energy spent during the path tracking

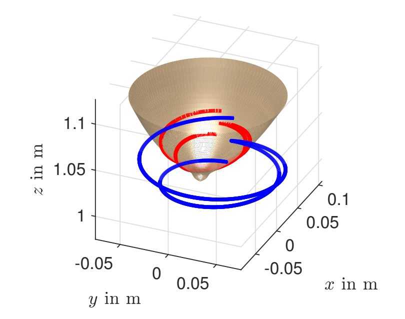
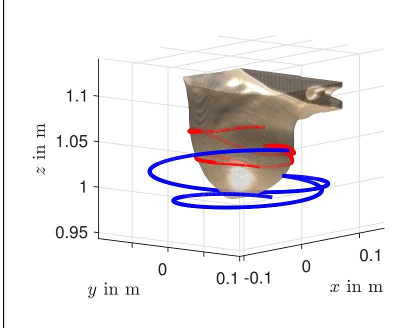
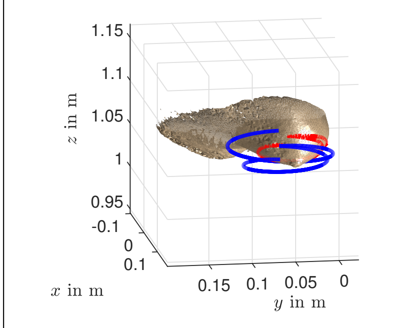
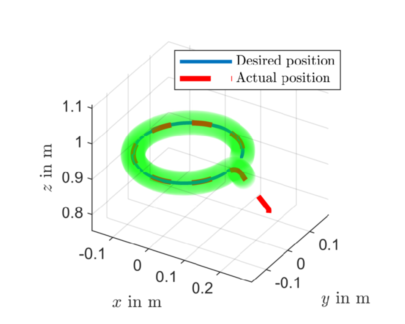
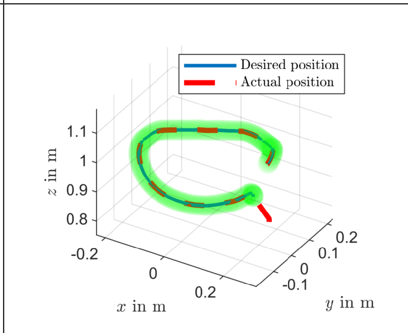
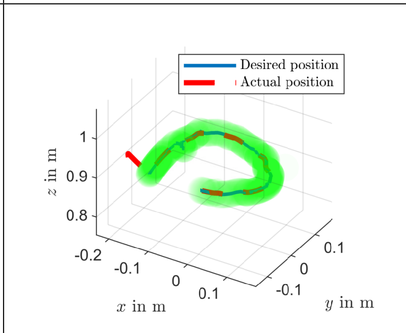
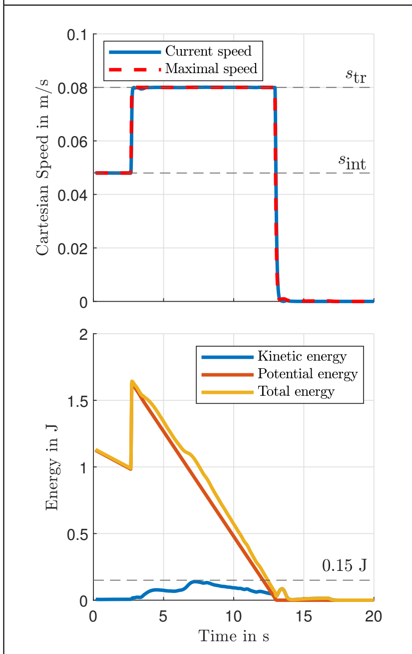
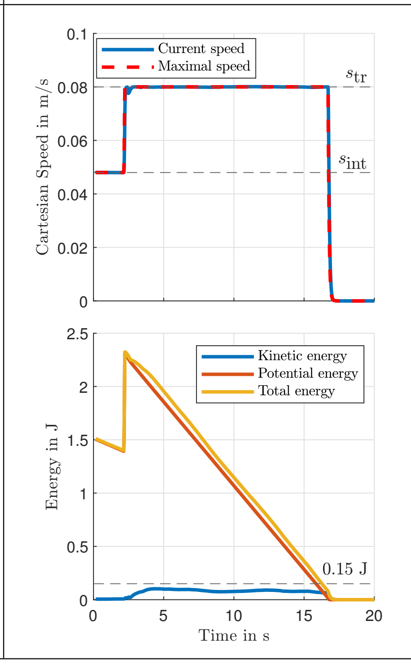
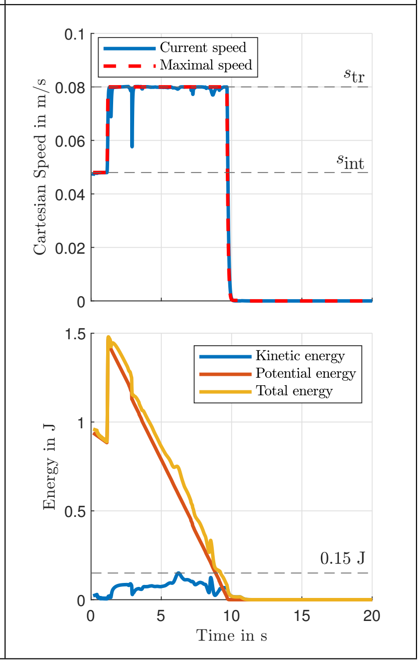
Model 1	Model 2	Model 3
		
		
		

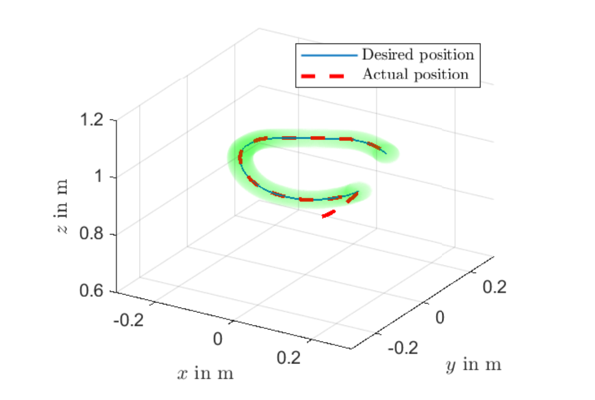
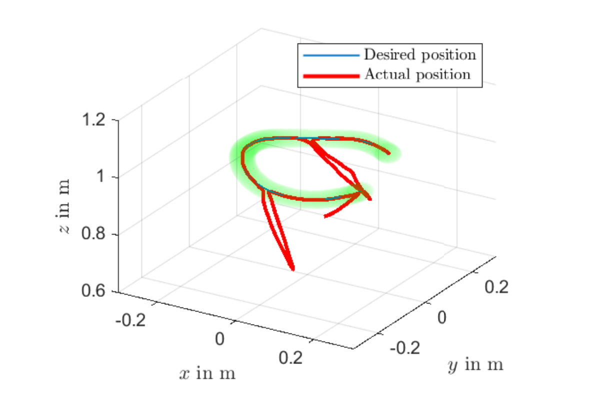
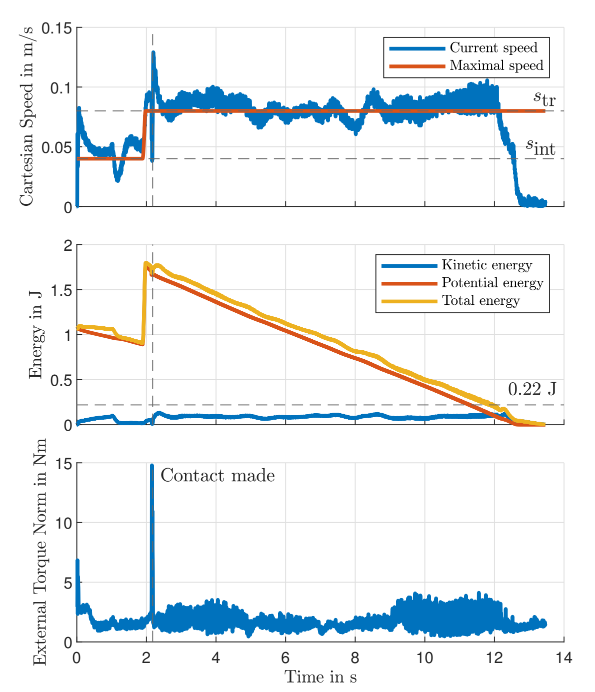
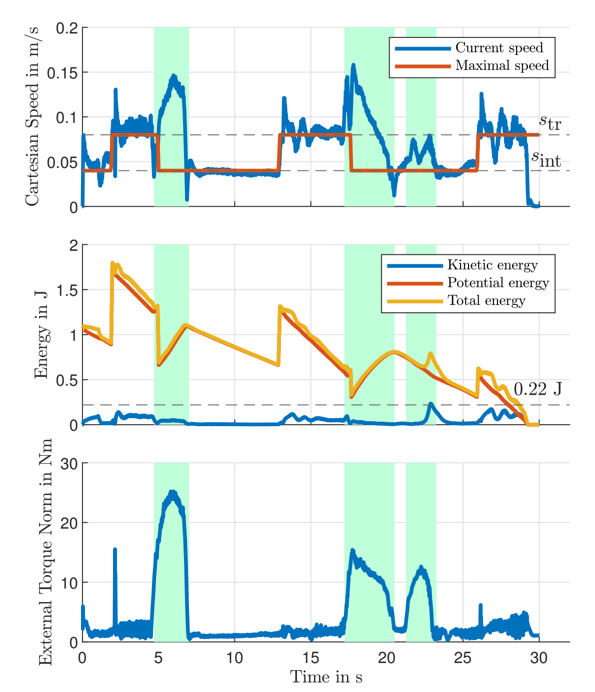
Table 5.4 shows the results of the experiments. When the robot enters the trajectory mode, the path is traced at the maximum speed s_{tr}. Different to the simulation, the speed saturates, but it exceeds sometimes the desired maximal speed. This was expected to happen because of uncertainties in the dynamic model of the robot and, in a bigger proportion, due to noise of the velocity measure (see the high frequency portion of the Cartesian speed signal in the plots). The control to trace the velocity is not robust enough to follow the velocity profile.

The magnitude of the external joint torques[3] shows when there is certain force/torque being applied to the robot. Without interaction, the torque magnitude remains at about 5 Nm. This torque is given by: uncertainties in the dynamic model and the constant contact that the dummy ultrasound sensor has with the phantom. The energy plot is similar to the one in simulation. The total energy decreases linearly and the kinetic energy remains below 0.22 J. In the second experiment, table 5.4 right, the human intervenes in three different instants on time. The green area in the plots of the bottom right of table 5.4 represents the intervals when an external force/torque was applied. After the human takes the end-effector out of the trajectory zone, the robot changes to interaction mode and the maximum velocity gets automatically reduced to s_{int}. Three different aspects can be noticed in the plots of the second experiment:

1. The velocity magnitude saturates only when the robot is not under the influences of external forces. The human can move the robot faster than the maximum established speed.
2. When the human intervenes, energy is injected to the system. Although the robot is taken far off the path, the energy does not significantly increase, as it would happen with fixed stiffness parameters.
3. Duration of the time, when the human holds the robot in the interaction mode, is not relevant. The tracking continues normally afterwards. This behavior is due to the time-invariant definition of the path.

[3]Only torques are considered because the LBRiiwa has only rotational joints and the external forces/torques are measured by the torque sensors of each joint

Table 5.4: Experimental results of scanning of the breast phantom without (left) and with interaction (right). The first row shows the path traveled by the ultrasound sensor tip. The green semi-transparent volume delimits the transition mode. The second row shows the speed profile of the ultrasound sensor tip, the energy spend and the external torques norm applied during the path tracking. The green semi-transparent zone in the bottom right figure denotes when a external force was applied

Without Interaction	With Interaction
	
	

5.3 Discussion

This chapter proposes a method to maximize pHRI for applications, where the human is expected to interrupt the task at any time, and move the robot away. To this end, a time invariant description of the path is presented. This description allows the fulfillment of the task despite human intervention. The algorithm was tested in a medical example application, where certain impedance is needed to perform robotic assisted ultrasound scanning. The time invariant description reduces the complexity of the path by defining it as a chain of waypoints that have to be followed in a specified order. The results show that physical human interaction is achievable and the robot continues with the task without changing the control law. The kinetic energy is maintained low (under 0.22 J). This maximum energy depends on the maximum speed chosen for each mode. The robot behaves "friendly" with the human, i.e., no high forces are exerted against him. The choosing of low speed for the interaction mode facilitates the manipulation by the human but reduces the time in which the robot comes back to finish the task. It was noticed during real experiments that there is no guarantee of having a maximum speed. Therefore, this can not be called a safe limitation. However, by setting the maximum velocities with certain tolerance below the limits given by the standards and regulations, the velocities remain within safe limits. A better way to respect this limit is to set a maximal Cartesian velocity as a constraint, as shown in chapter 4.

The TIMC can be modified to limit the maximal force that the robot can impose in the environment, as shown in [Cas18]. Although this modification must be further studied under different use-cases, it allows to limit the force to a maximal allowed impact force that is stipulated in the norm ISO/TS 15066 of 2016 [ISO16].

In future works, safety aware energy controllers have to be evaluated, and they can be combined with the time invariant description of the method. This method can be combined also with motion-force hybrid methods as [Kha95]. For the MURAB use-case, this method can guarantee the contact force between the sensor and the breast. An external force/torque sensor should be attached to the ultrasound sensor to directly measure the contact force/torque.

6 Evaluation of the framework under pHRI

In the last decades, a lot of research has been carried out towards controlling highly redundant robots to perform complex tasks in unstructured environments. As a result, several robotic systems are able to perform robust task execution and safe interaction with humans and the environment. Humanoid robots, such as ASIMO [SWA$^+$02], Robonaut 2 [DMA$^+$11], REEM [FM14], TORO [EWO$^+$14], ATLAS [KDF$^+$16], THR3 [Toy], NimbRo-OP2 [FFB$^+$18], among many others, are examples of such complex and highly redundant robotic systems. Robotic applications using these systems require handling several tasks simultaneously while not exceeding physical and virtual limitations.

Many different whole-body reactive controllers have been developed to control such systems [KSPW04, SK05, SGJG10, NKS$^+$10, BHG10, BOD10, SRK$^+$13, DWASH12, LVYK13, DS19]. The majority of the approaches are based upon the design of artifitial repulsive/attractive potential fields [Kha85]. Chapter 3 showed the drawback of using repulsive forces to avoid joint limits, and proposed the Saturation in Joint Space (SJS) approach. The SJS approach was extended to deal with any kind of constraints in Chapter 4. Results demonstrated that a stack of constraints can be included in the task hierarchy solvers based on null-space projectors (in section 2.2) or quadratic programming (in section 2.3). The redundancy resolution solves the hierarchy under the requirements of section 1.2 to have a constrained task hierarchical framework. However, a proper analysis of the robot reaction to external forces was not conducted. Specifically, requirement 4.3 in section 1.3.4 was not proven to be accomplished. This requirement specifies that the robot motion in the null-space must follow a desired dynamic response to external forces.

Previous works as [HSG$^+$14, Die16, HLM$^+$18, DSW$^+$18] focused on the response of the framework to external forces in the task-space. Experiments showed that the robot follows a desired dynamic behavior in the task space. However, none of these works did a proper comparison of the behaviour of the null-space dynamics when external forces are applied to the robot. Intuitive robot behaviour requires intuitive behaviour in the task space as well as in the null-space. Even more so, because the mentioned robots above have a lot of DOF and, therefore, a large null-space. Although the methods may follow a desired dynamic behavior in the null-space, they may not have an intuitive reaction to the force. An intuitive response is, as defined by requirement 4.3, a motion in the direction of the applied force. Redundancy must be solved as the nature would do, following physical laws. [BK00] demonstrated that solvers based on null-space projectors and dynamic consistent pseudoinverses minimize the "acceleration energy"[1] and follow the Gauss's principle

[1]Note that the concept of energy does not have the classical meaning due to the extra time derivative involved.

[Gau29]. Considering that Gauss's principle is equivalent to d'Alembert's principle for holonomic constraints. It can be deduced that the control force determined by minimizing the "acceleration energy" would be the force that nature would employ to solve the redundancy.[2] The drawback of these solvers is that they do not allow inclusion of inequalities, as discussed in section 2.5. Although these solvers can be modified to include inequalities at position, velocity and acceleration level (as shown in chapters 3 and 4), inclusion of torque limits is still not possible. Contrary to those solvers, QP-based solvers allow inclusion of torque limits. However, the quadratic problem must be properly formulated to solve the redundancy following Gauss's principle.

This chapter evaluates the reaction to external forces of QP and projector based hierarchy solvers, and proposes a mixed suitable solver for pyhsical Human-Robot interaction under unilateral constraints. Specifically, the contributions of this chapter are:

- a comparison of the dynamic behavior in the null-space of current state of the tasks hierarchy solvers.
- a hierarchy solver that solves the redundancy producing an intuitive reaction of the robot to external forces. The solver considers unilateral constraints at position/velocity/acceleration and torque level.

6.1 The dynamically-consistent constrained task hierarchy solver (DCTS)

Section 2.5 compared the different solvers. Advantages and disadvantages were presented in table 2.1. One aspect not remarked on that table is that QP-based and projector-based approaches use a different minimization function to solve the redundancy problem. This difference influences the behavior in the null-space and its response to external forces.

On the one side, [PMU+08] shows that the use of the dynamic consistent-pseudoinverse of the Jacobian (as given by [Kha80]) is the solution of minimizing the control input using the metric $\boldsymbol{M}^{-1}$ under the constraint of having zero error in the task

$$\min_{\boldsymbol{\tau}} \quad \boldsymbol{\tau}^T \boldsymbol{M}^{-1} \boldsymbol{\tau} \tag{6.1a}$$

$$\text{subject to} \quad \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\tau} = \ddot{\boldsymbol{x}}_{\mathrm{t}} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} - \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\nu}. \tag{6.1b}$$

Depending on the definition of $\ddot{\boldsymbol{x}}_{\mathrm{t}}$, the problem describes an impedance behavior without inertia shaping (IWIS) or with shaped inertia (OSC). A desired acceleration $\ddot{\boldsymbol{x}}_{\mathrm{t}} = \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{f}_{\mathrm{t}}$ solves the problem to have a dynamic behavior defined by IWIS. The force vector $\boldsymbol{f}_{\mathrm{t}}$ is computed as shown in section 2.1.2. To have an impedance behavior with inertia shaping as in OSC, the

[2] "*Since this principle underlies the evolution of constrained motion in mechanical systems in nature*" [Udw03].

desired acceleration must be computed as shown in section 2.1.1, and the external forces must be considered. The problem for OSC becomes:

$$\min_{\boldsymbol{\tau}} \quad \boldsymbol{\tau}^T \boldsymbol{M}^{-1} \boldsymbol{\tau} \tag{6.2a}$$

$$\text{subject to} \quad \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\tau} = \ddot{\boldsymbol{x}}_{\mathrm{t}} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}} - \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\nu} - \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T\boldsymbol{f}_{\mathrm{ext}} \tag{6.2b}$$

As stated in the introduction of this chapter, this solution follows the Gauss principle [Gau29] and minimizes the "acceleration energy" given by:

$$E_{\mathrm{acc}} = \frac{1}{2}\boldsymbol{\tau}^T \boldsymbol{M}^{-1}\boldsymbol{\tau}. \tag{6.3}$$

Minimizing the acceleration energy follows the principle of constrained motion in nature to solve the redundancy [UK92]. Thus, an approach has a more intuitive reaction when spending less "acceleration energy" during motion. The energy minimization can be, therefore, used as a metric to determine an intuitive reaction.

On the other site, QP-based solvers, in section 2.3, minimize the error between the current and the desired acceleration in the space of each task. As mentioned there, a regularization term is required to have a unique solution. In the literature, it is commonly used to minimize:

- the norm of the acceleration error for performing a joint position task $\|\ddot{\boldsymbol{q}}_{\mathrm{d}} - \ddot{\boldsymbol{q}}\|_2$, where $\ddot{\boldsymbol{q}}_{\mathrm{d}} = \boldsymbol{K}(\boldsymbol{q}_{\mathrm{d}} - \boldsymbol{q}) - \boldsymbol{D}\dot{\boldsymbol{q}}$ [DPNMN15] or
- the norm of the joint torques $\|\boldsymbol{\tau}\|_2$ [HLM$^+$18] or
- the norm of the joint accelerations $\|\ddot{\boldsymbol{q}}\|_2$ [QA19].

Having a desired joint position means that a desired robot's posture is enforced in the null-space. This choice brings two problems: changing the robot posture by means of external forces in the null-space of the tasks is not possible,[3] and a proper robot posture must be found depending on the application. The other two commonly used regularization terms are not meant to reduce the kinetic or the "acceleration energy", which implies that an intuitive behavior is not guaranteed. A better approach to define the regularization term is having a damping task in the joint space, which dissipates the kinetic energy: $\|\ddot{\boldsymbol{q}}_{\mathrm{d}} - \ddot{\boldsymbol{q}}\|_2$, where $\ddot{\boldsymbol{q}}_{\mathrm{d}} = -\boldsymbol{D}\dot{\boldsymbol{q}}$. However, the motion of the robot may not be intuitive by application of external forces, infringing the requirements established in section 1.3.4. A combination of the projector- and QP-based approaches takes the advantages of both.

It might appear simple to combine the two methods by solving the problem in 6.2 with a QP solver. That approach avoids numerical singularities and it opens the possibility to directly include unilateral constraints in the solver. However, two issues remain:

[3]The DOF not used to solve the main task do not behave as passive unactuated DOF.

- inclusion of new constraints is only possible at the same level of priority than the task described by $\ddot{\boldsymbol{x}}_{\text{t}}$ in eq. 6.2b and
- inclusion of multiple tasks with different levels of priority is not possible.

6.1.1 Separation of the task from the constraints

[FL13] treats the solution of redundancy at velocity level under inequalities for multiple tasks. The approach is formatted as a QP problem, where the tasks get different priority level by adding a task scaling factor s. This factor scales the task in case of joint limits hinder a fully accomplishment of the task. In this way, the direction of the task acceleration vector is maintained and the problem becomes solvable. The scaling factor is then maximized (i.e., equal to one if the task is feasible), and less than one if the saturated joints are required to fully perform the task. This approach enables the inclusion of new constraints in a different level of priority than the task. Applying this approach at torque level, the quadratic problem 6.2 becomes

$$
\begin{aligned}
\min_{\ddot{\boldsymbol{q}},\boldsymbol{\tau},s} \quad & \boldsymbol{\tau}^T \boldsymbol{M}^{-1}\boldsymbol{\tau} + w(1-s)^2 && (6.4a)\\
\text{subject to} \quad & \boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\nu} + \boldsymbol{g} && (6.4b)\\
& \boldsymbol{\tau}_{\text{min}} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\text{max}} && (6.4c)\\
& \ddot{\boldsymbol{c}}_{\text{min}} - \boldsymbol{J}_{\text{c}}\boldsymbol{M}^{-1}\boldsymbol{\tau}_{\text{ext}} \leq \boldsymbol{J}_{\text{c}}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\text{c}}\dot{\boldsymbol{q}} \leq \ddot{\boldsymbol{c}}_{\text{max}} - \boldsymbol{J}_{\text{c}}\boldsymbol{M}^{-1}\boldsymbol{\tau}_{\text{ext}} && (6.4d)\\
& \boldsymbol{J}\ddot{\boldsymbol{q}} = s\ddot{\boldsymbol{x}}_{\text{t}} - \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\nu} - \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\tau}_{\text{ext}} && (6.4e)
\end{aligned}
$$

where $w \gg 1$ is a large penalty term to give a higher priority to the scaling factor maximization than the minimization of the "acceleration energy". Note that at velocity level as proposed in [FL13], the scaling factor can never become negative. If a limit constraint becomes active, satisfying the constraint would not require the velocity to change its direction, $s < 1$. Not exceeding a position, velocity or acceleration limit requires either: to stop the motion in the operational space, $s = 0$ (position limit reached); to reduce the velocity at a saturated value, $0 < s < 1$ (velocity limit reached); or to remain with a constant velocity, $0 < s < 1$ (acceleration limit reached). At acceleration level, however, reducing the velocity or stopping the motion may require to decelerate the motion (acceleration must change to the opposite direction, $s < 0$).

The external torques in joint space $\boldsymbol{\tau}_{\text{ext}}$ are included in the constraint 6.4e instead of the external forces in the task space $\boldsymbol{f}_{\text{ext}}$. The external torques $\boldsymbol{\tau}_{\text{ext}}$ can be measured by joint torque sensors commonly available in collaborative robots as: the KUKA LBR iiwa [KUK], the Yuanda [YUA], the Panda emika [FRA], the sawyer [Ret] and the IRB 14000 YuMi [ABB]. On the contrary, the direct measurement of $\boldsymbol{f}_{\text{ext}}$ requires an external force/torque sensor.

6.1.2 Extension to multiple task of DCTS

Having multiple tasks increases the complexity of the problem, because every task should be solved minimizing the "acceleration energy". At the same time, tasks must be performed in a hierarchical way. Based on the mixed-task solver in section 2.4, a hierarchy scheme can be achieved while minimizing this energy. The inclusion of dynamically consistent null-space projectors guarantees also that kinetic energy is not inserted in the null-space. That consequence is explained by the use of the dynamically consistent pseudo-inverse in eq 2.5. This inverse is "*the unique operator that solves the redundancy without injecting energy in the null-space components of motion*" [BK00]. To include the null-space projectors, the minimization problem becomes

$$\min_{\ddot{\boldsymbol{q}}^{\text{aug}},\boldsymbol{\tau},\boldsymbol{s}} \quad \boldsymbol{\tau}^T \boldsymbol{M}^{-1}\boldsymbol{\tau} + \sum_{i=1}^{k} w_i(1-s_i)^2 \tag{6.5a}$$

$$\text{subject to} \quad \boldsymbol{\tau} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\nu} + \boldsymbol{g} \tag{6.5b}$$

$$\boldsymbol{\tau}_{\text{min}} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\text{max}} \tag{6.5c}$$

$$\ddot{\boldsymbol{c}}_{\text{min}} - \boldsymbol{J}_{\text{c}}\boldsymbol{M}^{-1}\boldsymbol{\tau}_{\text{ext}} \leq \boldsymbol{J}_{\text{c}}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\text{c}}\dot{\boldsymbol{q}} \leq \ddot{\boldsymbol{c}}_{\text{max}} - \boldsymbol{J}_{\text{c}}\boldsymbol{M}^{-1}\boldsymbol{\tau}_{\text{ext}} \tag{6.5d}$$

$$\boldsymbol{J}^{\text{aug}}\ddot{\boldsymbol{q}}^{\text{aug}} = \boldsymbol{S}^{\text{aug}}\ddot{\boldsymbol{x}}^{\text{aug}} - \dot{\boldsymbol{J}}^{\text{aug}}\dot{\boldsymbol{q}} - \boldsymbol{J}^{\text{aug}}\boldsymbol{M}^{-1}\boldsymbol{\tau}_{\text{ext}} \tag{6.5e}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{N}\ddot{\boldsymbol{q}}^{\text{aug}} = \sum_{i=1}^{k} \boldsymbol{N}_{\text{i}}\ddot{\boldsymbol{q}}_i \tag{6.5f}$$

with $\ddot{\boldsymbol{q}}^{\text{aug}} = \begin{bmatrix} \ddot{\boldsymbol{q}}_1 \\ \vdots \\ \ddot{\boldsymbol{q}}_k \end{bmatrix}$, $\ddot{\boldsymbol{x}}^{\text{aug}} = \begin{bmatrix} \ddot{\boldsymbol{x}}_1 \\ \vdots \\ \ddot{\boldsymbol{x}}_k \end{bmatrix}$, $\boldsymbol{N} = [\boldsymbol{N}_1 \ldots \boldsymbol{N}_k]$ and $\boldsymbol{S}^{\text{aug}}$ is a diagonal matrix with diagonal values from $s_1 ... s_k$. To scale every task in a prioritized order, the weights are set $w_i \gg w_{i-1}$. The strict hierarchy between tasks is guaranteed by using augmented null-space projectors. When a limit is encountered and is in a conflict with more than one task, tasks may be not scaled in a strict-priority order. The choice of the weights gives this strictness.

6.2 Simulation and experiments

This section evaluates the studied solvers and demonstrates the problematic of the hierarchy based on QP of section 2.3. The evaluation compares state-of-art implementations of QP solvers for control of redundant robots. Simulations and experiments show the advantages of the proposed DCTS. The tests are performed on a KUKA LBR iiwa. The values of the limits used in simulations and experiments are shown in table 6.1.

Table 6.1: Joint limits of the KUKA LBR iiwa

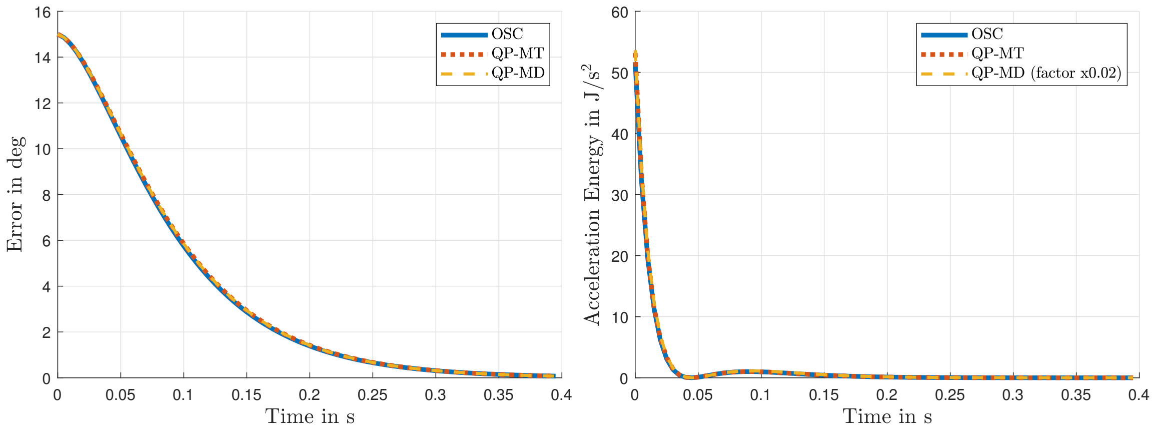
Limits \ Joint	q_1	q_2	q_3	q_4	q_5	q_6	q_7
$\boldsymbol{q}_{\text{max}}$ in deg	165	115	165	115	165	115	170
$\boldsymbol{q}_{\text{min}}$ in deg	-165	-115	-165	-115	-165	-115	-170
$\boldsymbol{v}_{\text{max}}$ in deg/s	100	110	100	130	130	180	180
$\boldsymbol{v}_{\text{min}}$ in deg/s	-100	-110	-100	-130	-130	-180	-180
$\boldsymbol{\tau}_{\text{max}}$ in Nm	100	100	100	100	100	100	100
$\boldsymbol{\tau}_{\text{min}}$ in Nm	-100	-100	-100	-100	-100	-100	-100

6.2.1 Simulation results

The simulations were run in Matlab/Simulink using the qpOASES library [FKP$^+$14] as a QP solver. A task for the end effector center is defined in two rotational coordinates ($m_\text{t} = 2$). The end effector is expected to rotate 15 degrees from the initial pose around the x-axis of the world frame, while keeping constant the initial rotation around the y-axis. The redundancy is solved by three approaches:

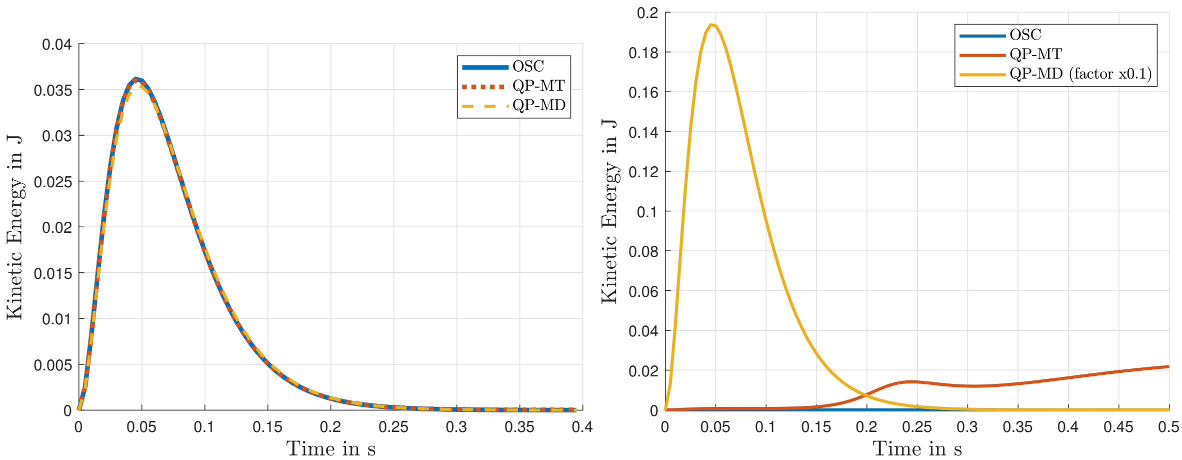
- the OSC, i.e., solving the QP problem in eq. 6.2
- solving the problem in section 2.3, with a regularization term that minimizes the norm of the torque $\|\boldsymbol{\tau}\|_2$ (QP-MT)
- solving the problem in section 2.3 with a regularization term that minimizes the kinetic energy in joint space $\|\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_\text{d}\|_2$, where $\ddot{\boldsymbol{q}}_\text{d} = -\boldsymbol{D}\dot{\boldsymbol{q}}$ (QP-MD).

All three approaches solve the task and minimize the error to zero in the desired coordinates (see fig. 6.1(a)), although the QP-MD spend more acceleration energy (about fifty times more). Figure 6.1(b) shows a similar behavior of the acceleration energy for the three methods, with the remark that the energy spent by the QP-MD had to be scaled by a factor of 0.02. Figure 6.2(a) shows that the kinetic energy in task space is approximately the same for all the approaches and converges to zero. However, the kinetic energy in the null-space behaves differently for each solver (see fig. 6.2(b)). While the OSC shows zero energy, QP-MT shows an energy that increases over time. This increment of the energy is because minimizing the torque does not guarantee that the velocity becomes zero, i.e., energy is not dissipated. If a damping task is used (QP-MD), the kinetic energy in the null-space will be reduced but it is much higher than minimizing the norm of the torque. These results demonstrate that the dynamically consistent pseudo inverse does not inject energy in the null-space components of motion. As QP-MT and QP-MD solvers provide a solution that minimizes the error in the task space without using this inverse, kinetic energy may be injected and it may not be dissipated. The dissipation depends on the chosen regularization term.



(a) (b)

Figure 6.1: (a) Orientation error of the 2D-rotational task. (b) Acceleration energy behavior



(a) (b)

Figure 6.2: Kinetic energy used to solve a 2D rotational task. (a) Energy in task space. (b) Energy in null-space

The second simulation illustrates the different responses of the approach to external forces. Envision a worker who must teach the robot different points located on a table, where the robot must automatically drill after. Moving the robot with a visual user interface may be slower and less intuitive than moving the robot hand-guided. To reduce the complexity of the teaching process, the end effector must hold a desired orientation around two axes to keep the drill perpendicular to the table. The position and the orientation around the drill axis are not controlled. Application of external forces should move the robot along these coordinates. The end effector starts with an orientation that locates the drill perpendicular to the table. An external force of 10 N is applied on the end effector in positive x-direction for 400 ms. The force disturbs the end effector making it to lose its desired orientation. The motion must follow the desired dynamic-behavior in task space. Such behavior follows the mass-spring-damper system motion-equation given in eq. 2.9 ($\boldsymbol{\Lambda}_{\mathrm{t,d}}\ddot{\boldsymbol{e}}_{\mathrm{t}} + \boldsymbol{D}_{\mathrm{t}}(\dot{\boldsymbol{e}}_{\mathrm{t}}) + \boldsymbol{K}_{\mathrm{t}}(\boldsymbol{e}_{\mathrm{t}}) = \boldsymbol{f}_{\mathrm{ext}}$).

Figure 6.3(b) shows that the external force injected much more kinetic energy in the robot using QP-MT or QP-MD than with OSC despite having the same task (note the factor x20 that amplifies the energy plot of OSC). As expected, QP-MT and OSC spent less "acceleration energy " than QP-MD (see fig. 6.3(a)), which leads to a more intuitive motion. With both approaches, the motion follows the force direction at the beginning (see fig. 6.4(a)). As the human expects by pushing a unit rigid body that is in an initial still state, the acceleration should produce a body motion in the direction of the force times the inertia matrix $\boldsymbol{\Lambda}$.[4] However, as OSC allows less injection of kinetic energy in the task and in the null-space, the motion is shorter and slower. Note that despite the difference of the motion in null-space of OSC and QP-MD, the task space-error behaves similarly for both approaches (see fig. 6.4(b)). In conclusion, OSC is the only approach that leads to the desired motion behavior. The end effector moves following the Newton's law $\ddot{\boldsymbol{x}} = \boldsymbol{\Lambda}^{-1}\boldsymbol{f}_{\mathrm{ext}}$ in the null-space (translation). In the task space, the motion follows the motion equation in 2.9.

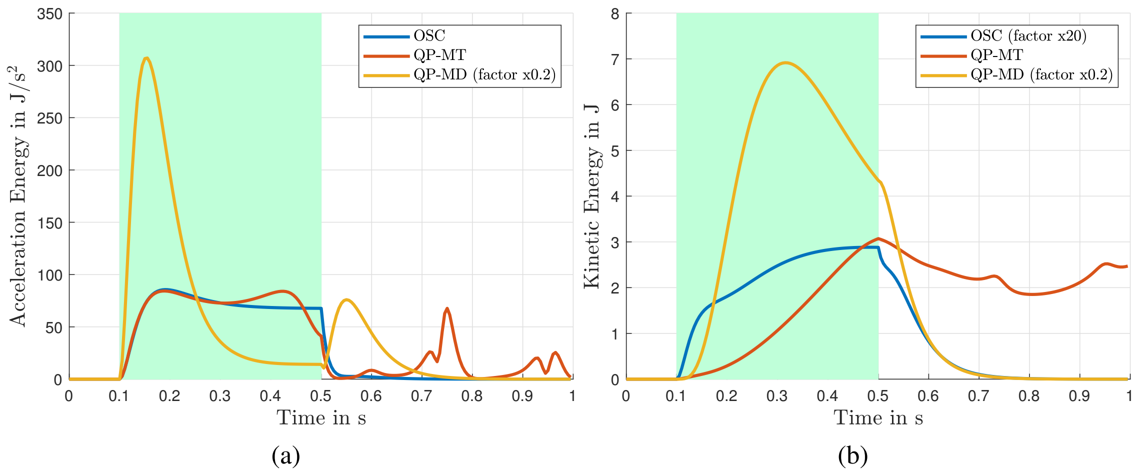


Figure 6.3: Comparison of minimization functions to solve redundancy. (a) Acceleration energy. (b) Kinetic energy

The next simulation demonstrates the advantages of the dynamically-consistent constrained task hierarchy solver (DCTS) over the projector-based approaches under unilateral constraints as in section 3.3.2 and 4.2. The center point of the end effector should go through a series of Cartesian points connected by linear paths, starting from $\boldsymbol{q}_{\mathrm{ini}} = [-0.78, 2.05, 2.07, -1.65, -2.08, 2.03, 0]^{\mathrm{T}}$ rad. The points are vertices of an octagon inscribed in a circle lying in the (y,z) vertical plane of

[4]The inertia matrix may have coupling terms that deviates the direction of motion from the direction of the force.

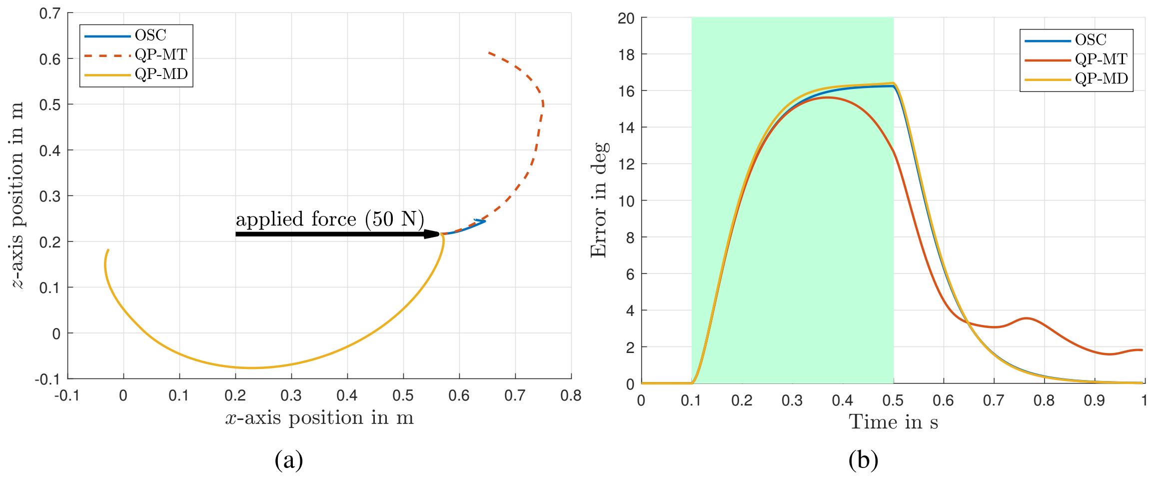


Figure 6.4: Comparison of minimization functions to solve redundancy. (a) Resulting motion in Cartesian null-space. (b) Task error. The green background zone shows the time interval in which the external force was applied.

radius 0.3 m. The center of the circle lies in [0.55,-0.05,0.22] m. The end effector starts in the center of the circle, goes to every vertice of the octagon, and it returns to the center every time.

A primary task of dimension $m_1 = 3$ is specified (only Cartesian positions). The desired acceleration $\ddot{x}_1$ is chosen so as to head towards the next Cartesian point with a speed 0.85 m/s, following the desired dynamic behavior given by eq. 4.27. This trajectory is particularly demanding at the vertices (reached with a tolerance of 0.001 m) of the octagon, where large accelerations are required to suddenly change direction. A limitation of the joint positions and velocities is included following the method in section 4.2. The computed command torque value $\tau_{\text{cmd},i}$ is saturated to its upper or lower limit and sent to the robot controller. If $\tau_{\text{cmd},i} > \tau_{\text{max},i}$, then $\tau_{\text{cmd},i} = \tau_{\text{max},i}$. If $\tau_{\text{cmd},i} < \tau_{\text{max},i}$, then $\tau_{\text{cmd},i} = \tau_{\text{min},i}$, with $i = 1 \ldots n$.

Figures 6.5 and 6.6 show the results of task execution using the projector-based approach of section 4.2. Figure 6.5 shows that the desired path in position is followed despite the limitation of joint positions, velocities and torques. However, there are two aspects to remark, given by the naive[5] saturation of the torques:

- when the repulsive torque, which keeps the joint velocities within their limits, exceeds its upper or lower limit, the velocity limit is not respected. The time-intervals represented by the gray zone in fig. 6.6 show these cases.
- when the joint torques to produce the desired acceleration in the operational space exceed their upper or lower limits, the acceleration error increases. See fig. 6.6

[5]Naive in this context means that the torques are saturated without computing a new solution after saturation to obtain an optimal result. Therefore, the desired accelerations in the operational space or in the joint space will not be produced.

The simulation was repeated using DCTS of section 6.1.2. Figure 6.7 shows the results of the position tracking of the star. The end effector followed the desired path despite all limitations. The error of the position tracking compared to the projector-based approach is slightly less (see table 6.2). However, the results in fig. 6.8 show that all limitations were respected. The error of the acceleration only increased when the task was scaled (scaling factor smaller than one). Such case came, when the torque limits were in conflict with the task, i.e., there was not solution to fully accomplish the task, and to keep the torques between their limits. In average, the acceleration error decreased compared to the projector-based approach (see table 6.2).

6.2.2 Experimental results

An experiment is carried out to evaluate the intuitive reaction of the DCTS in comparison to OSC and QP-MD. The task is, as in the first simulation, defined only in two coordinates ($m_\mathrm{t} = 2$) of the rotation. The stiffness matrix $\boldsymbol{K} \in \mathbb{R}^{2\times 2}$ is set to $1,000\boldsymbol{I}$ Nm/rad and the damping $\boldsymbol{D}$ to $63\boldsymbol{I}$ Nms/rad to keep the initial orientation during the experiment. To achieve a constant external force for all the experiments, a piece with approx. 4.1 kg of weight is attached to the end effector. The controller does not have knowledge of this piece, i.e., only the weight of the robot structure is considered to compute the gravity compensation torques. After starting the experiment, the end effector is expected to go down following Newton's law, while holding the orientation.

Figure 6.9 illustrates the robot motion after starting the experiment. In the one hand, the unintuitive motion of the QP-MD approach is visualized in fig. 6.9(a). The end effector starts to move up and backwards instead of falling down. In the other hand, an intuitive motion is shown in fig. 6.9(b) achieved by the OSC or DCTS. The end effector falls down due to the gravity force produced by the attached piece. Note that all approaches kept the desired rotation during the motion.

Figure 6.10(a) shows that the QP-MD uses more acceleration energy leading to a non-intuitive motion. The plot of the end effector motion in fig. 6.10(b) shows that the OSC and the DCTS delivered similar results despite different computation methods.

Table 6.2: Average errors during tracking of a star path

Approach	Position Error in m	Acceleration Error in m/s^2
Method in section 4.2	0.018	0.0294
DCTS	0.016	0.0248

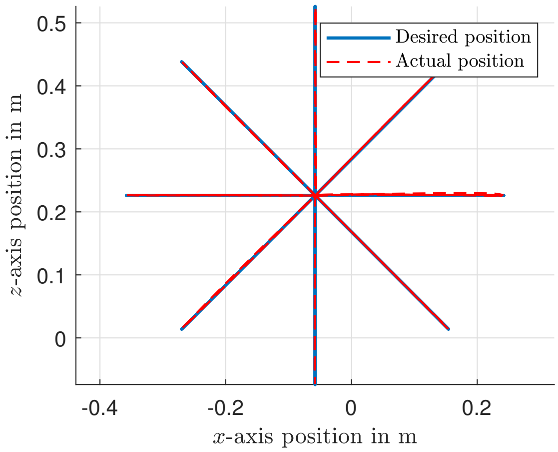


Figure 6.5: End effector Cartesian position using the projector-based approach

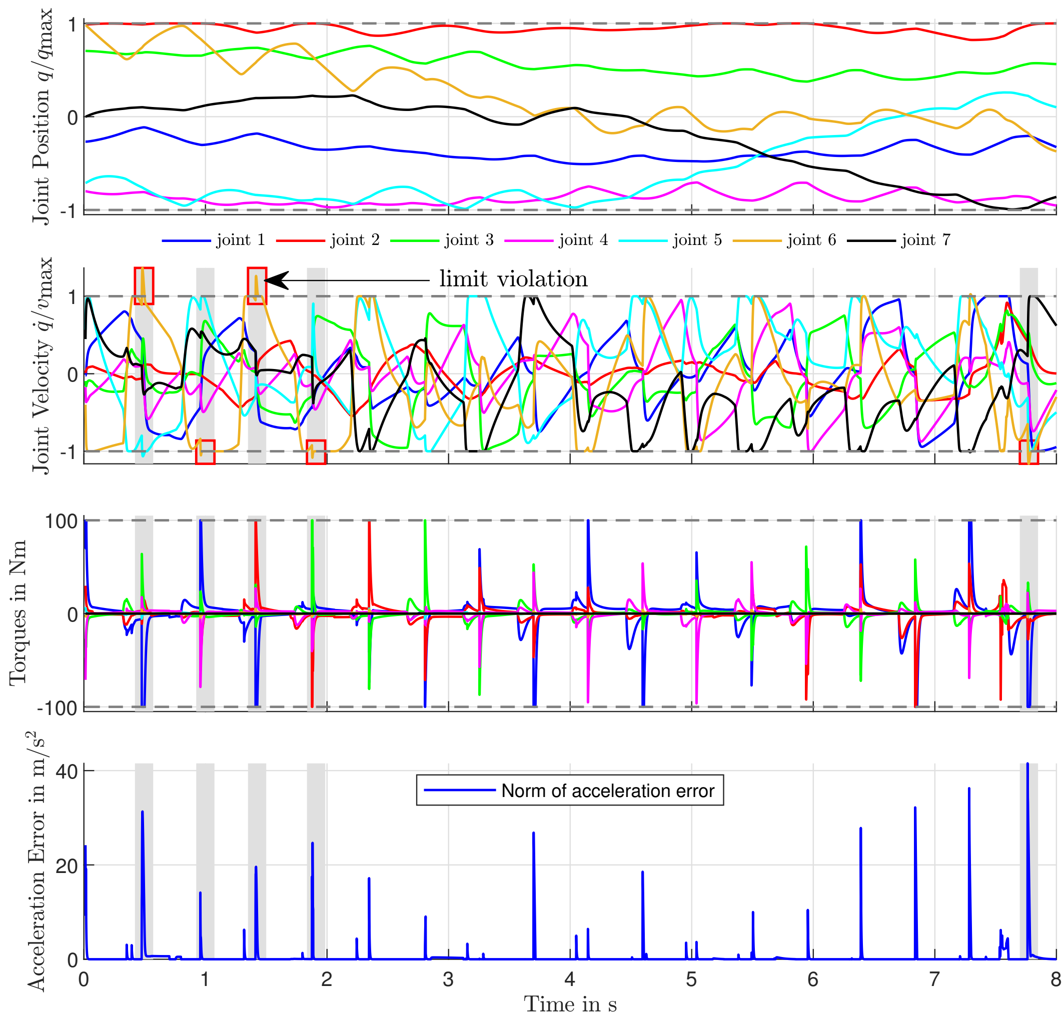


Figure 6.6: Results of drawing a star with the end effector using the projector-based approach.

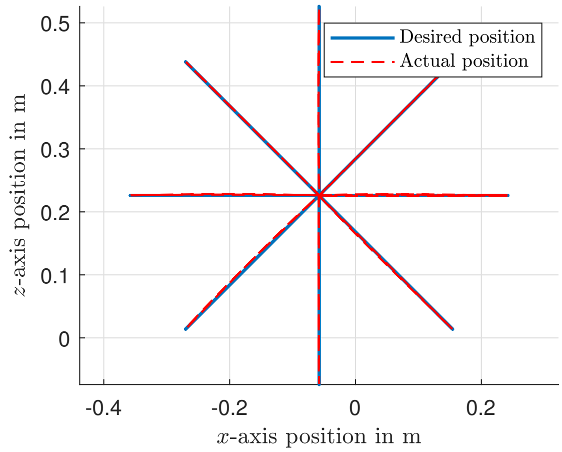


Figure 6.7: End effector Cartesian position using the DCTS

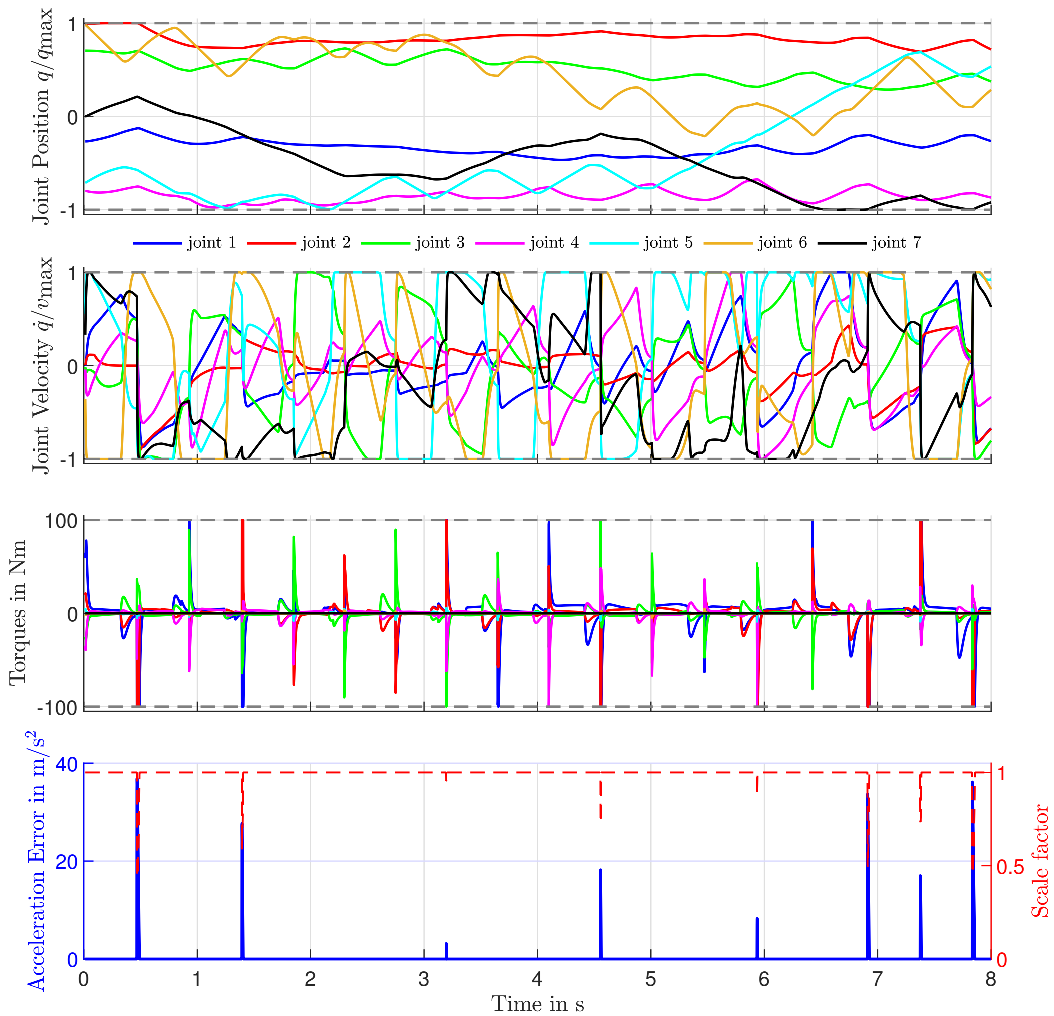


Figure 6.8: Results of drawing a star with the end effector using the DCTS

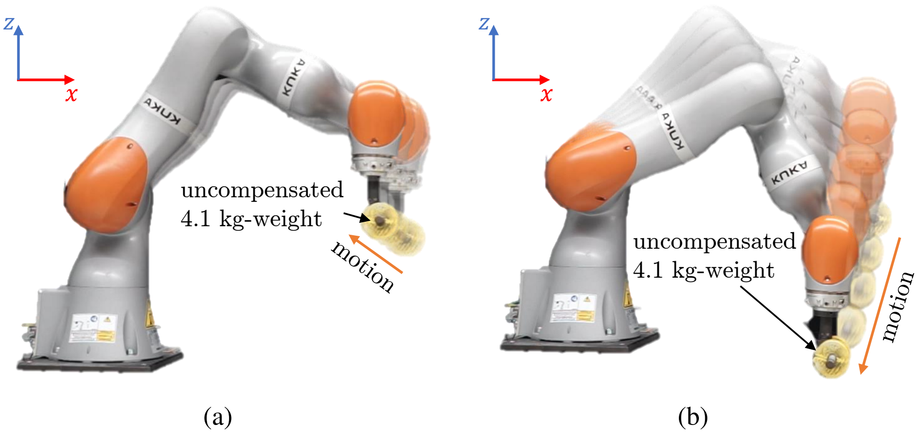


Figure 6.9: Video snapshots of the performed experiment. (a) Resulting robot motion using the QP-based solver and minimizing error for damping task as last task. (b) Robot motion using the DCTS or OSC. The orange arrow illustrates the direction of the motion.

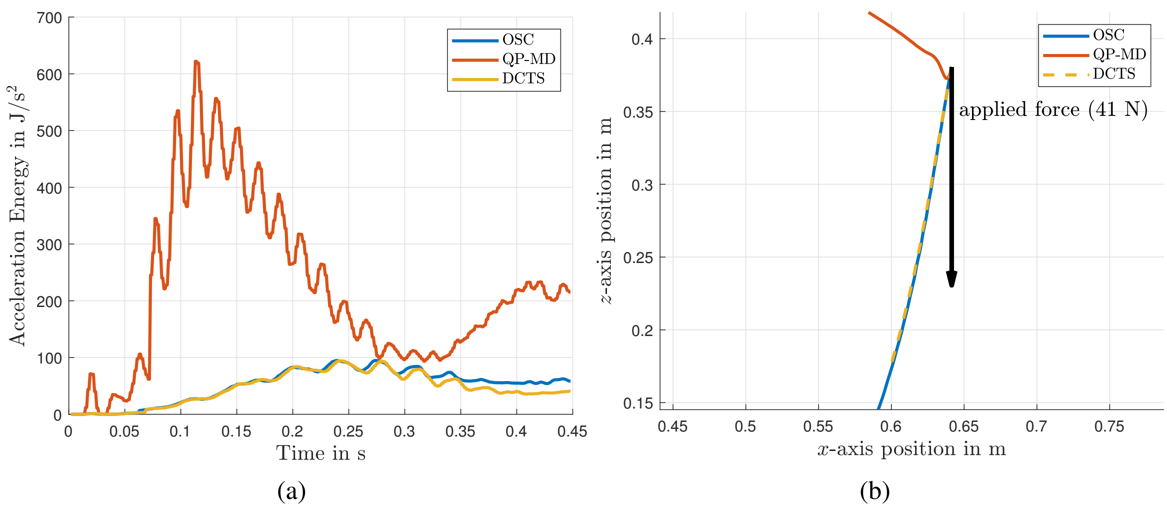


Figure 6.10: Comparison of redundancy resolution approaches under physical interaction. (a) Acceleration energy. (b) Resulting motion in null-space

6.3 Discussion

Experiments evaluated the different dynamic behavior in the null-space of QP-based and projector-based solvers. The evaluation demonstrated that QP solvers, as commonly used in literature, do not produce an intuitive reaction to external forces. Results showed that the robot motion does not always follow the direction in which the force is applied. Additionally, kinetic energy is injected in the null-space of the task. As a result of this injection of kinetic energy, the robot unnecessary moves in the null-space. In other words, the task could be accomplished spending less energy. Moreover, results showed that projector-based approaches solve the redundancy without injecting extra energy and produce an intuitive motion. The drawback of these approaches is the complex inclusion of inequalities at force level. Therefore, this chapter proposed a method to perform hierarchical control under unilateral constraints. The Dynamically-consistent Constrained Task Hierarchy Solver (DCTS) combines the advantages of state-of-the-art methods to solve redundancy:

- the natural solution[6] provided by approaches based on dynamically-consistent pseudoinverses and projectors,
- the direct inclusion of unilateral constraints (specially torque limits together with acceleration limits) and the singularity avoidance given by the use of QP solvers.

Experiments showed that contrary to the projector-based hierarchy solvers in section 4.2, the DCTS limits the commanded joint generalized forces while computing an optimal solution of the task. The framework using this solver unifies all constraints at different levels in different spaces, while solving the redundancy as the nature would do.

Although the computational effort was not deeply investigated in this chapter, the DCTS is expected to be computationally slower than the other solvers. This approach requires the computation of the dynamically consistent projectors in addition to solve the quadratic problem. Furthermore, the constraints of the quadratic problem include augmented matrices of the Jacobian, null-space projectors and desired accelerations. Such high-dimensional matrices increase the computational effort to find the optimal result of the command torques.

[6] Natural solution in this context is the solution that the nature would employ to solve redundancy for constrained motion in mechanical systems.

7 Summary

The increasing complexity in robotic systems has brought numerous challenges to achieve collaborative robotics properly. Many of these challenges are related to a large number of actuated DOF in robots such as humanoids. Even though the complexity of the tasks robots can perform has increased in various scenarios, they are still unable to perform every task autonomously and require humans to be involved in the work loop. Robot-human collaboration is, therefore, a topic of increasing interest in robotics. Collaborative scenarios together with complex robots require intuitive and safe controllers. Safety relates to numerous aspects of the robotic system, including: maximal allowed speed, maximal exchange of energy or power between the robot and the human, maximal forces applied to the human or the environment, etc. Many collaborative applications may require the robot to physically interact with the environment and the human, e.g., helping the human to carry heavy objects, medical applications as the MURAB project [MUR], and rehabilitation applications.

Sophisticated whole-body controllers must be employed to achieve complex constraint-task hierarchy schemes. Such controllers should efficiently and robustly command the robot to accomplish its goal without endangering the human, the environment, or itself. A key aspect for achieving such interaction is to have a compliant behavior of the robot. This thesis presents a framework to have a robotic co-worker with fully physical interactivity. Predominantly, the framework should manage a stack of tasks in a prioritized manner while the constraints are always respected. In the following paragraphs, an overview of the contributions of this thesis is given.

One particular case of constraints is unilateral constraints that are expressed as inequalities that must be upheld. Chapter 3 treats unilateral constraints in the joint space. Joint limits are especially relevant in redundant robotic systems. Making use only of the available joints brings an optimal redundancy resolution for the commanded torques. The idea is to create a repulsive force that prevents the joint from going to its limits, while the remaining joints are used to achieve the task as good as the remaining capability of the robot allows it. To this end, most of state-of-the art approaches rely on artificial repulsive potential fields to create the repulsive force. Such a technique requires hard-coded parameters to design the artificial potential field. On the contrary, the Saturation in Joint Space (SJS) approach in section 3.3.2 computes the repulsive force depending on the state of the robot. The avoidance of hard-coded parameters increases the robustness of the approach and makes possible a high-speed robot motion under unilateral constraints. In addition, SJS enables the inclusion of velocity and acceleration limits. When the saturation of the joint accelerations is carried out, the external forces are considered, and smooth behavior is assured in the proximity of the limits. This result enables physical Human-Robot Interaction (pHRI) while having an optimal result for the stack of tasks.

Chapter 4 extends the framework to handle any kind of unilateral constraints. The inclusion of rotational limits, for instance, opens the possibility to define the tasks in less DOF while remaining in a defined range of orientation, where the task is still achievable. The relaxing of the task increases the null-space that can be used to achieve secondary objectives. Another typical problem in robotics is to avoid collision with obstacles. Contact with some objects on the environment or parts of the human is not always desired in collaborative scenarios. The inclusion of limits that consider collision with moving obstacles solves that issue. To avoid parametrization of the repulsive force, the framework considers the velocity of the moving object. The method works even when the human intervenes, not allowing the robot to collide with some objects even if the human intends to do so.

Although having an impedance control with fixed parameters is enough for most of the applications that require contact, interactivity can be maximized by a variable impedance law. Chapter 5 proposes a novel definition of the task that maximizes the interaction. The MURAB project is taken as a use-case to prove the performance. The method avoids the need for path planning by having a time-invariant definition of the path. The variable stiffness reduces the energy that the robot could transfer to the human. This energy reduction, together with the assurance of the limits, makes the system safer for users who do not have much experience with robots and could unexpectedly disturb the robot.

The inclusion of the unilateral constraints and the definition of the task can be included in any of the task hierarchy solvers: hierarchy based on null-space projectors, hierarchy based on quadratic programming, and mixed task hierarchy (see Chapter 2). However, the choice of the solver implies different aspects. Chapter 6 makes a comparison of the solvers, especially of the behavior under pHRI. The comparison shows that QP-based schemes, as commonly used to perform torque control, do not present an intuitive behavior. On the other hand, projected-based methods require a singularity avoidance method. In addition, the inclusion of torque limits has not been conducted yet due to its complexity. A new approach is proposed in chapter 6 in order to have the advantages of both solvers. The Dynamic-consistent Constrained Task Hierarchical Solver (DCTS) brings all advantages together with the drawback of being computationally slow. No scheme works for all possible applications, and therefore, the choice should be according to the specific requirements of the use-case.

The proposed framework was implemented in different use-cases. Three proposed use-cases demonstrate the framework's usability. Use-case 1 was presented in Chapter 5, in which the framework is used in the MURAB project on-line [MUR] and off-line [GTW+20] as a motion planner. In use-case 2, the robot must autonomously learn the end-effector forces to insert a peg in a hole. The use of this framework gives the advantage of learning the forces in predefined coordinates, instead of learning the forces in the whole 6D Cartesian space. For instance, two rotational coordinates can be fixed, while the robot learns only the necessary translational forces and the rotation around the vertical axis. The framework includes limitations in any space allowing the learning process without damaging the robot or the environment. Results of this use-case are presented in [KMB20]. Use-case 3 is the teleoperation of a redundant robot with haptic feedback. The teleoperation device sends Cartesian space motions that the robot must perform. Having a kinematically redundant robot requires a proper redundancy resolution framework that considers Cartesian and joint limitations. [RPB+12] proposed a velocity-based controller to perform teleoperation under safe limitations. However, velocity or position control does not guarantee stability in contact situations. A stable interaction of the robot with the environment is guaranteed by using impedance control. Therefore, the framework proposed in this thesis is a suitable approach for this use-case. An implementation of the framework for teleoperation of a redundant robot and making passivity stabilization is presented in [Mar20]. In addition, two more use-cases for the framework were proposed but not intensely tested. The first one is the use of robotics in neurorehabilitation of impaired limbs. In this scenario, the patient must practice a pre-defined limb motion with the help of the robot. The robot motion is more or less assistive depending on the phase of the therapy [PCRBJ16]. A typical control strategy is *tunneling*, which consists in creating virtual channels for the end-effector in which the subject moves. This strategy requires physical interaction and limitation of joint and Cartesian coordinates. Using the framework with a path-dependent geometric representation of the task as proposed in [MA21] fulfills these requirements. The second proposed use-case uses a robot in 3D object modeling, as proposed in [MS20]. The framework facilitates the programming of complex motions to focus the object with different angles.

Although the framework demonstrated its usability, some aspects beyond the scope of this work could be further investigated:

- although the stability of impedance control and operational space control has been proven in [Die16] and [DOP18], respectively, it has not been proven with inequality constraints as defined in this thesis.
- an thorough assessment of the computational effort of the different solvers must be performed to find their limitations in the implementation.
- the proposed framework did not consider changing the priorities of tasks or the insertion of new ones. The addition of this feature can be included by using the method provided in [DS19].

In addition, future research can focus on implementing higher-level frameworks that provide the task definition from a semantic description. For instance, the description of grasping a cup would provide only the required coordinates and position limits, for which the task can be accomplished. [SGR+15] proposes a framework for intuitive programming of robots. Programming interfaces with semantically detailed descriptions allow non-experts to program the robot intuitively. Such interfaces can be extended by providing the hierarchical framework with the desired forces and force limitations instead of only providing geometric descriptions.

Implementing the previously discussed advancements in a follow-up work would extend the framework's usability beyond what is presented in this thesis. The framework could be used in real-world applications of highly redundant robots in challenging dynamic environments and reduce the programming complexity of the robot motions.

Appendix

A.1 Including gravity compensation

The computation of gravity term in the operational space in eq. 2.8 can be avoided, if a joint space task is assumed to be always the least important task. Consider only the gravitation compensation part

$$\boldsymbol{\tau}_{\mathrm{g,t}} = \boldsymbol{J}_{\mathrm{t}}^{T} \boldsymbol{p}_{\mathrm{t}}, \tag{A.1}$$

and two tasks where the last task is in joint space. The gravitational part of the torques are defined by:

$$\boldsymbol{\tau}_{\mathrm{g,1}} = \boldsymbol{J}_{1}^{T} \boldsymbol{p}_{1}, \tag{A.2}$$

$$\boldsymbol{\tau}_{\mathrm{g,2}} = \boldsymbol{J}_{2}^{T} \boldsymbol{p}_{2}. \tag{A.3}$$

As the second task is defined in the joint space $\boldsymbol{J}_2 = \boldsymbol{I}$. Equation A.1 becomes

$$\boldsymbol{\tau}_{\mathrm{g,2}} = \boldsymbol{g}. \tag{A.4}$$

The final torques to compensate the gravity can be then computed using eq. 2.3

$$\boldsymbol{\tau}_{\mathrm{g}} = \boldsymbol{J}_{1}{}^{T} \boldsymbol{p}_{1} + (\boldsymbol{I} - \boldsymbol{J}_{1}^{T} \bar{\boldsymbol{J}}_{1}^{T}) \boldsymbol{\tau}_{\mathrm{g,2}}. \tag{A.5}$$

Replacing eq. 2.8 in A.5

$$\boldsymbol{\tau}_{\mathrm{g}} = \boldsymbol{J}_{1}{}^{T} \bar{\boldsymbol{J}}_{1}^{T} \boldsymbol{g}_{1} + (\boldsymbol{I} - \boldsymbol{J}_{1}^{T} \bar{\boldsymbol{J}}_{1}^{T}) \boldsymbol{g}. \tag{A.6}$$

It is clear now that the total gravitational part is

$$\boldsymbol{\tau}_{\mathrm{g}} = \boldsymbol{g}. \tag{A.7}$$

This example demonstrates that the computation of the gravity term for every task is unnecessary, at least in the scope of this thesis. Note that the extension to more tasks does not change the result, because the gravitation compensation part for the joint space task is computed in the null space of all previous tasks, and this part is the complement of all superior gravitation part of the operational spaces (assuming that augmented projectors are used). In other words, adding the gravity term as proposed in this thesis, means a perfect projection of the gravity compensation part with less computational effort. The effect of a poor projection of the gravitational part when using successive null space projectors is therefore avoided.

A.2 The problem of torque limitation through inverse dynamics

Consider a two-link planar manipulator with rotational joints (q_1 and q_2) with links length l_1 and l_2. If each link is model as a rigid homogeneous rectangular bar with mass m_i and moment of inertia tensor

$$\boldsymbol{I} = \begin{bmatrix} I_{x,i} & 0 & 0 \\ 0 & I_{y,i} & 0 \\ 0 & 0 & I_{z,i} \end{bmatrix}, \tag{A.8}$$

the equation of motion can be found using Lagrange's equations[1]

$$\begin{bmatrix} I_{z,1} + I_{z,2} + m_1 r_1^2 + m_2(l_1^2 + l_2^2) + 2m_2 l_1 r_2 c_2 & I_{z,2} + m_2 r_2^2 + m_2 l_1 r_2 c_2 \\ I_{z,2} + m_2 r_2^2 + m_2 l_1 r_2 c_2 & I_{z,2} + m_2 r_2^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}, \tag{A.9}$$

where r_i are the distances from the joints to the center of mass for each link, and $c_2 = \cos(q_2)$. Coriolis and gravity terms are ignored to simplify the explanation.

For an example configuration $\boldsymbol{q} = \begin{bmatrix} 30 & 45 \end{bmatrix}^T$ deg and the data given in table A.1, the torques have to be limited to $\boldsymbol{\tau}_{\text{max}} = -\boldsymbol{\tau}_{\text{min}} = \begin{bmatrix} 10 & 10 \end{bmatrix}^T$ Nm. If the maximal and minimal joint accelerations are computed using the inverse dynamics as proposed in [FDK12]

$$\ddot{\boldsymbol{q}}_{\text{max}} = M^{-1}\tau_{\text{max}} = \begin{bmatrix} 0.1660 & -0.2364 \\ -0.2364 & 1.1367 \end{bmatrix} \begin{bmatrix} 10 \\ 10 \end{bmatrix} = \begin{bmatrix} -0.7042 \\ 9.0029 \end{bmatrix} \tag{A.10}$$

$$\ddot{\boldsymbol{q}}_{\text{min}} = M^{-1}\tau_{\text{min}} = \begin{bmatrix} 0.1660 & -0.2364 \\ -0.2364 & 1.1367 \end{bmatrix} \begin{bmatrix} -10 \\ -10 \end{bmatrix} = \begin{bmatrix} 0.7042 \\ -9.0029 \end{bmatrix} \tag{A.11}$$

This simple example shows that this approach does not result in proper joint acceleration limits,

Table A.1: Two-link planar manipulator data

Parameter	Value	Parameter	Value
l_1	1.5 m	m_1	2 Kg
l_2	1 m	m_2	1 Kg
r_1	1 m	$I_{z,1}$	2 Kg m^2
r_2	0.5 m	$I_{z,2}$	1 Kg m^2

[1] To see the complete procedure visit [SSVO10].

because $\ddot{q}_{\text{max},1} < \ddot{q}_{\text{min},1}$. One could reorder the maximal accelerations vectors to get:

$$\ddot{\boldsymbol{q}}_{\text{max}} = \begin{bmatrix} 0.7042 \\ 9.0029 \end{bmatrix} \tag{A.12}$$

$$\ddot{\boldsymbol{q}}_{\text{min}} = \begin{bmatrix} -0.7042 \\ -9.0029 \end{bmatrix} \tag{A.13}$$

However, if the task produces a joint acceleration $\ddot{\boldsymbol{q}} = \begin{bmatrix} 0 & 15 \end{bmatrix}^T$ rad/s^2, the saturation vector would be $\ddot{\boldsymbol{q}}_{\text{sat}} = \begin{bmatrix} 0 & 9.0029 \end{bmatrix}^T$ rad/s^2. This acceleration would produce the torques

$$\boldsymbol{\tau}_{\text{sat}} = M\ddot{\boldsymbol{q}}_{\text{sat}} = \begin{bmatrix} 8.5607 & 1.7803 \\ 1.7803 & 1.25 \end{bmatrix} \begin{bmatrix} 0 \\ 9.0029 \end{bmatrix} = \begin{bmatrix} 16.0282 \\ 11.2537 \end{bmatrix} \tag{A.14}$$

The torque exceeds the maximal torque limits, which demonstrates that the saturation of torques must be performed in a different way.

A.3 Cancellation of motion by a wrong inclusion of external torques in the optimization problem

Assume, only a joint space task is given to the robot. To produce a free motion when external forces are applied, the desired stiffness and damping is set to zero, i.e., $\ddot{\boldsymbol{q}}_{\text{d}} = \boldsymbol{0}$.

The optimal command torques to accomplish this task are computed under the constraint 13.b in [LTP16]:

$$\boldsymbol{\tau}_{\text{cmd}} = \boldsymbol{M}\ddot{\boldsymbol{q}}_{\text{d}} + \boldsymbol{g} + \boldsymbol{\nu} - \boldsymbol{\tau}_{\text{ext}} = \boldsymbol{M}\boldsymbol{0} + \boldsymbol{g} + \boldsymbol{\nu} - \boldsymbol{\tau}_{\text{ext}} = \boldsymbol{g} + \boldsymbol{\nu} - \boldsymbol{\tau}_{\text{ext}} \tag{A.15}$$

If the motion is computed following the inverse dynamics

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{\tau}_{\text{cmd}} + \boldsymbol{\tau}_{\text{ext}} - \boldsymbol{g} - \boldsymbol{\nu}) \tag{A.16}$$

then:

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{g} + \boldsymbol{\nu} - \boldsymbol{\tau}_{\text{ext}} + \boldsymbol{\tau}_{\text{ext}} - \boldsymbol{g} - \boldsymbol{\nu}) = \boldsymbol{0} \tag{A.17}$$

This result implies that there is no motion despite application of external forces, if the robot is in a static state.

Bibliography

Individual topics of this thesis were previously published during the work in the corporate research at KUKA Deutschland GmbH. Four international conference contributions [MFA18, MCAZ19, MAF$^+$19a, MAAZ20], three patents [MAF19b, Muñ19, MA21] and one patent application [MA20] were published as first author. Although this is own content, the references to these publications are neither omitted nor separately marked in the text. The same applies for publications as co-author [GTW$^+$20, KMB20, MS20], and supervised student works [Win18, Cas18, Abd20, Mar20].

[ABB] ABB: *Technical data IRB 14000 YuMi*. `https://new.abb.com/products/robotics/industrial-robots/irb-14000-yumi/irb-14000-yumi-data`. – (visited on: 2020-12-05)

[Abd20] ABDELAZIM, A. (supervisor: Munoz J. D.): *Optimized Redundancy Resolution for Torque-controlled Robots*. Dortmund, Germany, Technische Universitä Dortmund, Master Thesis, February 2020

[ASOFH03] ALBU-SCHÄFFER, A. ; OTT, C. ; FRESE, U. ; HIRZINGER, G.: Cartesian impedance control of redundant robots: recent results with the DLR-light-weight-arms. In: *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, IEEE, 14-19 Sept. 2003, pages 3704–3709

[ASOH07] ALBU-SCHÄFFER, A. ; OTT, C. ; HIRZINGER, G.: A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots. In: *The International Journal of Robotics Research* 26 (2007), Nr. 1, pages 23–39

[BHB84] BAILLIEUL, J. ; HOLLERBACH, J. ; BROCKETT, R.: Programming and control of kinematically redundant manipulators. In: *The 23rd IEEE Conference on Decision and Control*, IEEE, 12.12.1984 - 14.12.1984, pages 768–774

[BHG10] BEHNISCH, M. ; HASCHKE, R. ; GIENGER, M.: Task space motion planning using reactive control. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pages 5934–5940

[BK00] BRUYNINCKX, H. ; KHATIB, O.: Gauss' principle and the dynamics of redundant and constrained manipulators. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, IEEE, 24-28 April 2000, pages 2563–2568

[BK11] Bouyarmane, K. ; Kheddar, A.: Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 25.09.2011 - 30.09.2011, pages 4414–4419

[BOD10] Bin-Hammam, G. ; Orin, D. E. ; Dariush, B.: Whole-body humanoid control from upper-body task specifications. In: *2010 IEEE International Conference on Robotics and Automation*, 2010, pages 3398–3405

[Bro83] Brooks, R. A.: Solving the find-path problem by good representation of free space. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13 (1983), Nr. 2, pages 190–197

[Cas18] Castañeda, F. (supervisor: Munoz J. D.): *Time invariant motion controller for a serial manipulator*. Karlsruhe, Germany, Hoschschule Karlsruhe, Master Thesis, September 2018

[CDCK06] Chitrakaran, V. K. ; Dawson, D. M. ; Chen, J. ; Kannan, H.: Velocity and structure estimation of a moving object using a moving monocular camera. In: *2006 American Control Conference*, 2006

[CMAL07] Collette, C. ; Micaelli, A. ; Andriot, C. ; Lemerle, P.: Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts. In: *2007 7th IEEE-RAS International Conference on Humanoid Robots*, IEEE, 29.11.2007 - 01.12.2007, pages 81–88

[CY14] Chu, H. ; Yang, H.: A simple image-based object velocity estimation approach. In: *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*, 2014, pages 102–107

[DAHH06] De Luca, A. ; Albu-Schäffer, A. ; Haddadin, S. ; Hirzinger, G.: Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pages 1623–1630

[DASO+07] De Santis, A. ; Albu-Schäffer, A. ; Ott, C. ; Siciliano, B. ; Hirzinger, G.: The skeleton algorithm for self-collision avoidance of a humanoid manipulator. In: *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, IEEE, 04.09.2007 - 07.09.2007, pages 1–6

[DD97] Das, I. ; Dennis, J. E.: A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. In: *Structural optimization* 14 (1997), Nr. 1, pages 63–69

[Del18] Del Prete, A.: Joint Position and Velocity Bounds in Discrete-Time Acceleration/-

Torque Control of Robot Manipulators. In: *IEEE Robotics and Automation Letters* 3 (2018), Nr. 1, pages 281–288

[DF12] DE LUCA, A. ; FLACCO, F.: Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration. In: *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, IEEE, 24.06.2012 - 27.06.2012, pages 288–295

[Die16] DIETRICH, A.: *Whole-body impedance control of wheeled humanoid robots*. Bd. 116. Springer, 2016

[DLM92] DE LUCA, A ; MANES, C: Hybrid force-position control for robots in contact with dynamic environments. In: *Robot Control 1991*. Elsevier, 1992, pages 177–182

[DMA+11] DIFTLER, M. A. ; MEHLING, J. ; ABDALLAH, M. E. ; RADFORD, N. A. ; BRIDGWATER, L. B. ; SANDERS, A. M. ; ASKEW, R. S. ; LINN, D. M. ; YAMOKOSKI, J. D. ; PERMENTER, F et al.: Robonaut 2-the first humanoid robot in space. In: *2011 IEEE international conference on robotics and automation* IEEE, 2011, pages 2178–2183

[DMB93] DOTY, K. L. ; MELCHIORRI, C. ; BONIVENTO, C.: A theory of generalized inverses applied to robotics. In: *The International Journal of Robotics Research* 12 (1993), Nr. 1, pages 1–19

[DOAS13] DIETRICH, A. ; OTT, C. ; ALBU-SCHÄFFER, A.: Multi-objective compliance control of redundant manipulators: Hierarchy, control, and stability. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 03.11.2013 - 07.11.2013, pages 3043–3050

[DOAS15a] DIETRICH, A. ; OTT, C. ; ALBU-SCHÄFFER, A.: An overview of null space projections for redundant, torque-controlled robots. In: *I. J. Robotics Res.* 34 (2015), pages 1385–1400

[DOAS15b] DIETRICH, A. ; OTT, C. ; ALBU-SCHÄFFER, A.: An overview of null space projections for redundant, torque-controlled robots. In: *The International Journal of Robotics Research* 34 (2015), Nr. 11, pages 1385–1400

[DOP18] DIETRICH, A. ; OTT, C. ; PARK, J.: The Hierarchical Operational Space Formulation: Stability Analysis for the Regulation Case. In: *IEEE Robotics and Automation Letters* 3 (2018), Nr. 2, pages 1120–1127

[DPNMN15] DEL PRETE, A. ; NORI, F. ; METTA, G. ; NATALE, L.: Prioritized motion–force control of constrained fully-actuated robots: Task Space Inverse Dynamics. In: *Robotics and Autonomous Systems* 63 (2015), pages 150–157

[DR18] DONG, Y. ; REN, B.: UDE-Based Variable Impedance Control of Uncertain Robot Systems. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2018), pages 1–12

[DS19] DEHIO, N. ; STEIL, J. J.: Dynamically-consistent Generalized Hierarchical Control. In: *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 20.05.2019 - 24.05.2019, pages 1141–1147

[DSBS09] DECRE, W. ; SMITS, R. ; BRUYNINCKX, H. ; SCHUTTER, J. de: Extending iTaSC to support inequality constraints and non-instantaneous task specification. In: *2009 IEEE International Conference on Robotics and Automation*, IEEE, 12.05.2009 - 17.05.2009, pages 964–971

[DSW+18] DEHIO, N. ; SMITH, J. ; WIGAND, D. L. ; XIN, G. ; LIN, H. ; STEIL, J. J. ; MISTRY, M.: Modeling and Control of Multi-Arm and Multi-Leg Robots: Compensating for Object Dynamics During Grasping. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 21.05.2018 - 25.05.2018, pages 294–301

[DW95] DEO, A. S. ; WALKER, I. D.: Overview of damped least-squares methods for inverse kinematics of robot manipulators. In: *Journal of Intelligent and Robotic Systems* 14 (1995), Sep, Nr. 1, pages 43–68

[DWASH12] DIETRICH, A. ; WIMBOCK, T. ; ALBU-SCHÄFFER, A. ; HIRZINGER, G.: Reactive Whole-Body Control: Dynamic Mobile Manipulation Using a Large Number of Actuated Degrees of Freedom. In: *IEEE Robotics & Automation Magazine* 19 (2012), Nr. 2, pages 20–33

[DWT+11] DIETRICH, A. ; WIMBOCK, T. ; TAUBIG, H. ; ALBU-SCHÄFFER, A. ; HIRZINGER, G.: Extensions to reactive self-collision avoidance for torque and position controlled humanoids. In: *2011 IEEE International Conference on Robotics and Automation*, IEEE, 09.05.2011 - 13.05.2011, pages 3455–3462

[EMW14] ESCANDE, A. ; MANSARD, N. ; WIEBER, P.: Hierarchical quadratic programming: Fast online humanoid-robot motion generation. In: *The International Journal of Robotics Research* 33 (2014), Nr. 7, pages 1006–1028

[EWO+14] ENGLSBERGER, J. ; WERNER, A. ; OTT, C. ; HENZE, B. ; ROA, M. A. ; GAROFALO, G. ; BURGER, R. ; BEYER, A. ; EIBERGER, O. ; SCHMID, K. et al.: Overview of the torque-controlled humanoid robot TORO. In: *2014 IEEE-RAS International Conference on Humanoid Robots* IEEE, 2014, pages 916–923

[FDK12] FLACCO, F. ; DE LUCA, A. ; KHATIB, O.: Prioritized multi-task motion control of redundant robots under hard joint constraints. (2012), pages 3970–3977

[FDK15] FLACCO, F. ; DE LUCA, A. ; KHATIB, O.: Control of Redundant Robots Under

Hard Joint Constraints: Saturation in the Null Space. In: *IEEE Transactions on Robotics* 31 (2015), June, Nr. 3, pages 637–654

[FFB+18] FICHT, G. ; FARAZI, H. ; BRANDENBURGER, A. ; RODRIGUEZ, D. ; PAVLICHENKO, D. ; ALLGEUER, P. ; HOSSEINI, M. ; BEHNKE, S.: NimbRo-OP2X: Adult-sized open-source 3D printed humanoid robot. In: *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* IEEE, 2018, pages 1–9

[FKLK12] FLACCO, F. ; KROGER, T. ; LUCA, A. de ; KHATIB, O.: A depth space approach to human-robot collision avoidance. In: *2012 IEEE International Conference on Robotics and Automation*, IEEE, 14.05.2012 - 18.05.2012, pages 338–345

[FKP+14] FERREAU, H. J. ; KIRCHES, C. ; POTSCHKA, A. ; BOCK, H. G. ; DIEHL, M.: qpOASES: A parametric active-set algorithm for quadratic programming. In: *Mathematical Programming Computation* 6 (2014), Nr. 4, pages 327–363

[FL13] FLACCO, F. ; LUCA, A. de: Optimal redundancy resolution with task scaling under hard bounds in the robot joint space. In: *2013 IEEE International Conference on Robotics and Automation*, IEEE, 5/6/2013 - 5/10/2013, pages 3969–3975

[FM14] FERRO, F. ; MARCHIONNI, L.: Reem: A humanoid service robot. In: *ROBOT2013: First Iberian Robotics Conference* Springer, 2014, pages 521–525

[FRA] FRANKA EMIKA: *Panda Powertool*. `https://www.franka.de/technology`. – (visited on: 2020-12-05)

[Gau29] GAUSS, C. F.: Über ein neues allgemeines Grundgesetz der Mechanik. In: *Journal für die reine und angewandte Mathematik* 1829 (1829), Nr. 4, pages 232 – 235

[GKB11] GRIBOVSKAYA, E. ; KHEDDAR, A. ; BILLARD, A.: Motion learning and adaptive impedance for robot control during physical interaction with humans. In: *2011 IEEE International Conference on Robotics and Automation*, 2011, pages 4326–4332

[GTW+20] GROENHUIS, V. ; TAGLIABUE, E. ; WELLEWEERD, M. K. ; SIEPEL, F. J. ; MUÑOZ, J. D. ; MARIS, B. M. ; DALL'ALBA, D. ; ZIMMERMANN, U. ; FIORINI, P. ; STRAMIGIOLI, S.: Deformation Compensation in Robotically-Assisted Breast Biopsy. In: *Abstract from 11th International Conference on Information Processing in Computer-Assisted Interventions, IPCAI 2020, IPCAI 2020*, 2020

[HADH08] HADDADIN, S. ; ALBU-SCHÄFFER, A. ; DE LUCA, A. ; HIRZINGER, G.: Collision detection and reaction: A contribution to safe physical Human-Robot Interaction. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pages 3356–3363

[HASH09] HADDADIN, S. ; ALBU-SCHÄFFER, A. ; HIRZINGER, G.: Requirements for Safe Robots: Measurements, Analysis and New Insights. In: *The International Journal of Robotics Research* 28 (2009), Nr. 11-12, pages 1507–1527

[HB05] HUO, L. ; BARON, L.: Kinematic inversion of functionally-redundant serial manipulators: application to arc-welding. In: *Transactions of the Canadian Society for Mechanical Engineering* 29 (2005), Nr. 4, pages 679–690

[Her76] HERRMANN, L. R.: Laplacian-Isoparametric Grid Generation Scheme. In: *Journal of the Engineering Mechanics Division* 102 (1976), Nr. 5, pages 749–756. – ISSN 0044–7951

[HLM+18] HOFFMAN, E. M. ; LAURENZI, A. ; MURATORE, L. ; TSAGARAKIS, N. G. ; CALDWELL, D. G.: Multi-Priority Cartesian Impedance Control Based on Quadratic Programming Optimization. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pages 309–315

[HLP12] HAN, H. ; LEE, J. ; PARK, J.: A continuous task transition algorithm for operational space control framework. In: *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2012, pages 148–152

[Hog84] HOGAN, N.: Impedance control: An approach to manipulation. In: *1984 American control conference* IEEE, 1984, pages 304–313

[HP13] HAN, H. ; PARK, J.: Robot Control near Singularity and Joint Limit Using a Continuous Task Transition Algorithm. In: *International Journal of Advanced Robotic Systems* 10 (2013), Nr. 10, pages 346

[HRO16] HENZE, B. ; ROA, M. A. ; OTT, C.: Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios. In: *The International Journal of Robotics Research* 35 (2016), Nr. 12, pages 1522–1543

[HS87] HOLLERBACH, J. ; SUH, K.: Redundancy resolution of manipulators through torque optimization. In: *IEEE Journal on Robotics and Automation* 3 (1987), Nr. 4, pages 308–316

[HSG+14] HUTTER, M. ; SOMMER, H. ; GEHRING, C. ; HOEPFLINGER, M. ; BLOESCH, M. ; SIEGWART, R.: Quadrupedal locomotion using hierarchical operational space control. In: *The International Journal of Robotics Research* 33 (2014), Nr. 8, pages 1047–1062

[ISO16] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: Robots and robotic devices - Collaborative robots. Version: ISO/TS 15066. 2016. – Standard

[KB14] KRONANDER, K. ; BILLARD, A.: Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction. In: *IEEE Transactions on Haptics* 7 (2014), July, Nr. 3, pages 367–380

[KDF+16] KUINDERSMA, S. ; DEITS, R. ; FALLON, M. ; VALENZUELA, A. ; DAI, H. ; PERMENTER, F. ; KOOLEN, T. ; MARION, P. ; TEDRAKE, R.: Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. In: *Autonomous robots* 40 (2016), Nr. 3, pages 429–455

[Kha80] KHATIB, O.: *Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles*. 1980

[Kha85] KHATIB, O.: Real-time obstacle avoidance for manipulators and mobile robots. In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation* Bd. 2, 1985, pages 500–505

[Kha87a] KHATIB, O.: A unified approach for motion and force control of robot manipulators: The operational space formulation. In: *IEEE Journal on Robotics and Automation* 3 (1987), February, Nr. 1, pages 43–53

[Kha87b] KHATIB, O.: A unified approach for motion and force control of robot manipulators: The operational space formulation. In: *IEEE Journal on Robotics and Automation* 3 (1987), February, Nr. 1, pages 43–53

[Kha90a] KHATIB, O.: Motion/force redundancy of manipulators. In: *Japan USA Symposium on Flexible Automation, Kyoto, Japan, July* Bd. 23, 1990, pages 337–342

[Kha90b] KHATIB, O.: Reduced Effective Inertia in Macro-/Mini-manipulator Systems. In: *The Fifth International Symposium on Robotics Research*. Cambridge, MA, USA : MIT Press, 1990, 279–284

[Kha95] KHATIB, O.: Inertial Properties in Robotic Manipulation: An Object-Level Framework. In: *The International Journal of Robotics Research* 14 (1995), Nr. 1, pages 19–36

[KKB14] KHANSARI, M. ; KRONANDER, K. ; BILLARD, A.: Modeling robot discrete movements with state-varying stiffness and damping: A framework for integrated motion generation and impedance control, 2014

[KLT06] KAPOOR, A. ; LI, M. ; TAYLOR, R. H.: Constrained control for surgical assistant robots. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*, IEEE, May 15-19, 2006, pages 231–236

[KLW11] KANOUN, O. ; LAMIRAUX, F. ; WIEBER, P.: Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task. In: *IEEE Transactions on Robotics* 27 (2011), Nr. 4, pages 785–792

[KMB20] KASPAR, M. ; MUÑOZ, J. D. ; BOCK, J.: Sim2Real Transfer for Reinforcement Learning without Dynamics Randomization. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pages 185–191

[KSPW04] KHATIB, O. ; SENTIS, L. ; PARK, J. ; WARREN, J.: Whole-body dynamic behavior and control of human-like robots. In: *International Journal of Humanoid Robotics* 1 (2004), Nr. 01, pages 29–43

[KUK] KUKA AG: *LBR iiwa*. `https://www.kuka.com/en-se/products/robotics-systems/industrial-robots/lbr-iiwa`. – (visited on: 2020-12-05)

[KZK17] KHANSARI-ZADEH, S. ; KHATIB, O.: Learning Potential Functions from Human Demonstrations with Encapsulated Dynamic and Compliant Behaviors. In: *Auton. Robots* 41 (2017), Januar, Nr. 1, pages 45–69

[Lie77] LIEGEOIS, A.: Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms. In: *IEEE Transactions on Systems, Man, and Cybernetics* 7 (1977), Nr. 12, pages 868–871

[LME+11] LIU, M. ; MICAELLI, A. ; EVRARD, P. ; ESCANDE, A. ; ANDRIOT, C.: Interactive dynamics and balance of a virtual character during manipulation tasks. In: *2011 IEEE International Conference on Robotics and Automation*, IEEE, 09.05.2011 - 13.05.2011, pages 1676–1682

[LMP12] LEE, J. ; MANSARD, N. ; PARK, J.: Intermediate Desired Value Approach for Task Transition of Robots in Kinematic Control. In: *IEEE Transactions on Robotics* 28 (2012), Dec, Nr. 6, pages 1260–1277

[LP90] LOZANO-PEREZ, T.: Spatial planning: A configuration space approach. In: *Autonomous robot vehicles*. Springer, 1990, pages 259–271

[LTP16] LIU, M. ; TAN, Y. ; PADOIS, V.: Generalized hierarchical control. In: *Autonomous Robots* 40 (2016), Nr. 1, pages 17–31

[LVYK13] LENGAGNE, S. ; VAILLANT, J. ; YOSHIDA, E. ; KHEDDAR, A.: Generation of whole-body optimal dynamic multi-contact motions. In: *The International Journal of Robotics Research* 32 (2013), Nr. 9-10, pages 1104–1119

[MA20] MUÑOZ, J. D. ; ABDELRAHMAN, A.: *Robot Control*. Februar 20 2020. – European patent application EP3871841A1

[MA21] MUÑOZ, J. D. ; ALLMENDINGER, F.: *Bewegen einer roboterfesten Referenz*. Januar 28 2021. – German patent DE102019220619B3

[MAAZ20] MUÑOZ, J. D. ; ABDELAZIM, A. ; ALLMENDINGER, F. ; ZIMMERMANN, U. E.: Unilateral Constraints for Torque-based Whole-Body Control. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pages 185–191

[MAF+19a] MUÑOZ, J. D. ; ALLMENDINGER, F. ; FIORE, M. D. ; ZIMMERMANN, U. E. ; ORTMAIER, T.: Physical Human-Robot Interaction under Joint and Cartesian Constraints. In: *2019 19th International Conference on Advanced Robotics (ICAR)*, 2019, pages 185–191

[MAF19b] MUÑOZ, J. D. ; ALLMENDINGER, F. ; FIORE, M.: *Steuern eines Roboters*. Juni 19 2019. – German patent DE102018207921B3

[Mar20] MARTINEZ, M. (supervisor: Munoz J. D.): *Remote control of a redundant robot with haptic feedback*. Karlsruhe, Germany, Hoschschule Karlsruhe, Master Thesis, August 2020

[MCAZ19] MUÑOZ, J. D. ; CASTAÑEDA, F. ; ALLMENDINGER, F. ; ZIMMERMANN, U. E.: Time Invariant Motion Controller for Physical Human Robot Interaction. Volume 5A: 43rd Mechanisms and Robotics Conference (2019), 08

[Mes] MESHLAB: *MeshLab*. `https://www.meshlab.net`. – (visited on: 2020-12-05)

[MFA18] MUÑOZ, J. D. ; FIORE, M. ; ALLMENDINGER, F.: Operational Space Framework under Joint Constraints. In: *Proceedings. 2018 IDETC/CIE International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2018

[MKK09] MANSARD, N. ; KHATIB, O. ; KHEDDAR, A.: A Unified Approach to Integrate Unilateral Constraints in the Stack of Tasks. In: *IEEE Transactions on Robotics* 25 (2009), June, Nr. 3, pages 670–685

[MLOH03] MARAYONG, P. ; LI, M. ; OKAMURA, A. M. ; HAGER, G. D.: Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures. In: *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, IEEE, 14-19 Sept. 2003, pages 1954–1959

[Mor80] MORAVEC, H. P.: *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. Stanford, CA, USA, Diss., 1980. – AAI8024717

[MS20] MUÑOZ, J. D. ; SAFRONOV, K.: *Modellieren von Objekten mittels robotergeführter Kameras*. April 16 2020. – German patent application DE102020204829A1

[Muñ19] MUÑOZ, J. D.: *Robotersteuerung*. Juni 6 2019. – German patent DE102018207919B3

[MUR] MURAB CONSORTIUM: *MURAB Project MRI and Ultrasound Robotic Assisted Biopsy*. `https://www.murabproject.eu/`. – (visited on: 2020-12-5)

[Nat03] NATALE, C.: *Interaction Control of Robot Manipulators: Six degrees-of-freedom tasks*. Bd. 3. Berlin, Heidelberg : Springer Berlin Heidelberg, 2003

[NHY87] NAKAMURA, Y. ; HANAFUSA, H. ; YOSHIKAWA, T.: Task-Priority Based Redundancy Control of Robot Manipulators. In: *The International Journal of Robotics Research* 6 (1987), Nr. 2, pages 3–15

[NKS+10] NAGASAKA, K. ; KAWANAMI, Y. ; SHIMIZU, S. ; KITO, T. ; TSUBOI, T. ; MIYAMOTO, A. ; FUKUSHIMA, T. ; SHIMOMURA, H.: Whole-body cooperative force control for a two-armed and two-wheeled mobile robot using Generalized Inverse Dynamics and Idealized Joint Units. In: *2010 IEEE International Conference on Robotics and Automation*, 2010, pages 3377–3383

[ODAS15] OTT, C. ; DIETRICH, A. ; ALBU-SCHÄFFER, A.: Prioritized multi-task compliance control of redundant manipulators. In: *Automatica* 53 (2015), pages 416–423

[ORH11] OTT, C. ; ROA, M. A. ; HIRZINGER, G.: Posture and balance control for biped robots based on contact force optimization. In: *2011 11th IEEE-RAS International Conference on Humanoid Robots*, IEEE, 26.10.2011 - 28.10.2011, pages 26–33

[Ott08] OTT, C.: *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*. Bd. 49. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008

[PCRBJ16] PROIETTI, T. ; CROCHER, V. ; ROBY-BRAMI, A. ; JARRASSE, N.: Upper-Limb Robotic Exoskeletons for Neurorehabilitation: A Review on Control Strategies. In: *IEEE reviews in biomedical engineering* 9 (2016), pages 4–14. `http://dx.doi.org/10.1109/RBME.2016.2552201`. – DOI 10.1109/RBME.2016.2552201

[PMU+08] PETERS, J. ; MISTRY, M. ; UDWADIA, F. ; NAKANISHI, J. ; SCHAAL, S.: A unifying framework for robot control with redundant DOFs. In: *Autonomous Robots* 24 (2008), Nr. 1, pages 1–12

[QA19] QUIROZ-OMAÑA, J. J. ; ADORNO, B. V.: Whole-Body Control With (Self) Collision Avoidance Using Vector Field Inequalities. In: *IEEE Robotics and Automation Letters* 4 (2019), Oct, Nr. 4, pages 4048–4053

[RC81] RAIBERT, M. H. ; CRAIG, J. J.: Hybrid Position/Force Control of Manipulators. In: *Journal of Dynamic Systems, Measurement, and Control* 103 (1981), 06, Nr. 2, pages 126–133

[RCT+18] RAIOLA, G. ; CARDENAS, C. A. ; TADELE, T. S. ; DE VRIES, T. ; STRAMIGIOLI, S.: Development of a Safety- and Energy-Aware Impedance Controller for Collaborative Robots. In: *IEEE Robotics and Automation Letters* 3 (2018), April, Nr. 2, pages 1237–1244

[Ret] RETHINK ROBOTICS: *Sawer Black Edition*. `https://www.rethinkrobotics.com/sawyer`. – (visited on: 2020-12-02)

[Rob] ROBOCEPTION GMBH: *rc visard 65 monochrome*. `https://roboception.com/product/rc_visard-65-monochrome`. – (visited on: 2020-12-05)

[RPB+12] RUBRECHT, S. ; PADOIS, V. ; BIDAUD, P. ; BROISSIA, M. de ; DA SILVA SIMOES, M.: Motion safety and constraints compatibility for multibody robots. In: *Autonomous Robots* 32 (2012), Nr. 3, pages 333–349

[SC16] SIMETTI, E. ; CASALINO, G.: A Novel Practical Technique to Integrate Inequality Control Objectives and Task Transitions in Priority Based Control. In: *Journal of Intelligent & Robotic Systems* 84 (2016), Nr. 1-4, pages 877–902

[SCOL+04] SORKINE, O. ; COHEN-OR, D. ; LIPMAN, Y. ; ALEXA, M. ; RÖSSL, C. ; SEIDEL, H.-P.: Laplacian surface editing. In: BOISSONNAT, Jean-Daniel (Hrsg.) ; ALLIEZ, Pierre (Hrsg.): *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing - SGP '04*. New York, New York, USA : ACM Press, 2004. – ISBN 3905673134, pages 175

[SGJG10] SUGIURA, H. ; GIENGER, M. ; JANSSEN, H. ; GOERICK, C.: Reactive Self Collision Avoidance With Dynamic Task Prioritization For Humanoid Robots. In: *International Journal of Humanoid Robotics* 07 (2010), Nr. 01, pages 31–54

[SGR+15] SOMANI, N. ; GASCHLER, A. ; RICKERT, M. ; PERZYLO, A. ; KNOLL, A.: Constraint-based task programming with CAD semantics: From intuitive specification to real-time control. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 28.09.2015 - 02.10.2015, pages 2854–2859

[SK05] SENTIS, L. ; KHATIB, O.: Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. In: *International Journal of Humanoid Robotics* 2 (2005), Nr. 04, pages 505–518

[SK06] SENTIS, L. ; KHATIB, O.: A whole-body control framework for humanoids operating in human environments. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pages 2641–2648

[SPB11] SALINI, J. ; PADOIS, V. ; BIDAUD, P.: Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions. In: *2011 IEEE International Conference on Robotics and Automation*, IEEE, 09.05.2011 - 13.05.2011, pages 1283–1290

[SRG+16] SOMANI, N. ; RICKERT, M. ; GASCHLER, A. ; CAI, C. ; PERZYLO, A. ; KNOLL, A.: Task level robot programming using prioritized non-linear inequality constraints. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 09.10.2016 - 14.10.2016, pages 430–437

[SRK+13] SAAB, L. ; RAMOS, O. E. ; KEITH, F. ; MANSARD, N. ; SOUERES, P. ; FOURQUET, J.: Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints. In: *IEEE Transactions on Robotics* 29 (2013), Nr. 2, pages 346–362

[SS91] SICILIANO, B. ; SLOTINE, J.-J.E.: A general framework for managing multiple tasks in highly redundant robotic systems. In: *Fifth International Conference on Advanced*

Robotics 'Robots in Unstructured Environments, IEEE, 19.06.1991 - 22.06.1991, pages 1211–1216 vol.2

[SSB10] SCHREIBER, G. ; STEMMER, A. ; BISCHOFF, R.: The Fast Research Interface for the KUKA Lightweight Robot, 2010

[SSC14] STEINBRUCKER, Frank ; STURM, Jurgen ; CREMERS, Daniel: Volumetric 3D mapping in real-time on a CPU. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 31.05.2014 - 07.06.2014. – ISBN 978–1–4799–3685–4, pages 2021–2028

[SSVO10] SICILIANO, B. ; SCIAVICCO, L. ; VILLANI, L. ; ORIOLO, G.: *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010

[SWA+02] SAKAGAMI, Y. ; WATANABE, R. ; AOYAMA, C. ; MATSUNAGA, S. ; HIGAKI, N. ; FUJIMURA, K.: The intelligent ASIMO: system overview and integration. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* Bd. 3, 2002, pages 2478–2483 vol.3

[Tom20] TOM WEIDEMAN: *UCD: Physics 9A - Classical Mechanics*. 2020 https://batch.libretexts.org/print/url=https://phys.libretexts.org. – (visited on: 2020-11-14)

[Toy] TOYOTA GLOBAL NEWSROOM: *Toyota Unveils Third Generation Humanoid Robot T-HR3*. https://global.toyota/en/detail/19666346. – (visited on: 2020-12-02)

[Udw03] UDWADIA, F. E.: A new perspective on the tracking control of nonlinear structural and mechanical systems. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 459 (2003), Nr. 2035, pages 1783–1800

[UK92] UDWADIA, Firdaus E. ; KALABA, Robert E.: A new perspective on constrained motion. In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439 (1992), Nr. 1906, pages 407–410

[Wel13] WELLER, R.: *New geometric data structures for collision detection and haptics*. Springer Science & Business Media, 2013

[Win18] WINTER, M. (supervisor: Munoz J. D. ; Paris, J.): *Torque Control of a 3 DOF Surgical Tool*. Achen, Germany, RWTH Achen University, Master Thesis, April 2018

[Yan14] YANG, X.: Chapter 14 - Multi-Objective Optimization. In: *Nature-Inspired Optimization Algorithms*. Oxford : Elsevier, 2014, pages 197–211

[YUA] YUANDA ROBOTICS: *Designed to work*. https://www.yuanda-robotics.de/roboter/. – (visited on: 2020-12-05)

[ZF00] ZHANG, Shilong ; FASSE, Ernest D.: Spatial Compliance Modeling Using a Quaternion-Based Potential Function Method. In: *Multibody System Dynamics* 4 (2000), Nr. 1, pages 75–101

Curriculum Vitae

	Personal Information
Name	Muñoz Osorio, Juan David
Birth	31 March 1990 in Zipaquiá, Colombia
	Professional Experience
since Apr/17	Researcher and developer at Coporate Research of KUKA Deutschland GmbH
Nov/13 - Aug/14	Technical support and research in the project "Design and Construction of a parallel robot for machining in 5 Axes" at National University of Colombia, Bogotá, Colombia
	Education
since Apr/17	Ph. D. student Institute of Mechatronic Systems (imes), LUH
Oct/14 - Mar/17	Mechatronic engineering (M. Sc.), at LUH, Focus: Techniques of production, Automation and robotics
Nov/12 - Jul/13	Semester abroad at the University of Kassel (Germany)
Jul/07 - Jul/13	Mechatronic engineering (B. Sc.), National University of Colombia, Bogotá, Colombia
Jan/01 - Dec/06	High school at the Industrial Technical Institute of Zipaquirá
	Study-related activities and internships
Apr/16 - Jul/16	Internship at Corporate Research in KUKA Deutschland GmbH, Augsburg
Apr/15 - Mar/16	Auxiliary scientist at Institute for Mechatronic Systems, LUH
Aug/15 - Mar/16	Working student at Continental AG, Hannover
Nov/12 - Jul/13	Auxiliary scientist at Institute for Measures and Control at the Kassel University ,Kassel
Apr/12 - Aug/12	Working student at Bioinformatic center of the National University of Colombia, Bogotá, Colombia
Jan/11 - Feb/11	Member of CEIMTUN (student committee of mechatronics students) - Universidad Nacional de Colombia