# Industrial Human-Robot Collaboration:
## Maximizing Performance While Maintaining Safety

Dipl.-Ing. Roman Weitschat

**Ph.D. Thesis**

# Industrial Human-Robot Collaboration: Maximizing Performance While Maintaining Safety

Dissertation

zur

Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

am Lehrstuhl für Mechatronik

der Fakultät für Maschinenbau und Schiffstechnik

der Universität Rostock

vorgelegt von

Roman Weitschat, geb. am 20.03.1985 in Neubrandenburg

aus München

Rostock, den 14.05.2019

https://doi.org/10.18453/rosdok_id00002490

**Gutachter:**

Prof. Dr.-Ing. Harald Aschemann, Lehrstuhl für Mechatronik, Universität Rostock

Prof. Dr.-Ing. Alin Albu-Schäffer, Fakultät für Informatik, Technische Universität München

**Jahr der Einreichung:** 2018

**Jahr der Verteidigung:** 2019

# Abstract

For many years, separated autonomous robotic systems have been an essential component in industrial manufacturing. In particular, these heavy-payload robots perform a wide range of tasks, where high precision and repeatability is crucial. A flexible adaptation of fast changing tasks or environments as well as the interaction with humans can rather not be realized by these types of robots. Recently, a paradigm shift regarding customer demand could be observed. Short product life-cycles as well as increasing individualization of products require flexible manufacturing processes. Therefore, novel light-weight robot technology was developed, which enables the collaboration of humans and robots. In particular, highly productive robots are combined with the high flexibility of humans. However, only a few collaborative applications have been established in industry, which is mainly due to the low efficiency, i.e., large cycle times caused by safety regulations.

The goal of this thesis is to maximize performance in collaborative applications, while maintaining safety. For this, assembly workplaces are analyzed, typical tasks identified, and the potential of collaborative robots is elaborated. Current safety regulations are analyzed in order to identify the challenges in safe human-robot collaboration. Then, a novel control method is presented, which enables intuitive, safe, and efficient control of robots. The Mirroring Human Arm Motions approach presents a velocity-limited trajectory generation, in particular, for orientations in quaternion space. This method is extended to an online via-point trajectory generation in order to enable an adjustment of velocity limits for guaranteeing safety in real-time. Furthermore, in collaborative applications particularly collisions with the human arm are likely to occur. Therefore, human-arm performance is analyzed and experiments similar to typical collaborative scenarios are executed, to determine the dynamic properties. By exploiting the obtained information on human arm dynamics, a novel approach to improve the performance of robot motions is presented. From the experiments, a simplified human arm model is derived, which enables the calculation of movements of the human into the path of the robot. With this approach, a maximum robot velocity depending on kinematic limitations of robots and human-in-the-loop constraints can be determined. This idea is further developed into a nonlinear optimization problem, where minimal-time motions are found and applications with low-cycle times can be realized. In order to enable flexible robot motions within the entire workspace of the robot, a generalization method using Dynamic Movement Primitives is presented. It contains a novel real-time consideration of spacial and kinematic constraints, to fulfill the requirements on safe human-robot collaboration. Experiments on a collaborative workbench prove the effectiveness of the presented methods. Finally, a novel airbag technology is proposed, which enables a protective coverage of dangerous tools and objects and protects humans against injuries, caused by a collision with the robot. The so called Robotic Airbag is inflated with pressured air to create a cushion around sharp edges of tool and object. Intrinsic safety is guaranteed, as the airbag is always inflated before initiating a robot motion. In order to exclude an affect of the tool functionality, the Robotic Airbag can be deflated whenever required. Experiments with a crash-test dummy, and finally with a volunteer, prove the functionality and compliance with current safety standards. In Summary, the presented methods in this thesis enable a significant improvement of efficiency and safety in collaborative applications.

# Zusammenfassung

Seit vielen Jahren sind Robotersysteme ein wesentlicher Bestandteil in der industriellen Fertigung. Insbesondere klassische Industrieroboter übernehmen eine Vielzahl an unterschiedlichen Aufgaben, welche durch hohe Präzision, sowie eine hohe Anzahl an Wiederholungen ausgezeichnet sind. Durch die hohen Rüstzeiten dieser Systeme ist die Flexibilität bezüglich schnell wechselnder Aufgaben eher bedingt gegeben. Da allerdings die Nachfrage an individualisierten Produkten steigt und sich die Produktlebenszeiten verringern, werden anpassungsfähige Fertigungsprozesse benötigt. Neuartige Leichbaurobotertechnologie ermöglicht eine flexible Produktion durch die Kollaboration von Mensch und Roboter (MRK), die im Wesentlichen durch den schutzzaunlosen Einsatz von Robotern in unmittelbarer Umgebung eines Menschen charakterisiert wird. Die hohe Produktivität des Roboters soll mit der unbegrenzten Flexibilität des Menschen vereint werden. Jedoch haben sich kollaborative Applikationen bis jetzt noch nicht etabliert, da die Effizienz aufgrund von strengen Sicherheitsanforderung stark beeinträchtigt wird.

Das Ziel dieser Dissertation ist die Maximierung der Leistungsfähigkeit in kollaborativen Roboteranwendungen bei gleichzeitiger Einhaltung der Sicherheitsstandards. Dafür werden unterschiedliche Montagearbeitsplätze analysiert, typische Aufgaben identifiziert und das Potential von kolloborativen Robotern dargestellt. Aktuelle Sicherheitsstandards werden untersucht, um die wesentlichen Herausforderung zum effizienten Einsatz dieser Systeme zu identifizieren. Um effizente schutzzaunlose Roboteranwendungen zu ermöglichen, werden unterschiedliche Lösungsansätze präsentiert. Zuerst wird eine neuartige Methode beschrieben, die eine intuitive, sichere und effiziente Steuerung des Roboters ermöglicht. Mit dem „Mirroring Human Arm Motions" Ansatz kann eine geschwindigkeitsbegrenzte Trajektoriengenerierung realisiert werden, mit der in Echt-Zeit Geschwindigkeitsbegrenzungen angepasst werden können und somit Sicherheitsanforderungen garantiert werden. Eine weitere Entwicklung bezieht sich auf die direkte Koexistenz von Mensch und Roboter. Befinden sich beide gleichzeitig in einem gemeinsamen Arbeitsraum, dann ist die Wahrscheinlichkeit einer Kollision sehr hoch, insbesondere eine Kollision mit dem menschlichen Arm. Daher werden in einer Reihe von Experimenten die dynamischen Eigenschaften des menschlichen Arms analysiert. Aus diesem dynamischen Verhalten wird ein vereinfachtes dynamisches Modell erzeugt, um potentielle Armbewegungen simulieren zu können. Mittels dieser Simulation werden Bewegungen in Richtung der Roboterbahn projiziert und eine Dauer bis zu einer möglichen Kollision berechnet. Mit dieser Information wird die maximal zulässige Robotergeschwindigkeit in Abhängikeit der möglichen Bremsverzögerung angepasst bzw. erhöht, um schnellere und somit effizientere Roboterbewegungen zu ermöglichen. Diese Idee wird darauf hin als ein Optimierungsproblem formuliert, bei dem die benötigte Dauer einer Roboterbewegung minimiert wird. Dabei werden menschliche Armbewegung als Nebenbedingung in der Kostenfunktion mitbetrachtet. Mit dieser Methode können Taktzeiten reduziert und somit die Produktivität in MRK-Anwendungen gesteigert werden. Da Optimierungsmethoden schwer in Echtzeitanwendungen einbezogen werden können, werden Generalisierungsansätze unter Verwendung von Dynamic Movement Primitives entwickelt, damit eine höhere Flexibilität ermöglicht wird. Bei dieser Generalisierung wird im Wesentlichen auf die Einhaltung von Nebenbedingungen, wie räumliche Begrenzung, sowie die Limitierung der Geschwindigkeit eingegangen. In Experimenten wird der entwickelte Lösungsansatz überprüft und die Effektivität bewiesen. Um eine Effizienzsteigerung nicht nur auf der Steuerungsseite zu erhalten, sondern alle Möglichkeiten auszuschöpfen, wird ebenfalls eine Hardwareentwicklung präsentiert. Der sogenannte Robotic Airbag wurde entwickelt, um scharfkantige Roboterwerkzeuge sowie beförderte Werkstücke zu umschließen und sicher abzudecken. Eine intrinsische Sicherheit wird dadurch ermöglicht, dass der Airbag vor jeder unsicheren Roboterbewegung mit Druckluft befüllt wird und die scharfkantigen Gegenstände umschließt. Damit die Funktionalität des Roboterwerkzeuges weiterhin gewährleistet wird, kann die Druckluft wieder entlassen und die Umschießung des Werkzeugs sowie des Werkstücks aufgehoben werden, sobald der Roboter sichere, zumeist langsame, Bewegungen ausführt. In Experimenten mit einem Crash-Test Dummy, sowie einem Freiwilligen, wurde die Funktionalität getestet und die Einhaltung gegenwärtiger Standards bewiesen. Zusammenfassend ermöglichen die in dieser Dissertation präsentierten Methoden eine erhebliche Verbesserung der Effizienz, bei Einhaltung gegebener Sicherheitsanforderungen in kollaborativen Anwendungen.

# Acknowledgement

This dissertation was written during my employment at the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR) in cooperation with the University of Rostock. I am very greatful that I have been part of the best research environments in the field of robotics, which is also due to several persons. Accordingly, I would like to thank my supervisors during my work, Prof. Harald Aschemann and Prof. Alin Albu-Schäffer, who enabled me to elaborate this thesis with full liberty. Moreover, I want to thank my superiors Christoph Borst, Michael Suppa, and Freek Stulp for their unlimited support to realize my research ideas and always showing confidence in my work.

However, all of this would not have been possible without my highly esteemed colleagues and friends I made at the DLR Institute. In particular, I will always be deeply grateful to Hannes Höppner and Jörn Vogel, I mostly spend my time with, at work and after work, as well as Florian Schmidt and Alexander Dietrich, for their greatest and always immediate support, and the discussions late into the night. Additionally, I would like to thank my office colleagues Nico Mansfeld, Moritz Maier, and Felix Huber for many interesting discussions at the whiteboard and cooperation in dynamics and control theory.

Furthermore, I would like to thank Robert Burger, Franz Steinmetz and Antonio Gonzales Marin for their helpful assistance by configuring the robot system, which was the basis for successful experiments during my research. Many thanks to the mechanics and electronics laboratory, as well as the organization team at the DLR, who always supported my work.

Most importantly, I want to thank my family, especially my mom, for supporting me at any time and in every situation as well as my grandparents, which always have been there for me. Without them, my fantastic life and academic career would not have been possible. Finally, I want to thank my love Annette for her relentless patience, support, and love, she gave to me.

Munich, 14.05.2019 Roman Weitschat

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Industrial manufacturing is still dominated by high-precision, high-throughput, and heavy-payload robots. Especially in the automotive sector, this type of robots is integrated into the production lines. Due to safety reasons, the operational area is surrounded by fences, to obstruct the access to danger zones. On the one hand, the integration of robots enables a remarkable increase of productivity, but on the other hand the flexibility is limited and the integration requires very high set-up times. However, the increasing individualization and customization of products requires high flexibility in the manufacturing processes. Additionally, short product life cycles, the demographic change and the aging society demands a redesign of conventional manufacturing processes.

In the last years, robot hardware has made remarkable progress in order to enable a safe collaboration of humans and robots. Novel light-weight technology allows for slight collisions and prevents from severe injury caused by robots. Therefore, many companies would like to use collaborative robotic systems, where humans and robots are working side by side. Humans gain access to operating robotic system, as well as interaction possibilities and assistance scenarios can be realized, which can attain an improvement of working conditions and productivity. The combination of highly productive robots and humans with their infinite flexibility creates new opportunities in future manufacturing. Workforce deficiencies due to physical and psychological stress can be reduced by ergonomic and process-oriented support.

Anyhow, no great breakthrough occurred, yet, and only a few collaborative applications were established in industrial assembly or manufacturing areas. This is due to the stringent demand on safety during autonomous movements of robots in fence-less applications. The safety limits lead to a remarkable decrease of efficiency in collaborative applications, and an economic operation of these applications is not possible. Therefore, the objective of this thesis is:

***Maximizing Performance while Maintaining Safety***.

This thesis is motivated by the lack of efficiency in collaborative applications. Current international standards regulate safety requirements on fence-less autonomous robotic systems to protect humans from hazards. The specified limits for human-robot collaboration lead to a remarkable decrease of the operational velocity of a robot and, therefore, to very high cycle times. Previous research predominantly focused on safety to enable collaborative robot applications considering only the robotic system, or integrating unforeseeable reactive collision avoidance path planning methods. Therefore, this thesis is motivated to improve existing methods, and to develop novel control methods and hardware systems, in particular, with human-in-the-loop considerations, to get the maximum out of a collaborative robotic system.

This thesis addresses the optimal deployment of robots in collaborative workbenches in order to improve the usability robots. Given the developed collaborative work cell as a research-platform, the human is further taken into account. Human movements shall serve as basis for the development of novel control concepts in order to increase the operational robot velocity. Furthermore, human demonstration enables intuitive, safe, and efficient parameterization of robot skills. Novel optimization approaches are considering the human as influencing factor, and human-in-the-loop constraints enable an optimal robot motion for an increased efficiency, keeping the minimum safety requirements. In order to simultaneously facilitate the flexibility of the robotic system, generalization methods allow for an online application adaptation. Finally, innovative hardware developments pave the way to economically applicable industrial collaborative robot applications.

## Contribution to robotics research

This thesis contributes to the robotic research as follows:

1. **Application and Market trends studies**
   The preliminary work is a study of current customer needs and supplier trends. Human-robot collaboration is often used in different context. The market trend study investigates the demands on collaborative manufacturing. In an on-site analysis, industrial work places, in particular, industrial assembly applications are analyzed to identify suitable collaborative robot applications. Examples of possible realization of collaborative applications illustrate the opportunities of this technology.
2. **Development of a work cell for human robot collaboration**
   Originating from the work place analysis, an innovative work cell is developed to build a research platform for future scientific work in the field of human-robot collaboration. The workcell is used for the experiments, executed in this thesis, and an insight on possible collaborative workplace designs is presented.
3. **Human arm mirroring for intuitive programming**
   A novel method is presented, to generate smooth, and constrained motions, where human arm motions are mirrored. The online trajectory generation method, including velocity limitation, enables intuitive programming of for example pick and place applications in industrial scenarios.
4. **Online via-point motion generation**
   Initiated by the concept of mirroring human arm motions, online via-point trajectory generation can be derived from these method. The online-generated trajectories enable the adaptation of velocities and addition or removal of points on the path during the execution. The online velocity adaptation enables a remarkable improvement of performance for the novel human-in-the-loop concepts, presented in this thesis.
5. **Simplified dynamic human arm model and human motion experiments**
   To obtain real-time feasible simulation algorithms of the human arm, a simplified dynamic model is developed to provide a minimum computation time. Human arm motion experiments serve as basis for the validation of the dynamic model. Finally, the model can be used to pre-calculate human arm motions in order to predict the human safety situation.
6. **Human motion projection for safe human robot collaboration**
   Conceptual novel methods for commanding robots are developed, to improve performance during collaborative applications. For guaranteeing safety, every potential risk

has to be considered. This includes the human intention to hit the robot, or a reflex motion to a collision with the robot. Here, dynamic simulations and an analysis are performed to estimate a minimum time to a probable collision. This information can be used to increase the operational velocity of the robot and, therefore, the efficiency in the robot task execution.

7. **Optimal path planning under human-in-the-loop constraints**

   In this work, an optimization problem considering human-in-the-loop constraints is introduced. The efficiency of a robot motion is now calculated to a maximum performance in order to gain economic efficient robot applications in collaborative scenarios. These paths are time optimal motions under the consideration of the human motion projection to obtain fast, and simultaneously safe, robot motions.

8. **Adapted exponential trajectory generation**

   To generate time optimal trajectories to ensure path-following behavior, an exponential trajectory generation method is developed. The optimal path, obtained from the optimization, serves as input for the trajectory generation, which in turn serves as basis for online trajectory generalization methods, presented in this thesis.

9. **Generalization of optimal motions**

   To enable a flexible use of robotic systems regarding industrial human-in-the-loop applications, a generalization method is developed to generate motions online, learned from optimal motions, including constraints for shared workspaces. The novel constraint online motion generation method allows for efficient and safe motions for applications aiming at coexisting scenarios. Experiments validate the improvements in efficiency during collaborative tasks.

10. **Performance improvement by hardware development for safe human-robot collaboration**

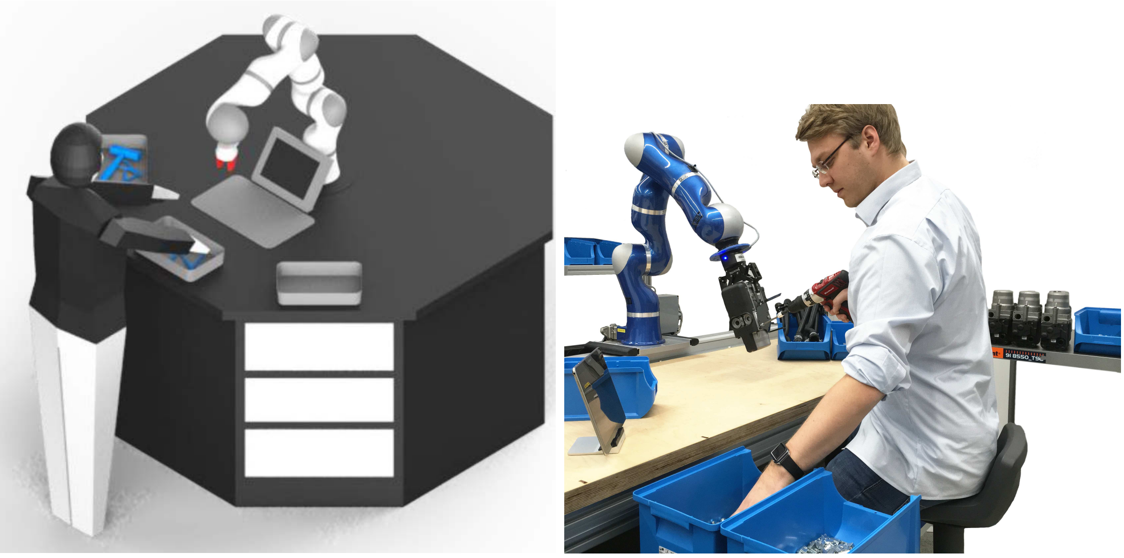
    This innovative development includes a hardware module for improving the efficiency in collaborative robotics. An airbag, developed to cover robotic tools, and simultaneously carried objects, allows for fast and safe robot motions. The functionality of this tool is proven by experiments with a crash test dummy, and finally tested with a human. The results promise a remarkable improvement of efficiency regarding international standards.

The author of this work is also the co-author of several publications, which are included in this thesis. Therefore, graphics, texts, contents, etc. have been used, which are cited whenever applicable.

# Chapter 2

# Human-Robot Collaboration

In the last years, the topic of human-robot collaboration became increasingly important. Market-leading robot-manufacturers, as well as several small enterprises, or start-ups, started to develop robots for workplaces, which are accessible by humans. The industry identified an enormous potential in collaborative manufacturing, as the productivity of robots and the flexibility of humans can now be combined. Collaborative applications range from pure co-existance to completely interactive industrial tasks.



(a) Collaborative application design

(b) Real collaborative assembly scenario

**Figure 2.1:** Human-robot collaboration in industrial applications. Left image: A collaborative application design in a medical scenario. Right image: A real collaborative assembly of an automotive component as illustrative example.

In this chapter, an overview is given about the state of the art in human-robot collaboration. In the first section, the subject *human-robot collaboration* (HRC) will be explained in detail and robot applications are classified. Then, different types of industrial robots are presented, and suitable robots for applications in physical human-robot interaction are described. Furthermore, the market relevance is analyzed with respect to the customer needs and the current robot supplier trends. In order to enable collaborative applications, safety requirements for industrial robots have to be considered, which are written to international standards and technical specifications. Additionally, recently developed control methods are presented, depending on human-in-the-loop parameters for path planning, i.e., including human position measurements into the robot control. Collision avoidance methods are proposed, and recent research in human motion and intention prediction. Finally, the developed Collaborative Workcell is presented.

## 2.1 Classification of Robotic Applications

Human-robot collaboration is used in many different contexts. In general, the demand in the industry aims at the co-existence between humans and robots, which means that the robots are operating autonomously and humans are allowed to enter the robot's workspace, without getting injured or harmed. Furthermore, the robot can be used as a static assistant, and is only moved in the absence of the human. In the following, different types of autonomous safe robot applications and human-robot collaboration are classified.

For the description of a robot application, the following classifications are defined, based on the definition in [1]:

**Physical working area separation**
In the physical working area separation, the deployment of protective fences physically prevent from contact between human and robot during the operation. Peripheral safety function or mechanisms, like contact buttons at the entrance of the robot workspace, guarantee the absence of a human being during a robot motion, as well as the access to this area. A contact with the robot during the operation should be prevented and is possible with a low probability. An interaction with the robot after the stop of the robot is not provided.

**Virtual working area separation**
In the virtual working area separation no real fences are mounted. A monitoring of the working area is executed by light barriers, ground sensors, or by vision based safety systems, which guarantee an absence of a human being. A contact with the robot during the operation should be prevented and is possible with a low probability. An interaction with the robot after the stop of the robot is not provided.

**Same workspace**
In a shared workspace tasks can be executed by the robot, as well as by the human shop floor worker. An autonomous motion, executed by the robot, is excluded, when the worker enters the workspace. The robot serves as a handling assistant, where a contact to the robot is wanted, exclusively with a non-moving robot.

**Shared workspace**
In a shared workspace, the area is used by the human and at the same time by the robot. Hence, the robot is in close proximity to the human and is operating while the human is located in the shared workspace. A contact between the robot and the human is not wanted, but an unwanted collision is highly probable. The designation for such application is a human-robot co-existance, where the risk analysis for human-robot collaboration, due to safety reasons, has to be examined. This describes the most common interest in collaborative robotics and is mainly treated in this thesis.

**Shared interactive workspace**
In a shared interactive workspace, robot and human work together in an application at the same time. Humans and robots are working very close and a frequent contact is desired, or allowed for. Every kind of dangerous operations have to be avoided.

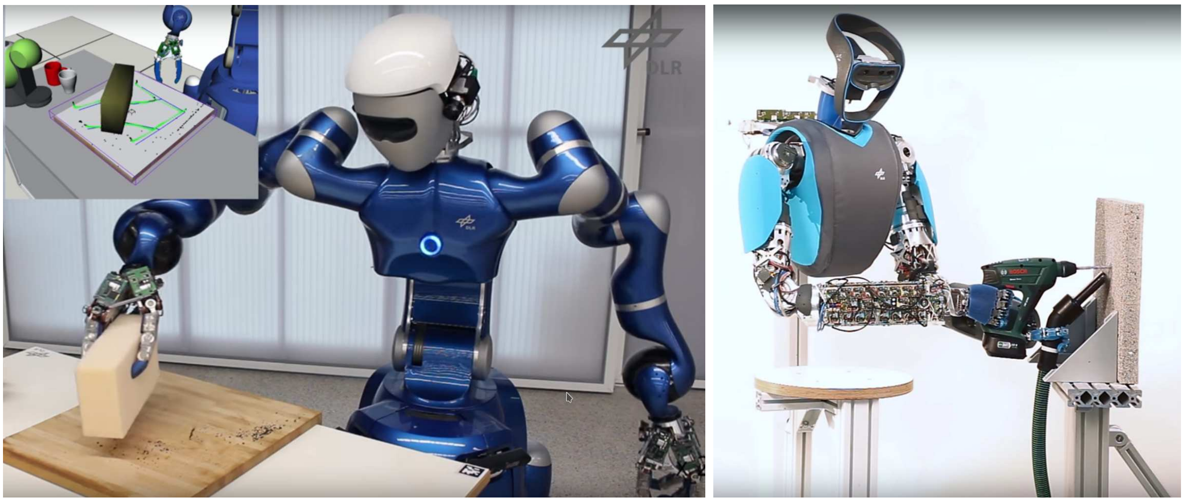
In this thesis the efficiency improvement of a shared workspace is in the focus, which also requires an optimal allocation of tasks that robot and human execute, and an ergonomic advantageous design of the collaborative workplace.

## 2.2 Robots for Industrial Use

In this section, robots are presented that are used in the industrial context. Endless new applications arise from robotics research. The major part in robotics is the industrial sector, where also many different types of robots are used. Latest challenges in humanoid robotics give an overview what is possible today, and where is a need of action for future research. In the following, a short overview is given about latest humanoid robots for the use of industrial applications. Then, common industrial robots are presented and disadvantages with respect to HRC described. Finally, collaborative robots are presented which fit to the content of this thesis.

### 2.2.1 Industrial Humanoid Robots

Humanoid robots are the long-term objective in robotics. Nowadays, this type of robot is exclusively used in scientific or research applications. The usage of humanoid robots in service robotics, elderly care, or also industrial robotics, provides a very high potential in flexibility in the automation process, as well as a high mobility, similar to a human. Up to now, the high complexity of such robots let the researchers face great technological challenges. First success was shown during the DARPA challenge, where humanoid robots had to complete various tasks, which also included sawing and drilling of holes [2]. Finally, this challenge showed that many problems and several hurdles exist for the successful deployment of humanoid robots.



(a) Rollin' Justin

(b) David

**Figure 2.2:** Humanoid robots

In Figure 4.4 two humanoid robots are depicted, which execute industrial like applications. The Rollin' Justin [3], illustrated in the left image, is cleaning a desk, and David [4], shown in the right image, is drilling a hole into a slab of stone. Generally, the complexity of humanoid robots exceeds the requirements for industrial applications. Therefore, in the following sections industrial robots are presented, which are suitable for industrial applications.

### 2.2.2 Industrial Heavy Robots

Industrial heavy robots have been used in industrial manufacturing for decades. Especially, in automotive industry the deployment of industrial heavy robots is dominated, but also the supplier, the metal and material producer, the food production, and the entertainment sector uses such robots.

(a) KR 120 R2500 ©RobotWorx (b) M 2000iA/2300 ©FANUC (c) IRB 6620 ©ABB

**Figure 2.3:** Selection of industrial heavy robots

In Figure 2.3 a KUKA robot, a FANUC robot, and an ABB robot are shown. The essential features of these robots are the high precision, which is less than a millimeter, and the high stiffness. They only provide position sensors, but due to the accuracy gears, the robot is not manually back-drivable. With their enormous strength, these kind of robots can move objects up to a ton of weight [5]. The application in collaborative work cells needs a special attention, as the strength and position control can lead to severe injuries to a human. Heavy industrial robots can be used in hand-guided applications, or in collaborative applications with special sensor equipment.

For a very close collaboration between humans and robots, these kind of robots are not suitable. Therefore, novel robot types were developed, to enable safe human-robot collaboration. In the following section, collaborative robots are described that are currently available on the robot market.

## 2.2.3 Collaborative Robots

This section gives an an overview about light-weight robots, and robots suitable for the use in fence less applications. First, an overview is given about commercial collaborative industrial robots on the market. Then, the DLR Light-Weight Robot LWR III is described in more detail, as this robot was used for the experiments in this thesis.

### 2.2.3.1 Industrial Light-Weight Robots

Industrial light-weight robots are a very recently developed technology. Every year, new types of robots appear on the market and every major robot company develops novel collaborative robots. Here, a short summary of collaborative robots is given.

In Figure 2.4 a selection of collaborative robots is illustrated. Basically, they have one property in common: they are small and light weight. This is also due to the standards, defined in [6, 7, 8], where maximum power, pressure, and forces are important factors. The robots Yumy and Kinova, shown in Figure 2.4 (d) & (f), comply with these standards by providing low power robots. It is of high importance, to ensure that no sharp or dangerous objects are grasped and moved. Universal robot provides a collision measurement, by observing external torques. This

(a) Universal Robot UR 5 ©Universal Robot

(b) Fanuc CR-35i A ©Fanuc

(c) KUKA iiwa

(d) ABB Yumi ©ABB

(e) Yaskawa HC10 ©Yaskawa

(f) Kinova Jaco ©Kinova

**Figure 2.4:** A selection of collaborative industrial robots

is done via motor current measurement and a state observer algorithm, which detects irregularities. This idea is also part of the KUKA iiwa and the Yaskawa Motomann HC10, but they provide force-torque sensing in the joints, where directly external torques can be measured. This enables a more precise measurement of external forces and, therefore, provides a much better collision detection with regards to sensitivity and required time to recognize a collision. Bosch developed the APAS robot with sensor fields for safety for collaborative robots. Their idea is to use capacitive sensors as a skin surrounding the robot structure. The function to stop the robot before a collision occurs, which is triggered by a disruption of the capactive field, eliminates the possibilities of injuries, caused by collisions with robots. Additionally, the sensor skin is designed as a foam skin. This also prevents from injuries, when collisions with a non-moving robot occur.

To summarize, many collaborative robots are available on the market, complying with the ISO standards, which enable fence-less robot applications. One of the first collaborative robots was developed at the German Aerospace Center (DLR) and is presented in the next section in more detail.

### 2.2.3.2 DLR Light-Weight Robot III

The DLR Light-Weight Robot (LWR), see Figure 2.5, was initially developed for the use in space. Therefore, the main requirement within the development was the light-weight design. The total weight is 14 kg and the maximum payload is also 14 kg, which has been demonstrated with climbing up a ladder. It turned out that this design also is advantageous for terrestrial application, as the light weight behavior enables a close co-existence of robot and human [9]. The modular design also allows for the use in many different robotic sectors.

**Figure 2.5:** The DLR Light-Weigth Robot III

The human friendly design, which means no sharp edges, and a joint axis limitation, such that no clamping can occur, is the first important characteristic to enable human-robot collaboration. The robot kinematic is defined by seven joints, each equipped with a torque sensor. This enables an extended workspace with a redundancy similar to human arms. The key technology of this robot is the integrated joint torque sensing, which enables a completely new control design, and simplifies the interaction with humans. They are mounted on the link side, which enables a direct measurement of the effective forces. This characteristic is used to employ an external torque observer with high efficiency that enables collision detection mechanisms to stop the robot in case of a collision.

In the following section, the demand and possible usage of collaborative robots is analyzed.

## 2.3 Projects and Market Relevance

This thesis is influenced by two main projects. The European project SAPHARI (Safe and Autonomous Physical Human-Aware Robot Interaction) [10] was part of the human-centered robotics in the DLR. The focus of this project was to develop interaction interfaces, safe and optimal control strategies, perception and sensor based task monitoring, as well as use-cases and integration platforms [11]. A demonstrator platform with two robotic arms was developed, to prove the interaction strategies developed by DLR. A BRIO toy train scenario assembly task was the initial use-case, which has been extended to a sterilization application in a hospital scenario [12].

As the topic human-robot collaboration got very interesting for the industrial sector, the following project RACELab (Robot Application Certification Laboratory) was referring to industrial needs. The goal was to develop certifyable industrial applications by combining software and hardware certification. The idea of RACELab was to develop robot skills which are software modules in the sense of safe reusable robotic skills for intuitive interfaces. The main challenges in this project were to develop software for programming robots. The target customers were divided into three groups:

- robot and programming experts
- advanced robot programmers
- user without any experience.

The challenge is to bring robots to everyone, such that these robots can be reprogrammed by everyone. A more detailed overview to robotic skills and programming robots can be found in [13].

In the beginning of this work, it was necessary to find out, what the real needs in the industrial sector are, and, additionally, the future supplier trends are of interest. Therefore, in the beginning of this thesis, an overview of the market in the area of human-robot collaboration is given. The analysis is divided into two different parts, which gives an overview about real existing problems and needs in the area of manufacturing. These two analyses are:

1. On-site analysis of work spaces in medium sized companies
2. Market relevance in collaborative robotics - customer needs, supplier trends

In the following, the two analyses are described and analyzed.

### 2.3.1 On-Site Analysis

The idea of this analysis was to find out, how these manufacturing procedures are executed nowadays, what task is worthy of improvement, and how a robotic assistant could support in such tasks. In the on-site analysis two companies were investigated. The conditions for selecting appropriate companies are as follows:

- manufacturer of components, real hardware products
- geometric limitations of the products or handled components
- predominant manual assembly by humans

These conditions were chosen because the main target is the implementation of robots in industrial scenarios with the limitation that collaborative light-weight robots should be used. This, finally, also limits the weight and dimensions of the components to be handled or processed.

In the following, the analysis of two manufactures are presented and discussed for the ability of integrating collaborative robots. A predetermined choice of key facts of the task examines whether certain characteristics of the assembly steps prevail. Interviews with the workers substantiate the need of assistive robotics.

(a) Manual assembly workbench (b) Assembly task with ergonomic hurdles

**Figure 2.6:** Manufacturing area

#### Automotive Component Supplier

The first company is a component supplier in the automotive sector. The main assembly tasks are manually executed by humans, but also autonomous, or semi-autonomous workstations, support the workers. The assembly areas are designed as workbenches with component boxes, which surround the worker.

In Figure 2.6 (a) a workbench assembly area is shown. Here, up to 27 boxes are surrounding the worker. Special fixtures for the components enable an ergonomic comfortable assembly. Figure 2.6 (b) shows a grasp into a box, whereby picking can be quite difficult, which is due to the economization of the production area size. However, in total eleven assembly areas were analyzed with respect to their the working conditions and characteristics.

In Table 2.1 it is evaluated, whether heavy weights (> 0.5 kg) have to be transported, another human co-worker is assisting, or the worker is interacting with machines. In this manufacturing no dangerous tasks were executed or dangerous components handled. The temporary lot size of an individual product was quantified by for example 60 components until the version of the product changes. This also implies that a frequent change of the assembly process is part of manufacturing. The cycle time for one assembly process varied from 32 seconds up to 5 minutes. The potential of an improvement by the usage of collaborative robots is very high and is going to be discussed in the following section.

**Control Cabinet Manufacturer**

In a second company, the main focus is on manually built control cabinets for industrial machines. Here, four main workstations are studied and analyzed. From the interviews, it is conspicuous that the help of intelligent machines is explicitly wanted with respect to an improvement of working conditions. The most expressed wish of a worker is the assistance in showing how/where the current work step should be executed, i.e., where along a cable should be placed. In summary, the deployment of autonomous robots in control cabinet manufacturing provides to be rather difficult, because the sensitivity and agility of the human hand was repeatedly needed to enable the assembly routine. The deployment of collaborative robots assisting the worker promises an increase of the efficiency of the tasks.

| **Definition** | **Company 1 Matches / Total 11** | **Company 2 Matches / Total 4** |
|---|---|---|
| Heavy weights | 4 | 0 |
| Interaction between humans | 1 | 1 |
| Interaction with machines | 5 | 0 |
| Uncomfortable work steps | 2 | 0 |
| Uncomfortable posture | 2 | 3 |
| Dangerous applications/components | 0 | 0 |
| Long transportation | 1 | 1 |
| Much motion to execute the task | 1 | 0 |
| Number of Components, cycle time | 32s-5min/100 pcs/week | 2-31 pcs/week |
| Changing geometry of components | 8 | 3 |
| Neediness, support wanted | 3 | 2 |
| Documentation | 5 | 2 |

**Table 2.1:** Documentation of work station with the analysis of each to obtain the main

### 2.3.2 Potential of Improvement with HRC

The deployment of collaborative robots has the potential of remarkable improvement of the assembly process in the manufacturing. The manual scan of components can be performed by the robot, and simultaneously be used for documentation of the process steps, to ensure quality.

(a) Holding objects (b) Hand-over SLCs

**Figure 2.7:** Robot assistance in assembly tasks

In Figure 2.7 the assembly of the automotive components is implemented such that the advantages of the deployment of collaborative robots emerge. It is shown that the robot holds objects to enable that the human can use both hands for complex or tiny assembly steps. The right part is handed over at the right time, which ensures a defined assembly process. Special tools, designed for the fixture, can be omitted and several components can be manipulated by the robot. By grasping boxes and, therefore, handing over small components at an ergonomic suitable position, improves the working conditions. Furthermore, an exchange of storages could be executed by autonomous mobile robots.

Necessary steps to be considered are the performance and safety analysis of such systems, to ensure suitable cycle times in the assembly applications. These considerations are necessary, to achieve efficient and safe human-robot collaboration. Nevertheless, this thesis aims at the improvement of industrial production processes and the achievement of the requirements in future manufacturing. In the following, the sustainable use of collaborative robots is further discussed by analyzing the current market trends.

### 2.3.3 Market Trends and Needs

In addition to the on-site analysis, this section gives an overview over the needs of the industrial manufacturer with respect to robotics, as well as the requirements and key market trends. To get an overview about the supplier trends and customer needs, interviews with ten robot experts from the customer side and robot supplier side are evaluated and yields to the following conclusions.

Automation in production is continuously increasing, especially in the region of Germany. This is caused by numerous macro-level trends like:

- Demographic changes / aging society
- The need of higher degree of automation
- Decreasing life-cycle of products
- Increasing individualization and customization

In the near future, the baby boomer generation is getting into the age of retirement, which will result in a drastic decrease of workers in the manufacturing. The increasing age also requires a facilitation of the working conditions, to ensure a robust and increasing productivity.

From these analysis, the main research topics for the human-centered robotics are determined as:

1. Enable human-robot collaboration with an optimal shared work space
2. User interfaces for task-level programming / non-expert programming tool
3. Modular design of skills for re-usability
4. Ensure safe and efficient robot operations shared work spaces
5. Enable certification of hard and software regarding ISO

In this thesis, the main focus aims at point 1., 4., and 5., in order to *maximize performance while maintaining safety* in collaborative applications.

In the following section, safety considerations and recent research is presented that analyses the risk of injuries during collaborative robot applications.

## 2.4 Safety in Physical Human-Robot Interaction

To bring robots fence-less into the production, the ISO standards are a prerequisite for licensing robot applications. However, a liability is not excluded, when following all rules from the ISO. Every possible hazard has to be considered, beyond the determined regulations, to do everything feasible to avoid accident with injuries [1]. In this section, an overview is given that contains preliminary research to safety in human-robot collaboration. This research resulted in international standards[2], which are explained and constitute the main motivation for this work. Finally, an economic overview is given to show the relevance of the topic.

### 2.4.1 Safety - The Basis for Human-Robot Collaboration

Asimovs first law [14]:

> *“A robot may not injure a human being or, through inaction, allow a human being to come to harm”,*

is one of the three robot laws and contains the main statement: a human shall not be injured by a robot. Figuratively, the meaning is that the robot shall not “decide” to harm a human, but it also includes that passively the robot can harm a human, during a motion. Up to now, robots are mainly deployed in industrial applications, where repeatedly tasks are executed with high efficiency. But what if a human got hit by such robot?

In [15] undesired collision scenarios are classified as follows:

- unconstrained impact
- partially constrained impact
- constrained impact
- clamping in robot structure

---

[1] Prof. Dr. Thomas Wilrich, Rechtsanwalt und Fachanwalt fuer Verwaltungsrecht, Muenchen

[2] ISO -International Standards Organization

- secondary impact

This separation is an important distinction and can result in different injuries in case of a collision, for example at the same relative robot-human velocity. The human can be clamped between the robot and an object, or within the robot structure, being not able to relieve itself. The hit of a robot can, for example, also cause the loss of balance, and the collision with another object can cause severe injuries. All these possibilities have to be considered for collaborative robot applications. The impact at the head can also lead to additional improvement of hazard. To analyze the effect to the human head, the Head Injury Criterion (HIC) can be used to quantify the injury potential after a collision [16].

Several safety analysis, experiments, and crash tests were executed to study hazards of robots, where pioneering work was done by Yamada *et al.* [17]. They introduced a novel human pain tolerance criterion, to obtain an impact behavior that is safe for the human. With this criterion, the separation of humans and robots was repealed and many collision tests were executed. Haddadin *et al.* analyzed injuries caused by a collision with a robot for different types of robots [18, 19, 20, 21, 22, 23]. Furthermore, the robot design and control was evaluated for safe human-robot interaction for human-care robots [24]

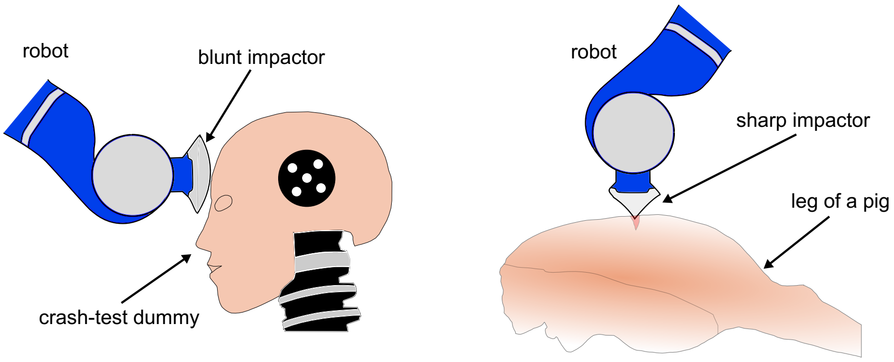


**Figure 2.8:** Collision Experiments

A schematic overview of experiments is illustrated in Figure 2.8. On the left side, collision experiments with the crash test dummy are shown, and on the right side impacts with a leg of a pig are analyzed. Special drop tests with different geometric objects are executed and the consequenses studied [25], [26]. The resulting information exhibits that the severity of an injury depends on the mass, geometry, and velocity of the colliding object.

With the obtained knowledge about injuries from collisions, and the resulting requirements [27], novel control methods and motion schemes were developed that adapt the velocity, depending on the potential risk [28], or being underneath desired thresholds [29], to prevent from pain in a collision with the robot. However, quantitative safety guaranties are given in [30] and new methods for intuitive behavior for safe human-robot collaborative applications are presented [31].

The analyses formed the basis for standards in human-robot collaboration, which are described in the following section.

### 2.4.2 Standards for Human-Robot Collaboration

To control, utilize, and define certain rules or guidelines for every kind of machines and procedures, a recognized organisation authors standards, which are based on scientific results, novel technology, and experiences to generate advantages for the society [32]. Especially in the protection of humans during the manufacturing process, such standards are from high importance and can (but not must) define a minimum requirement for a liability in case of an accident [33]. Most manufacturers follow these requirements and standards, to ensure suitable working conditions and a safe production environment.

For the deployment of machines, where robots are included, they are generally subject to the machinery directive 2006/42/EG, which defines a standardized level of protection for preventing from accidents with machines. In particular, for collaborative robots the DIN EN ISO 13489 defines the "safety-related design principles of employed control systems" [34]. With this standard, the risk of an application is determined and safety requirements are identified. Additionally, for collaborative robots the ISO 10218 is very important, which regulates the current industrial robot safety [6]. In the first part, the requirement for a robot in industrial environments is defined, where the demand on sensors and control strategies are defined. The second part ISO 10218–2 treats the robotic systems and their integration into the manufacturing processes. Especially for the human-robot collaboration, the deployment of a collaborative system is differentiated in

*5.10.2*: Safety-rated monitored stop - robot is not moving when human enters the shared workspace

*5.10.3*: Hand-guiding - robot can be moved by direct input including a dead man's switch

*5.10.4*: Speed and separation-monitoring - given a minimum distance and a maximum velocity, the robot is allowed to move

*5.10.5*: Power and force limiting by inherent design or control - collisions are limited by force and pressure given to the human body.

Additionally, the maximum velocity is limited to 0.25 m/s, the maximum affordable force to 150 N and the maximum dynamic power is bounded by 80 W, when collaborative robots are deployed [7].

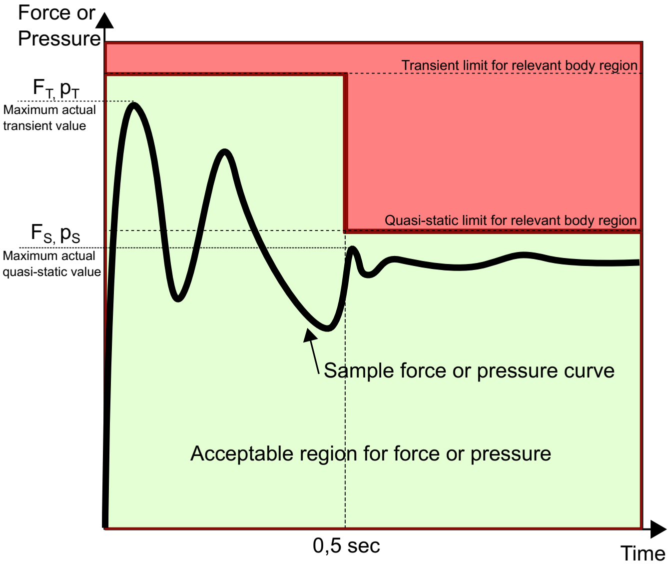


**Figure 2.9:** Force or pressure limits

Since human-robot collaboration became very popular, the international standards organization has discussed about new standards for human-robot collaboration, and defined novel rules in a technical specification ISO/TS 15066, which appeared in the year 2016. Here, acceptance of collaborative systems is defined by measurements of impact in case of collision.

Two types of collisions between robots and humans are distinguished:

1. transient contact and
2. quasi-static contact.

In Figure 2.9 the differentiation of both contacts is illustrated [8]. Furthermore, limits of pressure and force are defined given to a specific body region of the human. An overview is given in Table 2.2, where force and pressure limits for a transient or quasi–static impact are illustrated. These standards serve as basis for the following research in this thesis and are especially used for the contribution of human-in-the-loop control, hardware development, and evaluation for safe and efficient human-robot collaboration. In the following section, the state of the art in path planning and human prediction, as well as collision avoidance, is presented.

**Table 2.2:** ISO TS 15066

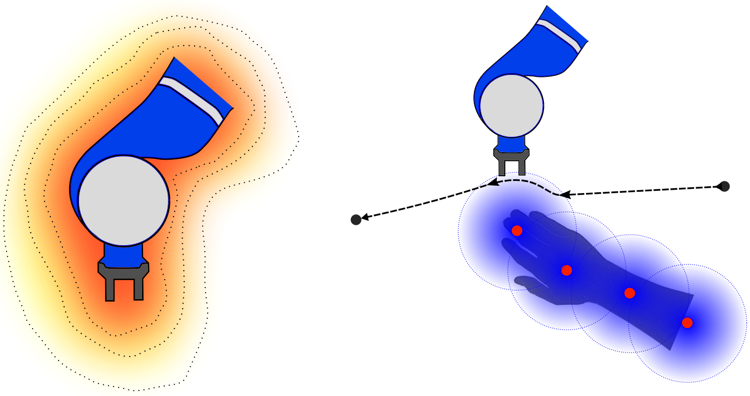
| | | Quasi-static contact | | Transient contact | |
|---|---|---|---|---|---|
| Body region | Specific body area | Maximum permissible pressure [a] $N/cm^2$ | Maximum permissible force [b] $N$ | Maximum permissible pressure multiplier [c] $p_T$ | Maximum permissible force multiplier [c] $F_T$ |
| Skull and forehead | Middle of forehead | 130 | 130 | not applicable | not applicable |
| | Temple | 110 | | not applicable | |
| Face | Masticatory muscle | 110 | 65 | not applicable | not applicable |
| Neck | Neck muscle | 140 | 150 | 2 | 2 |
| | Seventh neck muscle | 210 | | 2 | |
| ... | ... | ... | ... | ... | ... |

## 2.4.3 Path Planning, Collision Avoidance, and Motion Prediction

To prevent from collisions and, therefore, comply with the international standards, actuation mechanisms [35], [24], collision free path planning [36, 37, 38], collision avoidance methods, control side danger classification, and human motion estimation algorithms have been developed. Novel technology for detection and measurement of the human position enables the development of methods for collision avoidance. In [39] time-of-flight sensors are used to determine distance feedback and avoid collisions with obstacles. In the two dimensional space, multiple images can form an area measurement to perform collision free motions [40]. Another vision based sensor technology measures a point cloud at high frequencies to prevent from possible collisions [41, 42]. To prevent from shadows in the measurement area and, therefore, losses of signal an optimal placement of depth/presence sensor is analyzed [43].

In the depth space, information of contact points or collisions can be determined, and collision avoidance methods applied [44, 45]. Of high importance are is the collision avoidance and path adaptation in real-time [46, 47, 48, 49, 50, 51]. For the realization of collision avoidance and dynamic path generation, potential fields can be used to adapt the robot motion [52, 53, 54, 55]. An exemplary illustration is shown in Figure 2.10 (b). In Figure 2.10 (a) danger fields are illustrated, which define a certain danger value depending on the configuration and generate a danger assessment for articulated robots [56]. Furthermore, the human reachablity can be considered to define a possible danger [57].

A novel extension to real-time collision avoidance is the prediction, or motion estimation, of humans [58, 59, 60]. A major contribution to human motion estimation is done by the mobile robotics research community. As humans repeatedly pass or cross the robot path a collision between robot and human has to be avoided. Therefore, novel methods include the prediction of human motions, or the future path, respectively. In [61] human existence-probability is



(a) Danger Zones (b) Collision avoidance

**Figure 2.10:** Control side safety approaches

calculated for every location or, alternatively, a motion prediction framework presented in [62] is able to predict human motions in urban or social environments. Another approach is based on early prediction of human motions, where the human workspace occupancy is predicted using offline learned human trajectories. A trajectory generator generates motions using a minimized penetration cost, where multiple calculations are executed in parallel and the least cost is chosen to generate the final motion [63]. An extension of human motion prediction can be found in [64]. The environment of a human affects the final motion and the assumption is made that the human will not change the motion direction and speed significantly. These assumptions can only be made for mobile robots, where the entire human is treated as one black box of motion. For human-robot collaboration in the industrial assembly context, this assumption cannot be made as motions of the human, e.g. human arm motions, in a small operation area empirically are very fast.

From another point of view, the human intention is of interest, and can give information about the future action of a human [65]. In [66] they bemoan that conventional human motion prediction algorithms assume constant velocity and acceleration based trajectories. They remark that human motions depend on the behavior or the intention of the human, which is much more complex. Their assertion is that future human motions are influenced by various factors like the desired goal, the current environment, and social factors. In [67] they anticipate the human intention to make a decision about the next step of the robot. Further applications including human motion predicition, which enable a more efficient and transparent human-robot collaboration, can be found in [68].

In the following section, a novel workbench for efficient human-robot collaboration is presented.

## 2.5 The Collaborative Workcell - A Collaborative Workspace Design

In this section, preliminary considerations to shared work spaces will be presented. Human-robot collaboration is a very recent research topic, and only a few collaborative work cells are existing in industrial applications at present. Some robots were integrated in existing work places, like for example the assembly of automotive gears, or only concept studies were realized. Therefore, there is poor experience existing in collaborative robotic work cells. The integration of collaborative robots into existing work places often can lead to non-optimal usage of the robots. This affects the performance of the robot, which includes the usable work space, the flexibility for the case of a reconfiguration of the workplace, and, therefore, the efficiency of the collaborative work cell is not entirely considered. Anyhow, in this work an analysis of collaborative work cells is the basis of the research lab. The main consideration is on the work space of the robot. In former work, as for example in the EU project SAPHARI, the work space of the LBR III limited the integration into existing work cells enormously. The kinematic constraints led to a *proof of concept integration* of scenarios. This means that storages are kept small and they were mounted very closely to the robot. The content of the storages was a small number of objects. These problems are the basis of the following considerations and analysis for the collaborative work cell.

The development of workspace design is based on the on-site analysis and experience from existing applications. This is extended to a development of a shared work space for human-robot collaboration in assembly tasks. Finally, the human centered lab is presented including hardware setup, software, and control scheme as a basis for the following work on safe and efficient human-robot collaboration.

## 2.5.1 Shared Workspaces Design

A continuously recurring issue is the installation of the robot and the corresponding placement in the workcell. Additionally, the alignment of objects and storages has to be considered in order to enable robot applications. An essential property of such type of robots is a small work space, which results in the necessity of a workplace analysis and deployment optimization. In the on-site analysis, the main working positions were identified directly in front of the human, on a standardized work bench with a height of 90 cm. The assembly of components is executed on the workbench's surface. The movement space is limited caused by storages placed around the human worker. In Figure 2.11, upper image, the work space of human worker in a plane is illustrated by a blue circle, and simultaneously the work space of the robot for manipulating objects on a plane is illustrated by a red circle. The shared work space, regarding a non-moving human, is marked by the yellow area. As can be seen, the work space is significantly limited and leads to a non-optimal cooperation of human and robot.

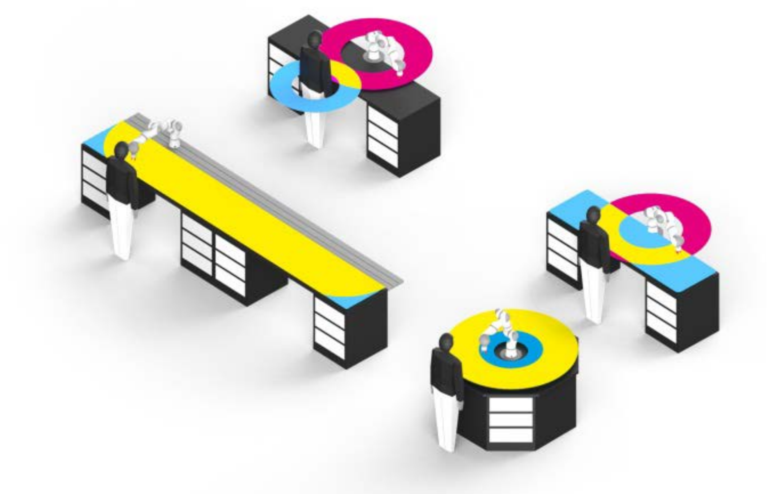


**Figure 2.11:** Study of shared work spaces by Thilo Wüsthoff and Roman Weitschat

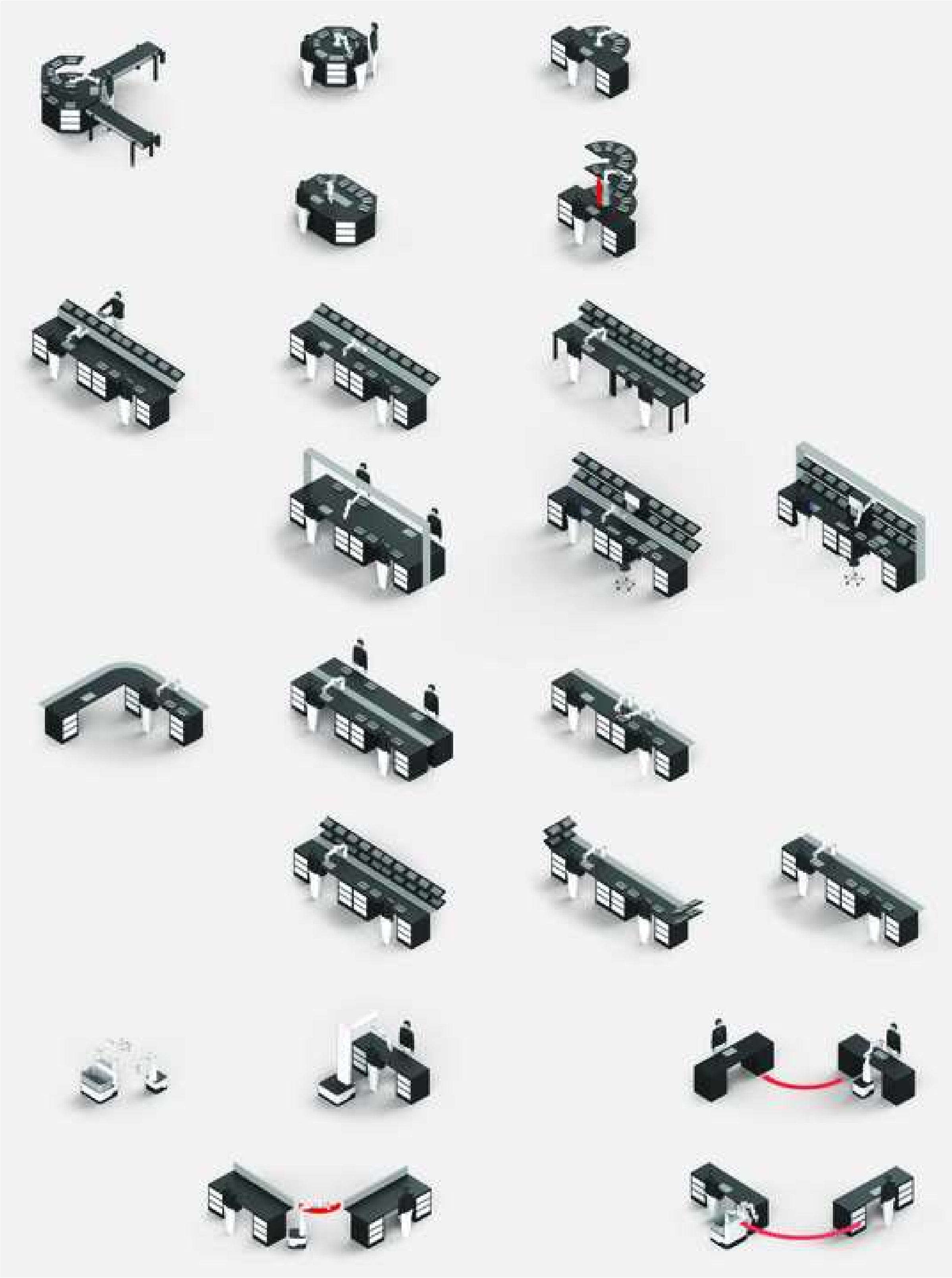


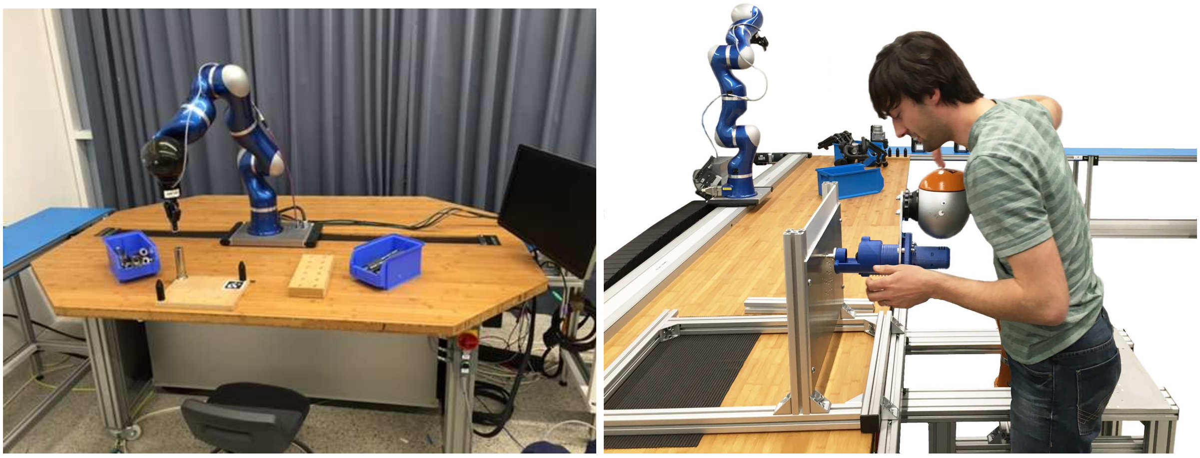
**Figure 2.12:** Study of shared work spaces by Thilo Wüsthoff and Roman Weitschat

Usually, the reachability of a human exceeds the circle-area due to the mobility and agility of the human body. Anyhow, this does not extend the shared workspace remarkably, as shown in Figure 2.11 right image - blue area. The goal is to find a workbench-design with maximum collaborative workspace including every position of the human worker. A variation of the concepts for collaborative workplace design is shown in Figure 2.12.

This underlying study served as basis for the development of the DLR Collaborative Workcell, which is presented in the following section.

### 2.5.2 Human-Centered Robot Systems - The Collaborative Work Cell

Finally, the collaborative research laboratory was extracted from these considerations about optimal structure or configuration in a human-robot collaborative scenario. A laboratory was developed, which allows for extended research in the field of human-robot collaboration. The resulting research lab - called *Collaborative Workcell* - is inspired by industrial processes, to ease the integration of scientific results into real industrial manufacturing processes. Additionally, a result of the the on-site analysis is the integration of flow production concepts in the manufacturing process. Therefore, this concept was endorsed and integrated into the research lab.


(a) RACELab Octagon

(b) RACELab Longtable

**Figure 2.13:** RACELab work benches. Right: the 4 m Longtable work bench for the assistance of two worker. Left: Octagon work bench with round shape for better work space usage and optimal cooperation.

The laboratory consists of two main components. First, a long assistive workbench - *Longtable* - as shown in Figure 2.13 (b), which has a length of $4$ m and a depth of $2$ m and is equivalent to a real industrial workbench. On the workbench, a LWR III is integrated, which is described in Section 2.2.3.2. The robotic system was extended with a linear axis, for optimizing the shared work space, as described in Section 2.5.1. The linear axis has a stroke of total $3.4$ m and a maximum velocity of $1$ m/s, whereas the axis is driven by a toothed belt, which simultaneously covers the gap to prevent from clamping of human body parts, as well as damage caused by small objects. The limits are secured by a magnet strip that stops the entire system in case of violation of these limits. Link-side and motor-side position sensors ensure a redundant measurement of the velocity, which is a first requirement for a collaborative system. To prevent from severe injuries, caused by the linear axis, contact sensors are integrated in the direction of the motion to stop the robotic system in case of an emergency.

The second component is shown in Figure 2.13 (a), where the design is changed to the second identified approach. Here, the reachability is optimized to a surrounding area. The robot is placed in the middle of the table, which can be used for two different cases. First, the robot can interact with multiple humans standing around the table and second, the robot has non-collaborative zones for dangerous applications. The internal communication to peripheral is realized by Beckhoff EtherCat solutions, which also enables to command the attached two-finger gripper from Robotiq. This workbench serves as hardware-platform used for the following research.

In the following section, the integrated software is summarized to give an overview about the internal control and programming of the created system.

## 2.5.3 System, Control and Programming

This section gives an overview about the software integrated or developed for the Human-Centered Robot Systems and the Collaborative Workcell at DLR. The robot control is is separated into two parts, the robot control and the programming part of the robot. Additionally, a system overview is given for the used or implemented software structure, as these are not commercially available.

### Low-level control

The robot low-level control is a development by DLR based on a RT Linux operating system (OS). This low-level control is a modular control scheme with the aim to change control modules depending on the current problem definition for the development of innovative methods, control strategies, safety algorithms, path planning and fast sensor data analysis. Robotic applications and new robot skills can be developed in a very flexible architecture. The Modular Robot Control (MRC), as shown in the RT area of Figure 2.14, is an entirely new development of robot control with individual processes compiled in Simulink and predefined interfaces for the communication between each process.

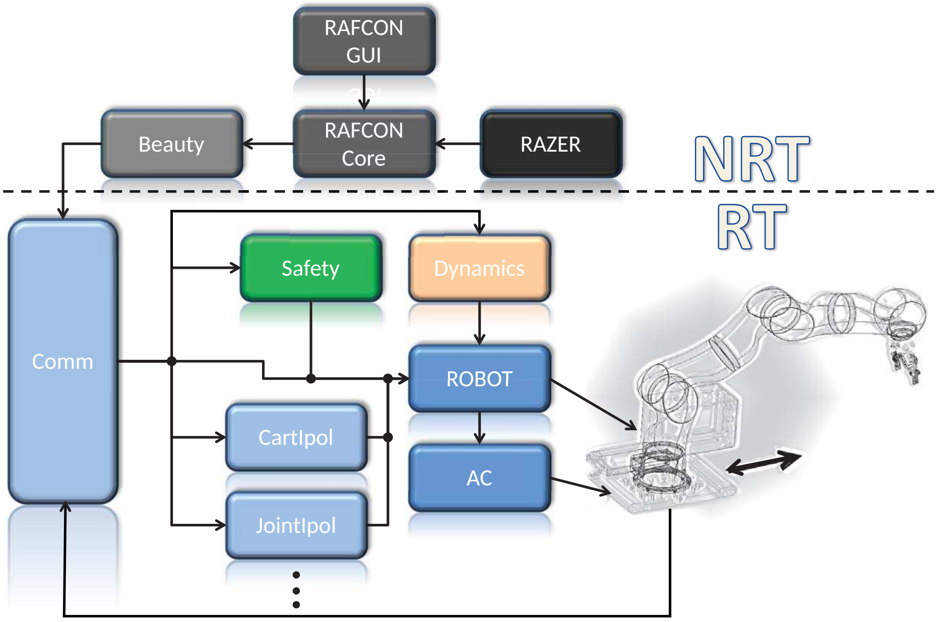


**Figure 2.14:** Control design developed for the human-centered robotics lab

In Figure 2.14 a schematic overview is illustrated. As mentioned above, the system is split in two parts, the upper is the non-real-time part, where commands can be executed on every kind of operating system, and at the bottom, there is the real-time part. The real-time processes are executed on a real time operating system, as for example vxworks, qnx, or rt-linux os. The latter is used for this thesis with a total rt-frequency of $1000$ Hz.

The communication between each process is realized by links and nodes, which is a DLR internal development. It provides a process management system and a real-time capable communication middleware. The low-level control is connected to the robotkernel, which enables to configure, manage, and interconnect with the driver modules of the robot.

For the collision detection used in this thesis, the momentum observer is of importance. Therefore, an overview is given, which clarifies the functional principle in the following. The overall dynamic system of the robot is described by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau_{ext}} \tag{2.1}$$

where $\mathbf{M} \in \mathbb{R}^{8\times 8}$ denotes the mass matrix, $\mathbf{C} \in \mathbb{R}^8$ is the Coriolis and centrifugal vector, $\mathbf{g} \in \mathbb{R}^8$ is the gravity vector and $\boldsymbol{\tau} \in \mathbb{R}^8$ is the control input. From the impedance controller, the control input is given by $\boldsymbol{\tau} = \left[\boldsymbol{\tau_{q_{1-7}}}\ f_{ac}\right]^T$, which determines the torques for the extended moving base. Since, the linear axis can only be controlled by desired position values, an at least two times differentiable trajectory has to be defined. Therefore, an admittance controller design is used for the linear axis by

$$M_{ac}\ddot{x} + D_{ac}\dot{x} = f_{ac} \tag{2.2}$$

where $M_{ac}$ denotes the virtual mass of the admittance controller and $D_{ac}$ is the damping term. To always ensure safety and being able to detect collisions, the robot control includes a momentum observer, as proposed in [69], which can be calculated by

$$\dot{\mathbf{p}} = \boldsymbol{\tau} - \boldsymbol{\beta}(\mathbf{q}, \dot{\mathbf{q}}) - \boldsymbol{\tau}_{ext}, \tag{2.3}$$

$$\dot{\hat{\mathbf{p}}} = \boldsymbol{\tau} - \boldsymbol{\beta}(\mathbf{q}, \dot{\mathbf{q}}) - \hat{\boldsymbol{r}}, \tag{2.4}$$

where

$$\boldsymbol{\beta}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) - \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} \tag{2.5}$$

includes the dynamics from Equation (2.1). The resulting observed external torques can finally be determined by

$$\hat{\mathbf{r}} = K_O \int_0^T (\dot{\hat{\mathbf{p}}} - \dot{\mathbf{p}}) dt, \tag{2.6}$$

where $\hat{\mathbf{r}} := \boldsymbol{\tau}_{ext}$. A more detailed overview on the observed generalized momentum can be found in [69]. The momentum observer is used to detect collision within the experiments, executed with a crash-test dummy presented in Chapter 7.

### Programming

To communicate with the low-level control, a middleware called *Beauty* is used, which translates the programming commands into controller values. With the standard programming language python, the robot can be programmed. For simplification, the RAFCON software is used [70], which is a state-machine-based programming software for experts and advanced robot user. This system allows for easy development of new robotic skills and provides a perfect research platform, as the system enables a fast adaptation of new methodologies. Using this software the generation of flexible robot skills [13] becomes possible, which can be easily parameterized in the non-expert user interface RAZER [71].

## Chapter 3

# Efficiency by Online Adaptable Robot Motions

In the field of robot motion planning, interactive motion generation becomes increasingly important. As an insight from the interviews and on-site analysis, presented in Chapter 2, shop floor workers expect to interact with robots similar to interacting with humans. This implies that humans want to gesticulate, talk to robots, or show the task to the robot, for parameterizing the entire robot application. From the robot side, visual and acoustic feedback is necessary. In the robotics community, this is known as *learning by demonstration*, which is a potentially intuitive method to program the robot or a robotic application. There are two typical approaches of learning by demonstration. For the first, the human operator moves the robot physically and records the desired motion and target position. In a second approach, human motions are recorded by a visual tracking system, where the position and velocity data is transformed into a robot trajectory offline.

In this chapter, a real-time method for trajectory generation is presented, which is applicable for interactive programming of robots, or demonstrating tasks in a safe way, without being in the danger zone. Therefore, the chapter gives an overview about path planning methods with a primary consideration of orientations. Then, the idea of mirroring human arm motions is presented, focusing on the realization of online trajectory generation, with a limitation of velocities. Finally, this method is transferred to a real-time trajectory generation for via-point motions, which are used to enable safe online motion generation, using online changeable via-points.

## 3.1 Orientations for Trajectories

In this section, a trajectory generation method is presented that enables online motion generation for orientations of the robot tool. First, a short overview is given for orientation representations, in particular quaternions. Then, the spherical linear interpolation (SLERP) is presented, which enables human arm mirroring for intuitive robot programming.

### 3.1.1 Quaternions

In general, a single rotation can be described as a circular movement around a defined axis [72]. For a rotation around multiple axes, the right hand rule is a well known method to define a Cartesian coordinate system that describes a rotation in the SO3 space around a specific axis by a desired angle. For the mathematical description a rotation matrix in $\mathbb{R}^{3\times3}$ is defined as

$$\mathbf{R}(\boldsymbol{\psi}) = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha s\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha c\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}, \tag{3.1}$$

where the rotation around the x-axis is given by $\alpha$, y-axis by $\beta$ and z-axis by $\gamma$. The $s$ and $c$ are the abbreviations of Sine and Cosine. The rotation matrix, as defined above, is a required input for the Cartesian impedance controller, as used for the *Collaborative Workcell*, to move the robot to a desired orientation. To finally calculate the orientation of a robot, different methods have been developed. Regarding the general rotation matrix, the role-pitch-yaw (RPY) formalism, as shown in Equation (3.1), enables a clearly defined description of an orientation and the generation of a path. Furthermore, Euler angles or angle-axis descriptions can be used, which is a formulation of a rotation in a straight forward manner.

A less intuitive description of orientations are *quaternions*, which are closely related to the angle-axis formalism. Quaternions are a popular choice for representing orientations, due to the reason that quaternions enable easy composition of rotations and coordinate transformation [73]. Additionally, a main characteristic of quaternions is that they are continuous in contrast to, for example, Euler angles, where a continuity of the path cannot be guaranteed. The quaternion representation can be calculated by transformation from the rotation matrix with

$$\begin{aligned} \eta &= \frac{1}{2}\sqrt{R_{11} + R_{22} + R_{33} + 1}, \\ q_1 &= \frac{1}{2}\text{sign}(R_{32} - R_{23})\sqrt{R_{11} - R_{22} - R_{33} + 1}, \\ q_2 &= \frac{1}{2}\text{sign}(R_{13} - R_{31})\sqrt{-R_{11} + R_{22} - R_{33} + 1}, \\ q_3 &= \frac{1}{2}\text{sign}(R_{21} - R_{12})\sqrt{-R_{11} - R_{22} + R_{33} + 1}. \end{aligned} \tag{3.2}$$

The inverse transformation from quaternion into a Rotation matrix is given by

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} 2(\eta^2 + q_1^2) - 1 & 2(q_1q_2 - \eta q_3) & 2(q_1q_3 + \eta q_2 \\ 2(q_1q_2 + \eta q_3) & 2(\eta^2 + q_2^2) - 1 & 2(q_2q_3 + \eta q_1) \\ 2(q_1q_3 - \eta q_2 & 2(q_2q_3 + \eta q_1) & 2(\eta^2 + q_3^2) - 1 \end{bmatrix}. \tag{3.3}$$

The Rotation matrix serves as input to the robot controller, and, therefore, a forth and back-transformation is necessary. A more detailed description of complex numbers and quaternions can be found in [74].

In the following section, quaternions are used to generate orientations online to control a robot and mirror the human arm motions. An appropriate algorithm, called SLERP, realizes minimum-time motions and a continuous path.

### 3.1.2 SLERP

The SLERP algorithm was initially introduced by Shoemake [75]. The formalism describes a linear path given the initial orientation and the desired goal position. Usually, by generating a path, one is interested in the shortest way of the path because the result is a time minimal motion for the robot. Having a look at the Euclidean space, obviously the shortest path is given by a straight line. In the non-Euclidean spaces like the SO(3) orientation space, a straight line cannot be found intuitively. With the work of Shoemake, a straight line in the orientation space can be represented using the quaternion based linear interpolation. As mentioned above, the SLERP formalism inputs are defined by an initial orientation defined by $\mathbf{q}_0$ and a desired goal orientation $\mathbf{q}_g$. The linear interpolation between these two orientation can be calculated as

$$\text{Slerp}(\mathbf{q}_0, \mathbf{q}_g, u) = \mathbf{q}_0(\mathbf{q}_0^{-1}\mathbf{q}_g)^u, \tag{3.4}$$

where the interpolation parameter is given by $0 \leq u \leq 1$. This value can be described as a time parameter for the duration of the interpolation. A more detailed calculation of this parameter is explained in the following.

In quaternion space, two possible descriptions of the same orientation exist. For example, the goal orientation $\mathbf{q}_g$ can also be described by $-\mathbf{q}_g$. This means, there are two solutions which describe a counter-clockwise or a clockwise rotation. The calculation of the shortest rotation is depending on the full rotation from the initial quaternion to the goal quaternion, i.e., $\mathbf{q}_0 \cdot \mathbf{q}_g$. The shortest path can finally be calculated by using the SLERP interpolation given by

$$\mathbf{q}_g = \begin{cases} \mathbf{q}_g & , \ \mathbf{q}_0^T\mathbf{q}_g \geq 0 \\ -\mathbf{q}_g & , \ \mathbf{q}_0^T\mathbf{q}_g < 0 \end{cases}, \tag{3.5}$$

to obtain a continuous behavior. Equation (3.4) can be reformulated in order to generate a desired rotation by

$$\tilde{\mathbf{q}}_\mathbf{d}(t) = \frac{sin[(1-u(t))\Theta]}{sin(\Theta)}\mathbf{q}_0 + \frac{sin(u(t)\Theta)}{sin(\Theta)}\mathbf{q}_g, \tag{3.6}$$

which describes the analytic time-dependent path for the orientation of the robot. The total amount of rotation is given by

$$\Theta = acos(\mathbf{q}_0 \cdot \mathbf{q}_g), \tag{3.7}$$

which describes the angular distance of the path.

In the following section, human arm motion mirroring is presented, where the SLERP algorithm allows for velocity-limited online trajectory generation.

## 3.2 Mirroring Human Arm Motion

Mirroring human arm motions posses tremendous potential to decrease the set-up time for robot implementation. Initially, the mirroring is from the field of telepresence, which considers the motion generation on certain distances. The main problem in control theory relies in the delay over a far distance. For terrestrial industrial applications the delay is inconsiderably small and can be ignored. Regarding the application analysis in Chapter 2, three main applications can be derived:

1. Online imitation of the human, in particular grasping into storages
2. Assembly of objects, while numerous robots imitate the motion and assemble the same object
3. Easy programming by non-physical hand guidance

Mirroring the human arm motions is motivated by the last point - safe and efficient easy-to-use programming inputs.

In order to cope with the safety requirements of ISO/TS-15066, the trajectory has to be limited in the velocity. In Figure 3.1 a schematic overview of a pick and place skill programming procedure is illustrated. Usually, the skill is taught by moving the robot in gravity free mode. Robots, which do not have the ability of a gravity-free torque-control, cannot be taught by moving the robot. Alternatively, robot skills can be parameterized by mirroring the human arm motion. In this context, the robot can be moved to the pre-pick-pose, pick-pose,

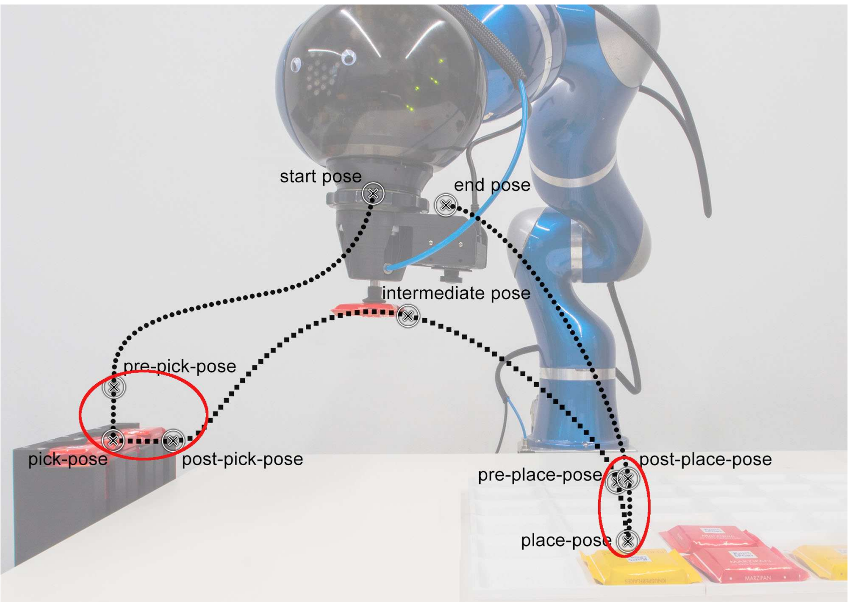


**Figure 3.1:** Robot skill taught by hand guiding of the human worker. The Pick & Place skill is one of the most common robot application. The most important parameters are the pick approaching direction and post motions, the obstacle-avoidance to place-position, approaching the place-position and post-place-position. For the handling of dangerous objects, or for intuitive and easy programming, novel methods for human arm mirroring are presented below. Image ©IEEE

post-pick-pose, intermediate points, pre-place-pose, place-pose, and post-place-pose, to fully define the pick and place skill. The mirroring human arm motion approach enables an intuitive programming of the robot.

In the following section, the human arm mirroring approach is presented.

### 3.2.1 Approach

Mirroring the human arm motions requires a robust and asymptotically stable trajectory generation, with the ability to limit the velocity of the robot. It is assumed that the human hand orientation and position is measured, for example, by a marker-based visual tracking system. The key feature of the approach is that the velocities are limited by the SLERP algorithm, and a smooth non-stick-slip trajectory is generated.

From the position measurements, the orientations are obtained in form of a rotation matrix, which are transformed into quaternion space. A frequency-estimation calculates the signal frequency input from the sensor. Using this information, a path is generated with the SLERP algorithm for a smooth motion without stick-slip effect. As the SLERP trajectory generation provides a non-differential motion, a quaternion filter design is developed that generates a two times differentiable smooth trajectory, which serves as input to the impedance controller of the robot. Figure 3.2 illustrates the schematic overview of the procedure.

The following section describes an overview of the functionality of the SLERP algorithm.

### 3.2.2 SLERP Linear Interpolation - Smooth Trajectories with Limited Velocities

The SLERP formulation describes an interpolation between two orientations, which is by definition a given path. Since this formulation describes a linear interpolation, it is expected that the resulting trajectory contains steps in the velocity. As it is well-known, steps in

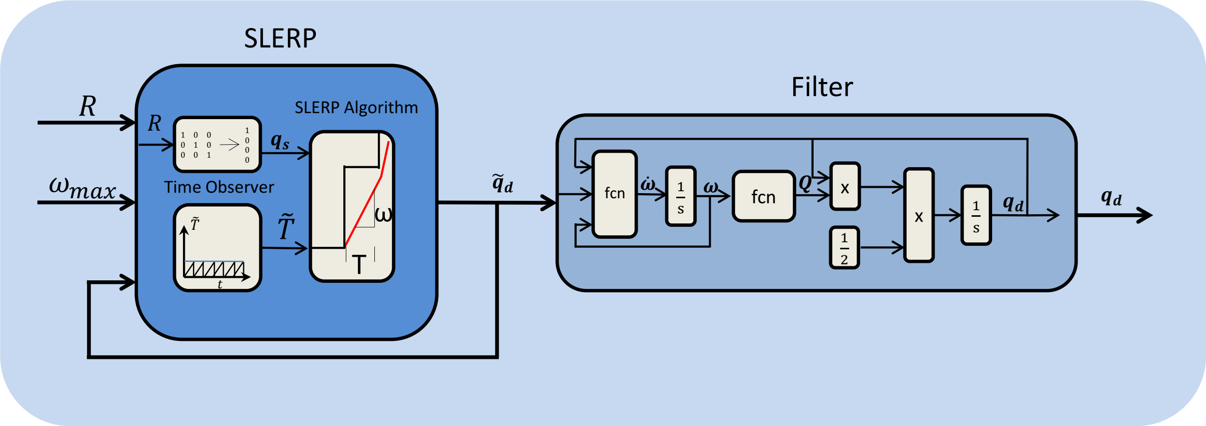


**Figure 3.2:** Approach - Combining two methods for smooth motion generation in quaternion space. A low-frequency sensor signal is used as input command for the robot. In combination with a method known from the dynamic movement primitives paradigm, a suitable orientation generation allows to smoothly and constrained move the robot in Cartesian space. Image ©IEEE

the velocity lead to impulses in the torques, which can damage the hardware. The goal of a smooth trajectory generation includes the smooth generation of velocity and acceleration-profiles. Therefore, the SLERP formulation has to be filtered, or smoothed, in order to obtain an applicable signal for the robot control.

As described above, the SLERP formulation includes an interpolation parameter $u$, which influences the progress of the trajectory. In order to limit the velocity of the robot, a time parameter $T_m$ is defined, which is dependent on the maximum rotational velocity [76]. The time parameter is described as follows:

$$T_m = \frac{\Theta}{\omega_{max}}, \tag{3.8}$$

where $\omega_{max}$ describes the maximum desired angular velocity, and $T_m$ describes the time over the full angle $\Theta$.

Anyhow, to prevent the robot from stick-slip motions caused by a low frequency sensor signal, a final interpolation time $T_q$ is defined with

$$T_q = max(T_m, T_s), \tag{3.9}$$

where $T_s$ denotes the sampling time of the source signal, as for example the tracking system. The maximum of both time parameters is chosen because the goal is to obtain a smooth motion of the robot. This implies, that reaching the desired goal position has to be avoided, until new signal data is obtained. This would lead to a slip stick behavior, which is not desirable for a robot motion.

The time constant of the source signal is given by

$$T_s = \frac{1}{\mathrm{f}_s}, \tag{3.10}$$

where $\mathrm{f}_s$ is the specified frequency of the used sensor.

A continuously determination of the frequency can often not be guaranteed, which is caused by the usage of low-cost solutions to enable human tracking, e.g., Microsoft Kinect Sensors. In combination with a non-real time operating system, a constant signal-frequency is often not obtained. In order to solve this issue, a frequency-estimation is implemented. The estimated frequency $\mathrm{f}_{obs} \approx \mathrm{f}_s$ can be obtained by

$$\mathrm{f}_{obs} = \frac{1}{\int_0^{\hat{t}} 1 \; dt + \epsilon}, \tag{3.11}$$

where $\hat{t}$ is the required time to a new signal input. The parameter $\epsilon$ denotes a small threshold, which enables the circumvention of a robot stop in case of small deviations.

Finally, the interpolation parameter $u$ can be formulated as

$$u(t) = \begin{cases} \zeta \Delta u & , \; \zeta < T_q \\ 1 & , \; \zeta \geq T_q \end{cases}, \tag{3.12}$$

with $\zeta$ as the time duration between the changes of the orientation, which can be formulated as $\zeta = t - t_l$. The parameter $t_l$ is the time on which the desired orientation changed for the last time and the parameter $t$ is the current time. The resulting interpolation input is $\delta u = \frac{1}{T_q}$. The initial orientation is $q_0$, which is updated on every change of the desired orientation, where

$$q_0 = \tilde{q}_d(t), \tag{3.13}$$

and the interpolation parameter $u$ is set to zero.

The result of the concatenation of linear interpolated straights is a discontinuity of the velocity in the trajectory. This unwanted behavior has to be smoothed. Therefore, in the next chapter, an algorithm is presented, which filters the original path information to obtain a trajectory suitable for the controller input of a robot.

### 3.2.3 SLERP Filter Design

To achieve a smooth robot behavior, a continuous input trajectory is required, at least up to the third derivative. This behavior can be achieved by implementing a filter to the SLERP algorithm.

The Filter design initially originates from the field of dynamic movement primitives (DMP), introduced in Section 6.2. DMPs usually use the behavior of second order differential equations. An advantage of a second order differential equation is the possibility of an asymptotically stable behavior. Usually, DMPs are used to generalize motions in joint or Cartesian space [77]. For the second order filter design, the quaternion formalism, as described in [78], is applied to rotational motions and can be expressed by

$$\tau \dot{\boldsymbol{\omega}} = K \mathbf{e}_d(\{\eta_d, \mathbf{q}_d^*\}\{\tilde{\eta}_d, \tilde{\mathbf{q}}_d^*\}) - D\boldsymbol{\omega}, \tag{3.14}$$

where $\boldsymbol{\omega}$ describes the angular velocity, and $\mathbf{e}_d$ is the error between the current orientation and the desired orientation.

The quaternion vector can be written as $\mathbf{q} = [\eta \; \mathbf{q}^*]^T$, where $\eta$ describes the scalar vector element in the quaternion formulation and the vector $\mathbf{q}^* = [q_i \; q_j \; q_k]^T$. The final error between the current orientation and the desired orientation can be written as

$$e_d(\{\eta_d, \mathbf{q}_d^*\}, \{\tilde{\eta}_d, \tilde{\mathbf{q}}_d^*\}) = \eta_d \tilde{\mathbf{q}}_d^* - \tilde{\eta}_d \mathbf{q}_d^* - \mathbf{q}_d^{*\times} \tilde{\mathbf{q}}_d^*, \tag{3.15}$$

where $\tilde{\mathbf{q}}_d^*$ is the current orientation and $\mathbf{q}_d^*$ is the desired orientation. The skew-symmetric matrix can be written as

$$\mathbf{q}_d^{*\times} = \begin{bmatrix} 0 & -q_k & q_j \\ q_k & 0 & -q_i \\ -q_j & q_i & 0 \end{bmatrix} . \tag{3.16}$$

To calculate the desired quaternion from the angular velocity, the derivation of time can be described as

$$\frac{d\mathbf{q}}{dt} = \begin{bmatrix} \dot{\eta} \\ \dot{\mathbf{q}}^* \end{bmatrix} = \frac{1}{2}\mathbf{Q}(\boldsymbol{\omega}) \begin{bmatrix} \eta \\ \mathbf{q}^* \end{bmatrix} , \tag{3.17}$$

where $\mathbf{Q}$ is obtained by

$$\mathbf{Q}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -\boldsymbol{\omega}^\times \end{bmatrix} . \tag{3.18}$$

From these equations, the direct input to the torque controller of the light-weight robot can be obtained. In collaborative robotics, an essential requirement is the limitation of velocities, which is possible with the presented method. In order to determine the needed parameter, a maximum velocity proof is calculated.

In general, the SLERP algorithm enables the limitation of velocities, but as the filter design is including a second order differential equation, an overshoot of the resulting trajectory has to be avoided.

To achieve this behavior, the single input formulation of the differential equation is used by

$$\tau\dot{\boldsymbol{\omega}} + D\boldsymbol{\omega} + K\mathbf{e}_d(\boldsymbol{\omega}, \boldsymbol{\omega}_d) = 0 \, . \tag{3.19}$$

The error between the current rotation and the desired rotation can also be written as the difference of the angular velocity and the desired angular velocity, which is described by a derivative of the differential equation by

$$\tau\ddot{\boldsymbol{\omega}} + D\dot{\boldsymbol{\omega}} + K\boldsymbol{\omega} - K\boldsymbol{\omega}_d = 0. \tag{3.20}$$

The transformation into the Laplace domain yields

$$\frac{\Omega(s)}{\Omega_d(s)} = \frac{K}{\tau s^2 + Ds + K} . \tag{3.21}$$

As it is well-known, a non-oscillating and asymptotically stable dynamic system is defined by negative and real poles, which can be obtained by

$$p_{1,2} = -\frac{D}{2\tau} \pm \sqrt{\frac{D^2}{4\tau^2} - \frac{K}{\tau}} \tag{3.22}$$

Using a constant stiffness parameter $K$, the damping term can be calculated as $D \geq \sqrt{4K\tau}$ to achieve a stable and non-oscillating system. Using this damping coefficient, it is guaranteed that the maximum angular velocity is never exceeded, which is given by $\omega \leq \omega_{d_{max}}$.

In the following section, the results of the simulation show the adherence and the suitable trajectory generation of this algorithm.

### 3.2.4 Cartesian Trajectory Generation

To obtain a 6-DoF end-effector motion, an online Cartesian trajectory generation is necessary. It is inspired by a common interpolation method between the current position and the goal position, similar to the SLERP algorithm. The desired filter input can be calculated by

$$\tilde{\mathbf{x}}_g(t) = \mathbf{x}_i(1 - u(t)) + \mathbf{x}_g u(t), \tag{3.23}$$

with $\mathbf{x}_i$ being the desired position during the motion, which is initialized by measuring the current position. The value $\mathbf{x}_g$ denotes the currently sensed position and $u(t)$ is the interpolation parameter. The time value to limit the velocity can be calculated by

$$T_x = \frac{\|\Delta\mathbf{x}\|}{\dot{x}_{max}}, \tag{3.24}$$

with $\Delta\mathbf{x}$ being the distance between the current desired position $\mathbf{x}_i$ and the new goal position $\mathbf{x}_g$. To obtain a maximum velocity, the method follows the same approach as for orientations, and the maximum time constant $T_{qx}$ has to be found, see Equation (3.9). To synchronize the overall motion, the total maximum time constant $T_Q = max(T_{qx}, T_q)$ has to be calculated, to reach the goal position in orientation and Cartesian motion at the same time.

### 3.2.5 Simulation

This section presents simulation results using Matlab/Simulink, to evaluate the effectiveness of the trajectory generation using SLERP with the novel filter design presented above. In a first example, the ineffictiveness of a standard filter design is shown. The Laplace domain for a common second-order low-pass filter is given by

$$G = K\frac{1}{T^2s^2 + DTs + 1} \tag{3.25}$$

with $K$ being the gain, $T$ the time constant and $D$ the damping term. The result of a critically damped filter design with a signal input, simulated as a step trajectory of 25 Hz frequency, is depicted in Figure 3.3. The step response of the resulting position signal shows the behavior of a critically damped system. The error between the signal input and the filtered value is

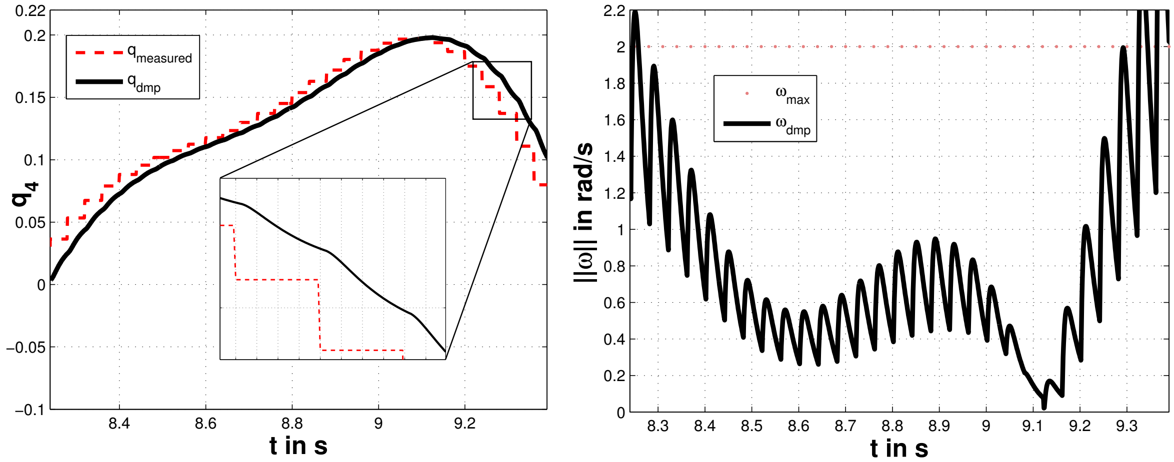


**Figure 3.3:** Direct signal input to the quaternion filter. (left) One component of the resulting quaternion is depicted. The red dashed line depicts the original sensor signal. The black solid line depicts the resulting trajectory. (right) The maximum desired velocity is depicted as dotted magenta line. The resulting velocity of the trajectory generation exceeds the maximum velocity and a strong oscillating behavior is apparent.

obviously small. In Figure 3.3, right side, the velocity output denotes an oscillating behavior. This is expected because the low sampling of the source signal has an infinitely high acceleration in each signal step. The filtered signal can be smoothed to some extent, but finally the output is not applicable to a robot controller, when using this approach. The torques applied to the robot would be very high, and the motion of the robot would show an unnatural and jerky behavior, which is not desired. Mirroring a human arm is very difficult because with the jerky behavior the robot seems to be uncontrollable in a precise task. Using higher values of the damping parameters, this oscillating behavior can be reduced, but in the end, the error between the desired and the resulting trajectory is increasing remarkably. Additionally, a lower frequency, for example 10 Hz, leads to an unnatural stick-slip behavior, which is undesired for a robot motion. Furthermore, the velocity of the resulting trajectory cannot be constrained, which is a requirement in human-robot collaboration.

For the simulation of the presented SLERP algorithm, an artificial trajectory input is generated, using defined increments of the desired positions. The frequency of the goal-position update is set to 25 Hz, to analyze and compare the behavior of the desired trajectory, which is applied to the robot. Within the experiments, a defined sensor frequency cannot be guaranteed, as, for example, using the VICON system on a non-real-time operating system leads to unstable frequencies. In Figure 3.4, the simulation results by using the SLERP algorithm output are illustrated. The left column depicts the four related quaternions. The red dashed line illustrates the desired goal position as input. Additionally, an important requirement for human-robot collaboration is the limitation of velocities. Therefore, the velocity of the desired trajectory is designed to be higher as the maximum desired velocity. The blue solid line depicts the resulting quaternion obtained by the SLERP algorithm. In the right column, the upper three plots of Figure 3.4 illustrate the resulting velocity output. The bottom right plot depicts the total angular velocity. As can be seen at the beginning and in the end, the velocity limits are not violated by the SLERP output. However, the trajectory has a step-wise change of the desired velocity, which is a non-suitable input to the robot controller. Therefore, a quaternion filter method is implemented to enable a smooth motion behavior.

In Figure 3.5, the same input is used equal to the first simulation. The left column includes the desired orientation, whereas the black solid line depicts the final filtered output of the quaternion. Here, the typical off-set of a filtered trajectory can be observed, which results from an acceleration limit and does not affect the desired final orientation. In the right column of Figure 3.5, the resulting velocities are depicted. As can be seen in the bottom right plot, the desired velocity limits are not violated. The behavior of the velocity is smooth and two times differentiable. Therefore, the SLERP algorithm, in combination with the presented filter design, enables velocity limitation and a smooth trajectory generation applicable to a robot system.

After proving the applicability, the next section presents the experimental setup and the results on a real robotic system.

### 3.2.6 Experimental Setup

In this section, the experiments are executed on a DLR light-weight robot LBR III, as shown in Figure 3.6 (6), to evaluate the SLERP filter combination on a real robotic system. A DLR-HIT hand is mounted to the robot tool-center-point (TCP) to enable real grasping applications. The goal of the experiments is that the robot mirrors exactly the human arm motion and activity online. In particular, the main task for the human experimenter was to grasp a ball from a stand (4) in a dynamic motion without slowing the movement down. The task of the robot is to execute exactly the same motion and perform the same actions in parallel (3). To trigger a grasp command, the surface electromyography technology is used. Sensors are placed on the forearm of the operator, where the electronic signals in the muscles are measured, and

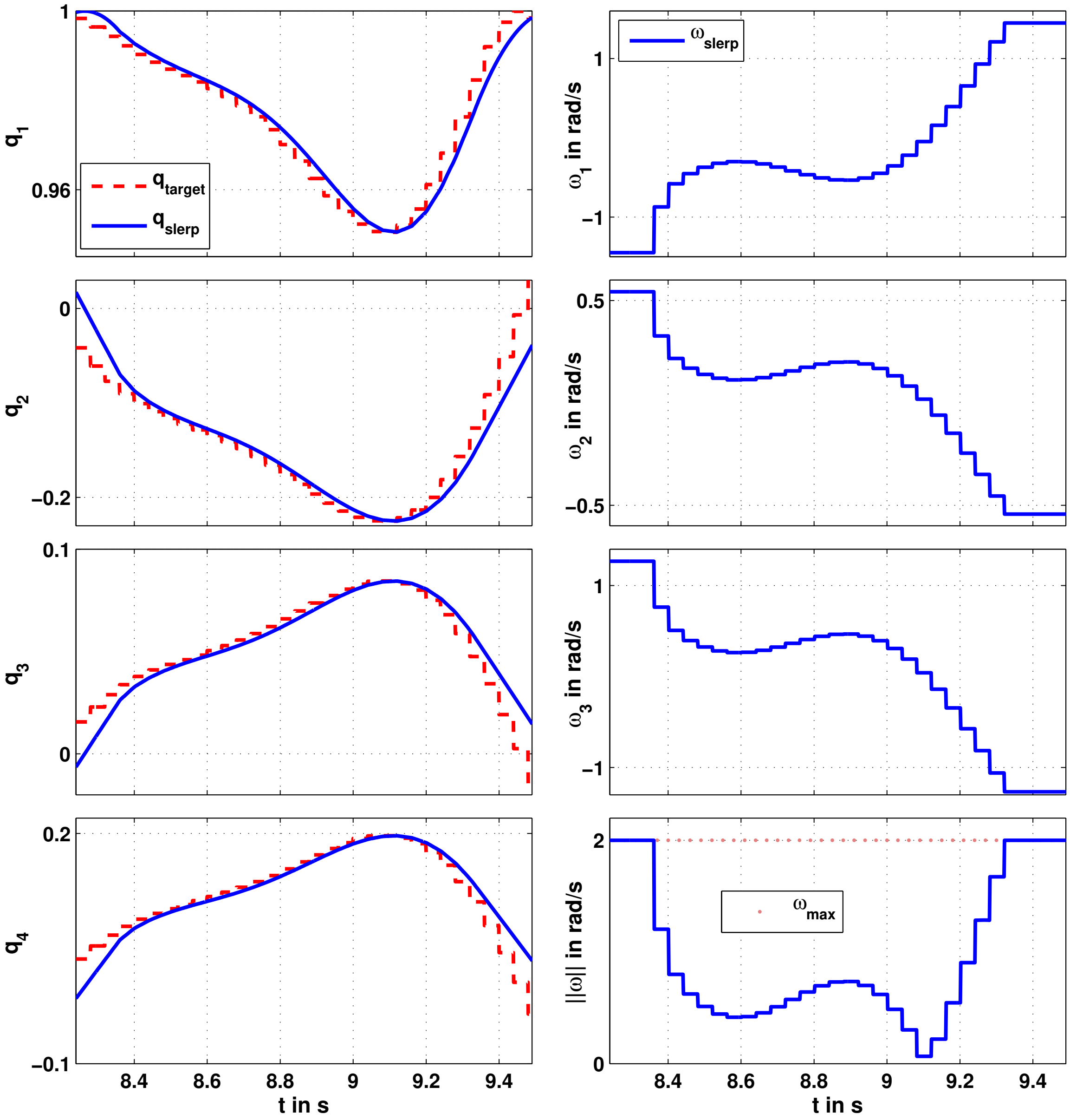


**Figure 3.4:** Left row: Comparison of artificially generated input and the generated trajectory using SLERP only. Right row: Angular velocities generated from the SLERP algorithm. The right bottom plot depicts the absolute angular velocity (blue) and the velocity limit of 2 $rad/s$ which is never exceeded. Image ©IEEE

the opening, or closing, of the hand can be detected (1). Thus, finally, the hand of the robot is commanded in order to close or open accordingly. For tracking the position of the hand, markers are placed at the back of the hand (2) and a VICON tracking sensor is located above the human (5).

The motion of the human experimenter should focus on the orientation, to obtain suitable results. Therefore, the initial orientation is at least 90 degrees different from the grasping orientation at the stand. Obviously, the human motion should not exceed the maximum velocity of the robot, but should be close to the limits. Two different experiments were executed; the first with a high sensor frequency of 100 Hz, and the second with a low sensor frequency of

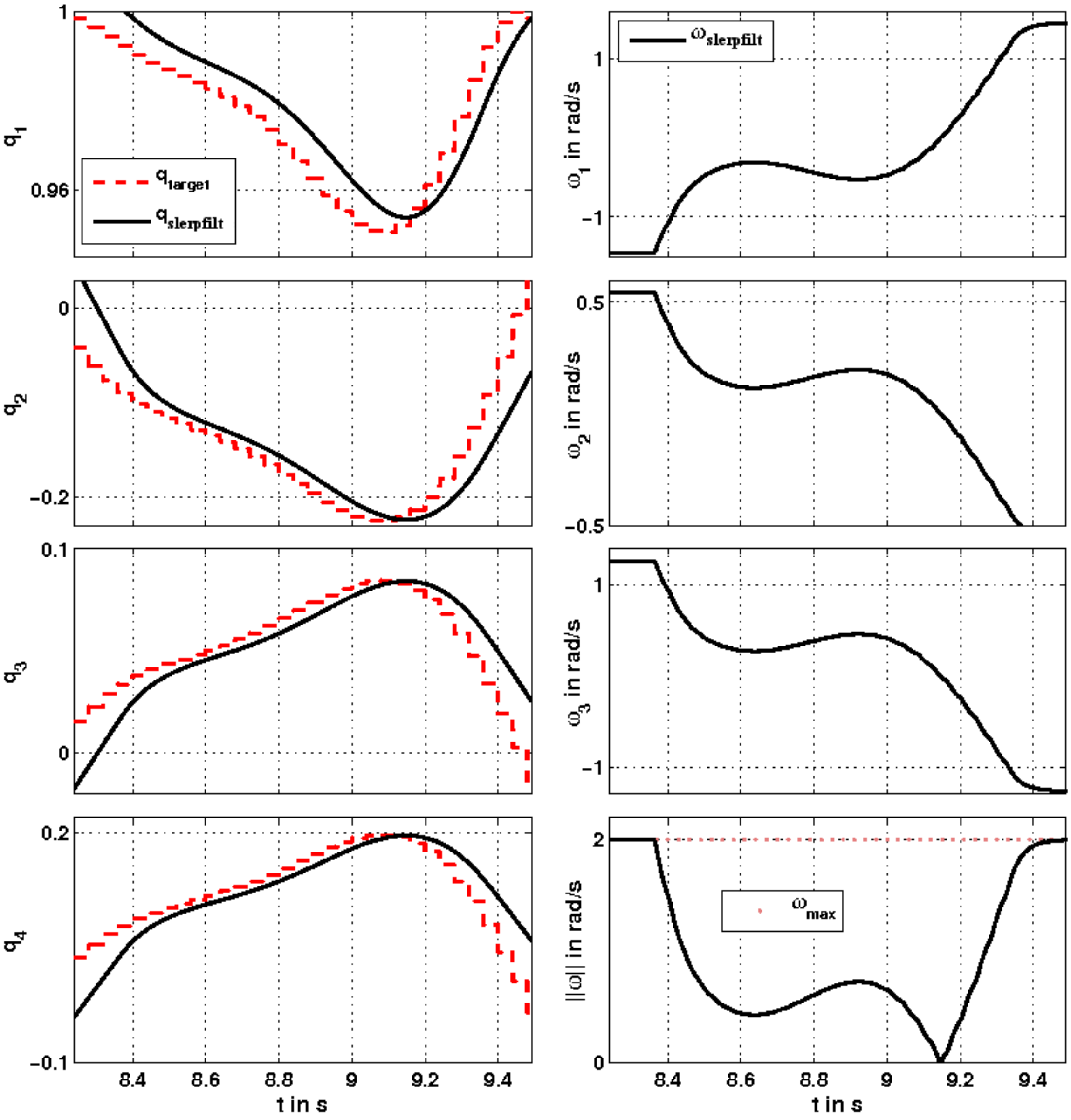


**Figure 3.5:** Left row: Comparison of artificially generated input data of low frequency with the filtered SLERP-generated trajectory. Right row: Angular velocities generated from the filtered SLERP algorithm. The right bottom plot depicts the absolute angular velocity (blue) and the velocity limit of 2 $rad/s$ which is never exceeded. Image ©IEEE

10 Hz, to compare the results, and to show that the presented method is also suitable for low frequency position measuring sensors.

In the following section, the experimental results are presented and discussed.

### 3.2.7 Experimental Results

The measurements of the human hand orientation show a remarkable noise. In Figure 3.7 the results of the first experiment are depicted. In the first experiment, a sensor frequency of 100 Hz is used, whereas the updating frequency of the trajectory generation input was deter-

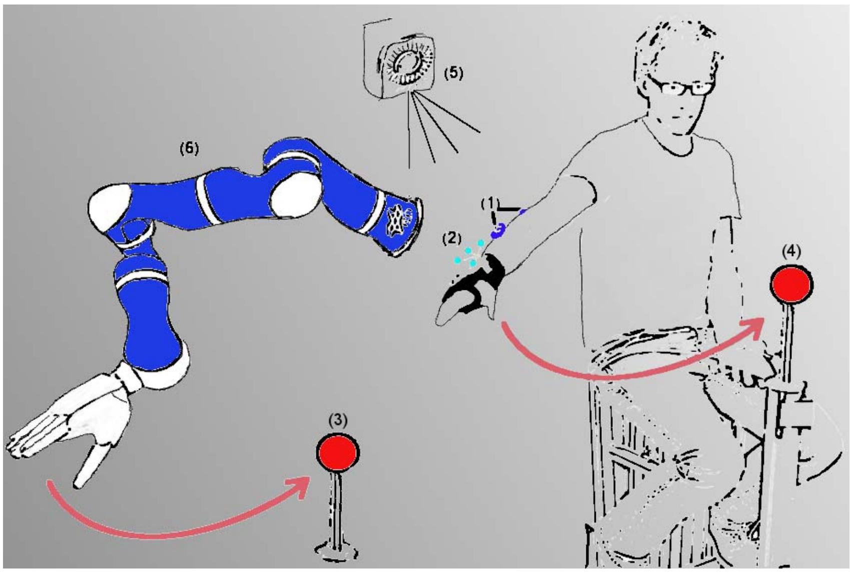


**Figure 3.6:** Schematic setup of the experiment - The human participant is equipped with EMG sensors (1) for triggering the grasp command and tracking markers (2) for tracking the position and orientation of the human arm with a VICON tracking system (5). The task in this experiment is to dynamically grasp the ball (4) without slowing down while the LWR III robot (6) is exactly mirroring the task (3). Image ©IEEE

mined by the frequency-estimation from Equation (3.11), which finally yields a lower sensor frequency output. The position measurements, obtained from the marker tracking sensor, serves as input to the impedance controller of the robot.

In the left column of Figure 3.7, the desired quaternions are depicted as red dashed line. The resulting unfiltered SLERP trajectory output is depicted as blue dashed-dotted line, and the final controller input, i.e., the filtered SLERP, is depicted as black solid line. As can be seen, the unfiltered signal of the sensor is used as input and, partially, outliers can be observed in the sensor signal, e.g., left column second row. In the right column of Figure 3.7, the resulting angular velocities are depicted. The upper three plots depict the angular velocity in each direction, and the bottom right plot depicts the total angular velocity. The maximum desired angular velocity is given by 2 rad/s, which is depicted as gray dashed line. Velocity steps close to 2 m/s can be observed by using only the SLERP algorithm output, which is illustrated as blue dashed dotted line. The black solid line depicts the filtered SLERP algorithm, as presented above. As can be seen, the trajectory behaves in a very smooth manner and no bend or step can be observed. The robot motion behavior was smooth and natural, which is the desired behavior for mirroring human arm motions, and for instance programming the robot in such way.

To also being able to use sensors with lower frequency, in a second experiment, the sensor frequency was limited to 10 Hz and given as input to the online trajectory generation algorithm. In this experiment, the noise was less and the position measurement much more robust. Therefore, the resulting trajectories by SLERP and the filtered variant map the sensor signal more exactly. Velocity limits are not exceeded, and the behavior is also smooth with a low frequency sensor signal.

To illustrate the experiments, a video sequence shows the exact process of the experiments. In Figure 3.9 can be seen that the experimenter starts at a very different initial orientation and the robot is mirroring the human motion with a little delay, which is the result of the sensor to input delay and the control chain of the robot.

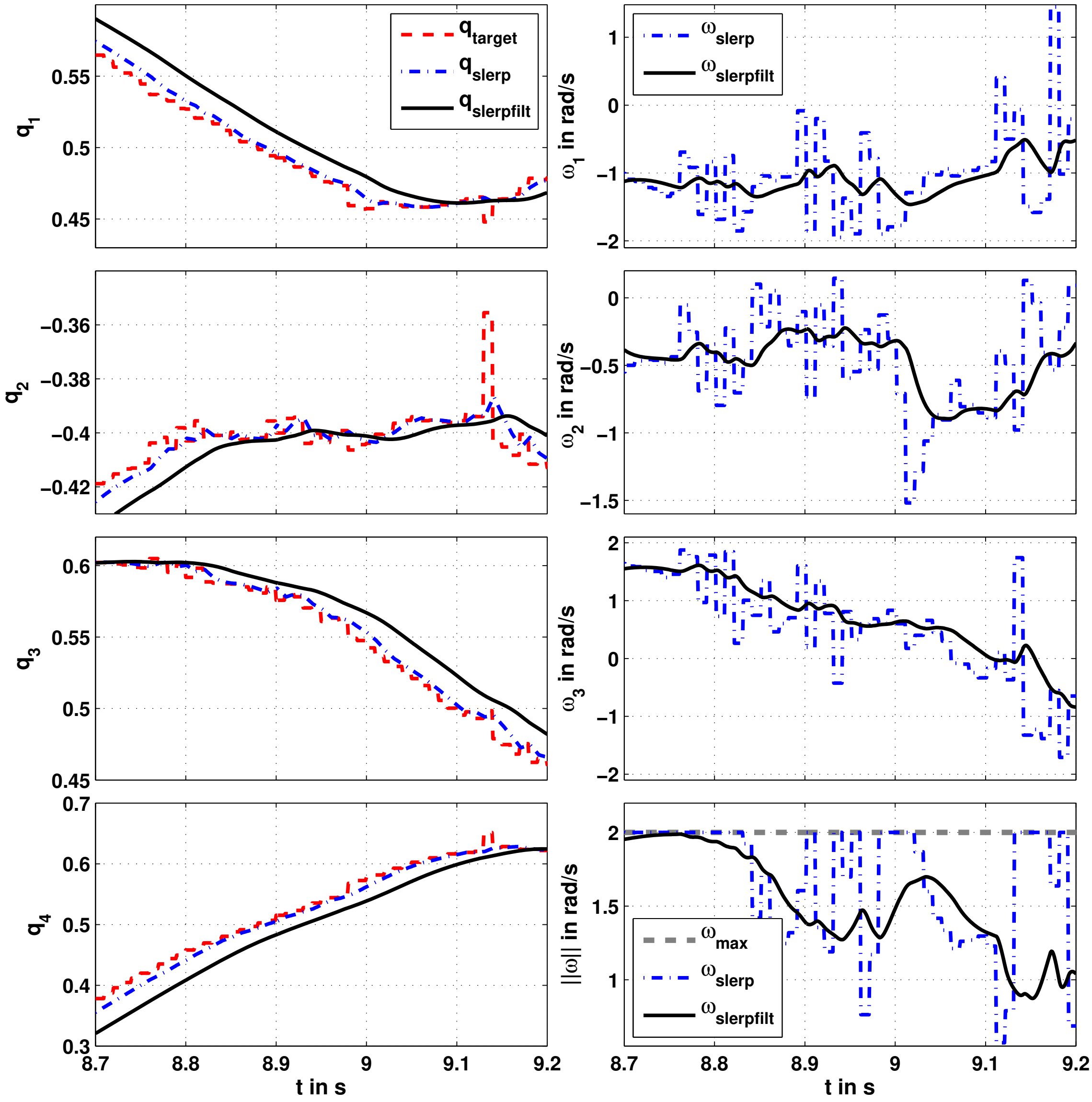


**Figure 3.7:** Left row: Comparison of 100 Hz sensor data (red), as received from the VICON system, the SLERP-generated trajectory (blue) and the final interpolator output (black). Right row: Comparison of angular velocities generated from the SLERP algorithm (blue) and the results of the filtered SLERP output (black). The right bottom plot depicts the absolute angular velocity (SLERP in blue, filtered SLERP in black) and the velocity limit of 2 $rad/s$ (grey), which is never exceeded. Image ©IEEE

To summarize, the presented method serves as suitable online trajectory generation and can be applied to real industrial applications. An extension to multiple robot applications, or especially separated online programming, provides tremendous potential in future robot programming.

In the following section, this approach is extended to an online via-point trajectory generation, to achieve a desired online velocity variation method for safe human-robot collaboration.

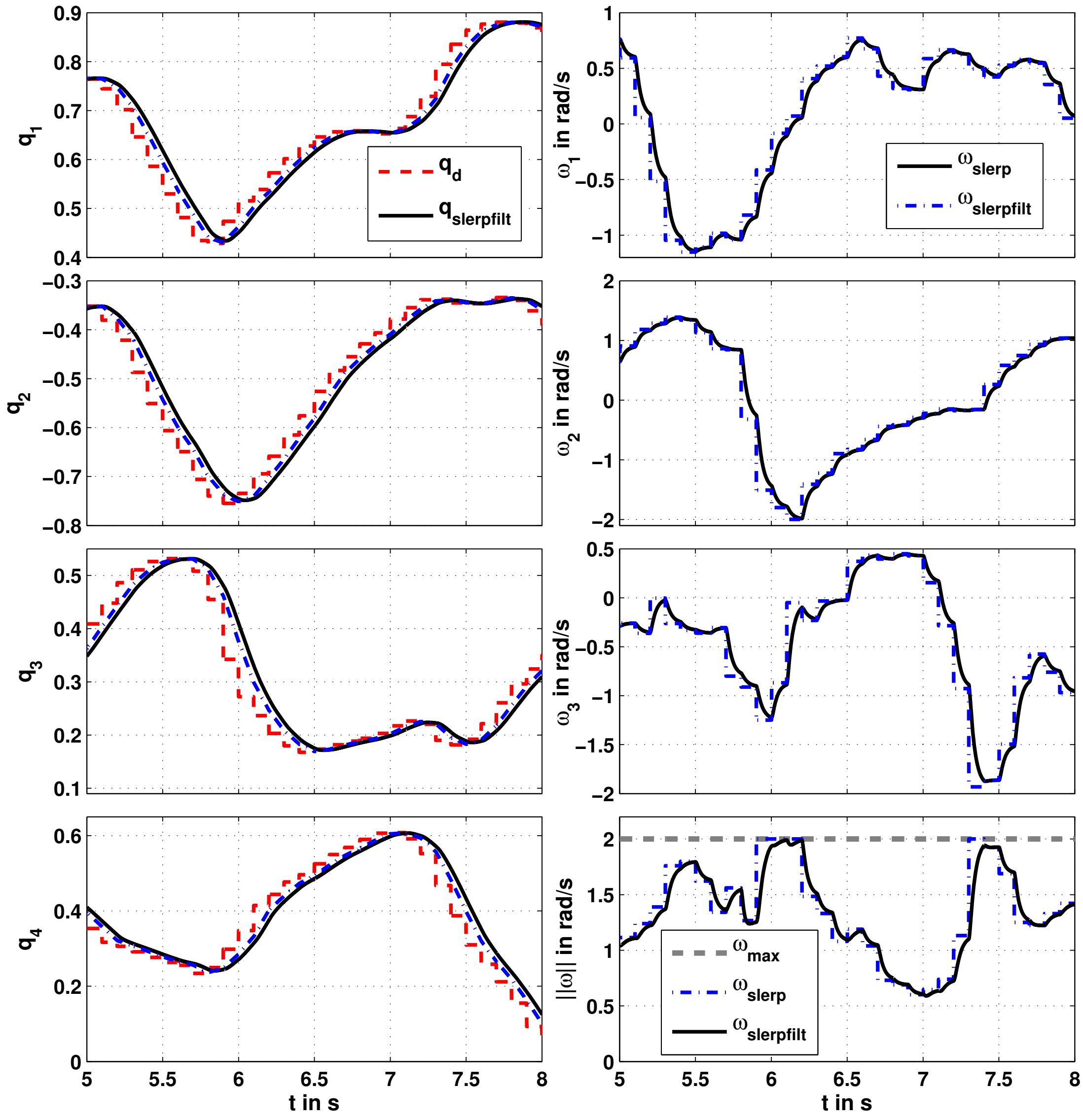


**Figure 3.8:** Left row: Comparison of 10 Hz sensor data (red), as received from the VICON system, the SLERP-generated trajectory (blue) and the final interpolator output (black). Right row: Comparison of angular velocities generated from the SLERP algorithm and the results of the filtered SLERP output. The right bottom plot depicts the absolute angular velocity (SLERP in blue, filtered SLERP in black) and the velocity limit of 2 $rad/s$ (grey), which is never exceeded. Image ©IEEE

## 3.3 Online Via-Point Motion Generation

In this section, the extension to online via-point trajectory generation is presented. In a first step, the approach is described, and, finally, the realization is explained.

### 3.3.1 Approach

In general, via-point motions enable more efficient robot motions. In most industrial applications, via-point motions are used to decrease the cycle-times and to avoid collisions with the environment within the task. The generation of motion using auxiliary points is usually executed offline. Lines between via-points are combined and blends are added to avoid jerky motions. Different methods to generate via-point motions exist, for example parabolic

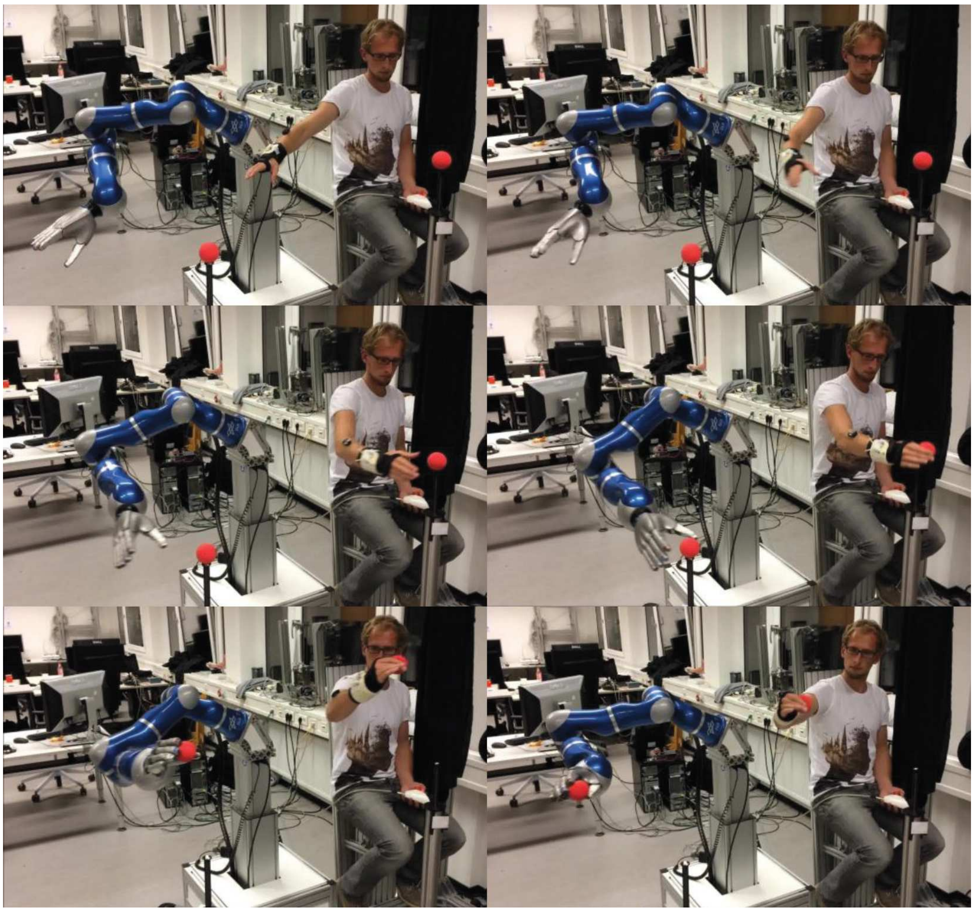


**Figure 3.9:** Grasping sequence where the human picks a ball from a stand and the robot mirrors the motion and the grasping. Image ©IEEE

blends, cubic or higher-order polynomials, splines, and B-Splines. A very good overview can be found in [79, 73]. In order to enable online adaptable via-point motions, the developed motion generation from Section 3.2 is extended from online mirroring human arm motions to a standard via-point trajectory generation with adaptable velocities. For improving the efficiency in collaborative robotics, the velocity adaptation during the motion, as well as the online path adaptation, are necessary characteristics.

Figure 3.10 gives an overview of possible or task-oriented via-point motions, respectively. An exact and direct motion to the via-point and goal position is not possible, if a stop-motion of the robot has to be avoided. In general, the three cases a)-c) are distinguished. In a) the general blend motion is illustrated, which is usually used for generating online motions. In robotic applications, often specific areas have to be avoided, in order to ensure safety or that no collision with objects can occur. Therefore, the case b) defines the shortest post-blend that passes the via-point. A very specific motion can be seen in c), which represents for example a dynamic grasp motion on a conveyor belt, a welding, or gluing operation with a robot. Cases d)-g) represent the turning motion, equivalent of the first three cases. Cases e) and g) occur in case a) and b) when no adaptation is performed. How to realize the different cases is presented in the next section.

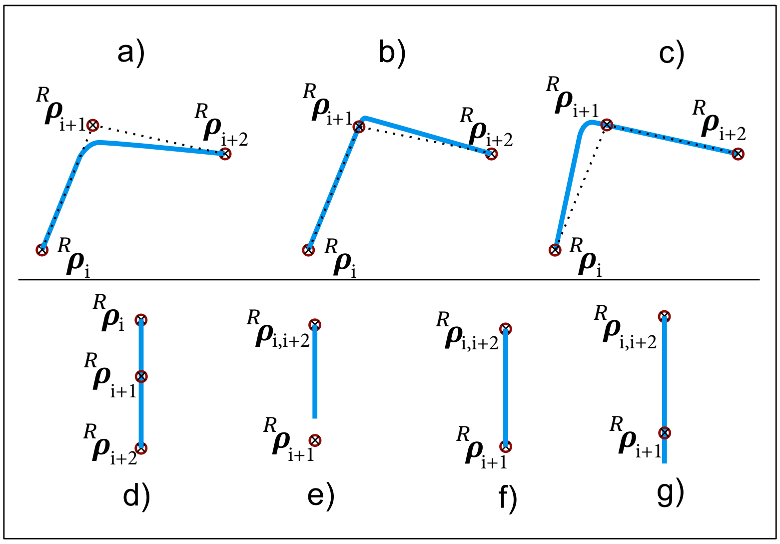


**Figure 3.10:** Behavior of different via-point motion with desired via-point behavior.

### 3.3.2 Transformation from Sensor Input to Online Via-Point Movements for Direct Paths

For the online trajectory generation, the algorithms, presented in Section 3.2, are used with minor modifications. The via-points are stored to a common ringbuffer [80] and the desired via-point is switched, when the current distance-to-goal parameter reaches the value one. Generating the motion with Equation (3.23), the distance-to-goal parameter can be calculated by[1]

$$\lambda(\tilde{\mathrm{x}}) = \frac{\tilde{\mathrm{x}}(t) - \boldsymbol{\rho}_i}{\Delta\boldsymbol{\rho}_i}. \tag{3.26}$$

with $\tilde{\mathrm{x}}(t)$ being the unfiltered output of the interpolator, see Figure 3.2. If $\lambda(\tilde{\mathrm{x}}) \geq 1$ the next via-point is taken from the ringbuffer and set as next goal position. This leads to via-point-behavior a), without affecting the velocity, such that the robot does not decelerate close to the goal position. As mentioned above, in collaborative applications it can be important that a defined zone is prohibited, or the path should exactly pass the via-point, as shown in case b). Then, an auxiliary via-point has to be adapted and can be calculated by

$$\boldsymbol{\rho}^*_{i+1} = \boldsymbol{\rho}_{i+1} + \xi \frac{\boldsymbol{\rho}_{i+1} - \boldsymbol{\rho}_i}{|\boldsymbol{\rho}_{i+1} - \boldsymbol{\rho}_i|} \tag{3.27}$$

with

$$\xi = 2\sqrt{\tau}\dot{x}_{max}. \tag{3.28}$$

To achieve a path accuracy, as shown in c), the via-points can be adapted by

$$\boldsymbol{\rho}^*_{i+1} = \boldsymbol{\rho}_{i+2} + \xi \frac{\boldsymbol{\rho}_{i+1} - \boldsymbol{\rho}_{i+2}}{|\boldsymbol{\rho}_{i+1} - \boldsymbol{\rho}_{i+2}|}. \tag{3.29}$$

To enable a direct and fast grasp motion, as shown in Figure 3.10 f), the auxiliary goal position $\boldsymbol{\rho}^*_{i+1}$ has to be adapted by the blend parameter, where the new blend parameter can be calculated by $\tilde{\xi} = \frac{1}{2}\xi$.

This online motion generation is implemented in the Collaborative Workcell and is used for experiments presented in the following chapters.

---

[1] for clarity the robot index is omitted: $^R\rho = \rho$.

## 3.4 Conclusion

For programming a robot, and staying out of the workspace of the robot, the mirroring of human arm motion approach provides fast and safe guidance of the robot. Here, the main focus is on the generation of continuous orientation with shortest distance to goal. This can be achieved, by using the SLERP algorithm, which also enables a limitation of velocities. Additionally, it enables the online adaptation of velocities, which is a requirement for efficient and safe robot trajectory generation. The developed filtering method, presented above, enables a smooth trajectory behavior, in particular for the desired velocity. Finally, the human can program the robot intuitively by executing the same application in a safe distance to the robot. The easy programming enables low set-up times, and improves the overall performance of the robotic system. The experiments, including the grasp of an object, prove the functionality of the presented method.

## Chapter 4

# Human Observation and Projection Performance Improvement by Attention

In this chapter, the idea of taking human dynamics into account is analyzed and discussed. The consideration of human safety is usually treated statically and the behavior of the human is not considered. In industrial standards, there are essential statements to describe dynamic consequences in case of a collision with a machine, but in fact there are two parameters treated in the case of a collision: impact force and impact pressure [8]. In the research community, these parameters can be ignored by introducing collision avoidance methods. Innovative human motion tracker systems enable the development of promising algorithms that improve the efficiency in fence-less human-robot collaboration. Human motions can now be tracked, or, in particular, the dynamics can be simulated. Hence, the prediction or projection of future human actions is possible.

In this chapter, the idea of considering humans will be further developed. A dynamic assumption of a human motion enables new technologies for more efficient human-robot collaboration. Therefore, the focus of this chapter is on the observation of the human and, thus, a projection of human motions for an improvement of performance is developed, without affecting the safety requirements. This constitutes the basis for further steps presented in this thesis.

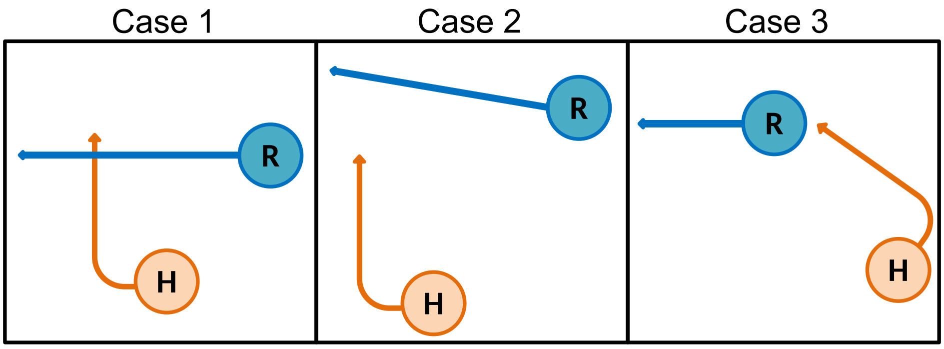


**Figure 4.1:** Differentiation between three cases: Case 1: The human is fast enough to cross the robot path. Case 2: The kinematic limits of the human are reached. Case 3: The robot is able to move very fast, the human cannot cross the robot path. Image ©IEEE

The main idea is illustrated in Figure 4.1, where three situations are depicted that show the scene before a collision between human and robot can occur. The left image depicts the scenario, where the human arm has the range to get into the path of the robot and a collision is quite possible. The middle image shows the situation, where the human is out of the range and the kinematic parameters of the human arm do not allow for reaching the robot's path. The last image, at the right, shows the robot in a motion away from the human. In this situation, there is a point at which the human cannot reach the robot anymore because the robot is moving too fast and is close to the kinematic limits of the human, or to its own desired goal position, respectively.

To obtain representative results, in a first step, human arm motions are analyzed and experimentally recorded according to general industrial scenarios. In a second step, a human arm model is developed, which is able to minimize the computation time such that it is suitable for the usage in real-time applications. The parameters of the arm are adapted and human arm motions are simulated. In the next step, a human arm state observer is presented, which is needed to transform Cartesian sensor information into joint values, to obtain current state information of the dynamic arm model. To calculate the maximum allowed velocity, motions of the human arm are projected into the path of the robot. The simulated results are compared to real human arm motions, as a validation of the simulation model.

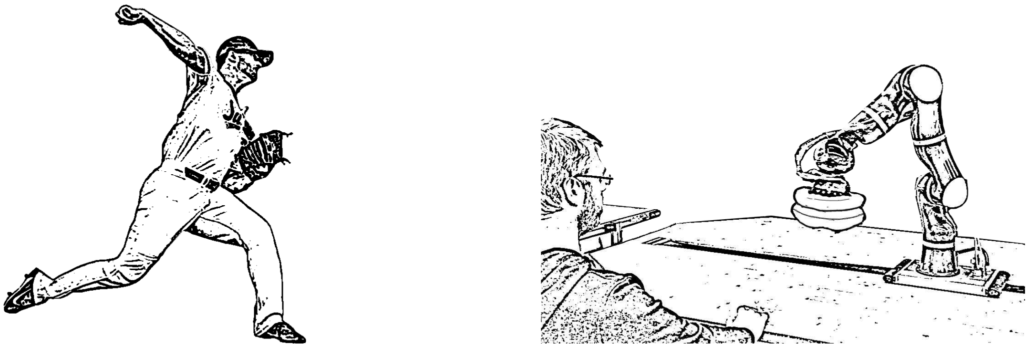
## 4.1 Human Arm Motions

The main idea of this section is to exploit the underlying dynamics of a human arm. The holistic goal is to obtain a time-span to a possible collision between robot and human, which requires an understanding of human arm motions. Therefore, common motions in collaborative scenarios have to be found and analyzed. In this section, experiments are executed to obtain real measurements of human arm motions, which can be used for further calculations in order to improve operating efficiency of a collaborative application.

### 4.1.1 Preliminary Human Arm Motion Considerations

Human motion prediction is an upcoming topic in the human-robot collaboration community. New sensor technologies and high resolution cameras enable the measurement of human positions, or, in particular, the human arm position. Initially, research on motion prediction was introduced by the mobile robotics community, in order to predict a possible future path of the human. This enables an optimal replanning of the future robot way, passing a crowd of humans. Since robots and humans get closer in industrial applications, the prediction, especially for human arm motions, emerged [81, 82]. Conventional approaches are using kinematic calculations in the operational space and predict a possible future motion of the human arm. Additionally, they use data-sets to learn particular motions, e.g., carrying a suitcase or doing an Indian dance.

In this thesis, the focus is on the maximum possible performance a human can afford regarding human arm motions. Considering professional sportsmen, humans can achieve enormous velocities regarding arm movements. In Figure 4.2 a pitch is illustrated, where a human can achieve a pitching velocity of more than 40 m/s [83]. Various human arm motions have been analyzed and considered in different situations. It indicates that the resulting motion depends on the configuration and initial conditions [84, 85]. However, in industrial scenarios, as illustrated in Figure 4.2 b), such high dynamics are not expectable. Therefore, motion experiments are conducted, to obtain the behavior of fast or reflexive human arm motions. The initial configuration is determined by standard positions of shop floor workers during an assembly application.



(a) Throwing a baseball

(b) Standard collaborative scenario

**Figure 4.2:** Human arm motions

In the next section, the experiments are described and executed to obtain real human arm motion measurements.

### 4.1.2 Experiments for Fast Human Arm Motions

In this section, human arm motions are recorded and analyzed. The goal of this research is to obtain the dynamics, and get an overview regarding the real performance of humans. Basically, a motion into the direction of the robot is of interest, in particular the duration of a change of the movement direction. The inertia is the decisive factor in this analysis, especially the behavior of a turning motion, regarding the Cartesian movement direction of the human hand, initially presented in [86].

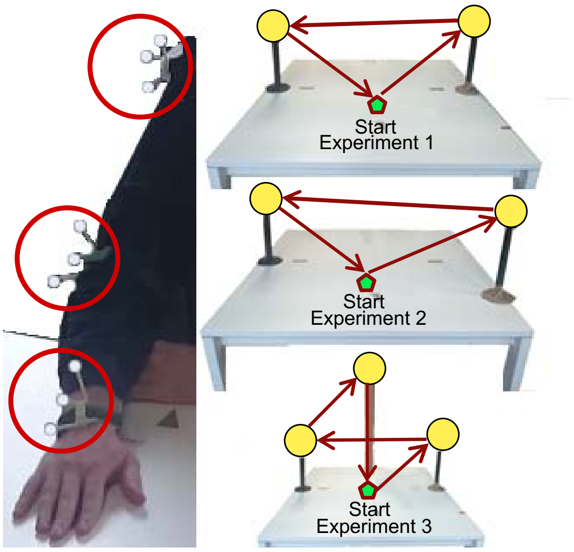
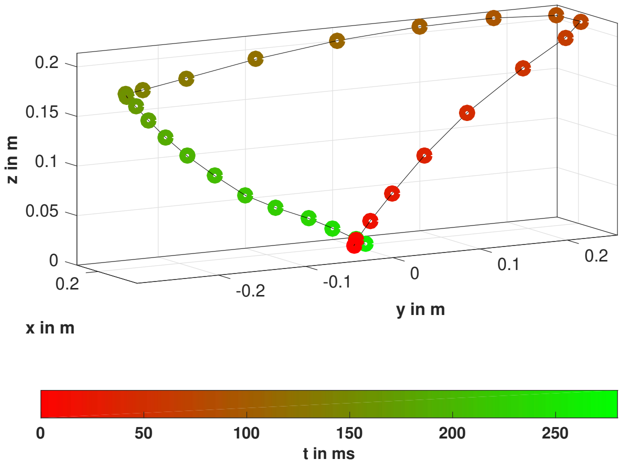
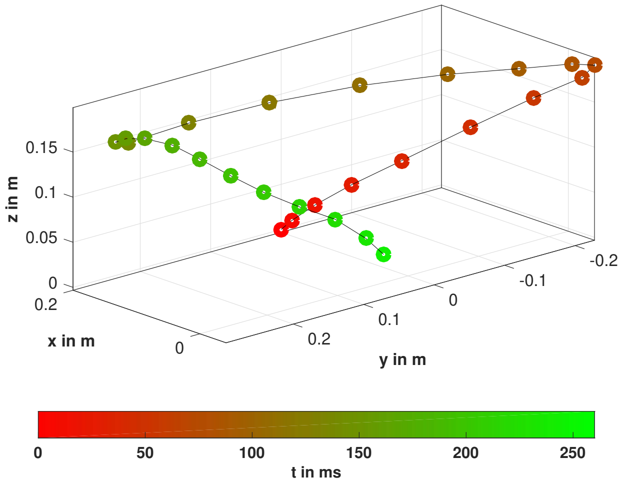


**Figure 4.3:** In these experiments the human experimenters should throw the objects from the pillows as fast as possible. Starting in a rest position the human arm has to be accelerated to get as fast as possible. Image ©IEEE

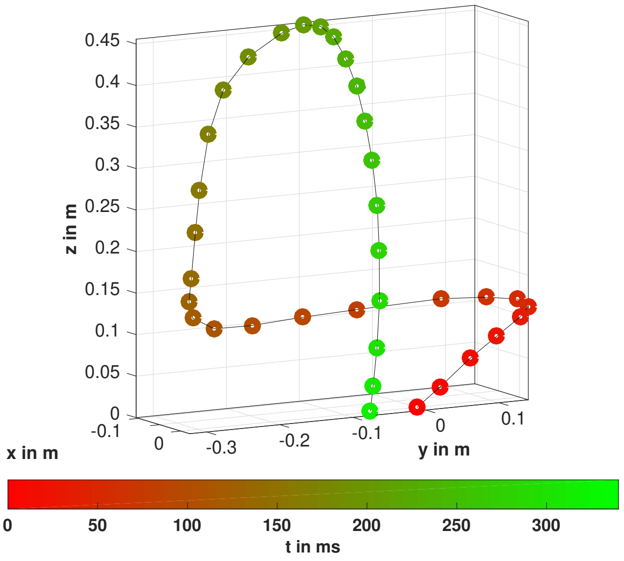
To record the human motions, the subject is placed in a usual industrial environment, i.e., sitting in front of a work bench. On the ceiling, VICON motion sensors are mounted that cover the entire table area. The VICON measurement frequency is fixed at 100 Hz, such that every 10 ms a new position is recorded. The special markers for the VICON are placed at defined positions of interest of the human arm. The first one is placed at the back of the human hand,



(a) Experiment 1



(b) Experiment 2



(c) Experiment 3 full

**Figure 4.4:** Measurements of the human motion from the experiments described in Fig. 4.3. Image ©IEEE

to obtain the end-position and the motion of the hand in Cartesian space, as shown in Figure 4.3 left column. The second marker is placed at the elbow, to obtain representative arm joint angles. The last marker is placed at the shoulder joint. In further considerations, these positions are necessary, because the human body is treated as static base for the human arm.

In Figure 4.3, right column, the experimental setup is illustrated. Three different tasks are executed, to obtain a certain range of motions, and more data for the evaluation of the dynamic model. In every experiment, a constant initial position is defined, which is in the middle of the table. Two or three pillars are placed in front of the human experimenter, where a foam ball is lying loose on top of each, which is illustrated as yellow circle. The task for the human experimenter was to slap down this softball from every pillar, with the aim to execute the motion as fast as possible. In total three different cases are specified. The first experiment includes two pillars, which are placed at the same distance to the experimenter, in particular, at position $P_1 = [0.4\ 0.2\ 0.15]\,\mathrm{m}$ and $P_2 = [0.4\ -0.2\ 0.15]\,\mathrm{m}$. The height of the pillar is chosen to be at a usual height considering the path of a robot. For the second experiment, the position of the first pillar is changed. Here, the motion of the human arm executes a back motion, to analyze the turning motion. The positions of the pillars in experiment two are therefore at $P_1 = [0.2\ 0.2\ 0.15]\,\mathrm{m}$ and $P_2 = [0.4\ -0.2\ 0.15]\,\mathrm{m}$. In the third, and last experiment, the setup is extended to a third pillar more far away from the human. Here, the motion to the end of the reachability of the human arm shall be analyzed. The positions for the pillars are $P_1 = [0.4\ 0.2\ 0.15]\,\mathrm{m}$, $P_2 = [0.4\ -0.2\ 0.15]\,\mathrm{m}$ and $P_3 = [0.5\ 0.0\ 0.45]\,\mathrm{m}$.

The measurement for the human arm motions are visualized in Figure 4.4, where the motion of the human hand is illustrated in Cartesian space. The three plots are equivalent to the described three experiments with different pillar positions. First of all, it is clearly visible that the human motion is much faster in comparison to the robot, where the time needed for a distance of $0.4$ m is about $60$ ms. The speed of the human hand can be calculated as $6.67$ m/s, i.e., the human hand is more than three times faster than the maximum robot velocity, which is $2$ m/s for the LWR III.

The motion to the first pillar needs an average time of $t = 70$ ms. The turning of the hand direction can be seen in this point. Additionally, it can be observed that the deceleration duration is equal to the acceleration duration. For the turning motion of the hand, the human needs around $20$ ms. In the second phase, which is the motion to the second pillar, a very short acceleration phase is visible but a constant velocity. This is probably caused by the different muscle strength as triceps and biceps are different. The third phase is clearly not executed as fast as possible, which is caused by a psychological point. The softballs are slapped down and the task is finished, so that the experimenter moved slowly back to the resting and starting position.

## 4.2 Model-Based Projection

In this section, a human arm model is generated and the kinematics of the arm model are defined. A human motion observer is presented to transform Cartesian information into the model joint angles and velocities to finally simulate possible human arm motions. Each model is simulated and experimentally evaluated. In the following, the main idea of this section is explained.

### 4.2.1 Approach

The human arm motion experiments indicate the influence of inertia onto the Cartesian path and the duration of motion. Especially the reverse motion shows the effect of the dynamic behavior. To reproduce this behavior, and exploit these data, a dynamic human arm model is

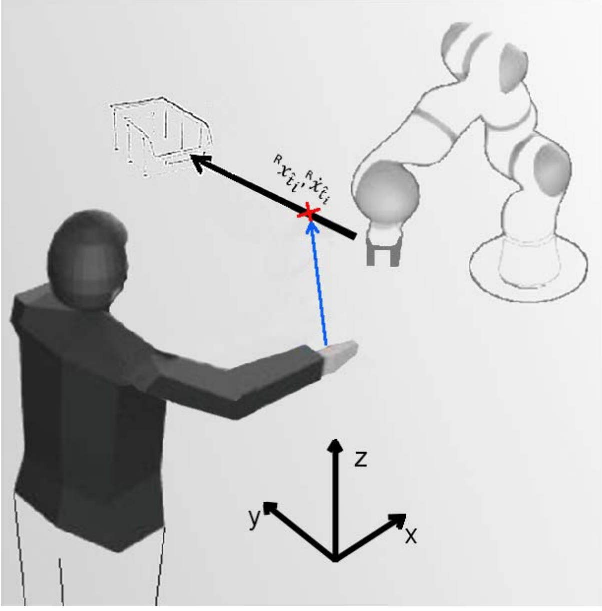


**Figure 4.5:** Human arm motion projection into the direction of the desired robot path. Calculation of the time up to a possible collision between the robot and the human arm by using a low-order dynamical arm model accounting for limited torques. Image ©IEEE

necessary. Usually, human positions are measured in Cartesian space by vision or marker-based sensors. In order to reduce set-up times, and enable a simplified sensing, the position measurement is limited to the human hand position, as shown in Figure 4.5. This enables the use of peripheral objects, or gloves, in this approach, respectively. However, this approach requires the transformation from Cartesian space into the joint space of the human arm model. Given the required initial values, a human arm motion is simulated into the robot's path. The resulting duration information can be used to increase the robot velocity and improve the efficiency of the application. A more detailed description is given in the following.

In Figure 4.6, the overall idea is illustrated. On the left side, the real-time observer is depicted. From a VICON position sensor, the current human hand positions $\mathbf{x}_s$ are measured and serve as input to a human arm observer. The observer estimates generalized coordinates of the arm model, by comparing the measurements and the transformed Cartesian positions of the human arm observer results. The observer is integrated to obtain smooth and accurate values for the position and the velocities of the current joint state, which serve as initial values for the human motion projection. In Figure 4.6, on the right, the non-real-time human arm motion projection is illustrated. The obtained values ${}^{H}\hat{\mathbf{q}}$ and ${}^{H}\dot{\hat{\mathbf{q}}}$ are the initial values for the dynamic model of the human arm. The projection of the human arm is not executed in real-time, because the entire motion has to be simulated as fast as possible. This implies a calculation at a frequency of 1 kHz up to 200 times. A duration of motion $\tilde{t}$ is obtained. From this time, the maximum allowed velocity for the robot can be calculated with respect to the possible breaking time of the robot.

In the following section, the human arm model is defined and the dynamic system presented.

### 4.2.2 Human Arm Model

In this section, the dynamic and kinematic modeling of a simplified human arm is presented. In previous research, often the dynamic human arm was considered in a plane, using two degrees of freedom (DoF) [87, 88]. Dynamics of higher DoF, up to precise multi-body dynamic arm models with learned parameters, provide accurate motion simulation [89, 90].

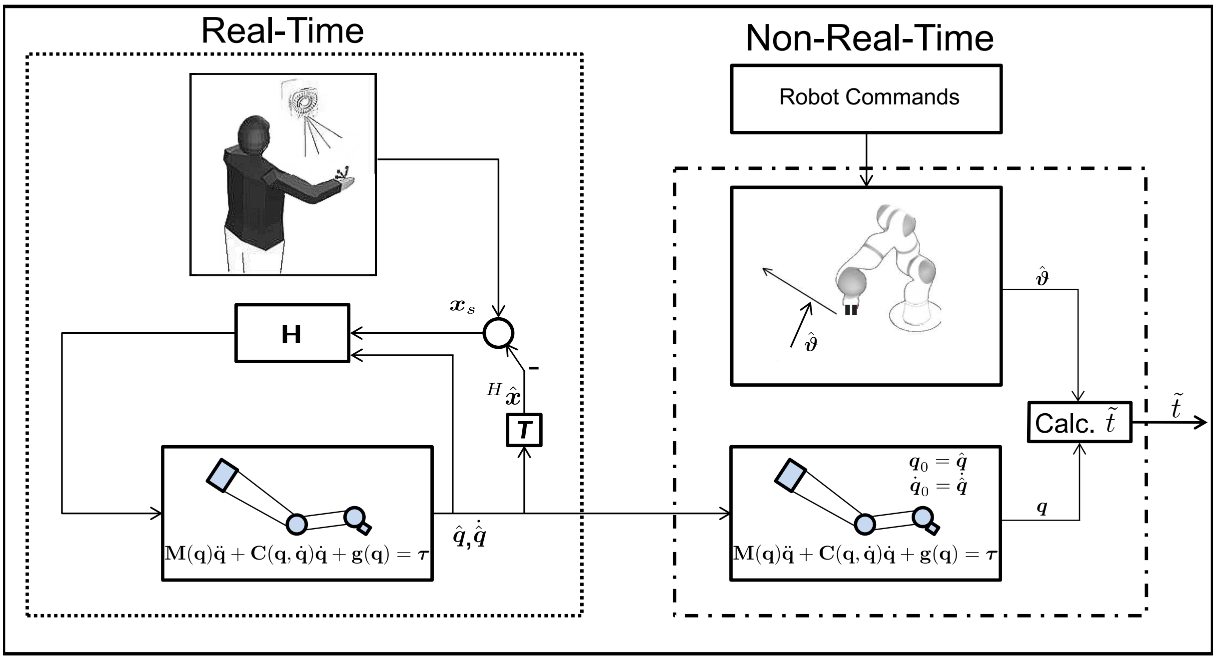


**Figure 4.6:** Structure of pre-calculated human motion into the direction of the robot path, calculating the time needed to get into the path considering the dynamics and limited torques of a 'human joint'. The calculated time to hit the robot defines the maximum velocity of the robot. Image ©IEEE

In order to ensure safety, full motions have to be calculated in the shortest possible time. Therefore, a reduction of the DoF is necessary, to enable a reduction of arithmetic operations. Hence, to achieve a three-dimensional Cartesian motion, a minimum of three DoF is required. The arm kinematics are defined by the length of the upper arm and the lower arm of the experimenter in Section 4.1.2. Average values can be taken from [91], where studies of multiple human kinematics are analyzed. Furthermore, the orientation is not of interest because the movement behavior is not influenced by the rotation of the hand, when considering the fastest motion to the path of the robot.

Applying three DoF yields an exclusion of redundancy, which is a specific property of the human arm and can be observed on elbow movements. Finally, the movement of the elbow cannot clearly be determined. Using a three DoF model is therefore an appropriate choice for a minimal model of the human arm. In the following, the human body is treated as fixed, comparable to a non-moving human. Hence, the human arm model possesses a fixed base.

In the following section, the kinematics of the human arm model are described, as well as the dynamic modeling of a human arm for a simplified dynamic model.

### 4.2.2.1 Kinematics Human Arm Model

For kinematic definition of a human arm, and the simplification to three DoF, three main rotational joints are used, to enable motions in the translatory $\mathbb{R}^3$ Cartesian space. Derived from the human arm structure, the kinematic definition includes a shoulder joint, consisting of two orthogonally arranged joints $q_1$ and $q_2$. In Figure 4.7, the configuration is illustrated. The third joint $q_3$ represents the elbow, which leads to the rotation definition of

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\mathrm{q}_1) & sin(\mathrm{q}_1) \\ 0 & -sin(\mathrm{q}_1) & cos(\mathrm{q}_1) \end{bmatrix}, \mathbf{R}_{y_1} = \begin{bmatrix} cos(\mathrm{q}_2) & 0 & sin(\mathrm{q}_2) \\ 0 & 1 & 0 \\ -sin(\mathrm{q}_2) & 0 & cos(\mathrm{q}_2) \end{bmatrix}, \mathbf{R}_{y_2} = \begin{bmatrix} cos(\mathrm{q}_3) & 0 & sin(\mathrm{q}_3) \\ 0 & 1 & 0 \\ -sin(\mathrm{q}_3) & 0 & cos(\mathrm{q}_3) \end{bmatrix}, \quad (4.1)$$

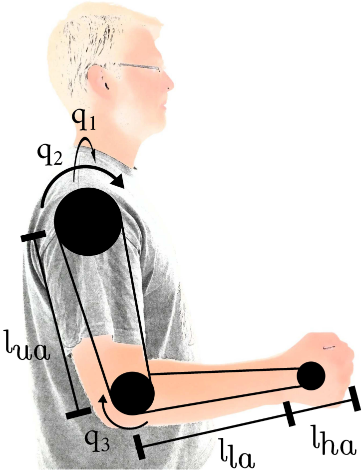


**Figure 4.7:** Human arm model. Image ©IEEE

requiring the length of each link. Each arm segment is defined by the

$$\mathbf{x} = \begin{bmatrix} 0 \\ 0 \\ l_{ua} \end{bmatrix}, \ \mathbf{y}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ \mathbf{y}_2 = \begin{bmatrix} 0 \\ 0 \\ l_{lat} \end{bmatrix}, \tag{4.2}$$

where the length of the upper arm is given with $l_{ua}$, and the length of the lower arm is defined by $l_{lat}$, which is the sum of the lower arm in combination with the hand position, i.e., $l_{lat} = l_{la} + \frac{1}{2}l_{ha}$. Due to the simplification, these two segments are combined, as the Cartesian position change is not significantly influenced by a human wrist motion. A classification and determination of length of segments, or human body parameters, can be found in [92, 91]

Using the general Denavit Hartemberg formulation, the transformation matrix is given by

$$\mathbf{T}_{hand} = \mathbf{R}_x \times \mathbf{R}_{y_1} \times \mathbf{T}_x \times \mathbf{R}_{y_2} \times \mathbf{T}_{y_1} \tag{4.3}$$

to obtain the Cartesian hand position from joint angles. The Jacobian matrix can be calculated by the partial differentiation as

$$\mathbf{J}_v = \frac{\partial \mathbf{v}_{hand}}{\partial \dot{\mathbf{q}}}, \tag{4.4}$$

where $\mathbf{v}_{hand}$ describes the Cartesian velocities at the hand frame. To enable the mapping from forces to torques, or from joint velocities to Cartesian velocities, respectively, the Jacobian matrix can be used as follows:

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{J}_v^T \mathbf{F} \\ \dot{\mathbf{x}} &= \mathbf{J}_v \dot{\mathbf{q}}. \end{aligned} \tag{4.5}$$

In the following section, the dynamics for the simplified human-arm model are described.

#### 4.2.2.2 Dynamics for Human Arm Model

The human arm dynamics are derived from the kinematics to a three DoF model, including three angular joint motions and two links, the lower arm and the upper arm. To obtain the dynamics of the human arm, the Lagrangian

$$L(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - V(\mathbf{q}) \tag{4.6}$$

can be written as a differentiation of the kinetic energy and the potential energy, as described in [73]. The kinetic energy of the rigid body is given by

$$T_b = \frac{1}{2} m_b ({}_b\mathbf{v}_{sb})^T ({}_b\mathbf{v}_{sb}) + \frac{1}{2} ({}_b\boldsymbol{\omega}_{sb})^T \boldsymbol{I}_b ({}_b\boldsymbol{\omega}_{sb}), \tag{4.7}$$

where ${}_b\mathbf{v}_{sb}$ and ${}_b\boldsymbol{\omega}_{sb}$ are the translational and the angular velocity vectors, $m_b$ is the mass, and $\boldsymbol{I}_b$ the inertia tensor. For the potential energy, only the gravity is of interest given by

$$V_g(q) = -g\mathbf{e}_g^T \sum_{i=1}^{n} m_{l,i} \mathbf{p}_{sl,i}(\mathbf{q}), \tag{4.8}$$

with $\mathbf{p}_{sl,i}$ being a vector pointing from the origin to center of mass, $\mathbf{e}_g$ being a unitvector pointing into the direction of the gravity, $m_{l,i}$ is the mass of the link, and $g$ is the acceleration of gravity. A more detailed overview can be found in [93]. The resulting equations of motions can be calculated by solving

$$\frac{d}{dt}\left(\frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} = \boldsymbol{\tau}, \tag{4.9}$$

with $\boldsymbol{\tau}$ being the torques [73]. This results in the standard dynamics form, given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \tag{4.10}$$

where M denotes the mass matrix, C is the Coriolis and centrifugal matrix, and g is the gravity vector. The generated dynamics serves as basis for the following simulation and controller design.

In the following section, a human arm state observer is presented that maps human motions into this dynamic model.

### 4.2.3 Human Arm State Observer and Motion Projection

In this section, the human arm motions are observed and projected, by using the simplified dynamic model in order to obtain the motion duration. In the first step, the observer-like design is presented and validated in simulation. Then, human arm motions are projected and compared to the measured real human arm motions described in the experiments above.

#### 4.2.3.1 Human Arm State Observer

The human arm state observer (HASO) is used, to obtain joint position information, derived from Cartesian position measurements of the human hand. The Cartesian human hand position serves as input for the HASO, which has an observer-like structure. Different technologies allow for measuring these values, e.g., by a markers-based tracking system such as VICON, a acceleration-based suit or the Microsoft Kinect, which is a vision-based system and can mostly be found in the field of gaming. For further validation and simulation, the VICON marker tracking system is used, as it provides higher accuracy and, therefore, enables a more precise evaluation of the developed method.

The illustration in Figure 4.6 shows the measured Cartesian position of the human hand, which is given by $\boldsymbol{x}_s$. These sensed position information serve as input for the impedance-based feedback-controlled system, which also provides the current angular velocity of each joint. Therefore, the mapping from the Cartesian space to joint space is necessary. To obtain the joint states, the equations for the HASO are given by

$$ {}^{H}\dot{\mathbf{q}}_t^{*} = \begin{bmatrix} {}^{H}\dot{\hat{\mathbf{q}}}_t \\ {}^{H}\ddot{\hat{\mathbf{q}}}_t \end{bmatrix} = \mathbf{f}({}^{H}\hat{\mathbf{q}}_t, \mathbf{u}_t) + \mathbf{H}(\mathbf{x}_{\mathbf{s}}, {}^{H}\hat{\mathbf{x}}_t, {}^{H}\hat{\mathbf{q}}_t, \boldsymbol{\tau}_H), \tag{4.11} $$

$$ {}^{H}\hat{\mathbf{x}}_t = \mathbf{T}({}^{H}\hat{\mathbf{q}}_t), \tag{4.12} $$

where ${}^{H}\hat{\mathbf{q}}_t \in \mathbb{R}^3$ is the vector of the observed generalized coordinates in joint space, ${}^{H}\mathbf{x} \in \mathbb{R}^3$ denotes the Cartesian coordinates of the position of the hand, and $\mathbf{u} \in \mathbb{R}^3$ describes the control input of the low-order dynamical system, whereas the input is not measurable, yet, and ,therefore, $\mathbf{u}_t = \mathbf{0}$. The HASO is illustrated in Figure 4.6, left side, in the real time part. The non-linear differential equation from Equation (4.10) is given by f with respect to ${}^{H}\hat{\mathbf{q}}_t$. The mapping from joint space into Cartesian space is performed with $\mathbf{T}$.

The control torques withing the HASO are based on an impedance-based controller design presented in [94], to observe the joint angles and velocities. The formulation for a general impedance control is given by

$$ \begin{aligned} \boldsymbol{\tau}_H &= \mathbf{g}_q({}^{H}\hat{\mathbf{q}}_t) + \mathbf{J}_v^T({}^{H}\hat{\mathbf{q}}_t)\mathbf{F}_d && (4.13) \\ &= \mathbf{g}_q({}^{H}\hat{\mathbf{q}}_t) + \mathbf{J}_v({}^{H}\hat{\mathbf{q}}_t)^T(\boldsymbol{\Lambda}({}^{H}\hat{\mathbf{x}}_t)\ddot{\mathbf{x}}_d + \boldsymbol{\mu}({}^{H}\hat{\mathbf{x}}_t, {}^{H}\dot{\hat{\mathbf{x}}}_t))\dot{\mathbf{x}}_d && (4.14) \\ &- \mathbf{K}_d(\mathbf{e}(\mathbf{x}_s, {}^{H}\hat{\mathbf{x}}_t)) - \mathbf{D}_d(\dot{\mathbf{e}}(\dot{\mathbf{x}}_s, {}^{H}\dot{\hat{\mathbf{x}}}_t))), && (4.15) \end{aligned} $$

where $\mathbf{g}_q({}^{H}\mathbf{q}_t)$ denotes the compensation of gravity, and

$$ \boldsymbol{\Lambda}({}^{H}\hat{\mathbf{x}}_t) = \mathbf{J}_v^{-T}({}^{H}\hat{\mathbf{q}}_t)M({}^{H}\hat{\mathbf{q}}_t)\mathbf{J}_v^{-1}({}^{H}\hat{\mathbf{q}}_t) \tag{4.16} $$

denotes the inertia, mapped into the task space, which is multiplied with the desired acceleration $\ddot{\mathbf{x}}_d$. Together with $\boldsymbol{\mu}({}^{H}\hat{\mathbf{x}}_t, {}^{H}\dot{\hat{\mathbf{x}}}_t))$ and the desired velocity $\dot{\mathbf{x}}_d$, these terms denote the feed forward terms. As no desired values exist in this case, the desired acceleration and velocity is set to zero, which eliminates the terms $\boldsymbol{\Lambda}\ddot{\mathbf{x}}_d = 0$ and $\boldsymbol{\mu}\dot{\mathbf{x}}_d = 0$, see [94]. The impedance control describes a spatial spring behavior, which is related to the potential $\mathbf{K}_d$, and a positive definite matrix $\mathbf{D}_d$ is given as a damping term. The damping matrix is a constant diagonal matrix, or it can be calculated by double diagonalization [94], respectively. These parameter are empirically evaluated.

In the following section, simulations are presented to prove the functionality of the motion observer and the transformation to the joint coordinates.

#### 4.2.3.2 Simulation HASO

In order to evaluate the presented method, motions are simulated, whereas real human arm motions are used as input to the model. The resulting Cartesian positions are compared to the arm motion data, obtained from the experiments in Section 4.1.2.

In Figure 4.8, the results of the simulation in MATLAB/Simulink are illustrated. The measured Cartesian human hand position is depicted as black solid line in the upper plot. These position values are used as input to the HASO. The red dashed line depicts the transformed, or observed, Cartesian coordinates of the human arm model, respectively. It can be seen that after a short settling time, the behavior of the HASO is accurate according to the motion of the human arm. Therefore, the obtained joint values can be derived from this simulation. In the middle plot, the resulting observed joint positions are shown. The three lines depict the generalized coordinates $q_1 - q_3$ of the human arm model. In the bottom plot, the observed human arm angular velocities are illustrated. The settling time can clearly be observed, which finally results in a smooth behavior. To summarize, the observer-based method to obtain ini-

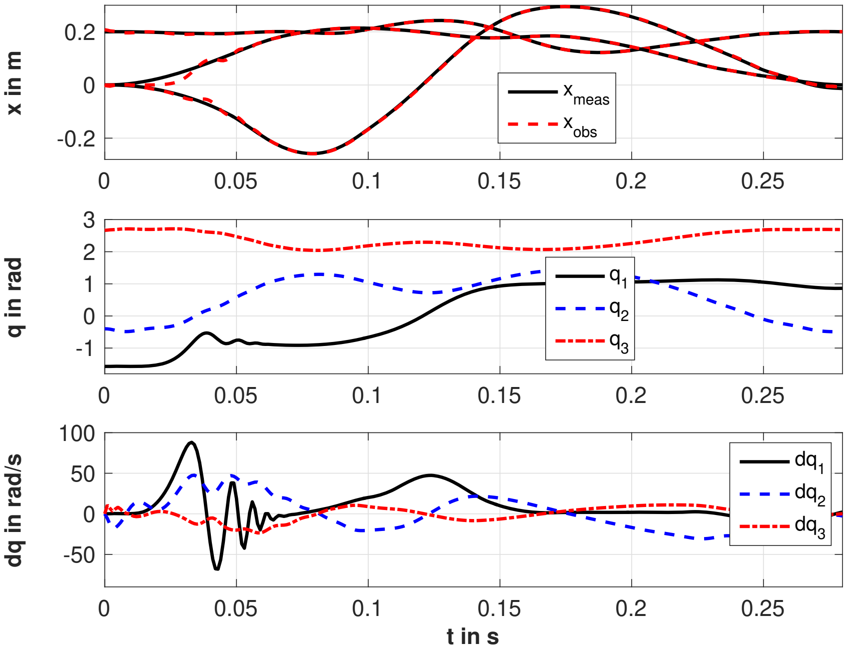


**Figure 4.8:** Mapping of Cartesian position information into joint angles provided by the HASO. The upper plot compares the measured human hand positions, depicted as black solid line, with the observed positions, red dashed line. The middle plot illustrates the mapped joint positions of the simplified dynamic model. The bottom plot depicts the angular velocities. Image ©IEEE

tial values yields suitable results at a high accuracy compared to real human arm motions. Therefore, the positions and velocities can be used as input for the human arm motion projection.

In the following section, the human arm motion projection is presented.

#### 4.2.3.3 Human Arm Motion Projection

As mentioned above, the projection of the human arm represents a pre-calculation of a human arm movement. A statement of probability of the human arm motion would not provide safety, in order to consider the worst-case scenario, which is required in a risk analysis. The human intention is not securely predictable, which is a residual risk for the human operator. The projection of the human arm serves the purpose that always a worst-case scenario is represented. Therefore, real dynamic information maximize the performance, whereas kinematic calculation do not represent human arm motions accurately. To obtain projections of the human arm, the simplified human arm model is controlled by an adapted controller design. The idea is to integrate a force pointing into the direction of the path of the robot. This can be compared to a spring behavior at the human hand. Due to the reason, a impedance controller could not be evaluated, an admittance controller design is developed.

In order to combine position and the force direction control scheme a PID controller is implemented as [79]:

$$\tau_H = \mathbf{K}_P \mathbf{e}(t) + \mathbf{K}_I \int_0^t \mathbf{e}(\tau) d\tau + \mathbf{K}_D \frac{d\mathbf{e}(t)}{dt} \tag{4.17}$$

with $\mathbf{K}_P$ being the proportional gain, $\mathbf{K}_I$ the integrator gain, $\mathbf{K}_D$ the damping term, and $\mathbf{e}(t) = {}^H\mathbf{q}_{des}(t) - \mathbf{q}(t)$ being the error between the desired joint position and the current joint position. The parameters are empirically adjusted with the information from the human arm motion experiment to obtain suitable results.

The direction vector for the admittance control is calculated by

$$\hat{\boldsymbol{\vartheta}} = \frac{{}^{R}\mathbf{x}_{\hat{t}_i} - \mathbf{x}_s}{\|{}^{R}\mathbf{x}_{\hat{t}_i} - \mathbf{x}_s\|}, \tag{4.18}$$

where ${}^{R}\boldsymbol{x}_{\hat{t}_i}$ describes the approximated position of the robot at an approximated collision time $\hat{t}_i$. This particular point can be achieved by a rough estimation of the collision point by calculating with the current velocity of the robot and the maximum velocity of the human arm. Given the desired direction vector from the current human arm position $x_s$ to the estimated collision point, the desired joint positions can be obtained by

$${}^{H}\mathbf{q}_{des} = \int_{0}^{t} \mathbf{J}_v^{*} \left(\kappa_{ad}\hat{\boldsymbol{\vartheta}} - \eta\dot{\mathbf{x}}\right) d\tau \ , \tag{4.19}$$

with $\eta$ being a damping term and $\kappa_{ad}$ a stiffness factor. In combination with Equation (4.17) the parameters are empirically defined to obtain an appropriate representation of the simplified dynamics with the real motions of the human arm, obtained from the experiments. Due to possible singularities at calculating the inverse Jacobian matrix, a pseudo inverse has to be calculated as

$$\mathbf{J}_v^{*} = \mathbf{J}_v^{T}(\mathbf{J}_v\mathbf{J}_v^{T} + \lambda^2\mathbf{I})^{-1}, \tag{4.20}$$

where $\mathbf{J}_v$ is the conventional Jacobian matrix, and the term $\lambda^2\boldsymbol{I}$ is a regularization term with identity matrix $\mathbf{I}$, which also avoids that the inverse Jacobian becomes zero [95]. The value $\lambda$ defines the gain for the regularization and is chosen as $\lambda = 0.3$. The proposed damped least squares solution was introduced by [96].

In the following, the results of the simulation are presented.

#### 4.2.3.4 Simulation

In this section, an exemplary use case is simulated to evaluate the functionality of the human arm motion projection. The simulation is based on the human arm motion experiments. The initial position of the human hand is located 0.2 m in front of the virtual human. The initial

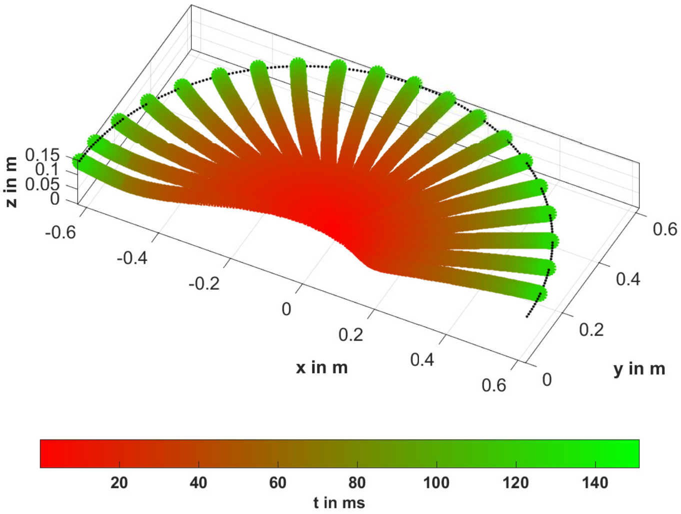


**Figure 4.9:** Human arm motion simulation sample. Multiple motions simulated within 180 $deg$ radius. Image ©IEEE

velocity is zero, as the simulation starts from a resting position. The direction vector is sampled to 20 simulations within 180 $deg$ from the human base.

The results of the simulation are visualized in Figure 4.9. A direct motion from the start position to an end position, up to the kinematic limits of the human arm, is depicted. As can be observed, the motion of the hand position in Cartesian space shows the same behavior as illustrated in the measurements of real human arm motions. It can be observed that the motion does not result in straight lines, as kinematic considerations would do. The kinematic limitation of the human arm can be reached within 140 ms, but the arm is also staying in that position. It is shown that the motion is similar to the measurements as a distance of $0.2$ m can be reached within $60$ ms, when no turning motion included. This simulation allows for motion from every start position to every end position, under consideration of the kinematic limits of the human arm. This enables a very flexible estimation of possible collisions with the human arm.

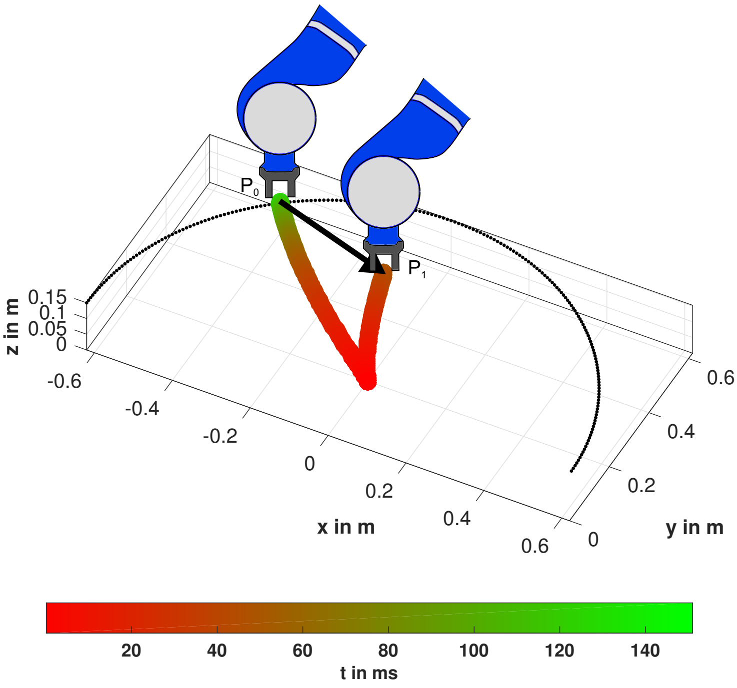
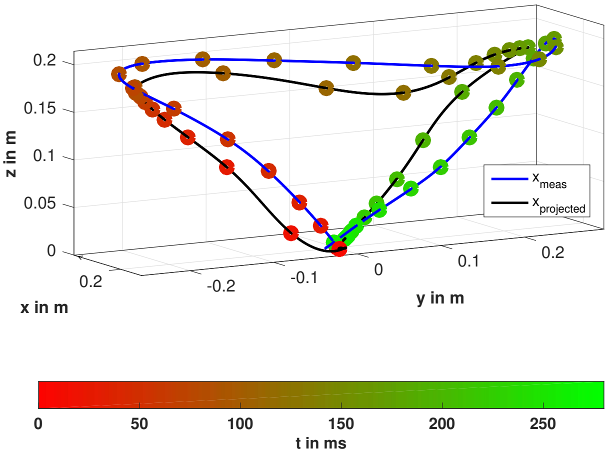


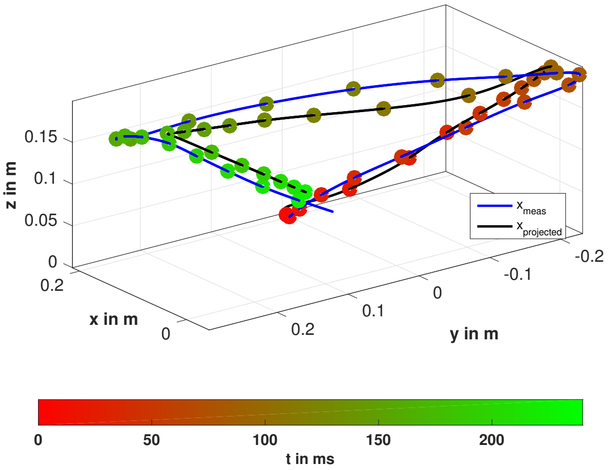
**Figure 4.10:** Simulation to a possible collision with the robot. The figure illustrates two different situation. The first shows that the robot is entering the shared workspace. From this point the robot velocity has to be decreased in order to guarantee safety. The velocity has to be further reduced depending on the duration of motion of the human arm, which is shown in $P_1$. Image ©IEEE

In Figure 4.10, an exemplary situation is illustrated, which shows the idea of this methodology. At position $P_0$ the borderline of the shared workspace, including the kinematic limits of the human arm, is depicted. Beyond this area, depicted as black solid line in a semicircle, the maximum possible velocity is limited by the kinematic properties of the robot. This statement assumes the limitation of a fixed base. In future work, the dynamic behavior of the human body has to be included. The sampled motion information is stored to look-up tables in order to use the human arm motion behavior in further research within this thesis.

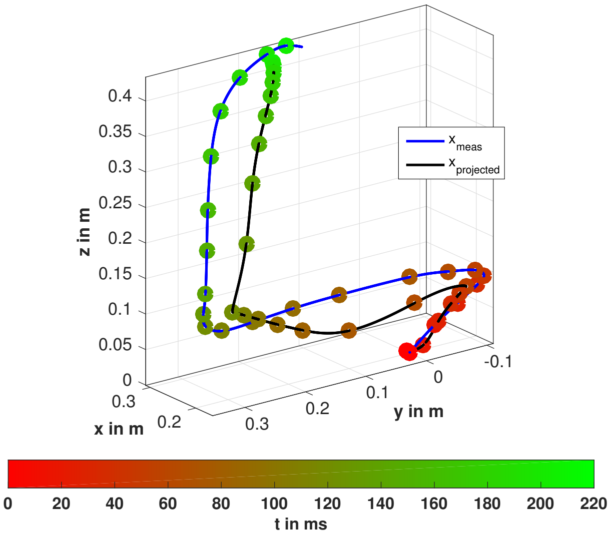
In the following section, the results of the motion projection are illustrated and compared to the human arm motion experiments, to evaluate the projection.



(a) Projection of the motion from experiment 1



(b) Projection of the motion from experiment 2



(c) Projection of the motion from experiment 3

**Figure 4.11:** Comparison of projected motions of the human model with the measurements from the experiments described in Fig. 4.3. Image ©IEEE

#### 4.2.3.5 Comparison to Real Human Arm Motions

To evaluate the presented motion projection method, the simulation results are compared to real human arm motions, which are recorded in Section 4.1.2.

In Figure 4.11, the comparison of motions is illustrated. In Figure (a) to (c), the same goal motions, i.e., virtually slapping the softball from the pillars, is simulated. The blue solid line depicts the measured human arm motions $x_{meas}$ in each plot. The black solid line depicts the projected motions $x_{projected}$ of the human arm in Cartesian space. The color scale illustrates the needed time with a sampling rate of $10$ ms. It can be seen that an approximation of human arm motion can be simulated using the human arm motion projection method. Smaller deviations are caused by the simplified human arm model and the adjustment of model parameters. Finally, the human arm motion projection enables a suitable approximation of human arm motions in order to estimate the duration to a collision of the human hand with the robot. This can be used to determine a maximum robot velocity, which is described in the following section.

### 4.2.4 Online Velocity Limitation

To enable the velocity limitation during the robot motion, the projection of human arm motions has to be executed within milliseconds. To ensure the minimal simulation time, a stop has to be triggered, when reaching the estimated collision position. The procedure of the calculation is described in the following:

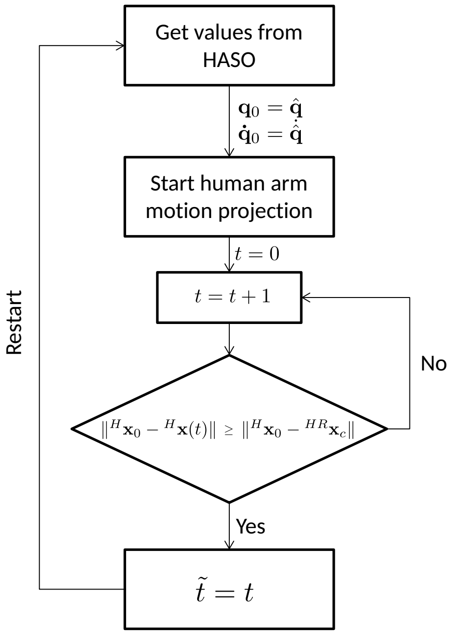


**Figure 4.12:** Block scheme human arm motion projection.

In Figure 4.12, the block diagram is illustrated, which depicts the process of the time calculation. After starting the projection, the time $\tilde{t}$ can be obtained with the stop criterion

$$\varsigma_c = \begin{cases} 0 & , \|{}^{H}\mathbf{x}_0 - {}^{H}\mathbf{x}(t)\| < \|{}^{H}\mathbf{x}_0 - {}^{HR}\mathbf{x}_c\| \\ 1 & , \|{}^{H}\mathbf{x}_0 - {}^{H}\mathbf{x}(t)\| \geq \|{}^{H}\mathbf{x}_0 - {}^{HR}\mathbf{x}_c\| \end{cases}, \tag{4.21}$$

where ${}^{H}\mathbf{x}_0 - {}^{H}\mathbf{x}(t)$ describes the total motion of the human arm in Cartesian coordinates, which was executed to time $t$ and ${}^{H}\mathbf{x}_0 - {}^{HR}\mathbf{x}_c$ describes to total distance between the initial human arm position to the estimated collision point ${}^{HR}\mathbf{x}_c$. It turned out that the motion of the human arm is not exact, and, therefore, the distances where chosen as stop criterion.

The final maximum velocity of the robot can be calculated as

$$ {}^{R}\dot{x}_{max} = {}^{R}\dot{x}_{ISO} + {}^{R}\ddot{x}_{max} * \tilde{t}, \tag{4.22} $$

where ${}^{R}\dot{x}_{ISO}$ describes the maximum allowed velocity in case of a collision, obtained by collision experiments with the evaluation of maximum force and pressure. The maximum deceleration ${}^{R}\ddot{x}_{max}$ of the robot can be taken from the robot specifications, as well as from breaking experiments with the robot.

In order to apply this methodology to the robot control, an online velocity adaptation is required. The velocities have to be adapted according to Equation (4.22), at any time $t$. To obtain this adaptation, in Chapter 3 online changeable velocity trajectory generation is described and used for the robot control.

In the next section, the robot motion is taken into account and the estimated time to goal of the robot.

### 4.2.5 Maximizing Velocity Limitations

Related to Case 3, as shown in Figure 4.1 in the beginning of this chapter, it has to be analyzed whether a collision can occur, when the robot moves at its maximum velocity. Therefore, the time of the robot needed to reach the goal position has to be calculated and compared to the time to collision a human would need. Imagine the robot is close to the goal and the human cannot reach the robot path anymore, such that no collision can occur, then, the limitation of the robot can be omitted and the robot velocity is only limited by its kinematic maximum, defined by the maximum robot velocity ${}^{R}\dot{x}_{rl}$.

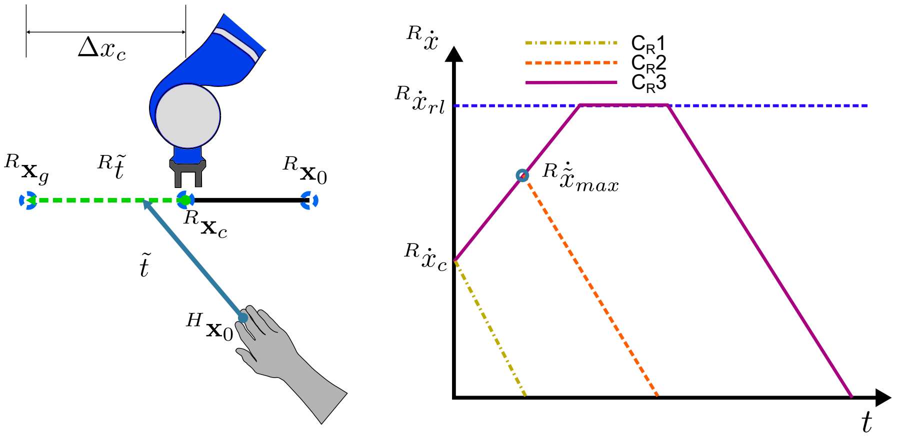


**Figure 4.13:** Robot motion time estimation.

In Figure 4.13, a robot motion is illustrated. The left plot shows the current hand position ${}^{H}\mathbf{x}_0$ and the current robot position ${}^{R}\mathbf{x}_c$. Obviously, when the robot is getting closer to the desired goal position, the time to finish the motion is decreasing, and an estimation of duration ${}^{R}\tilde{t}$ can be made. Three different possible actions can be executed:

**Case $\mathbf{C_R1}$**

The first case occurs, when the robot is close to the desired goal position, and in the deceleration phase to stop the robot. This can be calculated and examined by

$$\Delta x_c \leq \frac{3}{2}\frac{{}^{R}\dot{x}_c{}^2}{{}^{R}\ddot{x}_{max}}, \tag{4.23}$$

where the current robot velocity ${}^{R}\dot{x}_c$ needs to be decelerated to a robot stop. If this occurs, the trajectory generator is in the deceleration phase at the end of the motion. In this case, no limitation of the velocity is necessary.

**Case $\mathbf{C_R2}$**

In the second case, given the robot current velocity and the distance to goal, a time can be estimated that includes an acceleration phase to a certain velocity lower than the maximum velocity and the deceleration phase, to stop at the goal position. $\mathrm{C_R}2$ is assumed when

$$\Delta x_c \leq \frac{{}^{R}\dot{x}_{rl}^2 - \frac{1}{2}{}^{R}\dot{x}_c^2}{{}^{R}\ddot{x}_{max}} > \frac{3}{2}\frac{{}^{R}\dot{x}_c{}^2}{{}^{R}\ddot{x}_{max}}, \tag{4.24}$$

is fulfilled. The maximum reachable velocity can be calculated by

$${}^{R}\dot{\tilde{x}}_{max} = \sqrt{\Delta x_c {}^{R}\ddot{x}_{max} + \frac{1}{2}{}^{R}\dot{x}_c{}^2}, \tag{4.25}$$

which is necessary to obtain the minimal duration for reaching the goal position. The estimated time to goal of the robot can then be calculated by

$${}^{R}\tilde{t} = \frac{2}{{}^{R}\ddot{x}_{max}}{}^{R}\dot{\tilde{x}}_{max} + \frac{{}^{R}\dot{x}_c}{{}^{R}\ddot{x}_{max}}. \tag{4.26}$$

**Case $\mathbf{C_R3}$**

In the third case, the robot can accelerate to the maximum possible velocity, move at a constant velocity, and decelerate to the stop. Case 3 is fulfilled when

$$\Delta x_c > \frac{2{}^{R}\dot{x}_{rl}^2 - {}^{R}\dot{x}_c^2}{2{}^{R}\ddot{x}_{max}}, \tag{4.27}$$

then, the estimated time the robot needs to reach the goal is calculated by

$${}^{R}\tilde{t} = \frac{\Delta x_c}{{}^{R}\dot{x}_{rl}} + \frac{{}^{R}\dot{x}_0{}^2}{2{}^{R}\dot{x}_{rl}{}^{R}\ddot{x}_{max}} + \frac{{}^{R}\dot{x}_{rl} - {}^{R}\dot{x}_0}{{}^{R}\ddot{x}_{max}} \tag{4.28}$$

Now the estimated robot-time-to-goal can be calculated and the human-time-to-collision can be estimated, the final maximum applicable robot velocity can be determined by

$${}^{R}\dot{x}_f = \begin{cases} {}^{R}\dot{x}_{max} & , {}^{R}\tilde{t} \geq \tilde{t} \\ {}^{R}\dot{x}_{rl} & , {}^{R}\tilde{t} < \tilde{t} \end{cases} \tag{4.29}$$

In the following section, the human arm motion projection method is summarized and the idea discussed.

## 4.3 Summary

In this chapter, human arm motion dynamics are analyzed to enable an improvement of performance of the robot motions. Real human arm motions are experimentally executed. The experiments were chosen to include turning motions, such that indications to the dynamics of the human arm can be exploited. The goal of the experiments was to obtain fastest human arm motions in a common human position, as well as in collaborative applications, which is usually a sitting position in front of a workbench. In a second step, a simplified dynamic human arm model was developed with three DoF, to obtain a minimal computation time. Therefore, human arm kinematics where analyzed, as well as the inertia, which is finally mapped to the simplified 3-Dof model.

The final collaborative scenario contains only the measurement of the human hand, and excludes the determination of the elbow position of the human arm. Additionally, this is limited by the 3-DoF simplified model, where the elbow position is assigned to a certain position in Cartesian space. Therefore, a combined method is developed which enables the decoupling of the real human joint motions, which is shown in Figure 4.6.

To map the human arm motions to the simplified dynamic model, a human arm state observer (HASO) was developed. Here, the Cartesian hand position serves as input for the model and the observer or controller-like-scheme enables a mapping of the Cartesian positions to the representative joint angles of the simplified model. This is a necessary step, to obtain initial joint values for the projection. As the projection is executed with the same dynamic model, initial joint values are needed. Anyhow, in a next step human arm motion are simulated into the direction of the robot path. Here, a force vector is generated to a possible collision point, which spans a spring at the end point of the arm kinematics and generates desired positions in the joints. An admittance controller design is used because the motions of human arms are very fast and the representation of the human arm could be realized using the developed method. The result is a time $\tilde{t}$, which is a possible duration to a collision that can occur. However, the time-to-collision is the basis for further developments in this thesis, in order to enable safe and efficient robot motions.

Using dynamics for the representation of human motions enables a more precise representation of real human arm motions. Different configurations have different characteristics and lead to other motions and behavior of the arm, which can be represented by a dynamic model. In this thesis, a simplified model is used, which leads to an approximation and a suitable representation of the human arm. Nevertheless, simulating the human arm motions has a minor disadvantage, which is the computation time, where a kinematic consideration can be calculated analytically. Due to stability during the presented simulation the solver step should be at least 3 kHz. Additionally, the motion has to be simulated up to 140 times, in at least 2 ms, which necessitates a remarkable computation. Therefore, this chapter contains no experiments and for the following lookup tables or approximations of time needed to collision are used. For simplification, also a sampled workspace can be used for calculating the velocities, which also reduces the computation time drastically. However, an entire coverage of real projection can not be represented, because in a sampled look-up table the initial conditions are not treated, which could lead to the disadvantage that the sampled motions would be lower in time and the final robot velocity has to be slower. This could lead to a notable loss of performance.

In further research, the focus is on the minimization of computation time in order to develop real-time able dynamic models, to further improve the efficiency in collaborative applications.

# Chapter 5

# Performance Improvement by Optimization with Human-in-the-Loop Constraints

Robots usually move on a direct path, from an initial position, to a desired goal position. In shared workspaces, it is unavoidable that they pass the reachable area of the human co-worker within this motion. A reduction of the velocity, as presented in Chapter 4, based on human-in-the-loop considerations, is a necessary step in order to ensure safety for the human. Consequently, a motion on the direct path does not ensure the minimal duration of movement. Therefore, the challenge is to find an optimal path with the minimum time needed, to enable higher performance during the application.

In this chapter, a novel approach for path optimization is presented, extending optimal motion generation, as shown in Figure 5.1, which considers different types of motion optimization, with 1.) minimizing the error of the goal position $\vartheta_f$, considering the energy of the motion $E$, the joint torques $\tau_J$, and the time $t$ needed to execute the motion, 2.) reaching a maximum velocity $\dot{\vartheta}_{max}$, 3.) reaching the maximum velocity at an explicit target position, 4.) reaching a defined target motion (position and direction $\dot{z}$), 5.) minimum time for a breaking motion, 6.) following a path $\vartheta(t)$ and minimizing the error, 7.) minimizing the position error on a loop,

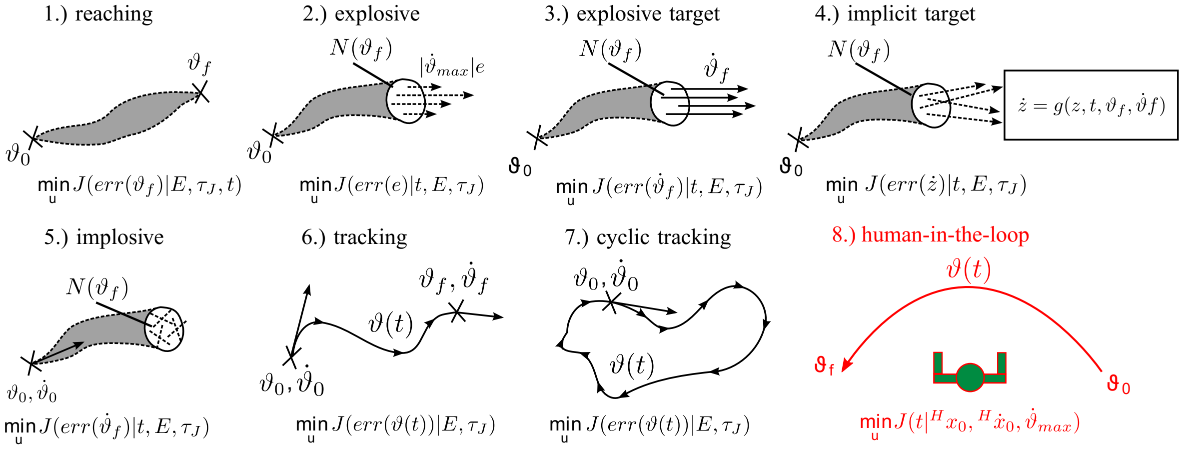


**Figure 5.1:** Robot motion types: for optimization the different robot motion types are presented in [97] that distinguish between different states, e.g. end states or initial states, regarding their cost function. This is extended with a new type of optimization, which is the main contribution of this thesis, industrial application optimization regarding efficiency with human-in-the-loop constraints.

8.) considering humans-in-the-loop with the current position ${}^{H}x_0$ and motion ${}^{H}\dot{x}_0$, minimizing the cycle time of the application. Therefore, human-in-the-loop constraints are described and used to find an optimal path to the desired goal position. This includes the reduction of velocity, depending on the distance of the human to the robot and total distance of the robot movement.

First, the approach is explained in more detail. Furthermore, motions that are on a direct path are analyzed, in order to generate fundamentals to enable a comparison of the optimized motions. Then, the nonlinear programming method, including human-in-the-loop constraints, is described. Finally, results are presented and discussed.

## 5.1 Approach

In a close collaboration of humans and robots, the path of the robot crosses the human reachability very often at hand. Therefore, assuming a direct path close to the human leads to a remarkable reduction of velocity. Consequently, the cycle times are significantly increasing, and the performance of the application is poor. To improve the performance, the optimization of the path could result in lower time to pass the shared work space.

Optimization methods usually encompass dynamics or torque constraints, as used in the optimal control community [98, 99, 100]. In this thesis, a novel approach is presented, which includes the projection of human arm motions as presented in Chapter 4. The gained safety considerations, to increase the robot velocity, are included into the cost function, as initially presented in [101].

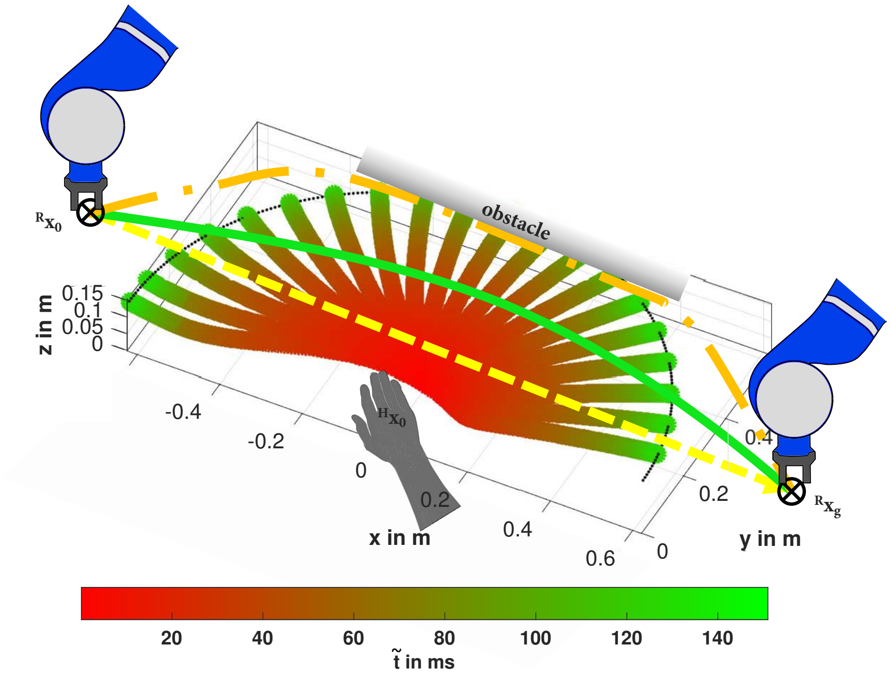


**Figure 5.2:** Path optimization approach with human-in-the-loop constraints. The red to green colored area depicts the duration of the human hand $\tilde{t}$ to this position, obtained by simulation of the simplified human arm model. Additionally, robot paths are depicted passing the human workspace. The yellow dashed line depicts a direct path from a starting position to the goal position, where usually the ISO velocities are used. The orange dashed-dotted line depicts a path for a possible very fast robot with spatial limitations, for example the own base. The green solid line depicts the optimal path considering the human and the possibility to get as fast as possible into this path. Here the best solution has to be found considering the length of the path and the duration of motion with adaptable velocities depending on the current human hand position. Image ©IEEE

In Figure 5.2, the approach to obtain a reduction of movement duration is illustrated. On the left side, the robot's initial position ${}^{R}x_0$ is depicted. The direct path to the goal position ${}^{R}x_g$, on the right side, is depicted as yellow dashed line. Obviously, the direct path leads to a strong decrease of the velocity, when taking the human hand position ${}^{H}x_0$ into account. The human agility is depicted as scale from red to green area, and illustrates the duration of motion to a grid in Cartesian space, as presented in Section 4.2.3.4. Additionally, the robot motions are often spatial limited, because they have a fixed base, as shown in Figure 2.11, fences limit the work space or the work space is additionally bounded by storages. Therefore, the path has to be adapted, which is depicted as orange dashed line.

The proposed approach divides the robot path into a various number $n$ of segments that are considered with the underlying constraints of the human agility. Each segment is analyzed regarding the distance to the human and the resulting time needed to reach each segment. Due to the non-linearity in the calculation of the human arm motions, a nonlinear programming algorithm is used to find optimal segments of the motion, with respect to minimize the time needed to reach the desired goal. Generally, the optimization of such motions can only be executed offline, since the computation power is not sufficient enough. Therefore, generalization algorithms have to be integrated to guarantee optimal motions also online, which are presented in the following chapter.

Collision avoidance methods usually react in the case of small distances to the human, without considering the overall situation from the beginning of the motion to end. The proposed method to optimize with human motion projections consists of three main parts:

1. generate optimal path with generalization possibilities offline
2. spatial online adaptation of path
3. online adaptation of velocity by allowing collisions after ISO standards

Common collision avoidance methods provide non-optimal path generation, as no cost function is considered during the motion. In this thesis, minimizing cycle times, considering worst case scenarios to ensure safety, is presented.

In the following section, the nonlinear programming algorithm used for the minimization of time is presented, as well as the analysis of the compared possible motions of the robot.

## 5.2 Nonlinear Programming for Efficient Robot Motions

From linear programming [102] to quadratic programming or nonlinear programming [103, 104, 105], respectively, different methods can be found [106] to solve an optimization problem. In this chapter, a nonlinear programming algorithm is chosen, to calculate the minimum duration of movement. As mentioned above, the projection of human arm motions is numerically obtained, and therefore a nonlinear optimization algorithm is necessary. First, a set of multiple auxiliary points defines the path from an initial position to a desired goal position. These auxiliary points define the parameters to be optimized in the nonlinear programming optimization, which is in particular a sequential quadratic programming procedure [107]. Human-in-the-loop information is used to find an optimum considering several constraints. Finally, the results are presented.

### 5.2.1 Robot Path Parameters

In order to find an optimal path for robots, including humans-in-the-loop constraints, requires the consideration of the human position and current motion. As the problem is defined, and the nonlinear programming solved offline, the robot path has to be verified in each path increment, to guarantee safety. Therefore, in this section the path parameters are chosen to be on a direct path, from an initial position, to a desired goal. The path consists of multiple auxiliary points, to obtain a segmented path for a closer examination of the constraints. Furthermore, a simple analytic path generation is desired, to ensure low computation time. This is also required, to enable online path examination in future research.

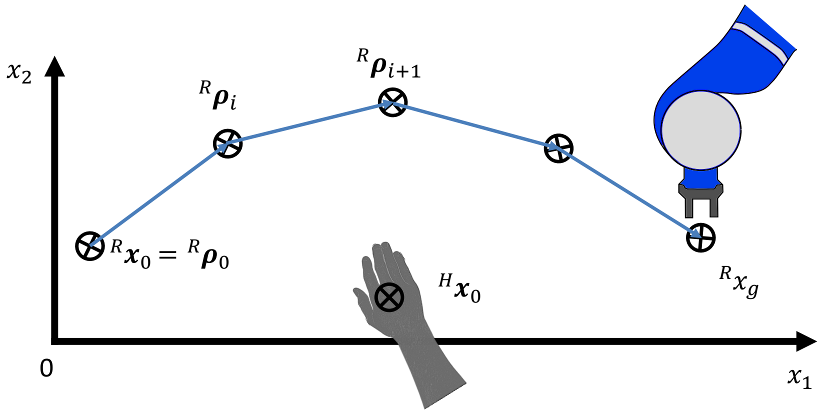


**Figure 5.3:** Robot path consisting of multiple segments with different allowed maximum velocities. The path consists of the starting position $^Rx_0$ and the goal position $^Rx_g$ with a defined number of auxiliary points $\rho_i$. Segmenting the path enables a flexible path creation to obtain optimal solutions with human-in-the-loop optimization. Image ©IEEE

In Figure 5.3, an exemplary path is depicted, which is an expected motion of the robot regarding the identified assembly applications, see Section 2.3. Multiple auxiliary points connected by lines define the entire robot path to a desired goal position. The division in segments enables an iterative calculation, to ideally solve the numeric nonlinear optimization problem.

Often direct robot motions are not applicable, which is due to possible collisions with objects in their environment. In particular, the path could directly pass a human body part, like for example the human hand position. The calculation of the segments, and their distance to the human body part, provides insight to the shortest possible motion of the human hand to the robot path. Each segment has to be considered including their boundaries, i.e., $\rho_i$ and $\rho_{i+1}$. Furthermore, the number of auxiliary points is initially not defined. Therefore, also the influence of number of auxiliary points is analyzed, too.

In Figure 5.4, two main cases are illustrated, that have to be differentiated. On the left side, the shortest distance $\Delta x_{min}$ to the segment is depicted. The point at $^Hx_0$ describes the current, or initial, human hand position on the workbench, respectively. The parameter $^R\rho_i$ and $^R\rho_{i+1}$ are the Cartesian positions of the beginning of each segment, as well as the end position. As the reachability of the human is limited in its kinematic constraints, the distance $\Delta x_{min}$ influences the time $\tilde{t}_min$ only if the segment is in the range of the human arm, i.e., the distance of the human base $^Hx_b$ is lower than $0.7$ m. In the other case, the time $\tilde{t}_{min}$ is infinite and the velocity of the robot is limited by its maximum.

To obtain the minimum distance to the segment, a line is generated through the auxiliary points by[1]

$$\mathrm{g} = \boldsymbol{\rho}_i + \lambda_g \Delta\boldsymbol{\rho}_i, \tag{5.1}$$

---

[1] for clarity the robot index is omitted: $^R\rho = \rho$.

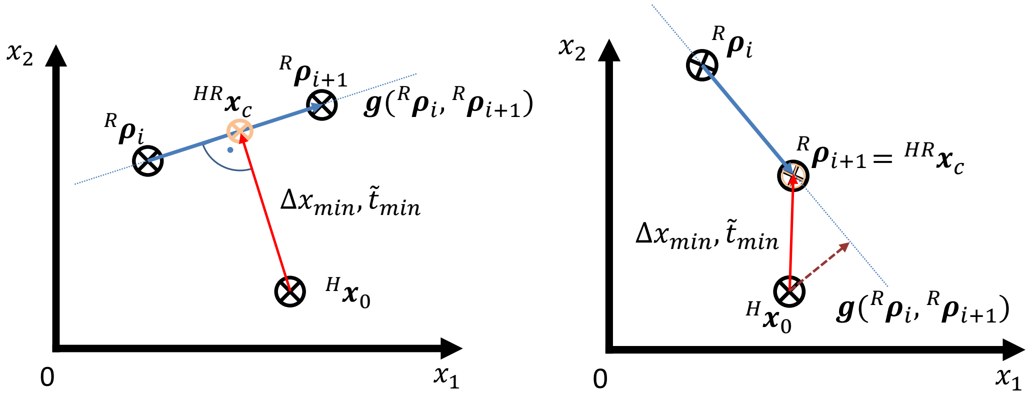


**Figure 5.4:** Schematic illustration of path segments and shortest time to path. The left plot illustrates the shorted distance from the human hand position to the line between the two auxiliary point, as the shortest distance to collision $^{HR}x_c$. As this collision point can be one of the auxiliary points, instead of being on the segment, in the right plot is illustrated that the shortest path has to be considered. Image ©IEEE

where $\Delta\boldsymbol{\rho}_i = \boldsymbol{\rho}_{i+1} - \boldsymbol{\rho}_i$ defines the direction vector of the line, $\lambda_g$ the line parameter, and $\boldsymbol{\rho}_i$ the position vector. To obtain the distance to the segment, an auxiliary plane is calculated, which is orthogonal to the line and goes through the human hand position $^{H}x_0$. The plane has to be converted, and the probable position of a collision $^{H}Rx_c$ has to be analytically solved.

Since lines have an infinite length, boundaries have to be considered, as illustrated in Figure 5.4 (right side). Depending on the orientation of the line, the distance to the line can be lower than the distance to the segment. Therefore, a differentiation has to be done to obtain the final distance to the segment. To find out whether the probable position of collision $^{H}Rx_c$ lies inbetween the two segments, the line parameter can be calculated as

$$\lambda_g(^{HR}\mathbf{x}_c) = \frac{^{HR}\mathbf{x}_c - \boldsymbol{\rho}_i}{\Delta\boldsymbol{\rho}_i}. \tag{5.2}$$

The final distance to the segment can now be calculated by

$$\Delta\mathbf{x}_{min} = \begin{cases} \|^{HR}\mathbf{x}_c - {}^{H}\mathbf{x}_0\| & , \ 0 < \lambda_g(^{HR}\mathbf{x}_c) < 1 \\ \|\boldsymbol{\rho}_i - {}^{H}\mathbf{x}_0\| & , \ \lambda_g(^{HR}\mathbf{x}_c) \leq 0 \\ \|\boldsymbol{\rho}_{i+1} - {}^{H}\mathbf{x}_0\| & , \ 0 < \lambda_g(^{HR}\mathbf{x}_c) \geq 1 \end{cases}, \tag{5.3}$$

where the final minimum distance is given by the path $\Delta\mathbf{x}_{min}(^{H}\mathbf{x}_0, {}^{HR}\mathbf{x}_c)$ from the human hand to the probable collision point. Alternatively, the distance is one of the auxiliary points, where $\Delta\mathbf{x}_{min}(^{H}\mathbf{x}_0, \boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1})$ is the lower value of the two distances. This applies, when the line parameter is lower 0 or greater 1.

The length of each segment is needed to calculate the time of motion and can be calculated by

$$s_i = \|\boldsymbol{\rho}_{i+1} - \boldsymbol{\rho}_i\| \tag{5.4}$$

The total length of the motion is the sum over all segments $s = \sum s_i$.

The resulting time of each segment can be calculated by

$$t_{s_i} = \frac{s_i}{{}^{R}\dot{x}_{max_i}}, \tag{5.5}$$

where the time is dependent on the maximum allowed velocity ${}^{R}\dot{x}_{max}$, which is obtained by Equation (4.22). The overall duration of motion can be calculated by

$$t_s = \sum_{1}^{n} t_{s_i}, \tag{5.6}$$

which is the sum of the individual segments.

In the next section, an approach is described to enable faster motions, with respect to the presented methodology of Section 4.1.

### 5.2.2 Sequential Quadratic Programming

The objective for the SQP is given by

$$y = \min_{\boldsymbol{\rho} \in \mathbb{R}^n} \Gamma(\boldsymbol{\rho}) \qquad \Gamma : \mathbb{R}^n \mapsto \mathbb{R}^m, \tag{5.7}$$

where $\Gamma(\boldsymbol{\rho})$ describes the cost function to be minimized, given a set of variable parameters $\boldsymbol{\rho}$ [108]. The cost function can represent an energy minimization, time to goal minimization, maximizing the speed and much more. The optimization problem is constrained by physical constraints as geometric limitations or maximum energy the system can afford. These limitations can be described as *inequality constraints* by

$$c_i(\boldsymbol{\rho}) \leq 0 \quad \text{with} \quad i = 1, ...m_c, \tag{5.8}$$

where $m_c$ describes the number of inequality constraints. These can be for example geometric limitations as the reachability of the robot, which means that a solution out of the range is not suitable. Another type of constraints are the *equality constraints*, as for example a position shall be reached. The equality constraints are given by

$$h_i(\boldsymbol{\rho}) = 0 \quad \text{with} \quad i = 1, ...m_h, \tag{5.9}$$

where $m_h$ denotes the number of all equality constraints. The overall Lagrange function

$$L(\boldsymbol{\rho}, \lambda, \mu) = \Gamma(\boldsymbol{\rho}) + \sum_{i=1}^{m} \lambda_i c_i(\boldsymbol{\rho}) + \sum_{i=1}^{q} \mu_i h_i(\boldsymbol{\rho}), \tag{5.10}$$

with the Lagrange parameter $\mu > 0$ and $\lambda > 0$ to activate the consideration of constraints. For optimizing the nonlinear optimization problem with human-in-the-loop constraints, solving a sequential quadratic programming (SQP) problem is applicable. For the optimization algorithm, the function is interpolated and the gradient calculated, which presupposes that the gradient vanishes for an optimal solution. The component-wise annotation is

$$\nabla L = \begin{pmatrix} \nabla_{\boldsymbol{\rho}} L \\ \nabla_{\lambda} L \\ \nabla_{\mu} L \end{pmatrix} = \begin{pmatrix} \nabla\Gamma(\boldsymbol{\rho}) + \sum\limits_{i=1}^{m} \lambda_i \nabla c_i(\boldsymbol{\rho}) + \sum\limits_{i=1}^{q} \mu_i \nabla h_i(\boldsymbol{\rho}) \\ \mathbf{c}(\boldsymbol{\rho}) \\ \mathbf{h}^{\mathrm{a}}(\boldsymbol{\rho}) \end{pmatrix} = \mathbf{0} \tag{5.11}$$

which is repeatedly executed for different parameter sets. The initial states serve for the Hessian matrices. Calculating the gradient, and, therefore, to find a local minimum, the

Taylor series can be used with integrating Equation (5.11) in

$$\min \frac{1}{2}\Delta\boldsymbol{\rho}^T \nabla^2_{\boldsymbol{\rho\rho}} L(\boldsymbol{\rho}^{(k)}, \boldsymbol{\lambda}^{(k)}, \boldsymbol{\mu}^{(k)})\Delta\boldsymbol{\rho} + \nabla\Gamma(\boldsymbol{\rho}^{(k)})^T\Delta\boldsymbol{\rho}, \tag{5.12}$$

where $k$ is the number of the iteration step.

To find an optimal path with human-in-the-loop constraints, initial values have to be found and the path has to be defined. In the following section, the path parameters are described, which serve as input for solving the optimization problem. As the algorithm should have a quadratic convergence, the SQP algorithm is used in the following, which is implemented in the nonlinear optimization of fmincon in MATLAB methods.

### 5.2.3 Optimizing with Safety Constraints

To obtain low production costs, the cycle times are a decisive factor to reduce personnel costs in the manufacturing process. Since current industrial collaborative applications do not achieve sufficient efficiency, and consequently lead to relatively high cycle times, compared to human productivity, the cost function is defined as the duration of motion:

$$\Gamma(\rho) = \sum_{i=1}^{n} t_{s_i}({}^{R}\dot{x}_{max_i}, s_i), \tag{5.13}$$

The sum of duration for each segment defines the minimum time to reach the desired goal position, where ${}^{R}\dot{x}_{max_i}$ can be calculated by Equation (4.22) considering the estimated time-to-collision $\tilde{t}$ obtained from human arm motion experiments. Approximately, this estimated time can be calculated by lookup tables from Chapter 4 or alternatively by assuming the maximum possible human arm velocity to define $\tilde{t}$. This finally reduces the computation time remarkably, and additionally includes the shortest possible time to collision. This implies that the initial human arm velocity is at its maximum and points into the direction of the robot path.

In collaborative applications, numerous limitation have to be considered, especially with a humans-in-the-loop. Equality constraints are not considered in this case, because the main equality constraints are defined by the start and end position, and they are fixed values in the optimization algorithm. However, to comply with the limitations, different inequality constraints are defined for the optimization process:

$$0 \leq {}^{R}\dot{x} \leq {}^{R}\dot{x}_l, \tag{5.14}$$

$${}^{R}\boldsymbol{x}_{min} \leq {}^{R}\boldsymbol{x} \leq {}^{R}\boldsymbol{x}_{max}, \tag{5.15}$$

$${}^{R}\boldsymbol{x} \geq {}^{H}\boldsymbol{x}_0. \tag{5.16}$$

Obviously, the maximum robot velocity ${}^{R}\dot{x}$ is limited by its physical constraints ${}^{R}\dot{x}_l$ of the overall robotic system. Additionally, the capability of manipulating is regional limited, and the Cartesian constraints with ${}^{R}\boldsymbol{x}_{min}$ and ${}^{R}\boldsymbol{x}_{max}$ have to be considered[2]. As, human-in-the-loop constraints in this thesis are basically defined as time-to-collision, which affects the cost and final motion, also spatial human constraints have to be treated. Using the presented method above, and the limitation to only measure the human hand position, does not exclude the human body. Assuming the human sitting in front of the workbench, the robot path shall not pass the region between the human hand position and the front of the workbench, which excludes a collision with the human body.

---

[2] Note that also the dynamics of the robot can be considered entirely, but here the focus is on safety, depending on the human capability simulation

For the nonlinear optimization process, initial values for each parameter have to be defined. To find a global minimum, usually concatenated, or random initial values, are used [104, 109], and the optimization process is executed multiple times. This prevents from getting stuck in a local minimum. For defining the initial values of a robot path, the direct line to the goal position is used. In particular, a certain number of auxiliary points is placed with a defined distance on the segment. They can finally be calculated by

$$\boldsymbol{\rho}_{ini_i} = {}^R\boldsymbol{x}_0 + \frac{i}{n}({}^R\boldsymbol{x}_g - {}^R\boldsymbol{x}_0) \ , \quad i = 1, 2, ...n-1. \tag{5.17}$$

Furthermore, the physical limits of the robot have to be considered. In particular, the possibility to reach a velocity at the beginning and in the end of the motion, caused occasionally minor problems.

Given the auxiliary parameter $\rho_1$ and $\rho_{n-1}$, and the maximum robot acceleration, the maximum possible velocity starting from a rest position, or getting into a rest position, can be calculated as

$${}^R\dot{\tilde{x}}_{max_{0,n}} = \sqrt{2\,{}^R\ddot{x}_{max} s_{0,n}} \tag{5.18}$$

for the first and the last segment, respectively. Finally, the resulting robot velocity can be determined by

$${}^R\dot{x}_{0,n} = min({}^R\dot{x}_l, {}^R\dot{x}_{max_i}, {}^R\dot{\tilde{x}}_{max_{0,n}}). \tag{5.19}$$

In the next section, the results from solving the nonlinear programming problem are presented and analyzed.

### 5.2.4 Results

For the optimization routine the numerical solver *fmincon* integrated in MATLAB is used to calculate the time-optimal motion with human-in-the-loop constraints. The human hand position is assumed as fixed value located at ${}^H\boldsymbol{x}_0 = [0.0 \ 0.05]^T$, which is a usual hand position in the work space, regarding the collaborative octagon workbench. The initial robot position is given by ${}^R\boldsymbol{x}_0 = [-0.4 \ 0.15]^T$, the destination is located at ${}^R\boldsymbol{x}_g = [0.4 \ 0.15]^T$, and the motion shall be in a constant height over the workbench with ${}^Rz = 0.1$ m. The scenario is chosen to be a motion inside the shared work space, from one side to the other side, and in front of the human with a total distance of $0.8\,$m.

In Figure 5.5, three different cases C1-C3 are illustrated. Since collision velocities in collaborative applications are determined by collision experiments, there are no explicitly defined velocity values. For example, the motion with a sharp knife has a $\mathrm{v}_{ISO} = 0\,$m/s and collisions with a cushion can enable impact velocities up to $\mathrm{v}_{ISO} = 1\,$m/s, see Chapter 7. Hence, for analyzing the results, three different cases are compared. The first case is depicted as blue dashed line with a maximum velocity of $\dot{x}_l = 1.5\,$m/s and an impact velocity of $\mathrm{v}_{ISO} = 0.33\,$m/s. The black dots illustrate the resulting auxiliary point on each path. For the second and third case, the impact velocities are chosen to be $\mathrm{v}_{ISO} = 0.66\,$m/s and $\mathrm{v}_{ISO} = 1\,$m/s, respectively.

Obviously, for every motion with $\mathrm{v}_{ISO} < \dot{x}_l$ the path distinguishes from the direct motion and executes a curved motion to keep distance to the human. If the allowed impact velocity converges to the maximum robot velocity, the distance to the human vanishes. In Figure 5.5, bottom plot, the related velocities are illustrated, where the steps are the resulting velocities for each segment. As can be seen, for a high value of $\mathrm{v}_{ISO}$ the maximum robot velocity can be reached also within the shared workspace. However, this is also depending on the maximum possible acceleration. For a low value of $\mathrm{v}_{ISO}$, the velocity during the motion can be improved significantly. It is finally shown that solving the nonlinear programming problem leads to an

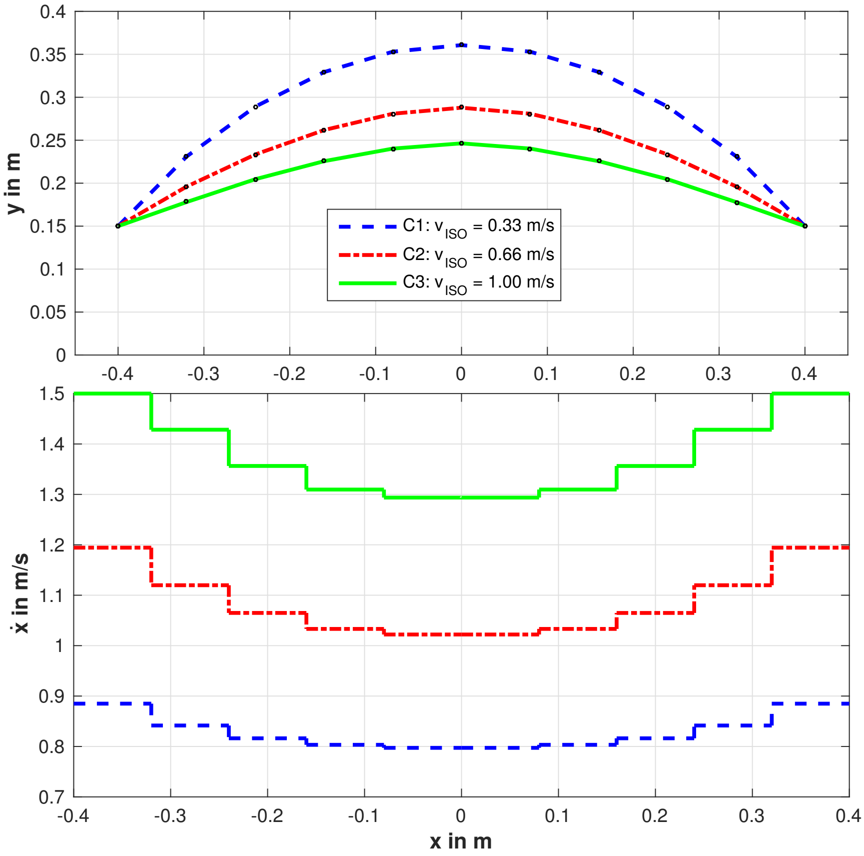


**Figure 5.5:** Optimization results illustrating the final path and the corresponding velocity. Three different cases C1 - C3 are considered, where different ISO velocities are defined. In the first case C1 the maximum allowed velocity is given by 0.33 m/s, C2 by 0.66 m/s and C3 1 m/s, which are possible collision velocities in real scenarios. The upper plot illustrates the resulting path in the x-y plane, which represents the surface of a workbench. The bottom plot illustrates the allowed maximum velocities depending on the position of the robot using the presented method with adaptable robot velocities. Image ©IEEE

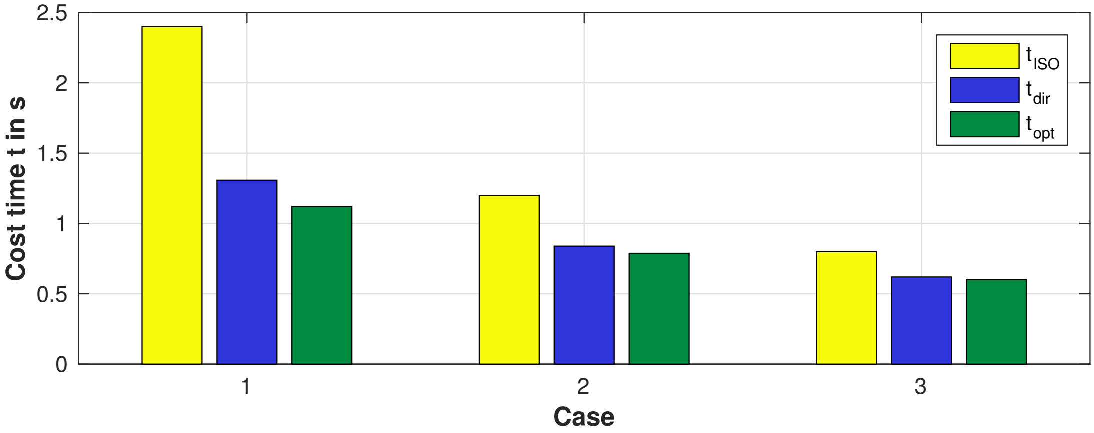


**Figure 5.6:** Cost Results obtained from solving the optimization problem. In this bar plot, the resulting cost values for each case are illustrated and compared. The yellow bar depicts the straight and direct motion from the starting position to the goal position and the duration which is needed. The blue bar depicts the cost value for the direct motion but with adaptable velocity as presented in this thesis. The green bar depicts the final duration solving the optimization problem. As proposed above this includes the minimal duration on a path which surrounds the human position.

adaptation of the path, with a curved behavior around the human hand position, and the

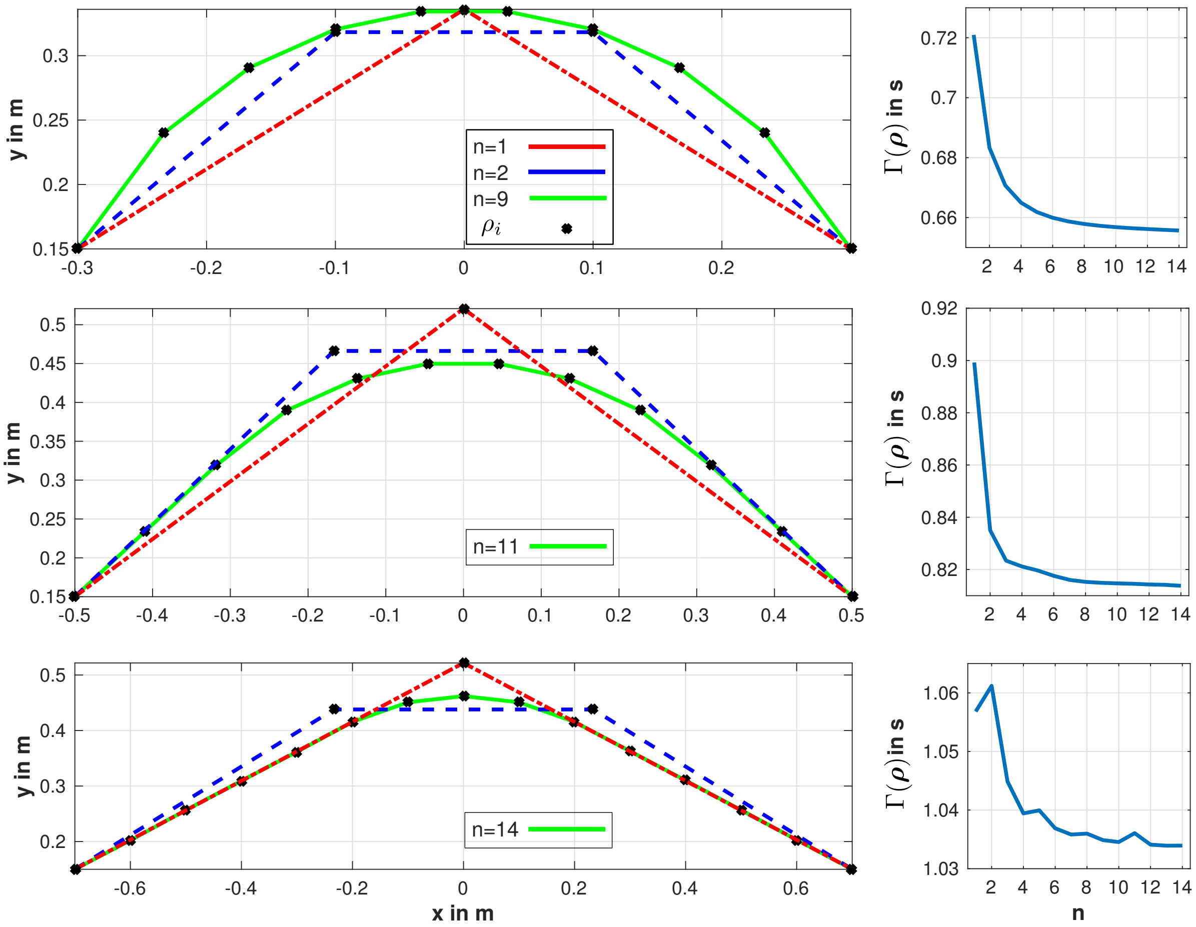


**Figure 5.7:** In this figure the dependency on the number of auxiliary points is depicted. Three different distances are considered to analyze the effect of number of auxiliary points. In the left column the paths are illustrated for the different case. The upper left plot depicts a motion of 0.6 m of distance with one, two and nine auxiliary points, the left middle plot up to eleven and the bottom left plot up to fourteen auxiliary points. In the right column the corresponding cost value is depicted depending on the numbers of auxiliary points. For the long distances the difference of cost value vanishes but on short paths inside the shared workspace and improvement of performance can be reached. Image ©IEEE

resulting velocity differs within the motion.

In Figure 5.6, a comparison of different motions regarding the three cases is illustrated. Each bar depicts the cost for each motion. The yellow bar illustrates the direct motion with the constant velocity of $\mathrm{v}_{ISO}$, the blue bar shows the duration of movement for the direct motion with a varied velocity, presented in Equation (4.22), and the green bar depicts the optimal motion results from above. It can be observed that for low values of $\mathrm{v}_{ISO}$ the difference between each duration of movement is significant. The direct motion with a constant velocity has a duration of motion $t_{ISO} = 2.4\,\mathrm{s}$, the method used in Chapter 4 leads to a duration of movement with $t_{dir} = 1.3077\,\mathrm{s}$ and the cost of the solved optimization $t_{opt} = 1.1211\,\mathrm{s}$. Finally, the improvement of performance is more than twice in comparison to a motion at a constant ISO-defined velocity.

To find out the necessity of a deep incremented trajectory, in a second optimization analysis the number of segments is varied. Three different paths are calculated with a total motion distance of $0.6$ m, $1$ m, and $1.4$ m. In every case, the number auxiliary points is increased and the resulting cost minimized, using the optimization routine. Obviously, the number of segments affects the path and the duration of motion, respectively.

In Figure 5.7, the results are illustrated for one and two auxiliary points for every motion. And alternatively the path for nine, eleven, and 14 auxiliary points is shown. Executing a motion with a shorter distance within the shared workspace, affects the spatial change significantly, when raising the number of auxiliary points. The path keeps distance in a balanced way, and the duration of movement can be reduced, in this case up to 0.08 s. Regarding a longer distance the differences between one and fourteen auxiliary points vanish. In total, a number of four to 8 via-points seems to be a suitable solution to describe the optimal path.

In the following section, the presented method is summarized, concluded, and the results are discussed.

## 5.3 Conclusion

In this chapter, a novel optimization method is presented which includes human-in-the-loop constraints inspired by Chapter 4, where the duration to the most possible collision is estimated and included in the calculation of the path. In a first step, a path of multiple lines is created and the minimum distance to the human is calculated, which usually describes the shortest duration to collision between the human hand and the robot end-effector. The presented method is limited to the human hand, but can also be transferred to the dynamics of the whole body of the human. This means, also head or body motions can be included using this methodology. For simplification, each human part could be treated separately, using a simplified dynamic model, because an entire model would probably lead to enormous computation times.

The obtained results are sufficient and increase the performance of the moving robot. In comparison to the previous defined velocity profiles, regulated by ISO, the optimization enables a significant reduction of cycle times. The difference between the optimal motion and the direct motion is higher, when lower velocities have to be applied on direct motions[3]. However, when the difference between the kinematic maximum and the allowed robot velocity vanishes, the path converges to the direct path. As in real industrial robot applications, usually the allowed maximum velocity is extremely low, e.g., 0.15 m/s and the maximum robot velocity much higher, e.g. 2 m/s, the difference will be remarkable and the cycle times noticeable reduced. The challenge now is to convert these motion into adaptable trajectory generation algorithm to enable a flexible motion planning within collaborative industrial application, to further improve the efficiency.

Additionally, another effect can be observed. As the optimal path surrounds the human worker, ergonomic advantages in a collaborative application arise. The human worker is not affected by the closely moving robot, which also can lead to a mental improvement of the work. In future work, the psychological improvement of the working conditions can also be analyzed which was previously done in collborative robotics research [38, 110, 111].

---

[3] ISO velocities are the maximum allowed velocities obtained from collision experiments.

# Chapter 6

# Generalization of Optimal Motions

A numerical optimization of motions can usually only be calculated offline, which requires numerous numerical operations that increase the computation time. Nevertheless, to enable flexible use of optimal motions, a suitable solution to generalize the motions has to be found. Different approaches exist, e.g., the whole paths recorded, or generated, for different situations and nearest neighbor interpolations allow for path adaptation, in terms of the desired conditions [112]. The generalization of motions is a necessary step in optimal online robot motion generation. Especially in human-robot collaboration, the flexibility is of major significance. For example, in a standard pick and place application, components are usually placed in patterns, and often have to be transported to different positions.

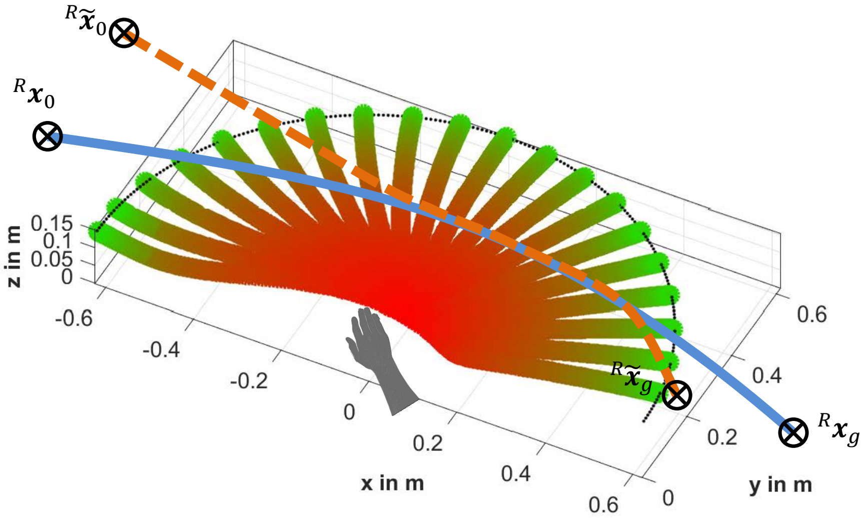


**Figure 6.1:** Generalization of motions for flexible robot applications. The original and optimal motion is depicted as blue solid line. The initial position $^R\mathbf{x}_0$ is changed to a new initial position $^R\tilde{\mathbf{x}}_0$ and the original goal position $^R\mathbf{x}_g$ is changed to $^R\tilde{\mathbf{x}}_g$. The resulting generalized motion is depicted as orange dashed line. Within the generalized motion the human-in-the-loop constraint have to be considered in order to ensure safety for the human co-worker. Image ©IEEE

An example of a generalized optimal motion is illustrated in Figure 6.1, which is extended with a major contribution of this thesis, i.e., the improvement of performance with human-in-the-loop constraints. The optimized motion, with a starting position at $^Rx_0$ and a desired goal position $^Rx_g$, differing from the original initial and goal position, are shown. The blue solid line depicts the optimal motion, passing the shared workspace. As mentioned above, the characteristics of being optimal shall be kept, but goal and initial position shall be adaptable within the generalized path, which is depicted as dark orange dashed line. Furthermore, the initial position ${}^{R}\tilde{x}_0$ and the desired goal position ${}^{R}\tilde{x}_g$ are changed.

A suitable solution is presented in [97], where robot motions are optimized and generalized by dynamic movement primitives. In this section, the generalization by using movement primitives is presented and analyzed for the case of a static human. In the first step, a fully described definition of a trajectory is required. Therefore, a time-minimal trajectory generation is presented using exponential functions. Then, general movement primitives are explained, and the implementation of the optimized and time-minimal trajectories explained. Furthermore, enabling the compliance of constraints is presented, which is a necessary condition in human-robot collaboration. Finally, the approach is evaluated with experiments on a real robot and the results are discussed.

## 6.1 Exponential Trajectory Generation for Generalizing Movements

To define a trajectory, positions, velocities, and accelerations have to be generated from the desired path information. These path information are defined by an initial position, a goal position and multiple auxiliary points in Cartesian space. These are, for example, obtained from the optimization as presented in Chapter 5.

Often, trajectories are generated by using polynomials like cubic splines, or B-Splines to generate velocities and accelerations [113, 114, 115, 116]. However, they are generally not suitable for time-minimal trajectory generation which is exemplary explained below. In order to obtain a smooth time-optimal behavior, a trapezoidal velocity and acceleration is required. Anyhow, the use of exponential functions for generating trajectories enables trapezoidal profiles also in the acceleration. Hence, the resulting velocities possess the same trapezoidal behavior, which has the characteristic of being time-optimal [117]. Additionally, continuity is given at least up to the third derivative, which enables smooth jerk behavior and, therefore, a smooth robot motion behavior.

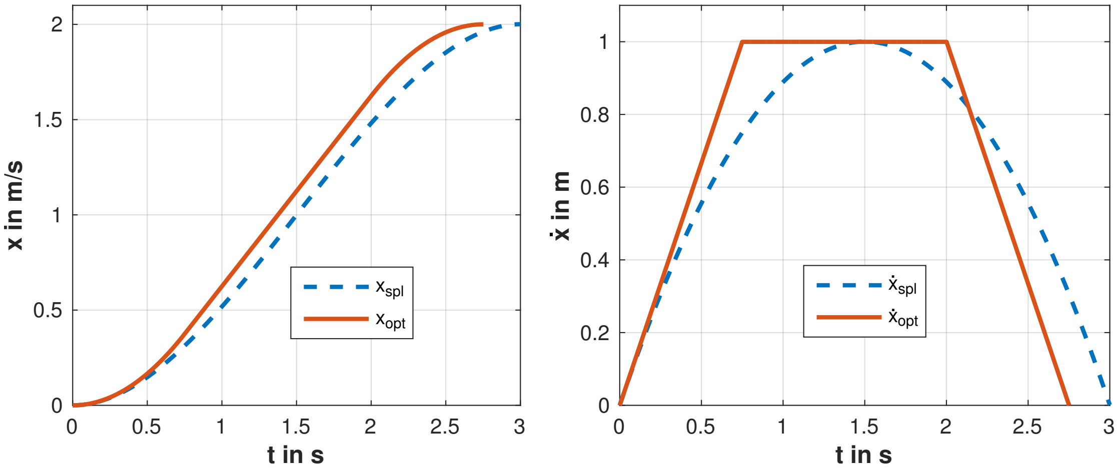


**Figure 6.2:** Comparison between the trajectory generated by a cubic spline and the time-minimal trajectory.

In Figure 6.2, a comparison of two different trajectory generation methods is illustrated. The blue dashed line depicts a cubic polynomial generated motion, as proposed in [118], with $\dot{x}(t) = 1.33t - 0.44t^2$ in the right plot. and $x(t) = \int \dot{x}(t)dt$ in the left plot. The orange solid line depicts the trapezoidal trajectory profile, generated similar to a bang-bang motion, with a maximum desired acceleration of $\ddot{x} = 1.33m/s^2$ given. It can be observed that the constant maximum acceleration behavior leads to a significantly faster motion. In this example, the time saving is 0.25 seconds, which is a reduction of motion duration by 8.33 %.

The trajectory generation, presented in this thesis, is inspired by [117]. Unlike the method presented there, the exponential function defines the acceleration profile instead of the velocity profile. The acceleration is calculated by

$$^{R}f_{\ddot{x}}(t) = {^{R}\ddot{x}_{max}} - {^{R}\ddot{x}_{max}} e^{-\eta * t^3}, \tag{6.1}$$

where $^{R}\ddot{x}_{max}$ represents the maximum desired acceleration of the robot, which is usually bounded by its kinematic constraints. The slope parameter $\eta$ allows for smoothing the jerk, but does not represent a jerk determination parameter. The defined trajectory profile is calculated numerically by

$$^{R}f_{x}(t) \;=\; \int_{0}^{t_f} {^{R}f_{\dot{x}}(t)}\mathrm{d}t \;=\; \int_{0}^{t_f}\int_{0}^{\bar{t}} {^{R}f_{\ddot{x}}(t)}\mathrm{d}t\mathrm{d}\bar{t}. \tag{6.2}$$

with $^{R}f_{x}(t)$ representing the path profile starting at the initial position $^{R}x_0$, and $^{R}f_{\dot{x}}(t)$ being the velocity, with an initial velocity $^{R}\dot{x}_0$.

The algorithm to calculate path parameters, presented in Chapter 5, requires multiple segments using auxiliary points. Each segment is defined by three main parameters:

1. initial position $^{R}\rho_i$
2. end position $^{R}\rho_{i+1}$
3. maximum velocity $^{R}\dot{x}_{max}$

From these definitions additional constraints can be defined:

4. $^{R}\dot{x}_{end} = {^{R}\dot{x}_0} = 0$
5. $^{R}\dot{x}_{max} > 0, {^{R}\ddot{x}_{max}} > 0$
6. $^{R}\dot{x}_{i_{end}} = {^{R}\dot{x}_{i+1_{max}}}$
7. $if\ {^{R}\dot{x}_{i_{end}}} > {^{R}\dot{x}_{i_{max}}}\ ,\ then\ {^{R}\dot{x}_{i_{end}}} = {^{R}\dot{x}_{i_{max}}}$

which are derived from the parameter definitions 1.-3.

An example is the end velocity of each segment. The end velocity has to be considered, otherwise the maximum velocity would be violated in the following segment, due to the fact that the acceleration can not be infinite. Therefore, the end velocity of each segment is adapted to the following segment. Additional constraints are that the maximum velocity or acceleration cannot be zero, which would lead to an infinite duration of motion and that the beginning and the end velocity of the overall via-point trajectory are at a rest position.

Algorithm 1 describes the procedure to calculate the entire trajectory. Sequentially, the seven sub-segments are calculated and concatenated to a trajectory from an initial position to a goal position. The constraint mentioned above have to be considered. If the generated trajectory exceeds the constraints, the maximum trajectory parameter, e.g. robot acceleration, are decreased. This procedure is iterated in order to obtain the whole motion.

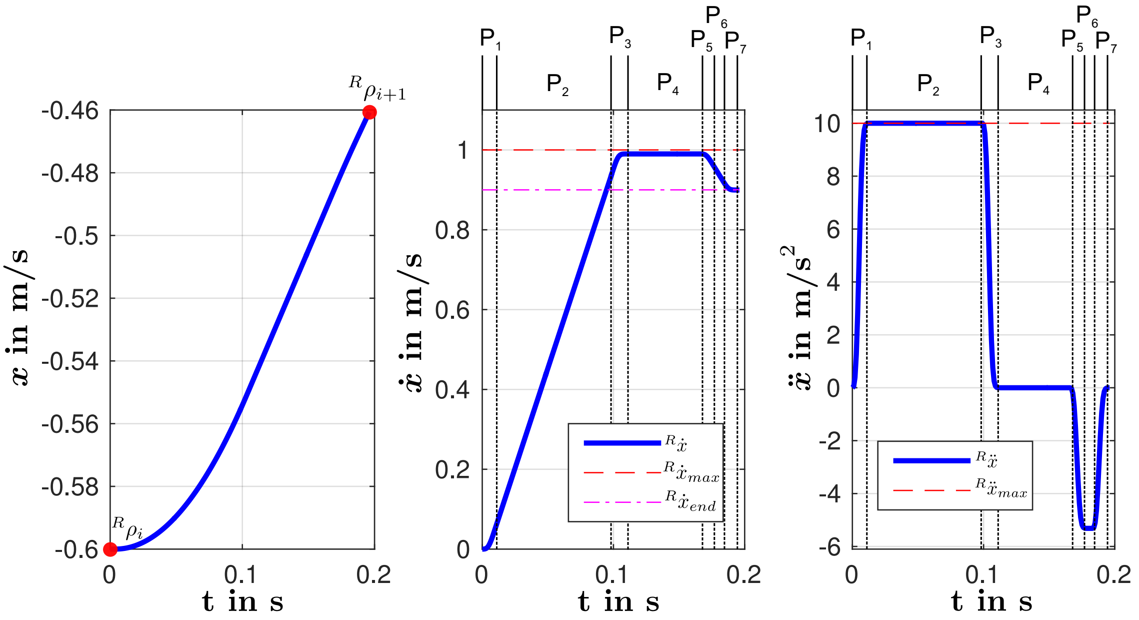


**Figure 6.3:** Example spline for via-point motions for one segment. The left plot depicts the position of the trajectory, the middle plot depicts the velocity and the right plot the acceleration. The trajectory generation is separated in seven segments $P_1 - P_7$. The desired end velocity is determined by the maximum velocity of the following segment. Image ©IEEE

In Figure 6.3 one part of the whole motion is illustrated and represents the trajectory for one segment. It is shown, that the trajectory is accelerating the motion until the maximum desired velocity is reached, depicted as sub-segment $P_1$-$P_3$. Then, this velocity is kept constant, sub-segment $P_4$, and the needed time or path to reach the desired end velocity is calculated, depicted as $P_5$-$P_7$. The composition of the individual segments yields to the final trajectory. In Figure 6.4, an entire path obtained from the optimization in Chapter 5 is transformed into

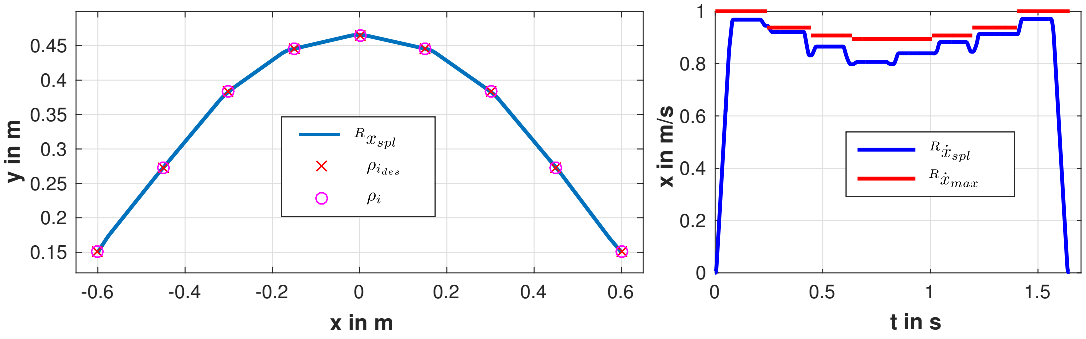


**Figure 6.4:** Example spline for via-point motions for one segment.

a trajectory. In the left plot, the Cartesian positions in the x-y-plane are illustrated, where the blue solid line depicts the incremental path positions, the red cross depicts the desired auxiliary point positions and the magenta circle the resulting endpoints of each segment, obtained from the exponential trajectory generation. It can be observed that the position error is at most zero and the desired goals are reached at a very small tolerance. In the right plot, the maximum desired velocities are depicted as red line segments and the resulting velocities are

**Algorithm 1** Calculate exponential trajectory

```
Require: ${}^R\ddot{x}_{max}$, ${}^R\dot{x}_{max}$, ${}^R\dot{x}_0$ , ${}^R\dot{x}_{end}$ , ${}^Rx_0$, ${}^Rx_g$
Ensure: ${}^Rf_{x(t_n)} \approx {}^Rx_g$, ${}^Rf_{\ddot{x}_{max}} \leq {}^R\ddot{x}_{max}$, ${}^Rf_{\dot{x}_{max}} \leq {}^R\dot{x}_{max}$
for P $\leftarrow$ 1 to 7 do
    while True do
        while ${}^Rf_{\ddot{x}_{max}} \leq {}^R\ddot{x}_{max}$ do
            ${}^Rf_{\ddot{x}(t_i)} = {}^R\ddot{x}_{max} - {}^R\ddot{x}_{max}e^{-\eta * t_i^3}$
            ${}^Rf_{\dot{x}(t_i)} = {}^Rf_{\ddot{x}(t_i)}\Delta t + {}^Rf_{\dot{x}(t_{i-1})}$
            ${}^Rf_{x(t_i)} = {}^Rf_{\dot{x}(t_i)}\Delta t + {}^Rf_{x(t_{i-1})}$
            if ${}^R\ddot{x}(t_i) > {}^R\ddot{x}_{max}$ then
                break
            end if
        end while
        if conditions & constraints ok then
            break
        else
            ${}^R\ddot{x}_{max} \searrow 0$
        end if
    end while
end for
```

depicted as blue solid line. It can be observed, that the velocities do not exactly fit to the maximum velocity because the combination of multiple trajectories generates various challenges, e.g., reaching the position at the same time. Other methods compensate these problems by deviating the positions, otherwise the desired velocity profile is not achievable. Here, only the limits are not violated, which does not lead to time-optimality. However, these problems are due to the limitation in trapezoidal acceleration profiles. A combination of different methods can lead to better results, and is part of future research.

In the next section, the generation of generalized trajectories is presented, which enables online motion generation and adaptation methods.

## 6.2 Dynamic Movement Primitives

The imitation of human motions is a suitable way to obtain robot motions for certain applications. For example, in table tennis, tennis, or golf, motions can be mapped to a robot motion [119, 120]. This comes initially from the learning-by-demonstration sector. The hand guided, or via sensor suit demonstrated, motions are mapped to a primitive motion generator. Dynamic Movement Primitives (DMP) are a promising approach for the generalizing of motions, and, in particular, optimal motions, to improve the performance of the robot system [97]. In the following, a short overview is given for DMPs and the generation of optimal motions. This is the basis of a high productivity in a manufacturing environment, and builds the basis for optimal safe motions, with a human in the work space.

The general formulation of dynamic movement primitives is given by an asymptotically stable second order differential equation [121, 122] as

$$\begin{aligned} \tau\dot{\boldsymbol{\vartheta}} &= K(\mathbf{g_d} - \mathbf{x}) - D\boldsymbol{\vartheta} + K\boldsymbol{f}(s) \\ \tau\dot{\mathbf{x}} &= \boldsymbol{\vartheta} \end{aligned} \tag{6.3}$$

where $\tau \in \mathbb{R}$ denotes the time factor, $K \in \mathbb{R}^{m \times m}$ a diagonal potential matrix, $D \in \mathbb{R}^{m \times m}$ a damping matrix. $\mathbf{g_d}$ is the desired goal position, which is usually treated as constant goal

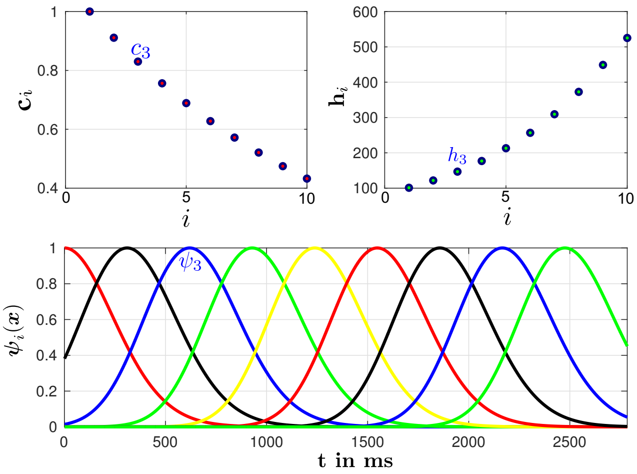


**Figure 6.5:** An example set of kernel functions with corresponding center points and widths of each kernel.

position before executing the motion. In this chapter, the goal constant is extended to a time varying variable, presented in Equation (6.8). $\boldsymbol{f}(s)$ is a time dependent force function with $s$ being a first order canonical system, which is given by

$$\tau \dot{s} = -\alpha_s s, \tag{6.4}$$

starting at one and approaching zero over time. The force term is usually learned from human demonstration. The recorded motion is represented by a set of sample points $S_d : \{[\gamma \; \dot{\gamma} \; \ddot{\gamma}]\}$, including position, velocity and acceleration data of the path. This data serves as input, to calculate the force function by inverting the differential equation from Equation (6.3). Finally, the force function is calculated by

$$\mathbf{f}^*(t) = \tau^2 \ddot{\boldsymbol{\gamma}} + D\tau\dot{\boldsymbol{\gamma}} + \kappa(t)(\boldsymbol{\gamma} - \boldsymbol{g}(t)), \tag{6.5}$$

with $\mathbf{f}^*(t)$ being the time varying force function, $\kappa(t)$ denotes a time varying potential function. In order to avoid an incremental force function, and to be able to adapt the force function, an approximation is made by Gaussian kernel functions given by

$$\mathbf{f}^*(t) \approx \mathbf{f}_{\approx}(x) = \left[ \frac{\sum\limits_{i=1}^{N} w^*_{i,l} \psi_i(x)}{\sum\limits_{i=1}^{N} \psi_i(x)} x \right]_{dim=M\times 1} . \tag{6.6}$$

To obtain a continuous distribution of Gaussian kernel functions, as shown in Figure 6.5, the Gaussian basis functions are calculated by $\psi_i(s) = e^{h_i(s-c_i)}$, with $h_i = \frac{2}{(c_{i+1}-c_i)^2}$ being the width of each kernel and $c_i = e^{-\alpha_x \frac{i-1}{N-1}}$ describes the time of the kernel peak [123]. The resulting path can finally be calculated online with the representing approximated force function.

$$\boldsymbol{\gamma}(t) = \frac{1}{\tau^2} \int\limits_0^{t_f} \int\limits_0^{\bar{t}} \mathbf{f}_{\approx}(\boldsymbol{x}) + \kappa(t)(\boldsymbol{g} - \boldsymbol{\gamma}) - D\tau\dot{\boldsymbol{\gamma}} \; \mathrm{d}t\mathrm{d}\bar{t} + \boldsymbol{\gamma}(0). \tag{6.7}$$

For an accurate representation, the number of weights matters, but also the positions and widths of the Gaussian basis functions. In order to obtain an optimal representation of the trajectory, the generalization can be expressed as in Algorithm 2.

**Algorithm 2** Parameter adaptation for a better representation of learned trajectories with DMPs

```
Input : $\mathbf{c}(\gamma)$, parameters $\{\boldsymbol{\xi}_k\}$
Output : $\mathbf{w}^*, \boldsymbol{\Phi}^*$
for $k \leftarrow 1$ to $m$ do
    $[\gamma_k^*, \dot{\gamma}_k^*, \ddot{\gamma}_k^*] = \min_{\mathbf{u}} J(\mathbf{c}(\gamma), \boldsymbol{\xi}_k)$
    for $i \leftarrow 1$ to $n$ do
        $\mathbf{f}_i^*(t_i) = -\tau^2 \ddot{\gamma}^*(t_i) + \kappa(t_i)(\gamma^*(\tau) - \gamma^*(t_i)) - D\tau\dot{\gamma}^*(t_i)$
        $x_i = \exp\left\{-\frac{\alpha_x}{\tau} t_i\right\}$
        $\mathbf{x}_i = [x_i; \cdots ; x_i]_{dim=M\times 1}$
        $\mathbf{F}_k^* = [\mathbf{F}_k^*;\ \mathbf{f}_i^{*T}(t_i)]$
        $\mathbf{X} = [\mathbf{X};\ \mathbf{x}_i^T]$
    end for
    $[\mathbf{w}_k^*, \boldsymbol{\Phi}^*] = \min \Gamma[(\boldsymbol{\Phi}^j, \mathbf{F}_k^*, \mathbf{X}) \rightarrow \mathbf{w}^j \rightarrow \mathbf{f}_{\approx}^j]$
end for
```

The algorithm, presented in Weitschat et. al [97], allows for minimizing the error between the original, optimized trajectory and the generalized trajectory in DMPs. In general, this algorithm contains the common calculation of DMPs using linear regression, in order to obtain optimal weights. In combination with a numerical optimization the parameter of the Gaussian basis function are adjusted, to optimize the approximation.

In the following section, an approach is presented to enable a constrained motion for safe human-robot collaboration.

## 6.3 Constraint Online Trajectory Generation

In this section, the constrained online trajectory generation is considered. In the general DMP formulation the limitation of velocities, spatial boundaries, etc., is not considered, and only the path of the trained motion using machine learning algorithms is optimized. Therefore, novel methods have to be found, to enable constrained online trajectories. As DMPs are a suitable solution for generalizing optimized motions, methods for the limitation of spacial constraints and velocity constraints have to be found. For human-in-the-loop applications, the distance to the human is of interest as it influences the allowed maximum velocity of the robot.

In this section, methods are developed to guarantee that the online trajectory generation, using dynamic movement primitives, complies with the limitation, given by the maximum velocity and the spatial constraints. The methods are simulated and experimentally validated as presented in [101].

### 6.3.1 Adaptation of Dynamic Movement Primitives for Spatial Constraints

Real industrial applications often contain multiple predominant spatial limitations, which influence the path of a robot. They can be caused by objects in the work space, fences that separate robots and humans in dangerous areas, or simply the own kinematic limitations of a robot. As the presented method includes the optimization of a path, and spatial constraints are considered, no changes have to occur during the online motion execution. The generation of a defined path can exactly be reproduced by using DMPs. The accuracy of the reproduction depends on the number of weights that are used and should finally be validated in a defined range.

An important characteristic of dynamic movement primitives is that the principle shape of path is maintained and different initial parameter and end-conditions can be changed online, without considerable expense. The importance of flexible path generation is significant, because in industrial scenarios, initial and goal positions have often to be adapted. Here, dynamic movement primitive allow for a flexible adaptation of path parameters.

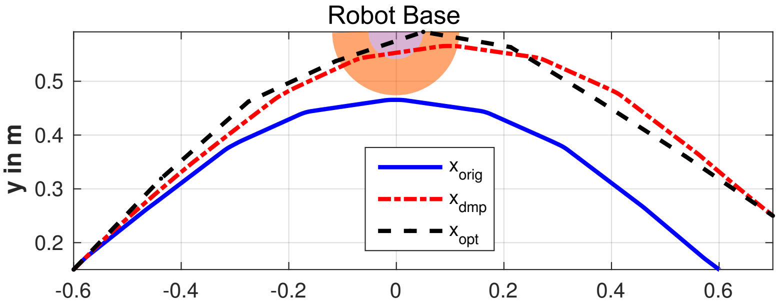


**Figure 6.6:** Generalization of motions with changed goal. The original optimal motion is depicted as blue solid line. The generalization using DMPs is illustrated as red dashed-dotted line. The optimal motion using the new desired goal without considering spatial constraints is depicted as black dashed line. The robot base and restricted area is illustrated as red semicircle. A violation of spatial constraints is obtained. Image ©IEEE

In Figure 6.6, a learned DMP motion is illustrated. The original path is generated as optimal path with human-in-the-loop constraints, as described in Section 5.2.3. The trajectory is generated offline as presented in Section 6.1 and the dynamic movement primitives learned following from Equation (6.5). The motion is chosen to be from the initial position ${}^{R}\boldsymbol{x}_0 = [-0.6\,0.15]$ to ${}^{R}\boldsymbol{x}_g = [0.6\,0.15]$ with a total distance of $1.2\,\mathrm{m}$, which is a usual motion at the octagon workbench and additionally is passing the human workspace. The original trajectory, from Section 6.1, is depicted as blue solid line. The goal position of the DMP motion is changed to ${}^{R}\boldsymbol{x}_g = [0.7\,0.25]$. The resulting motion is depicted as red dashed-dotted line. It can explicitly be observed that the shape of the path is kept, while the spatial motion has been extended significantly. Given the case, no spatial constraints exist, the motion would fit to the optimized motion, which is depicted as black dashed line. The same behavior is clearly a coincidence without any relation, but it is a desired effect.

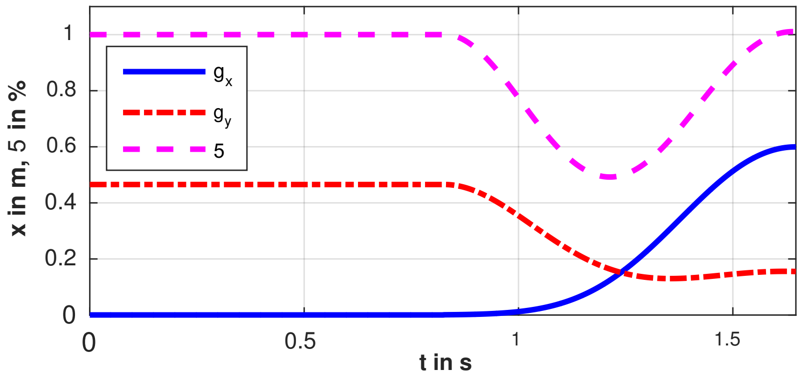


**Figure 6.7:** Via-kernel trajectory for a more robust path generation with different initial conditions. An auxiliary point is placed at turning point of path. Image ©IEEE

Since the position of the robot base is located in ${}^{R}x_b = [0.0\,0.6]$, as it is for example with a

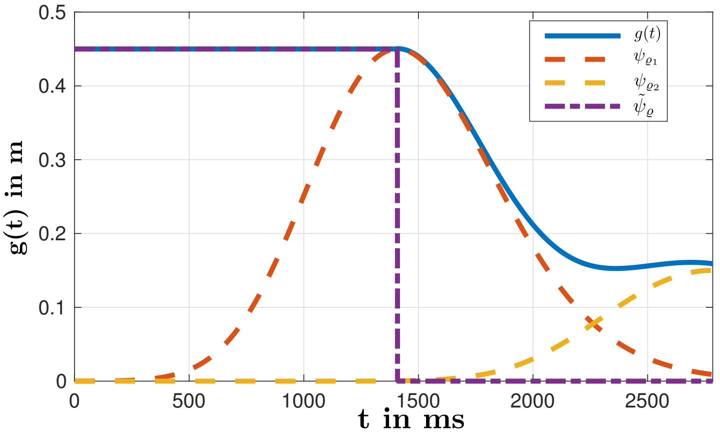


**Figure 6.8:** Example goal function concatenation, including the Gaussian kernels, the initial goal factor $\tilde{\psi}_varrho$, and the final goal function $g(t)$. Image ©IEEE

fixed base on the octagon workbench, the path would lead through its own base, which is kinematically not feasible. The robot path-constraints are given by ${}^{R}\boldsymbol{x}_y \leq 0.45\,\mathrm{m}$, which is considered in the optimization process. Therefore, spatial constraints have also to be considered during the online trajectory generation. The general DMP motion generation is extended with a time varying goal function, using auxiliary points placed on the original path. The path characteristics include special behavior in some positions, as for example turning points. Here, auxiliary points should be placed. As the auxiliary goals are changed directly, a function has to be used that is more than two times continuously differentiable. Simple ramps, anyhow, would generate an unwanted behavior and would cause jumps in the resulting velocity. Exponential functions, as also used for approximating the force function, provide the desired behavior, and can also be used to generate the goal functions online.

Therefore, the goal functions are calculated according to

$$\boldsymbol{g}(s_c(t)) = \sum_{i=1}^{m} \psi_{\varrho_i}(s_c)\boldsymbol{g}_{\varrho} \qquad i = 1, ...m, \tag{6.8}$$

where $\boldsymbol{g}_{\varrho}$ defines the via-goals of the motion. The Gaussian basis functions are given by

$$\psi_{\varrho_i}(s_c(t)) = e^{-h_{\varrho_i}(s_c(t)-s_{\varrho_i})^2}. \tag{6.9}$$

The positions of the basis functions can be determined by

$$s_{\varrho_i} = e^{-\alpha\, \varrho_i}, \tag{6.10}$$

where $\varrho_i$ denotes the percentage of the time duration, given in a range of $0 \leq \varrho_i \leq 1$. The width of each kernel is finally calculated by

$$h_{\varrho_i} = \beta_{\varrho}(e^{-\alpha(\varrho_i - \frac{\Delta t}{\tau})} - e^{-\alpha\varrho_i})^{-1}, \tag{6.11}$$

with a width parameter $\beta_{\varrho}$. To obtain a smooth behavior between the auxiliary goals and the final goal, the varying-stiffness function

$$\kappa = \sum_{i=1}^{m} \psi_{\varrho_i} \tag{6.12}$$

is weakened during goals shifts.

Figure 6.8 illustrates the resulting goal functions. The blue solid line depicts the x-goal-function, and the red dashed-dotted line illustrates the y-goal function, which are generated using Gaussian basis functions, similar to the approximated force function. The path is continuously and two times differentiable. What has to be considered using this method, is that the path has to be validated, as only using the goals can lead to undesired behavior. Additionally, an environmental observation should be integrated to check collisions also online. Anyhow, direct connections can be learned using multiple auxiliary points.

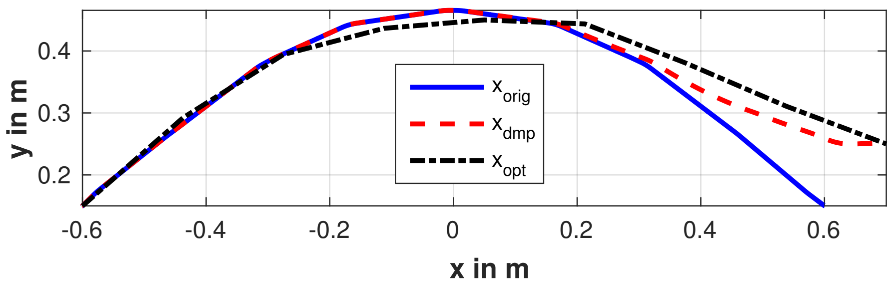


**Figure 6.9:** Generalization of motions with changed goal including the consideration of spatial constraints. The original optimal motion is depicted as blue solid line. The generalization using DMPs is illustrated as red dashed-dotted line. The optimal motion, using the new desired goal considering spatial constraints, is depicted as black dashed line. The generalized path does not exceed spatial constraints and is similar to the optimal motion. Image ©IEEE

Figure 6.9 illustrates the resulting path. The blue solid line depicts the original motion with unchanged goal positions. The red dashed line depicts the new generated constrained path with a changed goal. It can be seen that the spatial constraints are now not violated anymore, and the maximum path position is ${}^{R}\boldsymbol{x}_y \leq 0.45\,\mathrm{m}$, with a changed goal. For comparison with the optimal motion the black dashed-dotted line depicts the optimal path, which fits very well to the online generated path. Additionally, the auxiliary goals can be changed, which provides a higher flexibility in changing environments, as they often are in industrial scenarios.

In the following section, limiting the velocity by using dynamic movement primitives is presented.

### 6.3.2 Adaptation of Dynamic Movement Primitives for Velocity Constraints

Regarding safety, velocity is a main parameter to influence the all over safety of the robotic system. In the ISO-10218 the limit of 0.25 m/s can be found, when the dynamic power is limited by 80 W and the static force is below 150 N. In the extension of ISO/TS-15066, the velocity limit is not explicitly mentioned, but as the mass is a fixed parameter, the kinetic energy $E_{kin} = \frac{1}{2}mv^2$ can be influenced by the velocity. Therefore, the velocity has to be limited to obtain safe motions during the collaborative tasks.

For the generalization of motions, dynamic movement primitives are used, as presented above. As they are generated by second order differential equations, i.e., a dynamical spring damper system, the determination of the velocities can not be done directly. By applying forces or potential fields, the behavior of the generated trajectory is affected and can not exactly be predicted. Additionally, the velocity profile of the motion can be changed significantly, which also occurs, when changing the goal position ${}^{R}x_g$ of the DMP.

To ensure safety in collaborative applications, the velocities have to be limited, such that during the motion the maximum velocity is not exceeded in every single time step. Additionally, the velocity profile must be adaptable during the motion, as human-in-the-loop constraints are considered. A method in order to comply with velocity constraints with DMPs cannot be found in literature. As mentioned above, the velocity in the dynamic system can only be influenced, by adding forces to the system. Therefore, another force is added by

$$\mathbf{f}_c({}^R\dot{\boldsymbol{x}}) = \begin{cases} -K_c({}^R\dot{x}_t - (\dot{x}_{max} - \mu))^{\mathbf{R}}\dot{\boldsymbol{x}} & , {}^R\dot{x}_t > {}^R\dot{x}_{max} - \mu \\ 0 & , {}^R\dot{x}_t \leq {}^R\dot{x}_{max} - \mu \end{cases} \tag{6.13}$$

where ${}^R\dot{x}_t$ denotes the current measured robot velocity. This force vector always acts into the opposite direction of the current motion, and, therefore, can be applied online. As the velocity is limited by a force, the threshold parameter $\mu$ has to be defined and should be kept to a minimum required potential. The parameter $K_c$ represents the potential value. Both parameters can be determined by fulfilling the inequality constraint

$$\mathbf{f}_c({}^R\dot{x}) \geq \mathbf{f}_{\approx}(s_c) + \kappa(t)(\mathbf{g}(t) - {}^{\mathbf{R}}\mathbf{x}_{\mathbf{d}}) - D\,{}^R\dot{x}_{max}, \tag{6.14}$$

where each parameter is given by the dynamic movement primitive definition. The velocity can finally be limited by adding the force given by

$$\tilde{\mathbf{f}}_{\approx}(s_c) = \mathbf{f}_{\approx}(s_c) + \mathbf{f}_c({}^R\dot{x}), \tag{6.15}$$

to the dynamic system. This force is regulating the velocity online, such that the motion complies with the desired limits for a safe motion, defined by the collision experiments.

### 6.3.3 Simulation

To analyze the presented method, the different velocity profiles are compared in simulation using MATLAB/Simulink. The simulation results are illustrated in Figure 6.10 for a standard motion of 1.2 m, having the optimal motion learned, as presented in Chapter 5. The upper plot represents the motion in the x-y-plane over the table. The assumed human hand position is located in ${}^Hx_0 = [0.05\,0\,0]$ and the resulting path is avoiding a close motion to the human. The original, learned path, is shown as blue solid line, and the DMP motion with a changed goal is shown as green dashed line, using the presented method for limiting the velocity. The motion is spatially limited by $x < 0.45\,\mathrm{m}$, and the desired goal position is changed to ${}^Rx_g = [0.25\,0.7]$. It can be seen that there is no crucial change of the path, by using the velocity limitation method.

In Figure 6.10, bottom plot, the velocities of the motions are illustrated. The magenta dashed line depicts the maximum allowed velocity for the motion with a human-in-the-loop. The blue dotted line depicts the original motion, generated with DMPs, without a changed goal position ${}^Rx_g = [0.15\,0.6]$. It can be seen, that during the original motion no velocity constraints are violated, and the velocity profile is adapted to the profile of the maximum allowed velocity. Using the standard DMP formulation, and changing the goal, leads to the red dashed-dotted line. After changing the goal, the velocity profile of the generated trajectory exceeds the desired limits significantly. Consequently, safety is not guaranteed, such that the robot is able to decelerate its motion in time, and, therefore, impact forces and pressure can be higher than experimental evaluated. The green solid line depicts the motion with a changed goal and the regulated force vector. It can be seen that the maximum velocity is complying with the desired limits and keeps the behavior of the motion, i.e., reaching the desired goal position in shortest time.

With the implemented velocity regulation, motions on a robot can now be executed in a safe manner. Therefore, in the next section the presented method is experimentally validated on a robotic system.

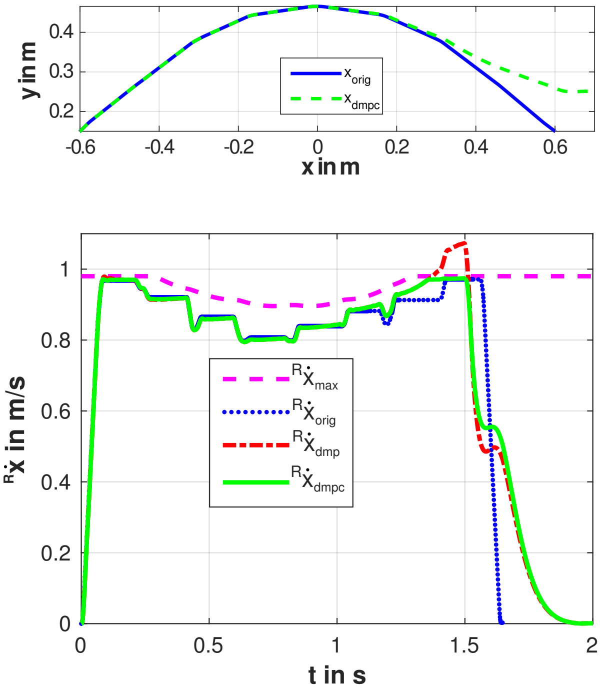


**Figure 6.10:** (Top) Generalization of motions complying with the desired velocity limits. No major deviations in the Cartesian motion can be observed. (Bottom) Comparison of velocity profiles generated by DMPs with changed goal positions. The magenta dashed line depicts the velocity limits over time, depending on the distance to the human hand. The blue dotted line depicts the original generated DMP trajectory. The red dashed-dotted line depicts the velocity profile of the motion with a changed goal position without considering velocity constraints. The green solid line depicts the presented method including the consideration of velocity constraints. Image ©IEEE

## 6.4 Experiments

In this section, experiments on the generalization of motions are presented. First, the experimental setup is described in detail. Then, three different trajectory generation methods are applied and compared to each other, aiming on the improvement of the application.

### 6.4.1 Experimental Setup

The experiments are executed on the collaborative octagon workbench with the extended robot workspace, where the robot is mounted on a linear axis. The maximum range is 2.6×1.6 m, and the entire 8-DoF system is controlled by a Cartesian impedance controller. The desired end-effector stiffness is parameterized by 800 N/m. As illustrated in Figure 6.11, the experimental setup is surrounded by Vicon a tracking system. A measurement of the human hand position is available at a frequency of up to 200 Hz, but for robustness during the experiment, an update frequency of 100 Hz is used. Special markers are placed on the back of the human hand, and the Vicon sensors are geared towards the workbench surface. For the experiments, the human is sitting in front of the workbench, the hand is placed on the workbench, and is moved during the motion and the reaction of the robot analyzed.

### 6.4.2 Experimental Results

In this section, three different methods, for moving the robot within a collaborative application, are presented. The maximum velocity is given by ${}^R\dot{x}_{ISO} = 0.15\,\mathrm{m/s}$, which is determined by collision experiments, see Chapter 7, where the maximum force and pressure is measured at a collision with a crash-test dummy. Motion records and breaking tests yield to a robot

**Figure 6.11:** The experimental setup on the octagon collaborative workbench is surrounded by VICON marker tracking system to measure the human hand position. Image ©IEEE

acceleration of $^R\ddot{x}_{max} = 3\,\mathrm{m/s^2}$. For simplification, and an intuitive interpretation, the motion is executed along the x-axis at an initial position $x = 0\,\mathrm{m}$ and a goal position $x = -1.2\,\mathrm{m}$. During the robot motion, a human is located in the shared workspace. Therefore, due to safety reasons, the maximum overall velocity is defined with $0.6\,\mathrm{m/s}$. The three experiments are presented in the following:

### Experiment 1: direct motion with ISO defined velocity

In the first experiment, the standard motion is executed, where the velocity is limited by the ISO with a maximum of $^R\dot{x}_{ISO} = 0.15\,\mathrm{m/s}$. The motion is illustrated in Figure 6.12, where the left plot depicts the position of the human arm over time. As mentioned above, the motion is executed into the x-direction, i.e., the red dashed-dotted, and the black solid line, are nearly constant. During the motion little deviations occur caused by the Cartesian impedance control. The green dashed line depicts the marker position, that is also constant, as there is no human-in-the-loop treatment. The right plot depicts the velocities and the distance during the motion. The path leads over the marker position, which is shown as black solid line, and depicts the distance between the human and the robot. The red solid line depicts the

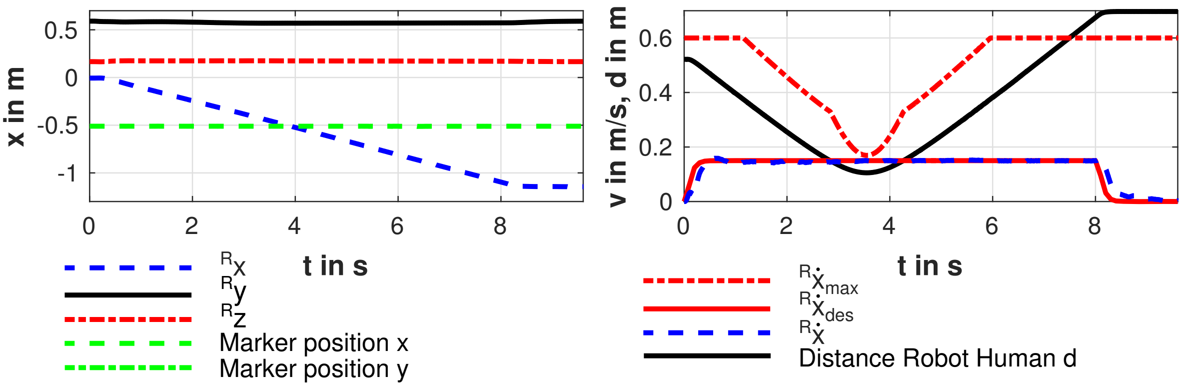


**Figure 6.12:** Experiment 1: ISO defined motions as usually used in collaborative applications.

desired velocity, which is commanded by the Cartesian trajectory generation, presented in Section 3.3.1. The blue dashed line shows the velocity, measured at the robots end-effector. The red dashed-dotted line shows the maximum allowed robot velocity, given the method for efficient robot motions, presented in this thesis. The total duration of the motion is eight seconds, for a distance of 1.2 m, which is only the motion in one direction. Obviously, the cycle times are very high. It can be assumed that the application in a real industrial scenario is very inefficient, and an amortization of such robotic system cannot be expected.

### Experiment 2: direct motion with ISO adapted velocity

In the second experiment, illustrated in Figure 6.13, the same motion is executed with an adaptable robot velocity during the motion. Here, the proposed calculation of the maximum velocity from Equation (4.22) is used, to determine the current desired velocity. The trajectory generation for adaptable velocity is used, as presented in Section 3.3.2. The estimated time to collision $\tilde{t}$ is calculated by using the duration data from the simulation, as the calculation of the dynamic model is not fast enough for real-time calculations. The marker is again placed, such that the robot passes the marker position, which is constant in the first motion. In the left column, again, the position over time is illustrated. The left column shows the velocities and distances. In the upper right plot, it can be seen that the desired velocity, depicted as red solid line, is reduced, when the distance decreases, depicted as black solid line. The blue dashed line shows the measured end-effector velocity. It can be seen, that the resulting robot velocity is violating the given limits, which is caused by the low-stiffness impedance control and the extended system using the linear axis. A better result can either be achieved by implementing a controller allowing for a higher stiffness, or a position control with inverse kinematics, which is future work for the octagon collaborative workbench. The total duration of motion is about $3.4\,\mathrm{s}$, which is a remarkable improvement of efficiency. Finally, it is 2.47 times faster, compared to the movement bounded by ISO. The risk of injuries during this motion is identical to the first experiment, as the velocity at a collision is equal.

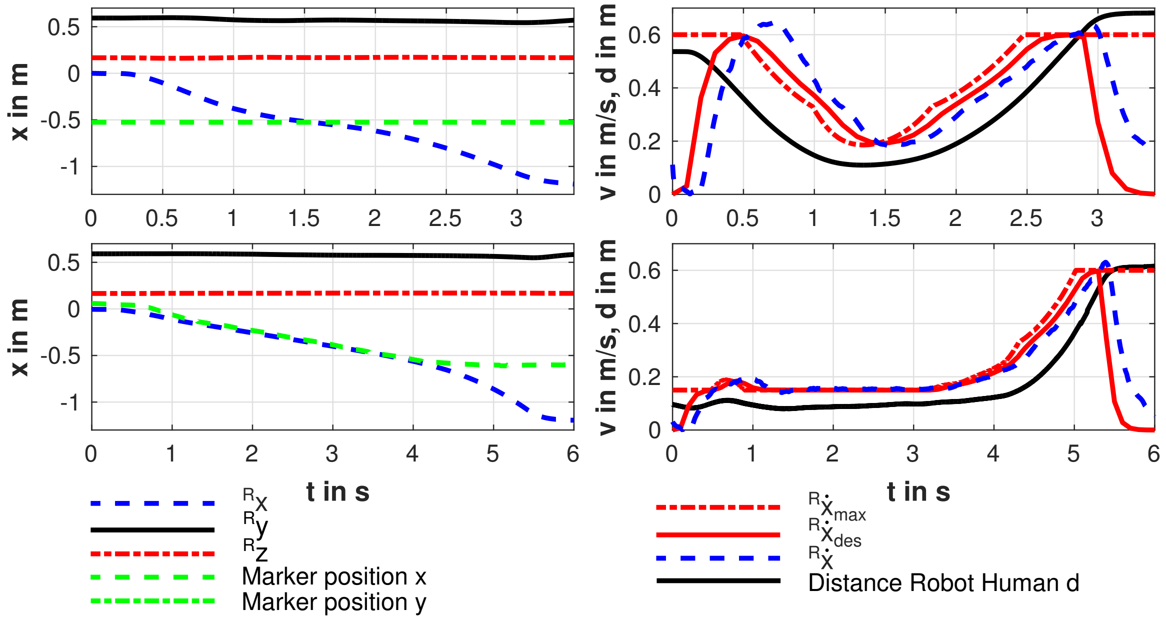


**Figure 6.13:** Experiment 2: Velocity adaptation on a straight path to improve efficiency.

In the second motion within Experiment 2, the marker was moved during the motion, which is shown in the left bottom plot, dashed green line. Here, the human hand is following the robots end-effector and the desired velocity is limited during the motion, shown in the right bottom plot. However, the duration of the motion is increased as the robot has to slow down the motion, which is a necessary reduction of the velocity, as the human is about 10 cm close to the robot end-effector.

As in this motion the desired path is on a direct straight, obviously, the motion is not optimal in the context of minimal cycle times. However, this solution provides a remarkable improvement of efficiency an due to the experimental results, a usage in real industrial applications is suitable.

**Experiment 3: direct motion with ISO adapted velocity**

In the third experiment, the presented method for optimal human-in-the-loop motions is used. Now, the robot is avoiding the direct motion and is moving on an optimal path generated in Chapter 5 and generalized with dynamic movement primitives. In Figure 6.14 the results are illustrated. In a first motion, the marker is placed at the same position as in the first and second experiment. In the upper left plot, is shown that the y direction of the robot path is now varying. The upper right plot illustrates the corresponding velocities. The velocity profile of the optimal motion, including human-in-the-loop velocity adaptation, reflects the learned trajectory. During the motion, the desired velocity profile, depicted as red solid line, is similar to results from the optimization. The distance, depicted as black solid line, is higher than in the experiments before, which enables higher motion velocities. Finally, the desired velocities are consequently below the limits that are calculated online.

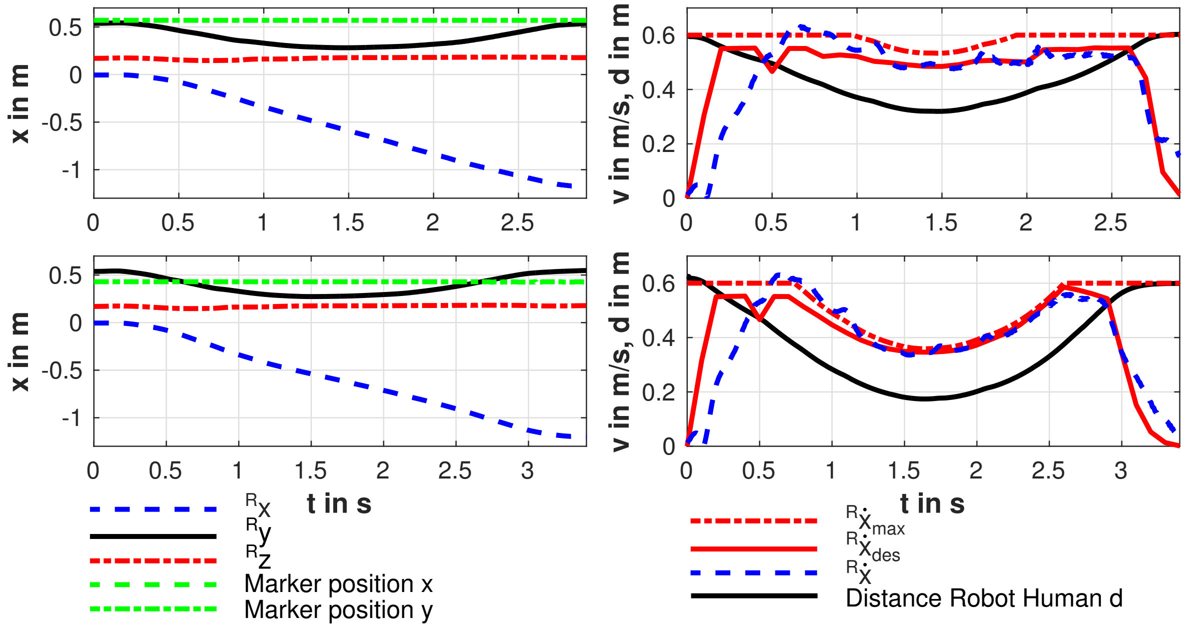


**Figure 6.14:** Experiment 3: Generalized optimal motions with human-in-the-loop constraints.

In the second motion, the robustness regarding complying with the limits is evaluated. During the second motion the marker is placed closer to the robot path, which has the spatial limitation of its own base and the velocity limitation. It is shown that with the presented method, the desired velocities are always below the desired limits. However, the duration of the motion is lower than in the other experiments, which shows that the developed method for efficient human-robot collaboration is applicable and improves the performance, whilst keeping injury risk at the same level.

## 6.5 Conclusion

In this chapter, novel methods for trajectory generation with humans-in-the-loop are presented. An optimization algorithm generates a desired path, including constraints for the motion, using multiple auxiliary points and connecting them by straights. This path provides only position information and is not applicable for execution at the robot. Hence, trajectories are generated with the focus on minimal time for the point to point motion and complying with the velocity and acceleration limits. Additionally, the trajectory is generated, such that the following velocity constraints are not violated.

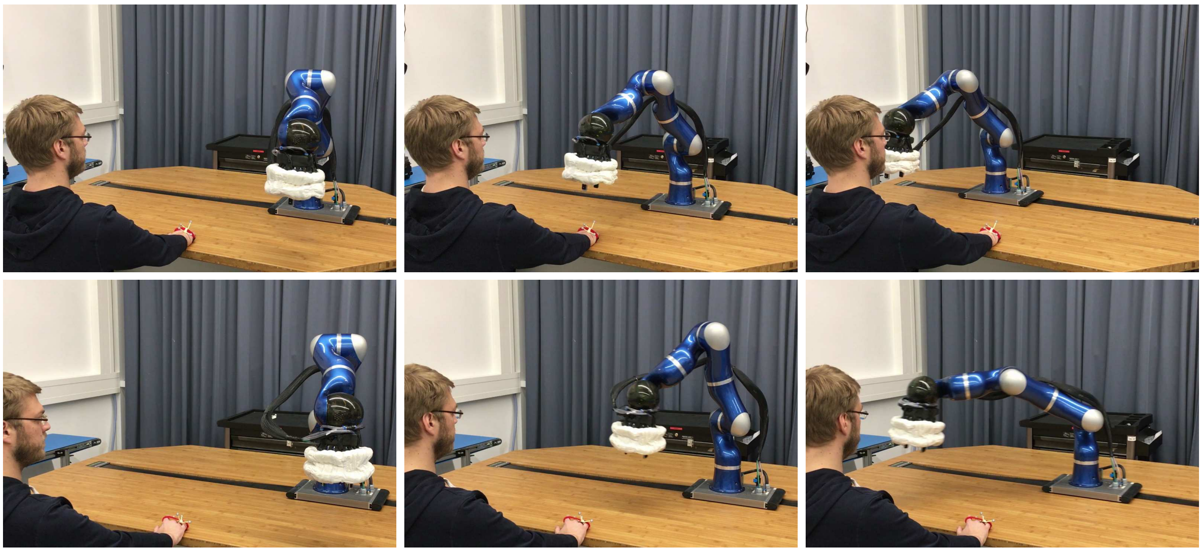


**Figure 6.15:** Video scene

The resulting path and trajectory information are static, and no flexibility is given within the application. Therefore, dynamic movement primitives are used to generalize the motions. For the motion generation, the general DMP formalism is used and adapted. As one main focus of this thesis is the improvement of performance in industrial applications, the optimized path and transformed trajectory information serve as input to generate and learn the weights of the DMPs, and then being able to generate the trajectory online. The second focus lies in the safety during the application and an operating robot. Hence, methods are developed that provide the compliance of constraints, in particular spatial constraints and velocity constraints. The necessity of spatial bounds is reasoned by the shape shift of DMPs all over the trajectory. A small change of the desired goal position keeps the characteristics of the path but also expands or reduces the motion significantly. The implemented method, using a time depending goal function, enables an adaptation of the path in a desired manner. Furthermore, the limitation of velocities online is implemented for the online trajectory generation method, which is necessary to ensure safe robot motions during the collaborative application.

To summarize, the optimization and generalization of robot motions can improve the performance of a collaborative robot work cell significantly. A requirement to ensure safety is depending on robust human detection sensor or vision based methods that always measure the correct human positions, and this is, anyhow, a conceivable technology in the near future.

# Chapter 7

# The Robotic Airbag Performance Improvement by Hardware

Safety requirements are subject to very strict rules in order to protect humans from hazards. Therefore, a robust behavior of the safety system is a very important property. Previously presented camera based systems do not provide robust human position measurements, and, therefore, no approval for operation in fence-less robot applications, where robots are sharing their work space with humans, can be granted. Often small light changes lead to an error in the detection, and, finally, a reliable safety is not provided. Operating robotic systems are surrounded by fences, or similar delimiting devices, to obstruct the access to the non-collaborative robot operation zone. As described in Section 2.4.2, different types of human-robot collaboration are available, but, in principal, the continuation of the robot application in the direct presence of a human is necessary. State-of-the-art safety systems do exactly the opposite: in that moment a human enters the work space of the robot, the application is stopped, and the robot system is switched into a safety mode. Usually, the entire task has to be started from the beginning. This leads to an enormous loss of production time, which finally causes high costs for the manufacturer.

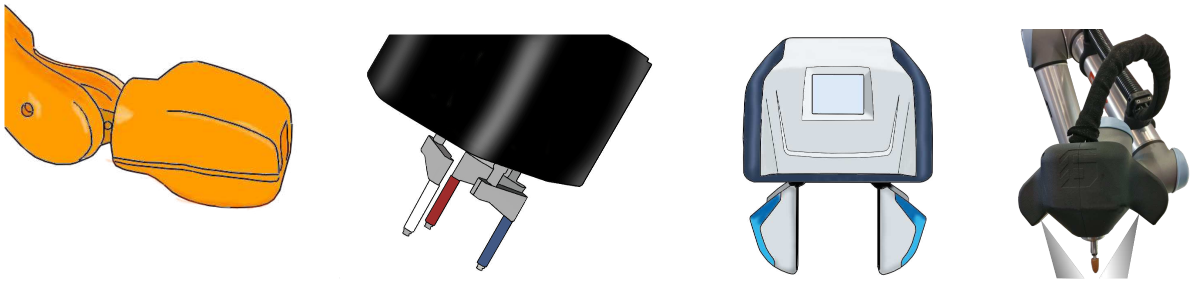
To address this problem, this chapter focuses on real physical hardware solutions that enable safe and efficient human-robot collaboration. First, state-of-the-art hardware safety systems are presented and discussed. The major disadvantages of these devices are identified and a suitable solution to solve that problem by adding a peripheral safety hardware, namely the *Robotic Airbag*, is presented. Safety tests and experiments, using a crash test dummy, evaluate the functionality of the Robotic Airbag. Finally, a benchmark analysis proves that the efficiency can be remarkably improved and the deployment in collaborative application is discussed.

## 7.1 Existing Hardware Solutions

In this section, existing technologies are described. The focus is on protection of robotic tools and peripheral objects that are unsafe in terms of collaborative robot applications. The solutions are discussed and the main advantages and disadvantages described. Furthermore, essential requirements for a protection system to secure robotic end-effectors are derived.

### 7.1.1 Hardware Safety Systems

Besides a robot control side collision avoidance system, based on vision technologies and corresponding algorithms, many different hardware or sensor solution exist, to enable fence-less robot work cells. The most popular one is a light barrier, which stops or slows down the robot, when somebody passes through this virtual fence system. The company PILZ GmbH & Co. KG developed solutions that are equal to virtual fences. The *SafetyEYE* is a vision based system that scans the robot work space, and in the case a human enters the work space, the robot goes into a safe mode. Additionally, the *PSENmat* is a mat, lying on the floor, in front of the collaborative system. In the case a human enters this work space and steps on the mats, the robot switches into a safe mode, too. This could also be a full stop of the robot system, which interrupts the production, and the productivity of the station is negatively influenced. Therefore, many companies are looking for solutions that allows for humans entering the robot work space, without a loss of performance of the robotic system. Thus, the main focus regarding HRC is given to the safety of humans, many companies started to bring solutions to the market, which cope with the safety standards described in Section 2.4.2 at the robot itself. The different types of solutions are described in the following:



(a) Foam cover (b) 1-DoF gripper (c) Sensor gripper (d) Ultrasound sensor

**Figure 7.1:** Schematic representation of state-of-the-art safety solutions for robotic tools to enable collaborative robotics. (a) Foam cover to ensure a cushion between human and tool in case of a collision. Usability with carried objects is severely restricted. (b) Gripper with a moving axis. Force given of the axis triggers a stop of the robot. No usability with carried objects and only in one direction. (c) Integrated capacitive sensors to stop the robot, additional rounding at the edges to lower pressure to the human skin in case of a collision. No usability with carried objects. (d) Ultrasound sensors that stop the robot before a collision occurs. Coverage of space and flexibility limited.

Robot companies claim that their robots are collaborative, which is true under constraints. The robot as collaborative machine is deemed to be collaborative, when the hardware system has an emergency switch, ensures a maximum velocity, which can be realized by two joint position sensors, and a mechanism or sensory trigger to stop the robot in case of a collision, to prevent from injuries to a human. However, collaborative robots are only safe systems without a tool attached to the robot. Generally, the tools of robots are a main bottleneck and prevent from safe collaborative robotic applications. The most metaphorical example is a very slow moving soldering iron, which moves very slow into the eye of a human, as well as a scalpel. The collision detection would stop the robot too late and an injury is unavoidable.

The first idea that came up, was to cover the tool. This is illustrated in Figure 7.1 (a). The foam provides protection for example from sharp edges of the tool, as it forms a cushion between the human and a dangerous part [124], [125]. In addition with sensor skin or capacity sensors it is a suitable solution in order to reduce impact forces and pressures [126]. In Figure 7.1 (b), a solution is presented, which focuses on clamping. The gripper is movably mounted and triggers a stop, when it is moved. The depicted solution in Figure 7.1 (c) is a gripper, which is developed by SCHUNK [127]. The characteristics of this gripper are big roundings on each edge and capacity sensors to stop the robot when an obstacle is getting into the robots path.

A very promising technology was developed by MAYSER GmbH & Co. KG, depicted in Figure 7.1 (d), who presented an ultrasound sensor, which stops the robot in the case an obstacle is getting into the path of the robot. These sensors can be attached to the robot and being programmed considering the environmental conditions. Additionally, trained specialists have to prepare a risk analysis.

All of these solutions, do have advantages and disadvantages but all these technologies do not finally solve the problem. In the following, the problems using conventional technology are discussed and essential requirements defined.

### 7.1.2 Discussion and Problem Definition

A large amount of collaborative robots is available on the market and all the robot manufacturers advertise with safe collaborative robotic systems. But, in fact a very few of these applications exist in real production lines. This is caused to two main reasons.

The collaborative application

1. does not comply with safety criteria and standards,
2. complies with the safety requirements, but leads to an uneconomical and inefficient process.

But, why do these solution do not allow for efficient HRC? This is because by operating a fence-less robotic systems always the potential risk on the entire robot application or workcell needs to be considered.

The solutions presented in Section 7.1.1 do not solve the entire problem. For example, supplying robotic tools with foam solves the problem of sharp edges and provides protection, but two main problems arise with this methodology. The foam attached to the tool leads to a much larger volume of the tool geometry. This causes that storages of objects have to be designed very large. Foam at gripper fingers do not allow for grasping closely positioned objects. The same problem is given with the *Co-act* gripper, where roundings prevent from severe injuries.

The 1-DoF gripper from Figure 7.1 (b) provides safety in one direction. However, a dangerous situation cannot be excluded during a standard motion from left to right because even a slow motion can lead to severe injury and the velocity of the robot has to be reduced dramatically. Additionally, the orientation of the robot tool is significantly bounded because gravity would trigger the safety stop.

The first three solutions of grippers, presented in Figure 7.1, have a common disadvantage: their main function is to transport objects, but they do not cover these objects. Most objects in industrial production have got sharp edges caused by the manufacturing process. The obtained results from Table 2.1 also include that almost every workstation has to handle different types of geometries. These different components are one main point to find a solution for safe collaborative robots with grasped objects. These objects cannot be changed in its geometry or design, because the needed surfaces to assemble two objects usually consist of sharp edges and no rounding can be added.

Until robot collision avoidance technologies are certified for collaborative solution a stopping robot can also lead to injuries. The ultrasound sensor provides protection if the human is treated as static, but in most cases the robot and the human are moving towards each other and sharp edged pieces can also lead to injuries. The little scattering angle of that sensor also leaves gaps and full safety cannot be provided.

Finally, three essential requirements can be formulated:

1. The permissible limits of the ISO standards must be complied with
2. The functionality of the tool mustn't be deteriorated, as well as the overall robotic system
3. Safety must be given, even if an object is carried by the robot and the brakes are activated

None of the existing technologies allows for safe and flexible HRC. In the next section, the *Robotic Airbag* is presented that complies with the defined requirements.

## 7.2 The Robotic Airbag for Human Robot Collaboration

In this section, an innovative hardware solution named Robotic Airbag is described, which enables economic collaboration between robots and humans. First the problems are defined in more detail and the disadvantages of state-of-the-art technologies discussed. Then, an approach to solve this problem is presented and detailed requirements to enable the functionality of such system are mentioned, as well as different versions or designs to extend field of applications, where the Robotic Airbag can be used for. The experimental results, including crash tests with a crash test dummy, are analyzed and discussed, regarding standards and requirements for safe operation of fence-less robotic systems.

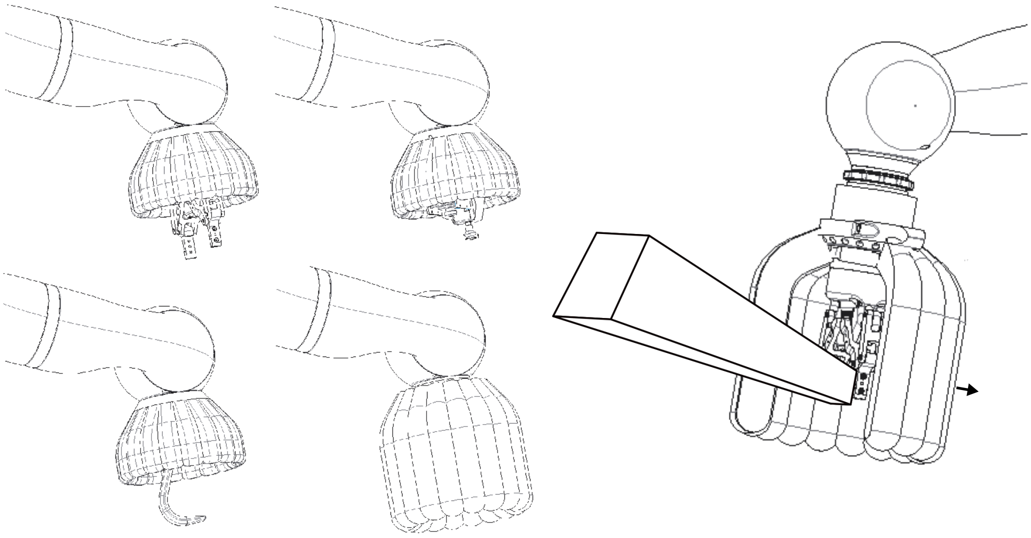
### 7.2.1 Approach

As mentioned in Section 7.1.2, the main issue of bringing robots into collaborative applications are tools with sharp edges that are not safe. Since airbags in cars and for many other situations are proven to prevent from severe injuries [128], [129], [130], airbags have the potential to also avoid injuries within collaborative applications. This section presents an innovative approach which enables safe human-robot collaboration in industrial manufacturing by covering dangerous edges by an airbag. The requirements given to a safety system for tools and objects as mentioned above are that the tool is covered and safe during a fast motion of the robot as well as the carried object.

Furthermore, no restrictions for the tool should be given, when it is needed for gripping or manipulating an object. To achieve this, a mechanism is needed that is able to extend around the tool in case of an insecure situation, which basically always occurs when a robot moves autonomously. Additionally, the mechanism has to be able to retract itself, when the full functionality of the tool is required.

This is the motivation for the following development at hand. An airbag-like safety-module was developed, which builds a cushion between the robotic tool and the human in case of a collision, also when an object is gripped. This innovative technology was at first presented in [131]. Usually, airbags are triggered in case of an emergency, but due to the fact that in collaborative robotics these collisions can occur very often, this would not cope with formulated requirements from Section 7.1.2. After the airbag would have been activated the overall robotic system would be unusable. Therefore, the main idea is that the airbag is not triggered at an upcoming danger, but is rather always inflated during unsafe motions of the robot. An unsafe robotic tool, or a gripper, carrying an object is constantly covered, which enables an intrinsically safe motion with an unsafe end-effector. The inflation is realized by pressured air, which is usually available in every industrial manufacturing.

The ability to deflate the airbag, when the robot is standing still, or moving very slow and safe, is the second very important feature of the *Robotic Airbag*. A deflated airbag is illustrated in Figure 7.2 (a). Elastic bands, or springs, allow for retracting the cover, and, therefore, unrestricted access to the tool can be provided. As mentioned in Section 7.1.2, mounting foam as safety cushion to the tool does not allow to free the tool.



(a) Full cylindrical design (b) Open part design

**Figure 7.2:** Functionality of the DLR safety module; a) standard two-finger gripper and deflated module; b) vacuum gripper with camera and deflated module; c) 3D-printed tools, hooks etc. (similar used in a known automobile manufacturing) and deflated module; d) inflated module covers a), b) and c) completely and ensures a safe and fast motion without harming a human. Image ©IEEE

The selected material for the *Robotic Airbag* is Nylon, since it is a well-proven and established material for airbags as used in bicycle helmets or cars, respectively. To bring the airbag into a desired shape, a special construct of chambers was developed, which simultaneously reduces the overall volume and fully covers the tool including their objects. Reducing the volume is an important aspect because a smaller volume leads to a shorter inflation and deflation duration, since it is dependent on the volume flow rate, which is physically limited. A limited diameter of the supply pipe, especially when it is led through the robot, limits the medium flow essentially. It is necessary to optimize the inflation time, to ensure fast cycle times, but it is recommended to limit the inflation, because an explosively inflation leads to potential injuries of humans or gripped objects can be dropped with high speeds, for example in the setup phase of a robotic system. However, this does not relate to the deflation of the airbag. The deflation can be executed as fast as possible, because no danger can emerge by deflating the airbag.

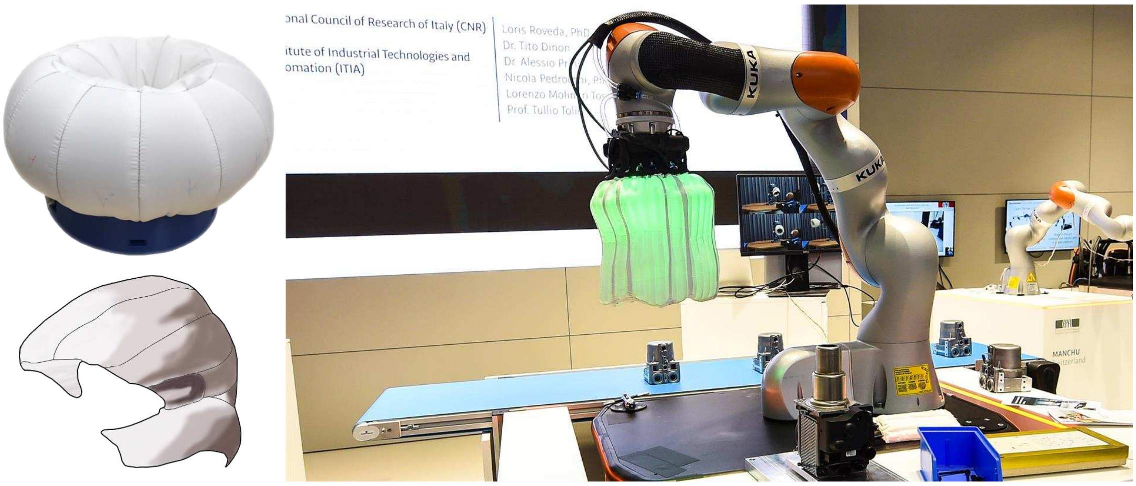
As illustrated in Figure 7.2, the *Robotic Airbag* is not depending on a defined set of tools. Rapidly new tools can be created even by 3D-prints or self created tools are covered by this safety module. Many different types of tools can be realized such that customized variants enable fast set-up times.

The decisive advantage of the developed *Robotic airbag* is that it covers both the tool and also a carried object. The customization of the airbag design can be adapted to the carried object. The form can be adjusted to the geometry of the object, too. Therefore, a safe motion with carried object is even possible.

The problem of clamping can also be eliminated with the *Robotic Airbag.* Given the case a human is clamped and is not able to free himself alone, which can quite happen when the joint-brakes are activated. The airbag is able to deflate at any time and free the human. This can be realized by sensor evaluation, a defined time after the collision or even by cutting the airbag.

### 7.2.2 Robotic Airbag Design

The form of the airbag is flexible designable, for a wide range of robotic applications. The final design required from the manufacturer is depending on the tool and carried component for this application. This is due to the geometric parameter sets of the application, which are in particular the size of the gripper and the size of transported the object.



(a) Possible designs (b) Final airbag design

**Figure 7.3:** Flexibility of Airbag Design. (a) Upper image illustrates a sphere design to surround objects. The lower image shows the Hövding[1] bike helmet airbag, which illustrates the possibilities of the design, depending on the seam design. (b) Show the final design for the airbag in the collaborative work cell and evaluated in collision tests and real industrial applications.

In Figure 7.3 different designs are illustrated. Inspired by the Hövding bicycle airbag, depicted in Figure 7.3 (a) bottom row, the designs are not straight and dependent on the same design adaptable. First, experiments with the designs are illustrated in Figure 7.3 (a) top row, which has a sphere design. However, it turned out that the sphere design is unsuitable for the retraction and the volume is too high. The final and suitable design is depicted in Figure 7.3. The cylindrical design enables a high flexibility for different tools and objects.

The transport of larger objects is still a special challenge. On this point, a closed airbag design is not suitable. To solve this problem, parts of the airbag can be left open as shown in Figure 7.2 (b) and the motion direction has to be adapted such that the airbag is always at front side. The airbag covers the tool and the object in the direction of motion, so that a collision between human and robot can only occur with the cushion of the airbag among them.

In the next section, the final design is experimentally evaluated and analyzed with respect to the ISO standards.

## 7.3 Experimental Validation for Improvement of Human Safety

This section describes the proof of the functionality and effectiveness of the *Robotic Airbag*. The prerequisites are a scenario, where a human and a robot collide with each other. To achieve applicable results, the crash tests were executed with a crash-test dummy, which has the similar shape and dynamic behavior like a human, and they are used to determine crash behavior of humans in car test for many years. As collisions usually not often take place at

---

[1] Source: `https://www.kunstform.org/en/hoevding-m-291`

a human head, this is finally the most dangerous zone on the human body. Integrated applications in the lab indicate that collisions with a head may happen, especially with a robot mounted on a workbench as assistive robot. Therefore, the experiments aim on the compliance with the ISO/TS-15066 on head limits, which can be found in Table 2.2.

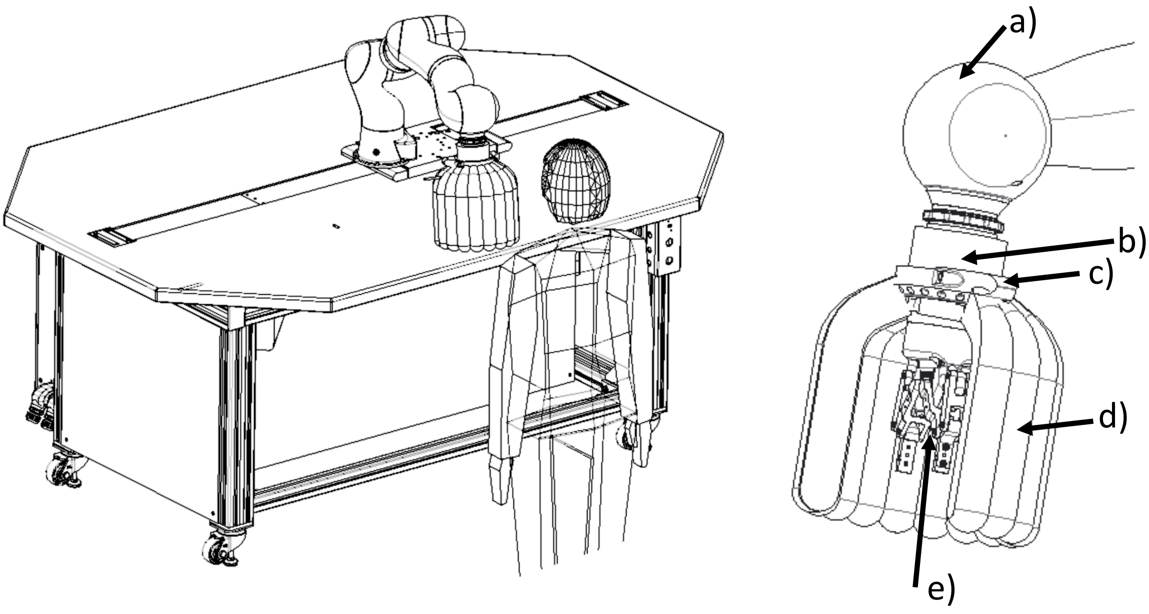


**Figure 7.4:** The setup includes the octagon workbench with a LWR III a) mounted on a linear axis, which is also used to reach the maximum velocities of the robot. A force torque sensor b) is attached to the robot tool center point (TCP) to determine the resulting forces during a collision. A flange c), that connects the airbag with the robot, and includes sensors, as well as valves to control the airbag d), is mounted to the force torque sensor. For the collision and injury experiments a standard industrial gripper (e) is used. Image ©IEEE

In Figure 7.4, the experimental setup is depicted and shows the RACELab octagon workbench, as described in Chapter 2.5. The exemplary setup is very close to a collaborative workbench that can be used in real industrial applications, as for example the table height and the operating space of the robot fit. For the experiments, the eight DoF octagon uses a KUKA LBR III, which is mounted on the linear axis for the extension of the work space. This is used to have longer acceleration phases, to get to the maximum speed of the robot system in the experiments. With their redundant position sensors link and motor sided, at the robot as well as at the linear axis, it is ensured that the correct velocity of the robot is given. A detailed overview of the inner control of the overall system is given in Section 2.5.3. The momentum observer from Section 2.5.3 in Equation (2.1) triggers the collision and forces of the robot to stop the motion. Usually, regarding the requirements of the ISO/TS 15066 the robot would enable the brakes to a hard stop. Due to the reason that for this robot system a compliant torque controller is implemented, this compliance is used to get the robot pushed away for safety reasons, for a later analysis of functionality and comparison to real human collisions, executed by the author of this thesis.

As in this thesis it is claimed that the tools are the main problem, the experiments focus on the tool side collisions. The mounted tool for the experiments is a common industrial gripper to proof the safety improvement by using an on the market available tool. The tool is also chosen without special roundings for collaborative applications, and the final collision tests are applied that way the edges of the tool hit the surface of the crash test dummy.

A detailed overview of the final end-effector design is given in Figure 7.4 right column, where

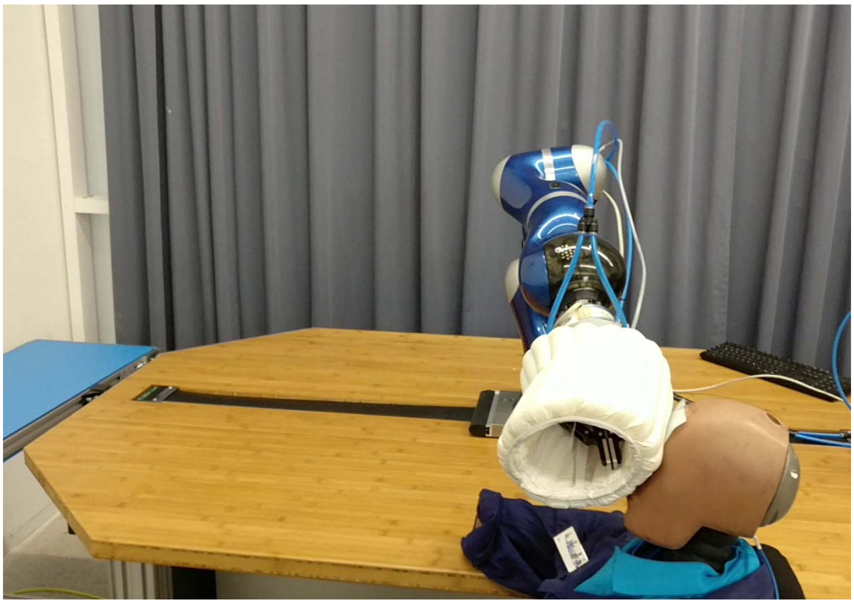


**Figure 7.5:** The experiments include collision tests with a crash test dummy. To measure the maximum pressure to the dummy skin a fuji pressure measurement film is attached to the dummies head, at the predicted position of collision. The entire experimental setup is illustrated in this image. Image ©IEEE

a) depict the used robot, b) is the force-torque sensor to obtain the impact forces and can be treated as extra payload as the weight of the sensor is around 700 g, c) is the flange mounted between the robot and the tool, where also the airbag is attached, the integrated sensors are placed and valves to control the inflation and deflation. As one can see, the airbag completely encloses the robotic tool, such that no collision can occur without the cushion of the airbag.

In Figure 7.5 the placement of the crash test dummy is illustrated. From table Table 2.2 it is known that an impact in the face of the human has the lowest limits for pressure and force, so the crash test dummy was placed such that a collision takes place in the area of the face, in particular a bit more on the forehead. This is due to the reason, that tests with a real human shall be comparable and the forehead is a more safe area in the face. For the analysis the maximum limits of the face, in particular the masticatory muscle, are used.

In the following, first the maximum forces within an impact are analyzed.

### 7.3.1 Maximum Force Analysis

To obtain usable results of the applicability of the *Robotic Airbag*, force experiments with the crash test dummy are executed. The force sensor is a *JR 3* sensor, based on strain gauge technology. The sensor is mounted between the airbag and the robot, to also measure the forces on the airbag itself. The inflated airbag has an internal stability due to the pressured air, which would imply that the inherent stability produces own forces which shall also be measured. Due to the equilibrium of forces, the location of the force torque sensor is suitable.

In Figure 7.6, the measurements of the forces are depicted, where a comparison of the impact with just the tool, i.e. no cushion from the airbag, and an airbag surrounded tool is shown. In the upper row, the impact forces are depicted depending on the time. The first impact experiment was executed with an end-effector Cartesian velocity of $0.4$ m/s, which is illustrated in the bottom row. The needed time to impact was around four seconds, and as can be seen the velocity was constantly given. Little deviations are caused by the impedance controller. It can

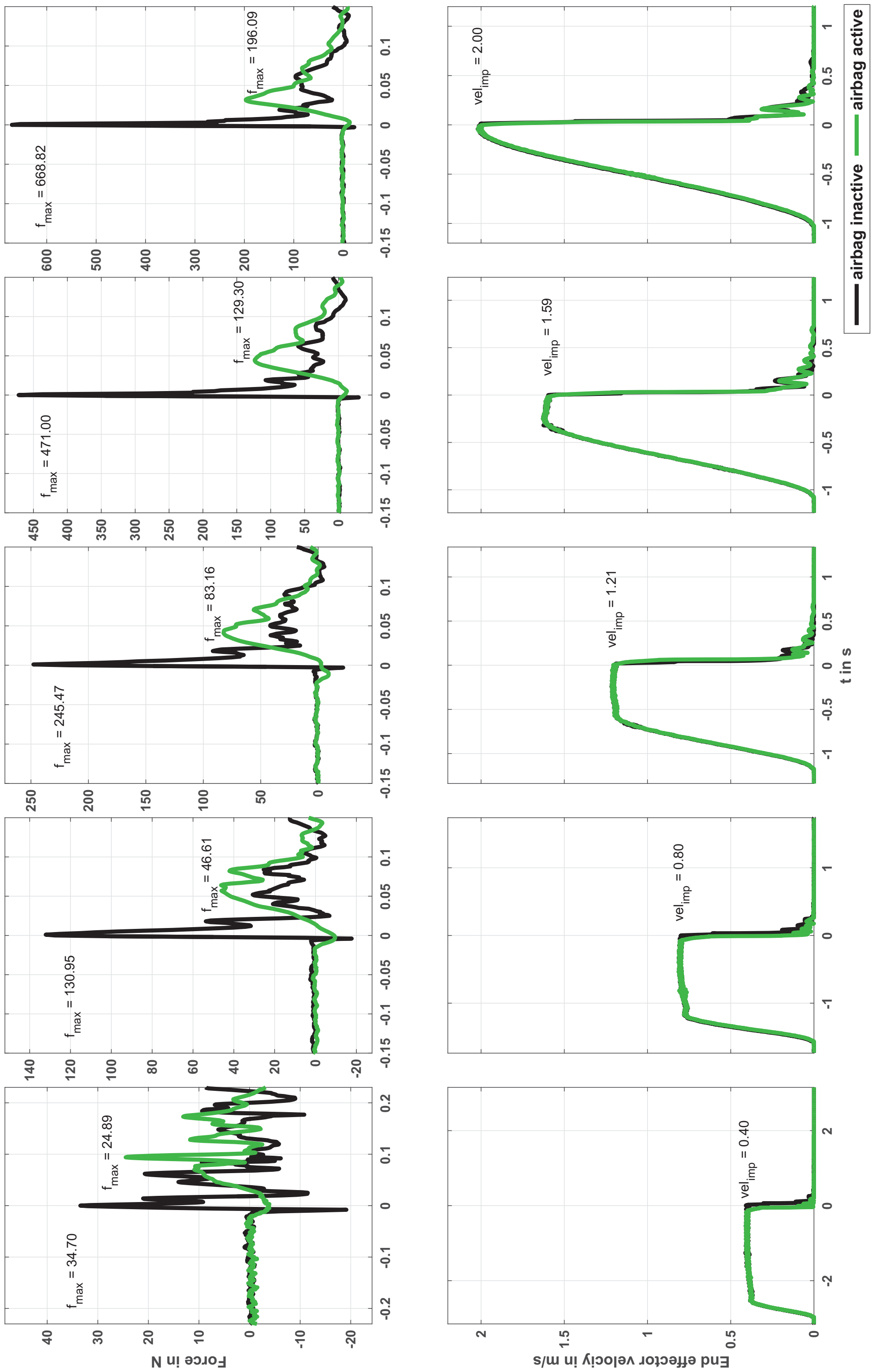


**Figure 7.6:** Upper row: Impact forces in N for a disabled (black line) and for an enabled (green line) safety module. It is clearly visible that the forces of an activated airbag are much lower than collisions with an inactive airbag; bottom row: the associated velocity of the end-effector. Image ©IEEE

also be seen, that the impact motions of the uncovered and covered tool are exactly the same motions, such that a comparison of both is feasible. The black solid lines depict the velocity of the motion with the uncovered tool and the green lines depict the velocity with an inflated airbag. Having a look at the forces, the difference between the covered tool and uncovered tool is $\Delta f_{max} = 9.81\ N$. Both maximum values are remarkably under the limits and do not distinguish much. As can be seen, in the force measurements at first a negative amplitude can be found. This phenomenon can also be found in other collision experiments, which could be a first impact reaction in combination with the momentum of the coupled system, but are not further analyzed. In the second column the impact was executed with a velocity of 0.8 m/s, as can be seen in the bottom row. At this point, a large difference between the two forces is obtained. The difference between these two forces is $\Delta f_{max} = 84.34\ N$. While the forces of the uncovered tool exceed the limits of the ISO standards, the forces of the covered tool are still under the limits. In the following experiment with higher velocities depicted in Figure 7.6 column three to five, the differences of the forces are $\Delta f_{max}(\dot{x} = 1.21\ \mathrm{m/s}) = 162.31\ N$, $\Delta f_{max}(\dot{x} = 1.59\ \mathrm{m/s}) = 341.70\ N$, $\Delta f_{max}(\dot{x} = 2.00\ \mathrm{m/s}) = 472.73\ N$.

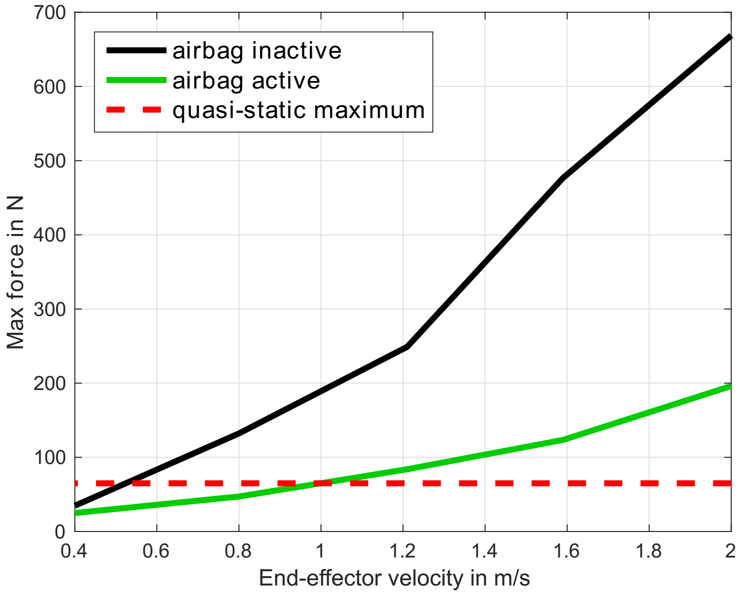


**Figure 7.7:** Maximum force measurements during contact with the crash-test dummy depending on the velocity of the robot end-effector. The black solid line depicts the maximum absolute measured contact force with an inactive safety module. The green solid line depicts the maximum force with an active safety module. Image ©IEEE

In Figure 7.7, a comparison of the slope depending on the velocities is illustrated. The black solid line depicts the forces over velocity of the uncovered tool. The green line depicts the forces of the covered tool over the velocity of the robot. To compare these values with the limits of the ISO the red dashed line depicts the maximum values, which are allowed for impact forces to the head. In Table 2.2, these limits are given by $F_{max} = 65$ N. Thus, the duration of the collision is less than half a second, the right column would be considered, where transient limits are listed. Transient impacts do have a multiplier, which can be found in the last row for the neck. As explained in Section 2.4.2 the limits can be doubled for transient contact, except the head is hit by an impact. Finally, in this experiments the *force multiplier* is *not applicable*.

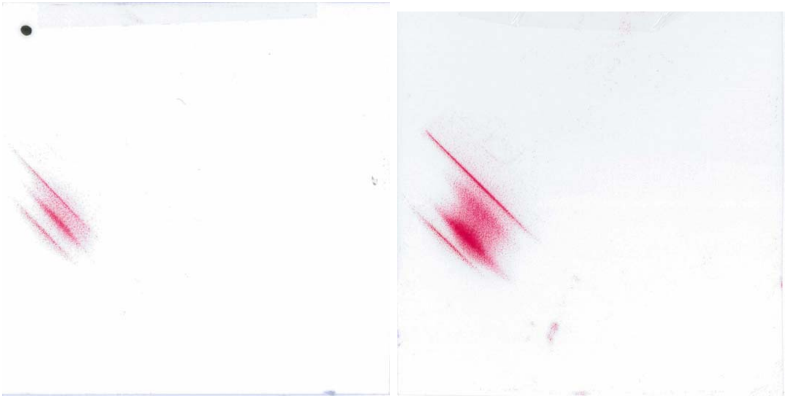
As a result of the force experiments, it is shown that the addition of an airbag can lower the impact forces enormously. The uncovered tool exceeds the limits at a Cartesian velocity of $0.5\,\mathrm{m/s}$, whereas the covered tool exceeds this limits only at $1\,\mathrm{m/s}$. This is an increase of

200 % of the maximum velocity. Additionally, it can be observed that the slope of the curves vary widely, such that the uncovered tool is increasing far over the limits, while the covered tool allows for much higher velocities, when for example other body regions are considered.

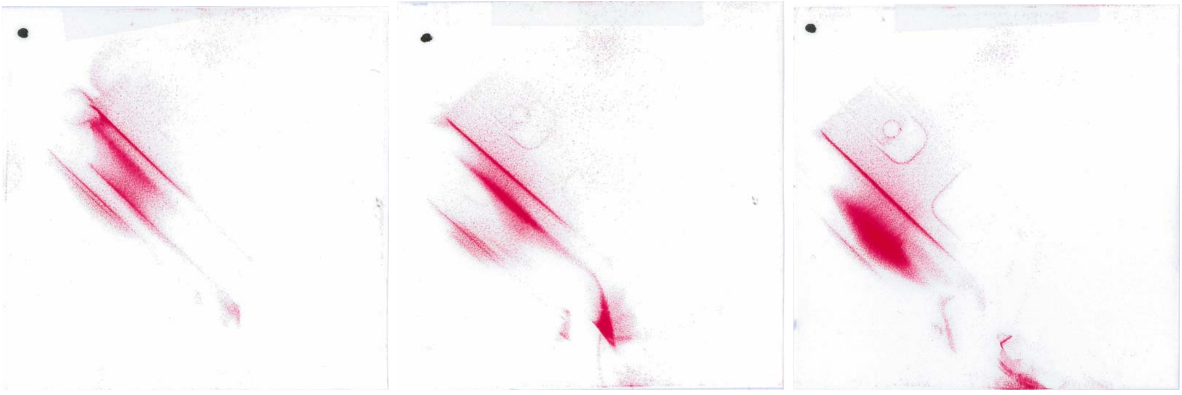
In the next section, the maximum pressure of an impact to the surface of the crash test dummy is analyzed.

### 7.3.2 Maximum Pressure Analysis

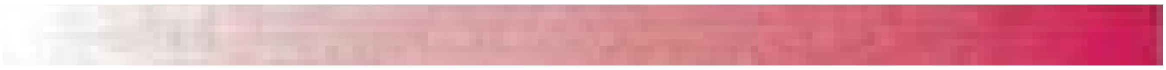
The second, and more pivotal, value is the pressure to the human skin. In the ISO/TS 15066 standards Table 2.2, a maximum pressure is defined depending on the specific body region. To obtain the the maximum pressure, there are two possibilities. The more expensive one is a sensor skin with electronic evaluation. An example of this technology is the Tactilus [2]. However, as many companies prefer the low cost version and the competent authorities accept this version, the prescale measurement films are used[3].



(a) 0.4 m/s (b) 0.8 m/s



(c) 1.2 m/s (d) 1.6 m/s (e) 2.0 m/s



(f) Range 250 N/cm$^2$ - 1000 N/cm$^2$

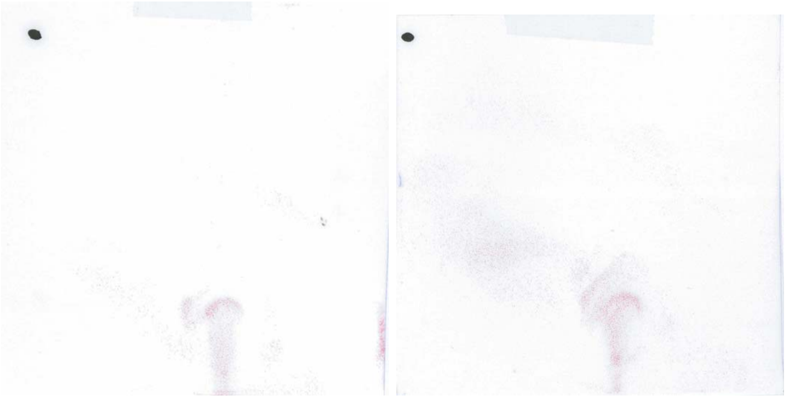
**Figure 7.8:** Experimental results of pressure measurements without an airbag. From (a) to (e) all five collisions with different velocities are illustrated. The range (f) is from 250 N/cm$^2$ to 1000 N/cm$^2$ and shows a color scale to compare with the films from the experiments. Already at the lowest velocity it can be seen that the strength of the color is close to the end of the scale, which means that even at a velocity of 0.4 m/s, the limits of the ISO/TS 15066 standards are widely exceeded.

---

[2] `http://www.sensorprod.com/glossary/matrix-pressure/matrix-pressure.php`

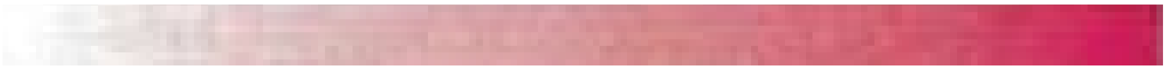
[3] `http://www.fujifilm.com/products/prescale/`

There exist two different types of measurement film, but finally the operating principle is the same. Is a pressure given to the film micro capsules, these are going to be damaged. If they are broken, a red colored fluid is leaking. Depending on the strength of the pressure, the red color is getting darker. In comparison with a scale, the maximum given pressure can be determined. The measurement film have to be attached, where the strongest impact can be assumed. Due to the fact the forehead is of interest, the film has to be attached very carefully, such that no micro capsules are damaged.



(a) 0.4 m/s (b) 0.8 m/s

(c) 1.2 m/s (d) 1.6 m/s (e) 2.0 m/s



(f) Range 50 N/cm$^2$ - 250 N/cm$^2$

**Figure 7.9:** Experimental results of pressure measurements with an airbag. From (a) to (e) all five collisions with different velocities are illustrated. The range (f) is from $50$ N/cm$^2$ to $250$ N/cm$^2$ and shows a color scale to compare with the films from the experiments. At the lowest velocity it can be seen that the strength of the color is close to the lower bound of the scale, which means that with a velocity of $0.4$ m/s, the pressure is far under the limit of the ISO/TS 15066 standards. Even with a velocity of $1.6$ m/s only appears a very light/bright color. With an airbag the velocity is not bounded by the pressure, as it is without an airbag, but by the force.

In Figure 7.8, the experimental results of the uncovered tool mounted to the robot are depicted. Preliminary investigations were executed to determine the needed range of the pressure for each experiment, since the fuji films are only available in specific ranges. The maximum pressure allowed regarding the ISO/TS 15066 is at the forefinger pad, with a value of $p_s = 300$ N/cm$^2$. As in the preliminary investigation evaluated, the impact pressure would be higher than $250$ N/cm$^2$. Therefore, the final range for the experiments is chosen to be at $250$ N/cm$^2$ to $1000$ N/cm$^2$. In Figure 7.8 (a), the impact was executed with a velocity of $0.4$

m/s, which is a slow motion for a robot in an industrial application and would often be uneconomic. On the pressure measurement film the edge of the tool is clearly visible. The color of the micro capsule reaction is dark red, which proves that the pressure to the dummy's skin is much higher than $300$ N/cm$^2$, rather close to $1000$ N/cm$^2$. The conclusion is that even at this slow robot velocity the human is not allowed to be hit by the robot. In Figure 7.8 (b)-(e), it is visualized that the edge of the tool is getting that deep into the dummy's skin, that the structure of the tool is already apparent. Therefore, it can be assumed that severest injuries may occur. It is urgently advisable to not operate such fence-less work cell, even with the velocity of the lowest in this experiments, at $0.4$ m/s.

In Figure 7.9, exactly the same experiments were executed with an airbag inflated and covering the tool. The preliminary investigations resulted in a probable range of $50$ N/cm$^2$ to $250$ N/cm$^2$ that are finally used for the impact experiments with an airbag attached. The lowest maximum pressure is given at $100$ N/cm$^2$ in the region of the head. As can be seen in Figure 7.9 (a)-(d), the maximum pressure is around $100$ N/cm$^2$, which is an enormous improvement of safety in comparison to a non-covered tool. Only with a Cartesian velocity of $2$ m/s, a little darker red color can be observed, which indicates that the limits are exceeded. The conclusion is, regarding only the pressure measurements that the robot is allowed to move at a very high and efficient velocity.

To sum up, the maximum allowed velocity with the given setup is much less than $0.4$ m/s limited by the maximum pressure, whereas the motion of the robot with a covered tool is limited at a velocity of $1$ m/s. This limit is given by the maximum force measurements. Finally, it can be assumed that the pressure is more important value as objects from the manufacturing are often sharp edged caused by the production process.

Since the *Robotic Airbag* needs time to inflate and deflate, and common robotic controls do not support online velocity adaptation, in the next section a benchmark analysis is presented, to proof the efficiency of an airbag attached to the robot.

### 7.3.3 Performance Analysis

The airbag system offers protection from severe injuries by a collision with a robot tool or object. The velocity can be improved up to five times of a motion without the airbag. But the airbag needs time to inflate and deflate. Since industrial robots do not offer online via-point trajectory generation as described in Section 3.3.1 the robot has to stop at any via point with an airbag attached. The efficiency of the task is depending on the allowed velocity, determined by ISO/TS 15066, and the total distance of motion. The benchmark of time saving and time loss can be approximately calculated as

$$t_b(s, v_{na}, v_{wa}) \approx \Delta x \cdot \frac{v_{na} - v_{wa}}{v_{na} v_{wa}} + t_{inf} + t_{def} \tag{7.1}$$

where $t_b$ is the time difference for an application, where the allowed velocity $v_{na}$ with no airbag and the allowed velocity $v_{wa}$ with an airbag attached is given by the ISO/TS 15066. The values $t_{inf}$ and $t_{def}$ are the needed inflation time and the deflation time of the airbag.

Having a look at a Pick & Place application, where an object is picked from a table and placed somewhere else. There are four main phases of robot motions. The first is, moving from the initial position to the pick pre-position. During the motion, the airbag is inflated to ensure safety. In Table 7.1, it is illustrated, how the application can be improved in its performance. It is assumed that the inflation time, as well as the deflation time, is given by $t_{inf} = t_{def} = 0.5$s. Depending on the distance, which can be found in the left column, and depending on the velocity, determined in experiments with the attached tool, which can be found in the top row, a reduction of duration can be achieved including an active airbag, as presented. The

| | | Velocity $v_n a$ in m/s | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 0.05 | 0.10 | 0.15 | 0.20 | 0.25 | 0.30 | 0.35 | 0.40 |
| Distance $\Delta x$ in m | 0.1 | -00.90 | 00.10 | 00.43 | 00.60 | 00.70 | 00.77 | 00.81 | 00.85 |
| | 0.2 | -02.80 | -00.80 | -00.13 | 00.20 | 00.40 | 00.53 | 00.63 | 00.70 |
| | 0.3 | -04.70 | -01.70 | -00.70 | -00.20 | 00.10 | 00.30 | 00.44 | 00.55 |
| | 0.4 | -06.60 | -02.60 | -01.27 | -00.60 | -00.20 | 00.07 | 00.26 | 00.40 |
| | 0.5 | -08.50 | -03.50 | -01.83 | -01.00 | -00.50 | -00.17 | 00.07 | 00.25 |
| | 0.6 | -10.40 | -04.40 | -02.40 | -01.40 | -00.80 | -00.40 | -00.11 | 00.10 |
| | 0.7 | -12.30 | -05.30 | -02.97 | -01.40 | -01.10 | -00.63 | -00.30 | -00.05 |
| | 0.8 | -14.20 | -06.20 | -03.53 | -02.20 | -01.40 | -00.87 | -00.49 | -00.20 |
| | 0.9 | -16.10 | -07.10 | -04.10 | -02.60 | -01.70 | -01.10 | -00.67 | -00.35 |
| | 1.0 | -18.00 | -08.00 | -04.67 | -03.00 | -02.00 | -01.33 | -00.86 | -00.50 |
| | 1.1 | -19.90 | -08.90 | -05.23 | -03.40 | -02.30 | -01.57 | -01.04 | -00.65 |
| | 1.2 | -21.80 | -09.80 | -05.80 | -03.80 | -02.60 | -01.80 | -01.23 | -00.80 |
| | 1.3 | -23.70 | -10.70 | -06.37 | -04.20 | -02.90 | -02.03 | -01.41 | -00.95 |
| | 1.4 | -25.60 | -11.60 | -06.93 | -04.60 | -03.20 | -02.27 | -01.60 | -01.10 |
| | 1.5 | -27.50 | -12.50 | -07.50 | -05.00 | -03.50 | -02.50 | -01.79 | -01.25 |
| | 1.6 | -29.40 | -13.40 | -08.07 | -05.40 | -03.80 | -02.73 | -01.97 | -01.40 |
| | 1.7 | -31.30 | -14.30 | -08.63 | -05.80 | -04.10 | -02.97 | -02.16 | -01.55 |
| | 1.8 | -33.20 | -15.20 | -09.20 | -06.20 | -04.40 | -03.20 | -02.34 | -01.70 |
| | 1.9 | -35.10 | -16.10 | -09.77 | -06.60 | -04.70 | -03.43 | -02.53 | -01.85 |
| | 2.0 | -37.00 | -17.00 | -10.33 | -07.00 | -05.00 | -03.67 | -02.71 | -02.00 |

**Table 7.1:** Benchmark analysis for airbag usage. The values in the table are times in seconds the cycle time of a motion can be reduced for a given distance and velocity allowed with no airbag attached.

green background illustrates that at this constellation a reduction of time is achieved, the orange background indicated that the deployment of an airbag provides no improvement for this cycle in the application. For example the robot is able to move $0.2$ m/s and the moving distance is $0.5$ m the reduction of time in this cycle is $1$ sec. Note that this considerations are based on the fact that the robot has to stop during every inflation or deflation period. The presented online velocity adaptation from Chapter 3 provides an improvement of efficiency using an active Robotic Airbag, especially, for movements with a duration of more than 1.2 seconds. This is by means the sum of the deflation and inflation time.

In the following section, the usage of such airbag technology is discussed.

## 7.4 Discussion

The presented solution to ensure safety, even if a sharp edged object is carried by the robot, offers new performance possibilities with collaborative robotic systems. The main issue of an object limitation can partially be eliminated by open airbag and motions in the direction of the covered parts. In the experiments, the obtained impact forces are analyzed. The used force torque sensor is based on strain gauge technology, which has proven characteristics of a first order low pass filter, and could influence the measurements, by filtering the curves. Having a look at the results, it is obviously that the behavior of the green curve has a much lower time behavior, which means, that the measurements of the green curves are influenced minimal, but the impacts of the uncovered tool all the more. It can be assumed that the forces of an uncovered tool are much higher, but the values of the covered tool are not. Another fact could be more essential, namely the mounting of the force sensor at the robot side. Dynamic forces given by inertia are not completely measurable. The measurements are also representative such that the results just apply to exactly the same setup. The dynamics of the robot, i.e. the given inertia parameterized by the mass of the tool, the mass of the carried object, and the collision detection and reaction of the overall system, do influence the final results.

The inflation and deflation of the airbag needs a given time. This is limited by the flow volume by time. An important characteristic is the chamber design, which enables the functionality of the airbag. The consequence is a stiff cylinder around the tools and object, whereas, the stiffness of an inflated airbag allows for an open bottom, whilst it also provides safety in that direction, for example clamping with a down moving robot. Internal sensors enable collision detection to stop the robot in case a human gets hit by the airbag.

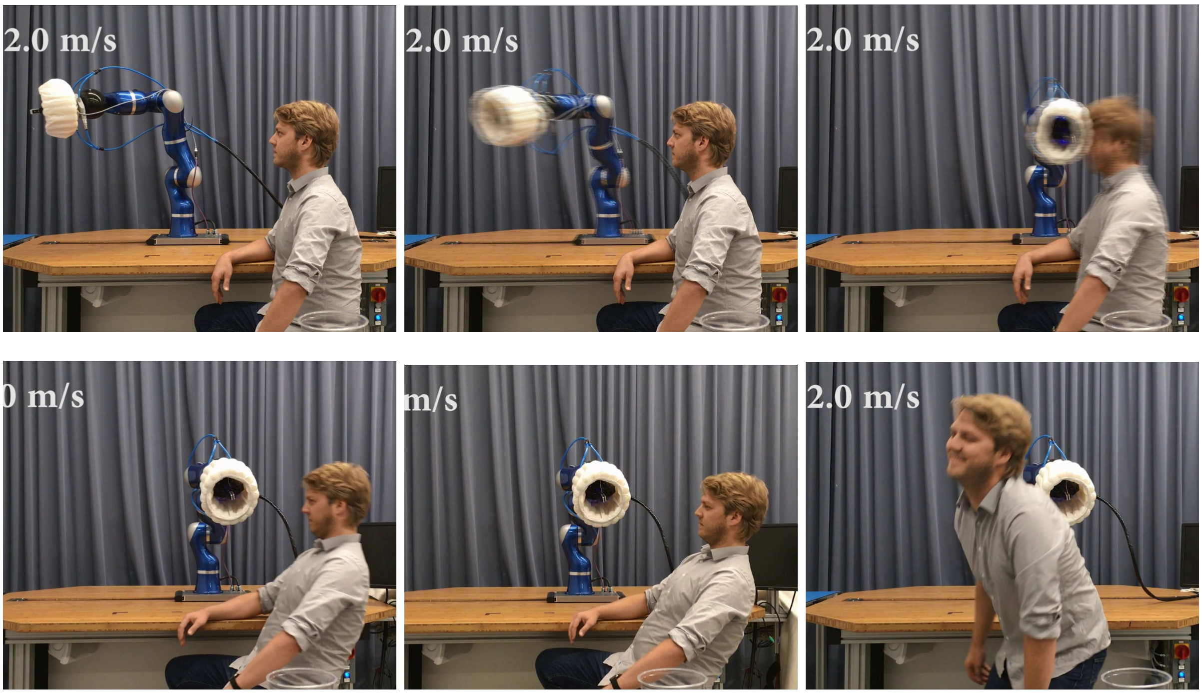


**Figure 7.10:** Video scene experiments at the author. Upper left image illustrates the deflated airbag, which inflates at the beginning of the motion. The tests were executed with a collision at the forehead, up the the maximum velocity of the robot without any injury.

A benchmark analysis was discussed above and the usefulness of the airbag is proven, such that the use of an airbag increases the performance enormously. The integrated user information via shining in colors of the airbag, see Figure 7.5, allows to signalize status information like safety errors or danger information, which improves safety in collaborative applications, too.

The results of the experiments are constitute values of exact this setup, such that different setups can generate different results. But generally it can be claimed, that the experiments are representative for all other applications that uses tools and objects with sharp edges. It is well known and proven for years that airbags in cars protect humans from severe injuries in a car accident and the same properties are expected to be in a collision with a robot. This means that an airbag for robots can revolutionize collaborative robotics in the near future. An international jury of robotic experts decided to award this application with the KUKA Innovation Award 2017.

The experiments with a crash test dummy proved the functionality of the *Robotic Airbag*. In order to achieve a feeling of the impact of such system, the author of this thesis did experiments at himself. The impacts took place at the same position as the dummy experiments were executed. Thus, an experimental validation of impacts with real human subjects is not executable, here just a short note of the impression of such impact can be formulated.

From the experiments it can be assumed that the pressure to the skin is lower than the limits of the ISO/TS 15066 so no injuries to the skin were noted. The maximum impact force was given at a velocity of 1 m/s. The impact with the head at this velocity was low and didn't feel bad at all. Up to a velocity of 1.6 m/s, the impact was ok and did not hurt or felt bad. At a velocity of 2 m/s a hit was clearly noticeable. This impact can be considered as unpleasant or uncomfortable, and can be compared as a hit in the boxing ring with an amateur boxer. Finally, also at a collision with the maximum robot velocity no injury could be observed.

In summary, this novel technology enables a lot of new applications, and the increased possible speed of the robot with an airbag attached, improves the efficiency of such systems enormously. The additional pressure sensors can be used for collision detection, and with an improved sensor data the robot can be stopped before the airbag is a cushion between the sharp object, as in the experiments only the internal sensors of the robot are used. With this extension of sensors and the integrated collision detection, the airbag technology can be attached to every robot on every link, such that many more non-collaborative robots can be made collaborative.

# Chapter 8

# Conclusion and Outlook

This thesis presents novel planning methods and tools, which improve the efficiency in collaborative robot applications, while maintaining safety. The on-sight analysis in Chapter 2 gives an insight about existing manual assembly applications and possible improvements of those applications using collaborative robot technology. From these insights the collaborative workcell with an extended workspace is presented, which serves as a research platform to develop novel and innovative technology in the field of human-robot collaboration.

A major challenge in robotics research is the intuitive control of robots. High set-up times and complicated handling of robots reduce the flexibility of robotic systems. Chapter 3 presents a novel method to intuitively control the robot. The SLERP algorithm enables movements on the shortest path possible, while enabling the limitation of velocities. As the output signal results in a step behavior in the velocities, a filter design is developed to smoothen the trajectory and enable an applicable control input to the robot. This method is extended to an online via-point trajectory generator, which offers the opportunity to adjust the auxiliary points online, and, additionally, the adaptation of velocity limits can be performed online. Human motions are mirrored in experiments, considering human-in-the-loop constraints, i.e., maximum velocities, which increases the performance and the intuitiveness of the collaborative application, remarkably. The experiments presented in Chapter 3 prove the usability on real robotic systems in common industrial applications.

Since the collaboration of humans and robots is becoming increasingly closer, novel methods for safe human-robot collaboration had to be developed. Chapter 4 addresses the dynamic behavior of human arm motions in typical industrial scenarios. Arm motion experiments for three different motions are presented that serve as basis for the development of a simplified dynamic model of a human arm. The simulation of human arm motions into the direction of the robot, enables the calculation of the duration to a potential collision. This information is used to increase the velocity of the operating robot, depending on the maximum deceleration of the robot and the current position and velocity of the human hand. Obviously, a robot movement on the direct path is not necessarily the ideal solution, which is shown in Chapter 5. Therefore, the robot path is separated into multiple segments, using auxiliary points to enable a segment-wise definition of the path, with respect to human-in-the-loop constraints. The aim is to achieve an optimal motion, i.e., a minimum-time motion to decrease cycle-times. The nonlinear programming problem is solved using a SQP algorithm. The results provide insights regarding the improvement on performance on an optimal path considering human-in-the-loop constraints. Since the optimization problem is solved offline, which makes is not applicable for online operations, generalization methods are developed in Chapter 6. As the obtained motion information from the optimizer is limited to Cartesian positions, first a trajectory has to be calculated containing velocities and accelerations. To ensure minimal time trajectories, an exponential function based trajectory generation is developed. For the generalization of movements, a DMP based approach is presented, which enables flexible adaptation of the desired initial and the desired goal position. The focus of the generalization is on the compliance of constraints, where methods to fulfill the requirements are developed, and experimentally analyzed. Especially, the compliance of velocity limits and spatial constraints, which is necessary to ensure safe human-robot collaboration, was proven in simulation and evaluated experimentally on a real robot.

Besides control methods, an improvement can also be achieved considering hardware impairments. Sharp edged robot tools and objects pose a potential threat for injuries in case of a collision between humans and robots. As an innovative technology to solve this problem, an end-effector airbag is developed and presented in Chapter 7. The Robotic Airbag enables human-robot collaboration for various existing robot tools and objects, without having an effect on the functionality of the overall robotic system. For this purpose, the Robotic Airbag is inflated with pressured air and covers both, tools and carried objects, always before the robot starts a motion. At the end of the motion, or during a manipulation task, the airbag is deflated and uncovers both, the tool and the object. Experiments regarding peak pressure and peak force were analyzed and evaluated, under consideration of the ISO/TS-15066. It has been proven that by using a conventional industrial gripper in combination with the Robotic Airbag, the permissible velocity limits could be extended up to five times compared to omitting the Robotic Airbag. Hence, cycle times in manufacturing processes are significantly reduced, and, thereby, the productivity is increased considerably. The amortization of such system is reached very quickly, as extensive enclosed work cells are unnecessary and a continuous operation of the robot is guaranteed even though humans are located in the workspace of the robot.

To sum up, the thesis presents effective methods and tools to improve the performance of a collaborative robotic system, remarkably.

## Outlook

In this thesis, different topics are treated, including the control of the robot, the integration of human-in-the-loop methods, and hardware developments to improve the performance of robots in industrial scenarios. Novel approaches are introduced and have to be improved in future research. The generation of trajectories online is a large field in robotics research, which enables a flexible and reactive motion generation for robots. The presented method of mirroring human arm motions, enables a very flexible and reactive motion generation, but requires an integration of acceleration limitation. The filtering of quaternions, and the resulting online trajectory generation methods, require an acceleration limitation for fast motions because the behavior of the robot can appear roughly without. Additionally, the path following behavior could be affected, which can be referred to the physical limits of the actuated system. This methodology necessitates robust vision or sensor systems, which enables robust information for the adaptation of the path online.

A very important future trend is the sensing of humans in shared workspaces. Many novel sensor technologies and algorithms are presented every year. Especially, in the field of industrial human robot collaboration, many challenges still exist, to enable an efficient integration of robotic systems in shared workspaces. In the area of human-in-the-loop control, future research will focus on the improvement of the human motion simulation, to enable an increasing precision in human motion projection. Higher accuracy will further improve the performance of robots within collaborative applications. Especially, the treatment of whole body motions are of interest in future research, which includes the dynamic behavior of the human body. Noteworthy here, is the integration of machine learning algorithms, including smart data, which are reducing the possible collision error to a very small proportion. Additionally, the psychological improvement of the working conditions can be integrated, which was previously introduced in collborative robotics research [38, 110, 111].

Nonlinear programming is a well known and continuously used method in robotics path planning. A future challenge will be the integration of more human based data, impressions, psychological behavior, or physiological data into the optimization routines. Anyhow, this will significantly improve the direct interaction between humans and robots, and, in particular, the trust in operating machines, not being in danger. A more challenging problem is the generalization of motions, in a very flexible manner, maintaining the constraints and the human intention, and keeping the efficiency of the application as appropriate as possible. The presented methods in this thesis provide a perfect base for generating motions in real time. A next step has to be the integration of these methods into industrial systems, proving the improvement of performance in field tests.

Certainly, the integration of the Robotic Airbag will be part of the field tests, as the improvement of efficiency is proven in this thesis. Obviously, this technology does not solve every problem, like the robot side collision detection, or the usage of the Robotic Airbag with big components. Therefore, novel methods, or combinations, of the airbag technology and additional sensor technologies seem very promising. The Robotic Airbag will further be developed to a market ready version to accelerate human-robot collaboration in industrial applications.

# Appendix

The modeling of the human arm, considering low computation time, and, therefore, limiting the degrees of freedom to three, is presented in Chapter 4. To allow for a proper re-implementation of the methods, the following content denotes the MATLAB code for the dynamic model. The values m_up and m_lo denote the mass of the upper arm, and the lower arm, respectively. The first representation is the mass matrix of the human arm model, which is designed as follows.

```
function M = fcn(q,lower_arm,upper_arm)

g = -9.81;

m_up = 4;
m_lo = 3;

q1=q(1);
q2=q(2);
q3=q(3);

M = zeros(3,3);

t1 = cos(q2);
t2 = upper_arm * t1;
t4 = cos(q3);
t6 = sin(q3);
t7 = sin(q2);
t9 = t4 * t1 - t6 * t7;
t10 = t9 * m_lo;
t16 = m_lo * lower_arm;
t19 = lower_arm ^ 2;
t24 = t1 ^ 2;
t26 = upper_arm ^ 2;
t30 = t4 * upper_arm;
t32 = t30 * t16 / 0.2e1;
t33 = t6 ^ 2;
t42 = m_lo * t19 / 0.3e1;
t46 = t32 + t42;
M(1,1) = -(-t2 * m_lo - t10 * lower_arm / 0.2e1) * upper_arm * t1
         +(t2 * t16 / 0.2e1 + t10 * t19 / 0.3e1) * t9 + t24
         * m_up * t26 / 0.3e1;
M(1,2) = 0.0e0;
M(1,3) = 0.0e0;
M(2,1) = 0.0e0;
M(2,2) = t32 + t33 * t26 * m_lo + (t16 / 0.2e1 + t30 * m_lo) * t4 * upper_arm
       + t42 + m_up * t26 / 0.3e1;
```

```
M(2 ,3) = t46;
M(3 ,1) = 0.0e0;
M(3 ,2) = t46;
M(3 ,3) = t42;
```

The following MATLAB code describes the Coriolis and centripetal forces, where l_oberarm = 0.2870 m and l_unterarm = 0.4160 m, denote the length of the upper arm, and the lower arm, respectively. Furthermore, the length of the lower arm is a concatenation of the lower arm and the hand.

```
function C = fcn(q,dq,l_oberarm,l_unterarm)

g = -9.81;

m_ob = 4;
m_un = 3;

q1=q(1);
q2=q(2);
q3=q(3);

dq1=dq(1);
dq2=dq(2);
dq3=dq(3);


C = zeros(3,3);

t1 = sin(q2);
t2 = t1 * dq2;
t3 = l_oberarm ^ 2;
t5 = cos(q2);
t9 = sin(q3);
t10 = t9 * t5;
t11 = cos(q3);
t12 = t11 * t1;
t13 = t10 + t12;
t14 = -dq2 - dq3;
t16 = t11 * t5;
t17 = t9 * t1;
t18 = t16 - t17;
t19 = t18 * m_un;
t20 = l_unterarm ^ 2;
t21 = t19 * t20;
t24 = l_oberarm * t5;
t31 = m_un * l_unterarm;
t36 = t9 * dq3;
t47 = t10 * dq1 + t12 * dq1;
t48 = m_un * t47;
t58 = -t16 * dq1 + t17 * dq1;
t69 = t24 * t48 * l_unterarm / 0.2e1;
t71 = m_un * t20;
t73 = -t18 * t47 * t71 / 0.3e1;
```

```
t76 = -t13 * t58 * t71 / 0.3e1;
t84 = t11 * l_oberarm;
t87 = t9 * l_oberarm;
t90 = -t31 * t47;
t105 = t9 * t3;
t106 = m_un * t14;
t109 = -m_un * t14;
t118 = t87 * t106 * l_unterarm / 0.2e1;
C(1,1) = -0.2e1 / 0.3e1 * t2 * m_ob * t3 * t5 + t13 * t14 * t21 / 0.3e1
         + (-t24 * m_un - t19 * l_unterarm / 0.2e1) * l_oberarm * t2
         + (t24 * t31 / 0.2e1 + t21 / 0.3e1) * (-t36 * t5 - t12 * dq2
         - t11 * dq3 * t1 - t10 * dq2);

C(1,2) = (-t24 * t48 - t19 * t47 * l_unterarm / 0.2e1) * t11 * l_oberarm
                 + (-t24 * m_un * t58 - t19 * t58 * l_unterarm / 0.2e1) * t9
                 * l_oberarm - t69 + t73 + t76;

C(1,3) = -t69 + t73 + t76;
C(2,1) = t5 * t1 * dq1 * m_ob * t3 / 0.3e1 - (-t84 * m_un * t47 - t87
                * m_un * t58 + t90 / 0.2e1) * l_oberarm * t5 + (-t84
                * t90 / 0.2e1 + t87 * t31 * t58 / 0.2e1 + t48 * t20
                / 0.3e1) * t18;
C(2,2) = t105 * t106 * t11 + (t84 * t109 + t109 * l_unterarm / 0.2e1) * t9
                * l_oberarm + t118 - (t31 / 0.2e1 + t84 * m_un) * t9 * dq3
                * l_oberarm + t105 * m_un * t11 * dq3;
C(2,3) = t118;
C(3,1) = t31 * t47 * l_oberarm * t5 / 0.2e1 + t48 * t20 * t18 / 0.3e1;
C(3,2) = t109 * l_unterarm * t9 * l_oberarm / 0.2e1 - t31 * t36
                * l_oberarm / 0.2e1;
C(3,3) = 0.0e0;
```

The following vector describes the gravity of the dynamic model.

```
function G = fcn(q,l_oberarm,l_unterarm)

G = zeros(3,1);

g = -9.81;

m_ob = 4;
m_un = 3;

q1=q(1);
q2=q(2);
q3=q(3);

t1 = cos(q2);
t3 = sin(q1);
t12 = cos(q3);
t14 = sin(q3);
t15 = sin(q2);
t17 = t12 * t1 - t14 * t15;
t25 = cos(q1);
```

```
t36 = (-t12 * t15 - t14 * t1) * t25 * g;
t45 = m_un * l_unterarm * t36 / 0.2e1;
G(1,1) = -t1 * m_ob * l_oberarm * t3 * g / 0.2e1 - l_oberarm * t1
                    * m_un * t3 * g - l_unterarm * t3 * g * t17 * m_un / 0.2e1;
G(2,1) = -m_ob * l_oberarm * t15 * t25 * g / 0.2e1 + t12 * l_oberarm
                    * m_un * t36 + t14 * l_oberarm * m_un * t17 * t25 * g + t45;
G(3,1) = t45;
```

Tx describes the transformation matrix, from joint space to task space (Cartesian space).

```
function Tx = fcn(q, l1, l2)

Tx1 = - l2*(cos(q(2))*sin(q(3)) + cos(q(3))*sin(q(2))) - l1*sin(q(2));
Tx2 = l1*cos(q(2))*sin(q(1)) - l2*(sin(q(1))*sin(q(2))*sin(q(3))
      - cos(q(2))*cos(q(3))*sin(q(1)));
Tx3 = l2*(cos(q(1))*cos(q(2))*cos(q(3)) - cos(q(1))*sin(q(2))*sin(q(3)))
      + l1*cos(q(1))*cos(q(2));


Tx = [Tx1;Tx2;Tx3];
```

Jx describes the jacobian matrix to enable impedance control, transform the forces into torques, and transform Cartesian velocities into joint velocities.

```
function Jx = fcn(q, l1, l2)

Jx1= [ 0,
       - l2*(cos(q(2))*cos(q(3)) - sin(q(2))*sin(q(3))) - l1*cos(q(2)),
       - l2*(cos(q(2))*cos(q(3)) - sin(q(2))*sin(q(3)))];

Jx2= [ l2*(cos(q(1))*cos(q(2))*cos(q(3)) - cos(q(1))*sin(q(2))*sin(q(3)))
       + l1*cos(q(1))*cos(q(2)),
     - l2*(cos(q(2))*sin(q(1))*sin(q(3)) + cos(q(3))*sin(q(1))*sin(q(2)))
       - l1*sin(q(1))*sin(q(2)),
     - l2*(cos(q(2))*sin(q(1))*sin(q(3)) + cos(q(3))*sin(q(1))*sin(q(2)))];

Jx3= [ l2*(sin(q(1))*sin(q(2))*sin(q(3)) - cos(q(2))*cos(q(3))*sin(q(1)))
       - l1*cos(q(2))*sin(q(1)),
     - l2*(cos(q(1))*cos(q(2))*sin(q(3)) + cos(q(1))*cos(q(3))*sin(q(2)))
       - l1*cos(q(1))*sin(q(2)),
     - l2*(cos(q(1))*cos(q(2))*sin(q(3)) + cos(q(1))*cos(q(3))*sin(q(2)))];


Jx = [Jx1;Jx2;Jx3];
```

# Bibliography

[1] "Wandlungsfaehige Automatisierungsloesungen mit MRK," https://mav.industrie.de/wp-content/uploads/K/U/KUKA-2017_03_22_mav_Wandlungsf%C3%A4hige_Automatisierungsl%C3%B6sungen_pdf.pdf, 2017, [Online; accessed 26-March-2018].

[2] "DARPA Robotics Challenge (DRC) (Archived)," https://www.darpa.mil/program/darpa-robotics-challenge, 2015, [Online; accessed 26-March-2018].

[3] C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schäffer, B. Brunner, H. Hirschmuller, S. Kielhofer *et al.*, "A humanoid two-arm system for dexterous manipulation," in *International Conference on Humanoid Robots*. IEEE, 2006, pp. 276–283.

[4] M. Grebenstein, A. Albu-Schäffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, R. Gruber, S. Haddadin, U. Hagn, R. Haslinger *et al.*, "The dlr hand arm system," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 3175–3182.

[5] S. Curtis, "Japanese robotics firm unveils 'world's strongest robot'," https://www.telegraph.co.uk/technology/technology-video/11666326/Japanese-robotics-firm-unveils-worlds-strongest-robot.html, 2015, [Online; accessed 26-March-2018].

[6] ISO, "ISO 10218-1:2012 Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots," http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=41571, 2016, [Online; accessed 13-September-2016].

[7] ——, "ISO 10218-2:2011 Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration," http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=41571, 2016, [Online; accessed 13-September-2016].

[8] ISO/TS, "ISO/TS 15066:2016 Robots and robotic devices - Collaborative robots," http://www.iso.org/iso/catalogue_detail?csnumber=62996, 2016, [Online; accessed 13-September-2016].

[9] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, "The dlr lightweight robot - lightweight design and soft robotics control concepts for robots in human environments," *Industrial Robot Journal*, vol. 34, no. 5, pp. 376–385, 2007.

[10] European Project, "SAPHARI," http://www.saphari.eu/index.php?option=com_content&view=article&id=189&Itemid=125, 2016, [Online; accessed 12-April-2018].

[11] M. Beetz, G. Bartels, A. Albu-Schäffer, F. Bálint-Benczédi, R. Belder, D. Beßler, S. Haddadin, A. Maldonado, N. Mansfeld, T. Wiedemeyer *et al.*, "Robotic agents capable of natural and safe physical interaction with human co-workers," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 6528–6535.

[12] S. Parusel, H. Widmoser, S. Golz, T. Ende, N. Blodow, M. Saveriano, K. Krieger, A. Maldonado, I. Kresse, R. Weitschat, D. Lee, M. Beetz, and S. Haddadin, "Human-Robot interaction Planning," https://youtu.be/h3OeTsaD6G4, 2014, [Online; accessed 12-April-2018].

[13] F. Steinmetz and R. Weitschat, "Skill parametrization approaches and skill architecture for human-robot interaction," in *International Conference on Automation Science and Engineering (CASE)*. IEEE, 2016, pp. 280–285.

[14] I. Asimov, "The Three Laws of Robotics ," https://en.wikipedia.org/wiki/Three_Laws_of_Robotics, 1942, [Online; accessed 26-March-2018].

[15] S. Haddadin, A. Albu-Schäffer, M. Frommberger, J. Rossmann, and G. Hirzinger, "The DLR Crash Report: Towards a standard crash-testing protocol for robot safety - Part I: Results," *International Conference on Robotics and Automation (ICRA)*, pp. 272–279, 2009.

[16] J. Versace, "A review of the severity index," SAE Technical Paper, Tech. Rep., 1971.

[17] Y. Yamada, Y. Hirasawa, S. Huang, Y. Umetani, and K. Suita, "Human-robot contact in the safeguarding space," *IEEE/ASME Transactions on Mechatronics*, vol. 2, no. 4, pp. 230–236, 1997.

[18] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Dummy crash-tests for the evaluation of rigid human-robot impacts," in *International Workshop on Technical Challenges for dependable robots in Human Environments*. Citeseer, 2007.

[19] ——, "Safety evaluation of physical human-robot interaction via crash-testing." in *Robotics: Science and Systems*, vol. 3, 2007, pp. 217–224.

[20] S. Haddadin, A. Albu-Schäffer, M. Frommberger, J. Rossmann, and G. Hirzinger, "The DLR Crash Report: Towards a standard crash-testing protocol for robot safety - Part II: Discussions," *International Conference on Robotics and Automation (ICRA)*, pp. 272–279, 2009.

[21] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "The role of the robot mass and velocity in physical human-robot interaction - Part I: Non-constrained blunt impacts," *International Conference on Robotics and Automation (ICRA)*, pp. 1331–1338, 2008.

[22] S. Haddadin, A. Albu-Schäffer, M. Frommberger, and G. Hirzinger, "The role of the robot mass and velocity in physical human-robot interaction - Part II: Constrained blunt impacts," *International Conference on Robotics and Automation (ICRA)*, pp. 1339–1345, 2008.

[23] S. Oberer and R. D. Schraft, "Robot-dummy crash tests for robot safety assessment," in *International Conference on Robotics and Automation*. IEEE, 2007, pp. 2934–2939.

[24] K. Ikuta, H. Ishii, and M. Nokata, "Safety evaluation method of design and control for human-care robots," *The International Journal of Robotics Research*, vol. 22, no. 5, pp. 281–298, 2003.

[25] S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schäffer, "A truly safely moving robot has to know what injury it may cause," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5406–5413, 2012.

[26] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Soft-tissue injury in robotics," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 3426–3433.

[27] ——, "Requirements for safe robots: Measurements, analysis and new insights," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, 2009.

[28] S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schäffer, "On making robots understand safety: Embedding injury knowledge into control," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1578–1602, 2012.

[29] R. Behrens and N. Elkmann, "Study on meaningful and verified thresholds for minimizing the consequences of human-robot collisions," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3378–3383.

[30] J. Heinzmann and A. Zelinsky, "Quantitative safety guarantees for physical human-robot interaction," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 479–504, 2003.

[31] D. Henrich and S. Kuhn, "Modeling intuitive behavior for safe human/robot coexistence cooperation," in *International Conference On Robotics and Automation*. IEEE, 2006, pp. 3929–3934.

[32] Wikipedia, "Normung," https://de.wikipedia.org/wiki/Normung, 2018, [Online; accessed 27-March-2018].

[33] T. Wilrich, "Die rechtliche Bedeutung technischer Normen," in *VDI-Spezialtag Mensch, Roboter und Sicherheit*, 2015.

[34] Wikipedia, "ISO 13849 General principles for design," https://en.wikipedia.org/wiki/ISO_13849, 2008, [Online; accessed 27-March-2018].

[35] A. Bicchi and G. Tonietti, "Fast and soft arm tactics: Dealing with the safety-performance tredeoff in robot arms design and control," *IEEE Robotics and Automation Magazine*, vol. 11, no. 2, pp. 22–33, 2004.

[36] D. Hsu, J.-C. Latombe, R. Motwani, and L. E. Kavraki, "Capturing the connectivity of high-dimensional geometric spaces by parallelizable random sampling techniques," in *Advances in randomized parallel computing*. Springer, 1999, pp. 159–182.

[37] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[38] P. A. Lasota and J. A. Shah, "Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration," *Human factors*, vol. 57, no. 1, pp. 21–33, 2015.

[39] L. Bascetta, G. Magnani, P. Rocco, R. Migliorini, and M. Pelagatti, "Anti-collision systems for robotic applications based on laser time-of-flight sensors," in *International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE/ASME, 2010, pp. 278–284.

[40] D. M. Ebert and D. D. Henrich, "Safe human-robot-cooperation: Image-based collision detection for industrial robots," in *International Conference On Intelligent Robots and Systems*, vol. 2. IEEE, 2002, pp. 1826–1831.

[41] J. Pan, I. A. Şucan, S. Chitta, and D. Manocha, "Real-time collision detection and distance computation on point cloud sensor data," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 3593–3599.

[42] M. Saveriano and D. Lee, "Point cloud based dynamical system modulation for reactive avoidance of convex and concave obstacles," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5380–5387.

[43] F. Flacco and A. De Luca, “Multiple depth/presence sensors: Integration and optimal placement for human/robot coexistence,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 3916–3923.

[44] F. Flacco, T. Kroeger, A. De Luca, and O. Khatib, “A depth space approach for evaluating distance to objects,” *Journal of Intelligent & Robotic Systems*, vol. 80, no. 1, pp. 7–22, 2015.

[45] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, “A depth space approach to human-robot collision avoidance,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 338–345.

[46] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[47] D. Kulić and E. A. Croft, “Real-time safety for human–robot interaction,” *Robotics and Autonomous Systems*, vol. 54, no. 1, pp. 1–12, 2006.

[48] L. Balan and G. M. Bone, “Real-time 3d collision avoidance method for safe human and robot coexistence,” *International Conference on Intelligent Robots and Systems (IROS)*, pp. 276 – 282, October 2006.

[49] Y. Yamamoto and X. Yun, “Coordinated obstacle avoidance of a mobile manipulator,” in *International Conference on Robotics and Automation (ICRA)*, vol. 3. IEEE, 1995, pp. 2255–2260.

[50] M. P. Polverini, A. M. Zanchettin, and P. Rocco, “Real-time collision avoidance in human -robot interaction based on kineostatic safety field,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4136–4141, 2015.

[51] A. De Luca and F. Flacco, “Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration,” in *International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2012, pp. 288–295.

[52] D. Kulić and E. Croft, “Pre-collision safety strategies for human-robot interaction,” *Autonomous Robots*, vol. 22, no. 2, pp. 149–164, 2007.

[53] M. Saveriano, F. Hirt, and D. Lee, “Human-aware motion reshaping using dynamical systems,” *Pattern Recognition Letters*, vol. 99, pp. 96–104, 2017.

[54] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, “Evolutionary artificial potential fields and their application in real time robot path planning,” in *Proceedings of the Congress on Evolutionary Computation*, vol. 1. IEEE, 2000, pp. 256–263.

[55] P. Ogren, N. Egerstedt, and X. Hu, “Reactive mobile manipulation using dynamic trajectory tracking,” in *International Conference on Robotics and Automation*, vol. 4. IEEE, 2000, pp. 3473–3478.

[56] B. Lacevic, P. Rocco, and A. M. Zanchettin, “Safety assessment and control of robotic manipulators using danger field,” *Transactions on Robotics*, vol. 29, no. 5, pp. 1257–1270, 2013.

[57] J. Lenarcic and A. Umek, “Simple model of human arm reachable workspace,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 8, pp. 1239–1246, 1994.

[58] Y. Kameda and M. Minoh, “A human motion estimation method using 3-successive video frames,” in *International conference on virtual systems and multimedia*, 1996, pp. 135–140.

[59] R. Poppe, "Vision-based human motion analysis: An overview," *Computer vision and image understanding*, vol. 108, no. 1-2, pp. 4–18, 2007.

[60] G. Ferrer and A. Sanfeliu, "Comparative analysis of human motion trajectory prediction using minimum variance curvature," in *Proceedings of the 6th international conference on Human-robot interaction*. ACM, 2011, pp. 135–136.

[61] H. Noguchi, T. Yamada, T. Mori, and T. Sato, "Mobile robot path planning using human prediction model based on massive trajectories," in *International Conference on Networked Sensing Systems (INSS)*. IEEE, 2012, pp. 1–7.

[62] G. Ferrer and A. Sanfeliu, "Behavior estimation for a complete framework for human motion prediction in crowded environments," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5940–5945.

[63] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 299–306.

[64] S. Hamasaki, Y. Tamura, A. Yamashita, and H. Asama, "Prediction of human's movement for collision avoidance of mobile robot," in *International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2011, pp. 1633–1638.

[65] J. Elfring, R. Van De Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 591–602, 2014.

[66] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 464–469.

[67] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 14–29, 2016.

[68] N. Jarrassé, J. Paik, V. Pasqui, and G. Morel, "How can human motion prediction increase transparency?" in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2008, pp. 2134–2139.

[69] S. Haddadin, *Towards safe robots: approaching Asimov's 1st law*. Springer, 2013, vol. 90.

[70] S. G. Brunner, F. Steinmetz, R. Belder, and A. Dömel, "Rafcon: A graphical tool for engineering complex, robotic tasks," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3283–3290.

[71] F. Steinmetz, A. Wollschläger, and R. Weitschat, "Razer - a hri for visual task-level programming and intuitive skill parameterization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 3, pp. 1362–1369, 2018.

[72] Wikipedia, "Rotation," https://en.wikipedia.org/wiki/Rotation, 2018, [Online; accessed 17-April-2018].

[73] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[74] T. Bajd, M. Mihelj, and M. Munih, *Introduction to robotics*. Springer Science & Business Media, 2013.

[75] K. Shoemake, “Animating rotation with quaternion curves,” *SIGGRAPH ’85 Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, vol. 19, no. 3, pp. 245–254, 1985. [Online]. Available: http://www.engr.colostate.edu/ECE555/reading/article_8.pdf

[76] R. Weitschat, A. Dietrich, and J. Vogel, “Online motion generation for mirroring human arm motion,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4245–4250.

[77] A. Ude, B. Nemec, T. Petrić, and J. Morimoto, “Orientation in cartesian space dynamic movement primitives,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2997–3004.

[78] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, “Online movement adaptation based on previous sensor experiences,” in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 365–371.

[79] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer Science & Business Media, 2008.

[80] Wikipedia, “Circular Ringbuffer,” https://en.wikipedia.org/wiki/Circular_buffer, 2018, [Online; accessed 03-April-2018].

[81] A. Pereira and M. Althoff, “Safety control of robots under computed torque control using reachable sets,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 331–338.

[82] ——, “Calculating human reachable occupancy for guaranteed collision-free planning,” in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4473–4480.

[83] B. D. Bushnell, A. W. Anz, T. J. Noonan, M. R. Torry, and R. J. Hawkins, “Association of maximum pitch velocity and elbow injury in professional baseball pitchers,” *The American journal of sports medicine*, vol. 38, no. 4, pp. 728–732, 2010.

[84] J. Van der Meulen, R. Gooskens, J. Denier Van der Gon, C. Gielen, and K. Wilhelm, “Mechanisms underlying accuracy in fast goal-directed arm movements in man,” *Journal of Motor Behavior*, vol. 22, no. 1, pp. 67–84, 1990.

[85] C. G. Atkeson and J. M. Hollerbach, “Kinematic features of unrestrained vertical arm movements,” *Journal of Neuroscience*, vol. 5, no. 9, pp. 2318–2330, 1985.

[86] R. Weitschat, J. Ehrensperger, M. Maier, and H. Aschemann, “Safe and efficient human-robot collaboration Part I: estimation of human arm motions,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1993–1999.

[87] A. Babiarz, A. Czornik, J. Klamka, M. Niezabitowski, and R. Zawiski, “The mathematical model of the human arm as a switched linear system,” in *International Conference On Methods and Models in Automation and Robotics (MMAR)*. IEEE, 2014, pp. 508–513.

[88] H. J. Nagarsheth, P. V. Savsani, and M. A. Patel, “Modeling and dynamics of human arm,” in *IEEE International Conference on Automation Science and Engineering*. IEEE, 2008, pp. 924–928.

[89] P. T. Katsiaris, P. K. Artemiadis, and K. J. Kyriakopoulos, “Modeling anthropomorphism in dynamic human arm movements,” in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 3507–3512.

[90] J. Rosen, J. C. Perry, N. Manning, S. Burns, and B. Hannaford, "The human arm kinematics and dynamics during daily activities-toward a 7 dof upper limb powered exoskeleton," in *Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on.* IEEE, 2005, pp. 532–539.

[91] R. Dumas, L. Cheze, and J. Verriest, "Adjustments to mcconville et al. and young et al. body segment inertial parameters," *Journal of Biomechanics*, vol. 40, no. 7, pp. 1651–1652, 2007.

[92] C. Simonidis, S. Gaertner, and M. Do, "Spezifikationen zu den ganzkoerpermenschmodellen im sfb 588," Technical report, Universitaets Verlag Karlsruhe, Tech. Rep., 2009.

[93] C. Ott, "Cartesian impedance control of redundant and flexible-joint robots," *Springer Tracts in Advanced Robotics*, vol. 49, pp. 16–30, 2008.

[94] A. Albu-Schäffer, C. Ott, U. Frese, and G. Hirzinger, "Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 3. IEEE, 2003, pp. 3704–3709.

[95] C. Wampler and L. Leifer, "Applications of damped least-squares methods to resolved-rate and resolved-acceleration control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 110, no. 1, pp. 31–38, 1988.

[96] A. S. Deo and I. D. Walker, "Overview of damped least-squares methods for inverse kinematics of robot manipulators," *Journal of Intelligent and Robotic Systems*, vol. 14, no. 1, pp. 43–68, 1995.

[97] R. Weitschat, S. Haddadin, F. Huber, and A. Albu-Schäffer, "Dynamic optimality in real-time: A learning framework for near-optimal robot motions," in *International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2013, pp. 5636–5643.

[98] S. Haddadin, F. Huber, and A. Albu-Schäffer, "Optimal control for exploiting the natural dynamics of variable stiffness robots," in *International Conference on Robotics and Automation (ICRA).* IEEE, 2012, pp. 3347–3354.

[99] D. J. Braun, F. Petit, F. Huber, S. Haddadin, P. Van Der Smagt, A. Albu-Schäffer, and S. Vijayakumar, "Robots driven by compliant actuators: Optimal control under actuation constraints," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1085–1101, 2013.

[100] D. Chen, B. Wang, and L. Wang, "Optimal control of robot manipulators via double generating functions," in *International Conference on Robotics and Biomimetics (ROBIO).* IEEE, 2017, pp. 2337–2342.

[101] R. Weitschat and H. Aschemann, "Safe and efficient human-robot collaboration Part II: Optimal generalized human-in-the-loop real-time motion generation," in *Robotics and Automation Letters (RA-L).* IEEE, 2018, pp. 3781–3788.

[102] D. G. Luenberger, "Introduction to linear and nonlinear programming," 1973.

[103] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Traces and emergence of nonlinear programming.* Springer, 2014, pp. 247–258.

[104] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters, "Trajectory planning for optimal robot catching in real-time," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3719–3726.

[105] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms.* John Wiley & Sons, 2013.

[106] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.

[107] J. Nocedal and S. J. Wright, *Sequential quadratic programming*. Springer, 2006.

[108] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2013.

[109] A. Werner, R. Lampariello, and C. Ott, "Optimization-based generation and experimental validation of optimal walking trajectories for biped robots," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4373–4379.

[110] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, "A human aware mobile robot motion planner," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[111] T. Arai, R. Kato, and M. Fujita, "Assessment of operator stress induced by robot collaboration in assembly," *CIRP annals*, vol. 59, no. 1, pp. 5–8, 2010.

[112] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[113] C. Wampler and L. Leifer, "Applications of damped least-squares methods to resolved-rate and resolved acceleration control of manipulators," in *J. Dyn. Syst., Meas. Contr*, vol. 110, 1988, pp. 31–38.

[114] V. Usenko, L. von Stumberg, A. Pangercic, and D. Cremers, "Real-time trajectory replanning for mavs using uniform b-splines and 3d circular buffer," *arXiv preprint arXiv:1703.01416*, 2017.

[115] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.

[116] T. Kröger, "On-line trajectory generation in robotic systems," *Springer, Tracts in Advanced Robotics*, vol. 58, 2010.

[117] Z. Rymansaib, P. Iravani, and M. Sahinkaya, "Exponential trajectory generation for point to point motions," in *International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2013, pp. 906–911.

[118] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Pearson, 1986.

[119] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *International Conference on Robotics and Automation (ICRA)*, vol. 2. IEEE, 2002, pp. 1398–1403.

[120] A. Billard and M. J. Matarić, "Learning human arm movements by imitation:: Evaluation of a biologically inspired connectionist architecture," *Robotics and Autonomous Systems*, vol. 37, no. 2-3, pp. 145–160, 2001.

[121] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in neural information processing systems*, 2003, pp. 1547–1554.

[122] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[123] A. Gams, T. Petrič, L. Žlajpah, and A. Ude, "Optimizing parameters of trajectory representation for movement generalization: robotic throwing," in *International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*. IEEE, 2010, pp. 161–166.

[124] B. Danube, “Airskin safety,” http://www.bluedanuberobotics.com/?page_id=87, 2016, [Online; accessed 14-September-2016].

[125] L. Zeng and G. M. Bone, “Design of foam covering for robotic arms to ensure human safety,” *Electrical and Computer Engineering*, pp. 1145–1150, 2008.

[126] BOSCH, “APAS workstation - flexible product assistant,” http://www.bosch-apas.com/en/apas/start/bosch_apas.html, 2016, [Online; accessed 14-September-2016].

[127] Schunk, “Co-act gripper meets cobots,” http://de.schunk.com/de_en/co-act/, 2016, [Online; accessed 14-September-2016].

[128] A. S. Jaffredo, P. Potier, S. Robin, J.-Y. Le Coz, and J.-P. Lassau, “Upper extremity interaction with side impact airbag,” in *Proceedings of IRCOBI Conference on the Biomechanics of Impact, Goteburg, Sweden*, 1998, pp. 485–496.

[129] T. Tamura, T. Yoshimura, M. Sekine, M. Uchida, and O. Tanaka, “A wearable airbag to prevent fall injuries,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 6, pp. 910–914, 2009.

[130] G. T. Loo, J. H. Siegel, P. C. Dischinger, D. Rixen, A. R. Burgess, M. D. Addis, T. O’quinn, L. McCammon, C. B. Schmidhauser, P. Marsh *et al.*, “Airbag protection versus compartment intrusion effect determines the pattern of injuries in multiple trauma motor vehicle crashes,” *Journal of Trauma and Acute Care Surgery*, vol. 41, no. 6, pp. 935–951, 1996.

[131] R. Weitschat, J. Vogel, S. Lantermann, and H. Höppner, “End-effector airbags to accelerate human-robot-collaboration,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.