TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Echtzeitsysteme und Robotik

Guaranteeing Safe Robot Motion

Aaron Paul Pereira

Vollständiger Abdruck der von der Fakultät der Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzende:	Prof. Gudrun Klinker, Ph.D.
Prüfer der Dissertation:	1. Prof. Dr.-Ing. Matthias Althoff
	2. Prof. Dr. Julien Provost

Die Dissertation wurde am 18. Juni 2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 9. Dezember 2018 angenommen.

Abstract

This thesis presents a framework for guaranteeing safety in human-robot co-existence in a practical and industry-applicable way. We show how to control a robot such that its movement is verifiably safe with respect to user-defined criteria, while maintaining efficiency such that the robot does not stop unnecessarily, and is able to replan around obstacles.

To be able to guarantee safety online during operation, we develop fast methods to calculate the volume both the robot and the human may occupy during their respective trajectories, in a conservative way. In the case of the human, since the intention is unknown, we also study human movement, collecting data from a range of humans. From this, we develop prediction models which account even for unexpected movements, but do not overestimate the occupied space. Using these spatial occupancies to calculate trajectories in the shared workspace which are safe for the robot to execute, we guarantee safety in all human-robot co-working scenarios.

Finally, the approach is tested on humans. Test subjects react well to this approach in terms of trust, and efficiency of the robot is higher than a comparable approach from the state of the art. Also, as subjects work more with the robot, they adjust to its behaviour, reporting better understanding of the robot and feeling safer after experience compared to at first sight.

Zusammenfassung

Diese Dissertation stellt einen praktischen und praxisnahen Ansatz für garantierte Sicherheit in Mensch-Roboter-Zusammenarbeit vor. Es wird ein gemäß benutzerdefinierten Sicherheitskriterien verifizierbar sicheres Steuerungskonzept für einen Roboter präsentiert, welches effizient ist und seine Bewegungen um Hindernisse herum planen und ausführen kann.

Um Sicherheit direkt während des Arbeitsvorgangs garantieren zu können wurden effiziente Algorithmen entwickelt, die das mögliche Ausmaß der Bewegungen von Mensch und Roboter schnell und konservativ berechnen. Da die Bewegungsintention des Menschen nicht vorab lesbar ist, wurden Bewegungsdaten von verschiedenen Probanden gesammelt um menschliche Bewegungen zu untersuchen. Basierend auf diesen Daten wurden Modellen zur Prognose von menschlichen Bewegung entwickelt, die auch unerwartete Bewegungen miteinbeziehen, den eingenommenen Raum aber nicht überschätzen. Mittels dieser Prognosen werden sichere Bewegungsabläufe für den Roboter berechnet und daher ständige Sicherheit für alle Szenarien, in denen Mensch und Roboter zusammenarbeiten, garantiert.

Schließlich wurde dieser Ansatz durch eine Benutzerstudie geprüft. Probanden zeigten Vertrauen in den Ansatz, und der Roboter bewegte sich effizienter als mit einem vergleichbaren Sicherheitsansatz aus dem Stand der Technik. Darüber hinaus passten die Probanden sich besser an den Roboter an, je mehr sie mit dem Roboter arbeiteten: ihr Verständnis des Roboters sowie ihr Sicherheitsgefühl verbesserte sich nach Arbeitserfahrung mit dem Roboter.

Note to Practitioners

As interest and opportunities for robots working in close proximity to humans grow in industry, the need arises for solutions to guarantee the humans' safety. Technical Specification 15066 [1] describes in detail the requirements of ISO 10218-1 [2] for robots working near humans. One mode of operation called speed and separation monitoring (SSM) allows the robot to monitor the human and adjust its behaviour to maintain a safe speed and distance.

Our implementation of SSM, based on partial motion planning, allows us to guarantee the safety of nearby humans according to various criteria. The desired movements of the robot can be preprogrammed or calculated on the fly. We show that our method of SSM performs 36% more efficiently than another mode of operation, the safety-rated monitored stop (SRMS), in a co-working scenario where a human continually enters the robot workspace.

This approach can avail of several and multiple sensing modalities, from industry-standard light curtains to cameras with depth sensors and motion tracking systems. It can be implemented on smaller, human-arm-sized robots such as the Schunk LWA-4P as well as on large, high-inertia robots that have traditionally been kept behind cages; it can also be adapted for mobile robots or manipulators on mobile bases.

Anwenderinformationen

Da in der Industrie Interesse und Gelegenheit für den Einsatz von Robotern in unmittelbarer Nähe von Menschen wächst, steigt auch der Bedarf nach Lösungen, die die Sicherheit des Menschen in einer Mensch- Roboter-Zusammenarbeit garantieren. Das erlaubte Verhalten von Robotern, die mit oder in nächster Nähe zu Menschen arbeiten, wird in ISO 10218-1 [2] beschrieben und in Technische Spezifikation 15066 [1] detailliert. Im Operationsmodus *Geschwindigkeits- und Abstandsüberwachung* (im Englischen, *speed and separation monitoring, SSM*), beobachtet der Roboter den Menschen und passt sich an sein Verhalten an, um sichere Geschwindigkeit und Abstand einzuhalten.

Basierend auf der sogenannten “partial motion planning” kann die hier präsentierte Implementierung von SSM die Sicherheit von in der Nähe des Roboters stehenden Menschen bei Berücksichtigung verschiedenen Kriterien garantieren. Die gewünschte Bewegungen des Roboters können in Voraus programmiert oder zeitgleich zur Operationsphase berechnet werden. In einem Zusammenarbeitsszenario, wo ein Mensch kontinuierlich in den Arbeitsraum des Roboters eindringt, ist der hier präsentierte Ansatz um 36% effizienter als ein anderer Operationsmodus, der sogennante *sicherheitsbewertete überwachte Halt* (im Englischen, *safety-rated monitored stop, SRMS*).

Dieser Ansatz kann sich einer Vielzahl von Sensorik-Modalitäten bedienen: industrielle Lichtvorhänge, Kameras mit Tiefensensoren sowie Motion-Capture Systeme. Er kann sowohl auf kleineren Robotern wie dem Schunk LWA-4P sowie auf größeren Industrierobotern, die bisher hinter Abzäunungen arbeiten mussten, angewendet werden. Der Ansatz kann auch für mobile Roboter oder Manipulatoren auf mobilen Plattformen angepasst werden.

Acknowledgements

Since a sole name appears as author of a work that is the fruit of several shared thoughts, the least I can offer a word of thanks to the many people who have contributed to this thesis. Foremost among them is my advisor Matthias Althoff. He has supported me continually, over and above the role of supervisor, offering both academic inspiration and guidance as well as personal encouragement and advice. I am extremely grateful.

To my fellow Smart-E ESR's, in particular my long-term office-mate Esra İçer for surviving two moves of office, the conversations on world politics and other matters and making sure our plants didn't die, and Andrea Giusti for teaching me low-level control and Italian and being detail-oriented.

To my colleagues Ahmed, Albert, Bastian, Christian, Dongkun, Hendrik, Markus, Sebastian, Silvia and Stefanie, for good times shared in the office and in northern Italy and Malta. I am lucky to have worked in such a fun research group!

Outside of my department, I thank my collaborators Jonas Schmidtler and Jakob Reinhardt from Lehrstuhl für Ergonomie, TUM, and I am grateful to Thomas Illa, Asuman Sezgin for capturing human movement data.

Further afield, working together with Martijn Zeestraten and Sylvain Calinon from IIT/Idiap was an absolute pleasure and I hope it will not be the last time. I am grateful to Mark Burgin and Geoff Pegman of RURobots Ltd., who provided and collaborated on an industrial use-case for testing of our verification algorithms. Thanks to Samia Nefti-Meziani and Laura Dawson from the University of Salford who hosted me during a secondment in 2016 and to my colleagues Roy Assaf, Saber Mehboubi-Heyderabadi and Stefania Russo there for their welcome.

I was lucky at TUM to work with some of the brightest minds internationally. I thank the students whose theses I supervised: Natalie Reppekus, Efe Bozkir (whose work appears in Sec. 2.3.1), and Florian Otti. Special thanks must go to Dario Beckert, who worked extensively on the study in Sec. 5.4 and on the verification of multiple safety criteria in Sec. 5.4, to Cédric Stark, whose thesis and later work resulted in Sec. 4.6, and to Stefan Liu, for the many thought-provoking discussions in the office and at conferences and whose work, while not described in this thesis, will drive forward formal approaches in robotics.

I was also very fortunate to have several motivated students from the practical course “Safe Human-Robot Co-Existence”, whose work contributed to this thesis: Mareike Baumann and Jonas Gerstner (Sec. 5.5), Jan-Hendrick Neumann (Sec. 2.3.1), Tim Salzmann (Sec. 2.3.2), Miguel Neves (adaptation of approach to modular robots). Thanks to all of you.

To Peer Lucas, Thomas Krieger, Michael Wolfram, Florian Böck in the seminar “Cyber-Physical Systems” – thank you for your contributions. I am grateful to former HiWis Daniel Spohr for human motion data processing, Christopher Kühn for video editing and Ertuğrul Karademir for the visualisation.

Research would not function without organisation: I thank Amy Bücherl, Gertrud Eberl and Ute Lomp from the Secretariat for their friendliness and helpfulness from the very start, and Marie-Luise Neitz and Federica Pepponi for coordinating Smart-E and UnCoVerCPS.

I thank the members of the robotics community who reviewed our papers, for their time and (predominantly) helpful comments. Not to be neglected are European Commission and the millions of EU taxpayers who finance invaluable independent research at institutions such as the TU Munich – so thank you dear reader!

I am grateful to my colleagues at DLR during my secondment there and afterwards when writing up. From the MoDex Lab members (and quasi-members) Annika M, Annika S, Andrea, Bene, Georg, Hadi, Hans, Julian Martin H, Martin P, Neal, Philipp, Qian, Ralph, Tom, Vanessa, Volker and Zhaopeng, from the Space Factory project Fabian, Ribin and Thomas H, from METERON SUPVIS Justin also Adrian, Alex, Daniel, Lio, Peter and Thomas G, and numerous others who contributed to a great work environment.

I also need to thank those who kept me sane and reminded me that thesis $\neq$ life: Lichthof (Andi, Johannes, Manu, Erik, Eva; the latter two also for proofreading), Shotstakovich (Alex, Felix, Jana, Johanna, Julie, Linards, Oskars, Raphael) my housemates Dom and Akhila and all my friends from UniChor and Refugio.

Finally, my family deserves the most gratitude: firstly, my grandparents and cousins in India who took me in and kept me sane while writing this dissertation. For the last twenty-four years of my life I have been lucky to have a brother, Jason, with whom I can have intellectually stimulating conversations and share ideas. Of course, it is my mother Tina and my father Melwyn who shaped us into what we are–encouraged and advised us on our life path, even when it was not smooth, and taught us how to think critically. *It is to my parents that this thesis is dedicated.*

8th February 2019 Aaron Pereira

Nomenclature

Conventions

Index of definitions

Sets

$[A]$	interval tensor/matrix
$\mathcal{G}_{\square}$	$\subseteq \mathbb{R}^3$, reachable occupancy of $\square$
Γ_{3DOF}	$\subseteq \mathbb{R}^3$, human reachable occupancy found with 3-DOF model (Sec. 4.3.2), in the shoulder coordinate system
Γ_{4DOF}	$\subseteq \mathbb{R}^3$, human reachable occupancy found with 4-DOF model (Sec. 4.3.1), in the shoulder coordinate system
Γ_{ACC}	$\subseteq \mathbb{R}^3$, human reachable occupancy found with acceleration limits model (Sec. 4.4.1)
Γ_{ISO}	$\subseteq \mathbb{R}^3$, human reachable occupancy found with ISO model (Sec. 4.2.2)
Γ_{JS}	$\subseteq \mathbb{R}^3$, human reachable occupancy found with 4-DOF model (Sec. 4.3.1), in the world frame
Γ_{POS}	$\subseteq \mathbb{R}^3$, human reachable occupancy found with model using rigid arm structure (Sec. 4.4.3)
Γ_{VEL}	$\subseteq \mathbb{R}^3$, human reachable occupancy found with velocity limits model (Sec. 4.4.2)
$\mathcal{F}$	$\subseteq \mathbb{R}^3$, robot reachable occupancy
$\mathcal{H}_t$	sensor data of the human at time t
Q	joint space of a kinematic chain
$\mathcal{Q}^{(\square)}([t_a, t_b])$	$\subseteq Q$, the reachable set in joint space from time t_a to t_b, found using model $\square$
$\mathcal{R}^{e}_{\boldsymbol{p}}([t_a, t_b])$	Exact reachable set of $\boldsymbol{p}$, from time t_a to t_b
$\mathcal{R}^{(\square)}_{\boldsymbol{p}}([t_a, t_b])$	Reachable set of $\boldsymbol{p}$, from time t_a to t_b (optional: found using model $\square$)
$\mathcal{S}$	symmetric matrix of intervals
$\mathcal{V}_r^{\mathcal{P}}$	sphere-swept volume (see Def. 3.5) with radius r and defining points $\mathcal{P}$
$\mathcal{X}$	state space of a kinematic chain
$\mathcal{Z}_{\boldsymbol{c}}^{G}$	zonotope (see Def. 3.6) with centre $\boldsymbol{c}$ and generators G

Indices

i	index of a joint or link on a kinematic chain
j	index of a point in a polytope
k	index of a point in time

Scalars

$[a]$	interval of a scalar
m	number of degrees of freedom of the robot
n	number of degrees of freedom of the human kinematic model
s	time parameter of a trajectory
t	time
ΔT	timestep of long-term planner
Δt	timestep of short-term planner/verifier

Vectors

$[\boldsymbol{a}]$	interval vector
$\boldsymbol{p}_{j,i}$	$\in \mathbb{R}^3$, the j^{th} point defining the sphere-swept volume (SSV) enclosing the i^{th} link on a manipulator.
$\boldsymbol{q}$	$\in Q$, the joint positions
$\boldsymbol{u}_k$	control input to robot at time t_k
$\boldsymbol{x}$	$\in Q \times \mathbb{R}^m$, the robot state, i.e. $[\boldsymbol{q}^\top, \dot{\boldsymbol{q}}^\top]^\top$
$\boldsymbol{\tau}$	joint torques of a manipulator

Parameters

$a_{\max,\square}$	$\in \mathbb{R}^+$, the maximum magnitude of acceleration of $\square$ in task space
$v_{\max,\square}$	$\in \mathbb{R}^+$, the maximum magnitude of velocity of $\square$ in task space
$[\boldsymbol{q}_{\inf}, \boldsymbol{q}_{\sup}]$	$\subseteq Q$, the interval of reachable joint positions of a kinematic chain
$[\dot{\boldsymbol{q}}_{\inf}, \dot{\boldsymbol{q}}_{\sup}]$	$\subseteq \mathbb{R}^m$, the interval of reachable joint velocities of a kinematic chain
$[\ddot{\boldsymbol{q}}_{\inf}, \ddot{\boldsymbol{q}}_{\sup}]$	$\subseteq \mathbb{R}^m$, the interval of reachable joint accelerations of a kinematic chain

Trajectories

$\boldsymbol{\xi}$	long-term plan; mapping from time parameter s to joint space Q
Ψ	short-term plan; mapping from time t to time parameter s
$\psi^{\eta}_{[t_a]}$	time-scaling manoeuvre; mapping from time t to time parameter s; starting at t_a and ending with $\dot{s} = \eta$

Superscripts and subscripts

$\square_A$	of the arm
$\square_E$	of the elbow
$\square_F$	of the forearm
$\square_H$	of the hand
$\square_{LA}$	of the left arm
$\square_{RA}$	of the right arm
$\square_S$	of the shoulder
$\square_T$	of the torso
$\square_U$	of the upper arm
$\square_W$	of the wrist
$\square_{e(,C)}$	at the time from which the robot on a short-term plan is safe (optional: according to criterion C)
$\square_f$	at the end of a long-term plan
$\square_h$	at the timestamp of the sensor data
$\square_k$	of the timestep currently executing

Marks and operations

$\dot{\square}$	derivative with respect to time
$\ddot{\square}$	2nd derivative with respect to time
$\dddot{\square}$	3rd derivative with respect to time
$\lvert\mathcal{A}\rvert$	$\in \mathbb{N}_0$ cardinality of the set $\mathcal{A}$
$\lVert\boldsymbol{a}\rVert$	$\in \mathbb{R}$ the Euclidean norm (magnitude) of the vector $\boldsymbol{a}$
$\lvert\boldsymbol{a}\rvert$	a vector of the absolute values of vector $\boldsymbol{a}$
inf(□)	infimum of interval □
sup(□)	supremum of interval □

List of Acronyms

AABB axis-aligned bounding box

ANOVA analysis of variance

CS Cartesian space

DB dominant-back-off

D-H Denavit-Hartenberg

DHM digital human models

DN dominant-non-back-off

DOF degree of freedom

GH glenohumeral

GJK Gilbert-Johnson-Keerthi

HMM hidden Markov model

HRCoex human-robot co-existence

HRI human-robot interaction

HRO human reachable occupancy

IA interval arithmetic

JS joint space

MPC model-predictive control

PoE Power over Ethernet

POMDP Partially Observable Markov Decision Process

QP quadratic programming

RO reachable occupancy

RRO robot reachable occupancy

RRT rapidly-exploring random tree

SB submissive-back-off

SME small and medium-sized enterprise

SN submissive-back-off

SRMS safety-rated monitored stop

SSM speed and separation monitoring

SSV sphere-swept volume

TS time-scaling

TTC time to completion

UDP User Datagram Protocol

TCP tool centre-point

Contents

Chapter 1

Introduction

Humans, since the beginning of history, have used tools to simplify daily life. Perhaps since the first human crushed their finger on a grinding stone or burnt themself on a forge, we have developed a healthy apprehension of the harm these tools could do to us.

This apprehension is well founded. Between 2014 and 2018 in the UK alone, over 13,000 injuries and 60 deaths were caused by contact with moving machinery [3]. Although thorough and rigorous safety procedures exist [4], [2], accidents occur nevertheless, often either due to misapplication of the safety measures, or humans engaged in behaviour unforeseen in the risk assessment or prohibited in the workshop code of practice.

If, on the other hand, machines were able to certify themselves safe, i.e. only carry out actions that they were certain would be safe, this human error could be avoided. The science-fiction writer Asimov postulated a built-in framework of meta-rules to which every robot must adhere, the highest being that "a robot may not injure a human being or, through inaction, allow a human being to come to harm" [5]. This has often been considered the "holy grail" of robotics, but its apparent simplicity belies a formidable technical challenge; not, of course, because a robot is inherently malignant, but because of its inability to perceive and interpret the environment in terms of the safety of nearby humans and therefore evaluate the effect of its actions on their safety.

Strategies for Safer Robots

What, then, is a dangerous action? We do not consider the social and psychological effects of robots in society, which requires a different kind of discussion, but rather the physical injury which results from collision of robot and human. The well-known study of Haddadin et al. [6] identifies properties of an impact which could cause serious harm to a human. Clamped impacts and prolonged impact are shown to be especially dangerous. Reactive control methods are proposed to reduce the time of impact and the risk of clamping [7]. However, if the inertia of the robot is high, it may not be able to change its motion in time to avoid harm, even if the collision is detected almost immediately. Other control methods limit the impact energy of the end-effector, e.g. [8]; the performance of the robot is then limited to the low speeds such methods require. Another approach to reducing the impact energy is to build robots with inherently compliant structures: “soft robots” [9]. Such robots can be made either from rigid links with inherent and deliberate compliance in the joints [10] or actuators [11], or from deformable materials [12]. Both concepts have the added advantage of protecting the robot: compliant joints absorb spikes in torque from impacts, which could otherwise damage delicate actuator components such as harmonic drives. Deformable structures “cushion” any electronics even from sustained applied force. While these remarkable capabilities are out of reach of traditional rigid-link robots, soft robots suffer the disadvantage of being more difficult to control due to the elasticity and deformation of the structure.

Collision Avoidance

Another approach is to try to avoid collision altogether. To this end, a number of approaches have been proposed. The concept of potential fields [13] where obstacles exert repulsive virtual forces on the manipulator is long established in motion planning, though the dynamic nature of the human obstacles and imperfect sensing mean that motion can be unsmooth and inefficient. A variety of approaches account for this challenge: Flacco et al. [14] account for the incomplete sensing of the robot’s environment when using commercially available Microsoft X-Box Kinect sensors, which only sense vision and depth information and do not give a full 3-D image. The authors also account for the velocity of the obstacle when generating repulsive virtual forces, a concept introduced in [15]. Researchers from the group of Rocco present the *Danger Field* [17] or *Safety Field* [16], turning the robot into the potential-generating objects and providing a measure of the robot’s danger to the human which can be minimised. In [17], quadratic programming (QP) is used to minimise this measure while adhering to the desired task.

Formal Methods

Such potential-field based methods have the advantage of smooth, continuous motion with easy spatial re-planning. However, it is hard to make formal guarantees of safety with such methods. One option is to use barrier functions to enforce virtual fixtures [18], or Invariance Control [19], which modulate the input to the manipulator to guarantee satisfaction of constraints in Cartesian space or velocity. The constraints must be defined analytically and the guarantees are subject to several assumptions on the dynamics. These methods also do not currently incorporate a model of human movement which is overapproximative, i.e. accounts for all possible movement; though it is conceivable to do so, it is not certain that robot motion would still be feasible under such constraints.

Motion Prediction

The approach proposed in this thesis endows the robot with an “awareness” of the human and its movements. The robot uses an understanding of human movement to calculate the area that it should avoid and hence choose movements that are safe to perform. Methods for predicting movement intention variously use electromyography [20], the beginning of the movement [21, 22] or contextual information from the environment [23]. Despite good prediction accuracy, the future movement cannot be known 100% accurately and reliably with any of these methods. In real-life manufacturing scenarios, humans may perform unexpected movements, for example when catching a falling object, or reflex movements when touching a hot or sharp object. In order to account for such movements we cannot rely on intention estimation, but must account for the entire range of dynamics possible in the human body. The predictions presented in this thesis are therefore *set-based*, aiming to find the entire set in space that could be occupied by the human until future time horizons.

Models of Human Dynamics

This presents some unique challenges which are addressed in this thesis. Using skeleton geometry, empirically based models of muscle dynamics and measured values of maximum muscle force from biomechanical studies in the last few decades, digital human modelss (DHMs) have been built [24, 25]. These can simulate the biomechanical structure—muscles, bones, tendons and ligaments—of an average human highly accurately. The inputs to such models are the excitations of the individual muscles, and the outputs are the joint positions. The dynamics of a DHM are highly nonlinear due to the complex muscle dynamics, high dimensional, and hybrid due to the presence of mechanical stops. This poses a challenge to obtaining set-based predictions: the reachable set, or image, of hybrid, nonlinear systems is impossible to obtain exactly [26]. To find a tight overapproximation (i.e. that includes all reachable states while excluding as many unreachable states as possible), one can linearise the nonlinear system and account for the linearisation error [27]; this process is not only computationally expensive but also introduces large overapproximations due to the huge uncertainty in the input. Furthermore, humans are variable in dimensions and physical capabilities, hence it is also nontrivial to adapt the DHM to predict the movements that humans are physically capable of performing.

In contrast, we present some simple, set-based models of human movement, which we use to predict the set of space possibly occupied by the human, faster than real-time. We also show that these predictions are *overapproximative*, meaning that they account for all relevant human movement from a range of humans, and *tight*, meaning that they do not include too much space which is physically unreachable by the human. Such predictions of the human can be used by the robot to plan a safe trajectory.

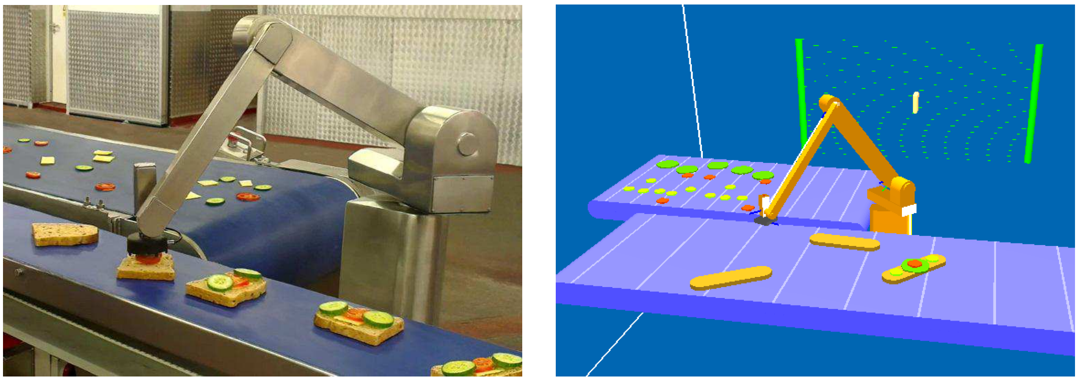


Figure 1.1: A use case for formally verified trajectory planning: left, the GRAIL sandwich assembly robot from R.U.Robots Ltd., and right, the same robot in simulation with a light curtain behind it to detect incursions into the workspace. Images courtesy of Mark Burgin and Geoff Pegman, R.U.Robots Ltd.

Sensing the Human

The robot's understanding of the human's current position comes from sensors and its internal model of the environment. Our approach can avail of both simple sensors (e.g. light curtains and alarm doors) as well as cameras with depth sensing and infrared motion-capture systems which can extract the pose of the human skeleton. Sensors certified for use in safety-critical systems are typically the former, although to date one commercially available camera system is available: the Pilz SafetyEye[1]. However, its latencies lag behind state of the art uncertified systems: its guaranteed latency of $100ms$ (at time of publication) compares to average latency of $20ms$ for the Xbox Kinect v2[2] and infrared motion tracking systems such as Qualisys (under $10ms$)[3] or Vicon (under $5ms$)[4]; furthermore, it is expensive.

Light curtains are cheap, reliable and widely used in factories. A method for formally-verified trajectory planning using light curtains is presented in [178]; Fig. 1.1 shows a use case from the food industry, where a light curtain is used to implement our approach in simulation. However, as a sensor for humans, they have certain drawbacks. One is their limited sensing: A light curtain senses only when the human is crossing its field of view. Similarly to cages, where the robot is interrupted when the cage door is opened, the robot cannot tell on which side of the light curtain the human is and how far across the light curtain the human has crossed. Dangerous situations can still occur if the human is already inside the cage, and the robot is turned on [28].

However, computer vision is a fast-developing field and recent years have seen advances in high-frequency object tracking [29], robust pose recognition from multiple cameras [30], and

[1] www.pilz.com/de-DE/eshop/00106002207042/SafetyEYE-Sicheres-Kamerasystem, accessed 24.2.2018
[2] http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx, accessed 1.10.2015
[3] www.qualisys.com/news/new-real-time-performance-test-with-50-oqus-cameras/, accessed 24.2.2018
[4] www.vicon.com/products/software/tracker, accessed 2.1.2018

new types of vision sensors such as event cameras, currently used for fast visual odometry of mobile robots [31]. It is likely that humans will be able to be tracked reliably and robustly in the near future, without using personal protective equipment such as reflective clothing, which would be necessary for current marker-based motion tracking systems. On this basis, the majority of the predictions of the human which are presented in this thesis assume that the pose of the human skeleton can be sensed with a reasonable degree of accuracy.

1.1 Human-Robot Co-Existence in Application

The work in this thesis is intended for contemporary practical application outside the laboratory, in collaborative manufacturing as well as in applications where human-robot co-existence is not yet common due to lack of guaranteed safety. Such areas may be medical robots, domestic robots and autonomous robots. Engineering research is only relevant inasmuch as it is practical, therefore it is useful to review the state of the art in technology for collaborative operation as well as current standards for safety.

Human-Robot Interaction (HRI) has been in existence for over half a century, with teleoperated robots being first used in the 1940's to remotely conduct hazardous chemistry and physics experiments, minimising the danger to the human. Teleoperation expanded into nuclear de-commissioning, surgical robotics and even space operations [32]. In such scenarios, the interaction between human and robot master device is controlled and, especially where force-feedback is not implemented, poses little physical danger to the human.

Robots which work alongside or interact with humans, also known as “cobots”, have only been of industrial relevance in the last couple of decades. Prior to this, human-machine interaction was limited to a computer interface, teleoperation, or a dashboard, rather than physical proximity in a shared workspace. One early example is found in [33], where robots provide assistance to assembly-line workers by lifting heavy components and guiding their placement, aided by virtual fixtures. Since then, robots have been used to lift and reorient heavy parts for human accessibility and ergonomics during welding operations [34], and to improve efficiency on the production line by delivering parts to a human worker, saving them from fetching the parts themselves [35, 36].

Industrial standards for safety in the workspace of machines has yet to account for human-robot coexistence or collaboration in a comprehensive way reflecting the developments in the last few years. The ISO standard ISO 10218-1 [2], which governs the safe operation of industrial robots, briefly mentions four modes of operation for safely controlling robots in collaborative workspaces. These are:

1. **Safety rated monitored stop** – the robot performs a controlled, category 2 stop[5] when

[5] Stop categories are defined according to IEC 60204-1 [37] as follows. A category 0 stop removes all power from the motors immediately and applies the brakes. A category 1 stop uses the motors to guide the robot to a stop, then removes power and applies the brakes. A category 2 stop uses the motors to guide the robot to a stop, but does

the human enters the collaborative workspace, and resumes work when the human is away.

2. **Hand guiding** – using a hand-held and hand-operated device to transmit motion commands to the robot.

3. **Speed and separation monitoring** – the robot limits its speed to a particular upper bound and its distance to the human to a particular lower bound. The approaches presented in this thesis are within the framework of this mode of operation.

4. **Power and force limiting** – the robot limits its power and potential impact force, so that collisions are limited in severity. This has been a topic of vigorous research over the last few years. Although collision detection without torque sensors on the robot joints is possible [38], the advent of reliable and relatively inexpensive robots with joint torque sensors, such as the Light-weight Robot (LWR) from the German Aerospace Centre (DLR) and KUKA, have made it possible to detect collisions fast [7]. Capacitative sensing such as artificial skins [39] can be useful in detecting collisions. In this way, the robot can react to collision quickly, limiting the duration of impact and hence its severity. For lower-inertia robots working in human robot collaborative scenarios, such approaches are promising.

However, detailed instructions on how to implement such methods are not in any ISO standards and compliance was until recently judged on a case-by-case using a risk assessment specific to the application. In early 2016 the Technical Specification 15066 [1] was published, a collaboration between researchers, industry, the ISO and other stakeholders. This shed more light on how the different modes of operation could be implemented.

In our case, we wish to avoid any collisions with the robot altogether. This is relevant when addressing the case of medium to high inertia robots in human-robot co-existence (HRCoex), or when the robot is carrying dangerous or sharp tools. For us, therefore, the *safety rated monitored stop* and *speed and separation monitoring* are relevant. Of these, the latter is more efficient (we test the efficiency of both methods in Chapter 5, and show that *speed and separation monitoring* has a significant advantage over the *safety rated monitored stop* in terms of robot idle time; the human time to completion (TTC) were not significantly different.)

In every case, a situation and application-specific risk assessment must be carried out before the robot is certified safe to operate. However, as robots become capable of learning skills autonomously, the entire behaviour of the robot cannot be exhaustively tested in advance. Furthermore, flexible manufacturing requires reallocation of resources, meaning that robots may often need to change their location and role on the workshop floor. Finally, the advent of modular robots mean that the entire *structure* of the robot can change. Such flexibility of function requires a corresponding paradigm shift in safety certification.

not remove power or apply brakes. According to ISO 10218-1 [2], a fault in the safety-related control system must automatically lead to a category 0 or 1 stop.

We look to the industrial standards [4, 2, 37] for inspiration, since guaranteeing that the robot behaviour conforms to that which is agreed upon by leading experts and stakeholders is an obvious strategy for making our work relevant to industry. However, the current pace of development in human-robot co-working means that acceptable behaviour can change, as evidenced by the publication of the new Technical Standard [1] during the preparation of this thesis.

The guiding philosophy of this thesis is therefore to use the current standards as a framework for the development of approaches to safety, while being mindful that the standards may change in the coming decades. Standards and industry practices therefore *guide* the direction of this research rather than *limiting* it. A further consideration, is to make this approach as general and as modular as possible, such that the guarantee of safety can "bolt on" to whichever robotic system a prospective user would like.

1.2 Thesis Outline and Major Contributions

In the following chapters we answer the questions:

- *How does the robot plan a safe trajectory?* (Chapter 2) We present algorithms for planning and modifying trajectories of a robot such that they are guaranteed safe with respect to certain criteria. These algorithms are based on conservative predictions of the future spatial occupancies of both human and robot. We test these algorithms, showing for the first time that formal verification is feasible for robot trajectories for two different safety criteria, and that we can plan and modify the spatial trajectory of the robot during operation, while also satisfying these safety criteria. This chapter is primarily based on the publications [178, 179, 180], the student projects [181, 182, 183], plus novel work.

- *How does the robot predict unsafe collisions?* (Chapters 3 and 4) We present novel, scalable methods for tightly and quickly bounding the future spatial occupancies of the robot (Chapter. 3). Also whereas previous work focusses on predicting expected movement or assumes a known intention, we show how to bound all relevant human motion where intention is unknown (Chapter. 4). In the case of the robot, we present novel scalable methods to enclose the spatial occupancy over sections of the robot's trajectory and compare these approaches with respect to their computational expense and tightness of the bounds. In the case of the human, we show how the prediction is dependent on sensing technology, but that we can still conservatively predict the reachable areas of a human subject during HRCoex, by abstracting the human body to simpler dynamical models and performing reachability analysis. As well as comparing the computational expense and tightness, we perform reachset conformance checking [40] to validate the conservativeness of our prediction. We check the conservativeness of the prediction against recorded movements and movements simulated from efficient exploration a biomechanical model.

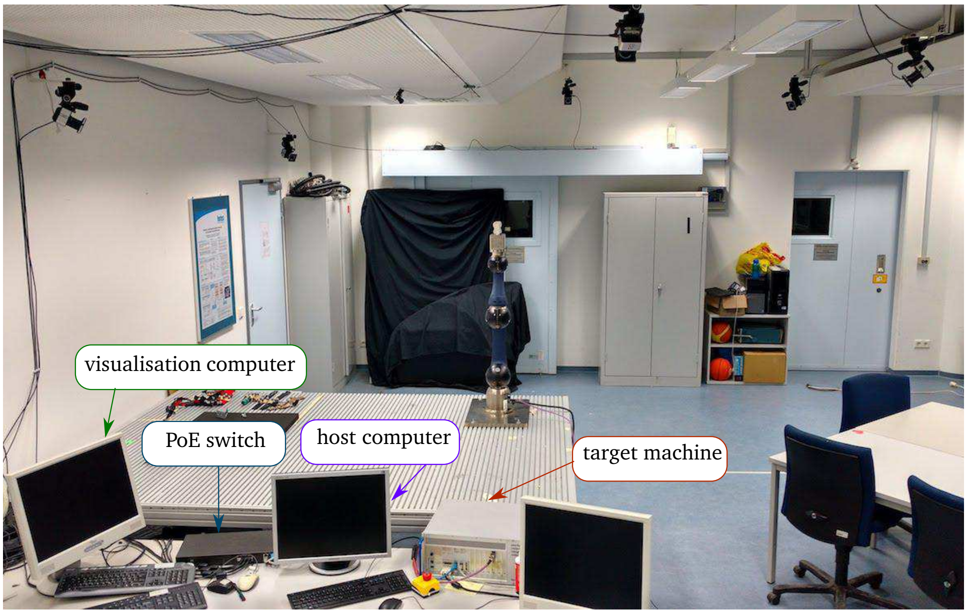


Figure 1.2: Photograph of setup

These chapters draw from publications [184, 185, 186, 187] and the student projects [188, 189].

- *How well does this approach work?* (Chapter 5) We evaluate our method in user studies, looking not only at the efficiency of the robot using our method but also the human factors. We find that our method is significantly more efficient than the state of the art in terms of robot idle time with no noticeable effect on human TTC, and that different robot behaviours can increase or reduce human trust in the system. We also found that the human's perception of safety improves with increased exposure to the robotic system. Further insights are discussed in this chapter. This chapter is based on the publication [190], the student projects [191, 192] as well as novel analysis.

1.3 Experimental Setup

The experimental setup for experiments in this thesis (with the exception of Sec. 5.4) is shown in Fig. 1.2 and in the schematic in Fig. 1.3. The robot is a Schunk LWA4P modular robot arm with 6 controllable degrees of freedom and an industrial gripper. It is controlled over CAN bus by a Speedgoat Performance real-time target machine with an Intel i7 quad-core processor ($3.5GHz$) and 2GB RAM running real time Simulink R2015b at $500Hz$.

The human is tracked by a 6-camera Vicon Vero 1.3 motion capture system, operating at $250Hz$. The cameras operate by shining infrared light on the workspace, which is reflected by clusters of retroreflective markers on the objects to be tracked. The 6 cameras are connected

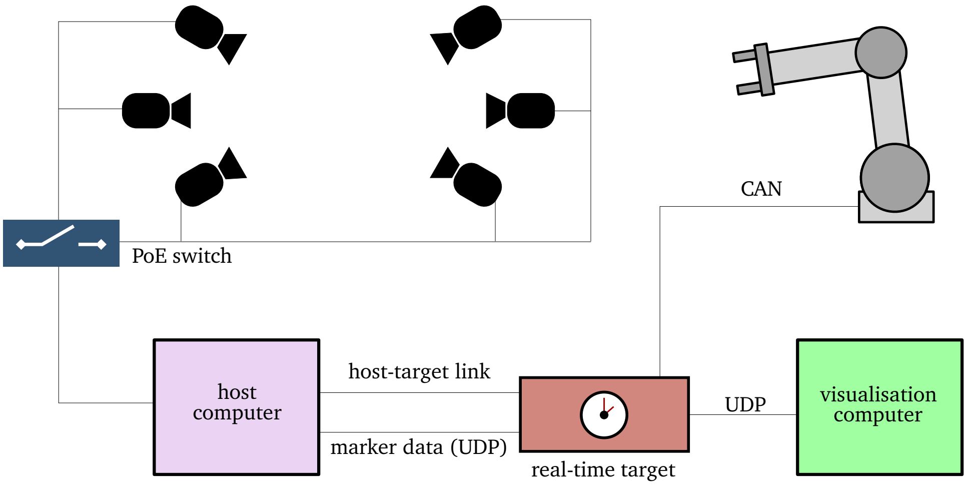


Figure 1.3: Schematic diagram of the setup

via a Power over Ethernet (PoE) switch to a Dell Precision Tower 3620 with an Intel Xeon E3-1270v5 processor ($3.6GHz$), 16GB RAM and NVIDIA Quadro K620 Graphics card running Windows 10. The camera data is processed by Vicon Tracker 3.5 software on this machine to triangulate the positions of the marker clusters; these are then sent via User Datagram Protocol (UDP), at $1kHz$, to the real-time target machine. The marker clusters on the human body are shown in Fig. 1.4.

In the experiments which involve replanning of trajectories in Sec. 2.4, we also have a visualisation in a virtual environment created in Coppelia V-REP and receiving information in real time from the target machine via UDP at a rate of 20Hz.

The processing of the data in Tracker 3.5 is the only non-real-time part of the system; to the best of the author's knowledge, no tracking system with deterministic latencies near comparable to the latencies of the Vicon system exist. According to the information from the developers (`www.vicon.com/products/software/tracker`, accessed 2.1.2018), the Tracker 3.5 software has a computation time of $1.5ms$ for 5 objects and $2.8ms$ for 10 objects. As we had 7 or 8 objects on the human (depending on whether the marker on the lower back was used), we took the latency at a conservative $5ms$, plus the frame rate of the cameras and the cycle time of the UDP connection, which were $4ms$ and $1ms$ respectively, i.e. $10ms$ overall.

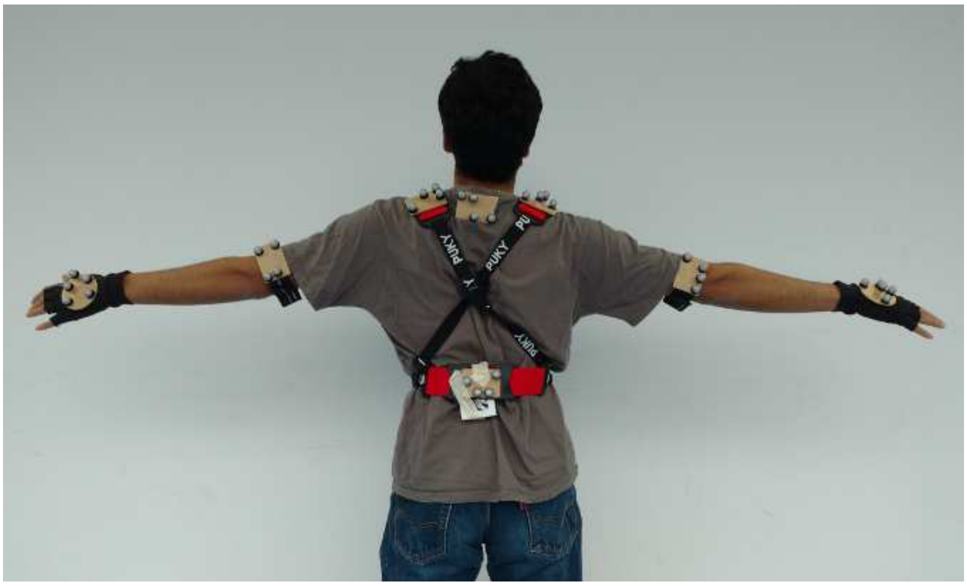


Figure 1.4: Marker cluster positions on a human subject

Chapter 2

Robot Planning and Verification

The principle behind guaranteeing safe robot motion is that the robot ***only performs actions that have been previously verified safe****.*

Verifying the safety of a global plan in the immediate vicinity of a human is impractical, since the environment is so dynamic: humans can move unpredictably and fast, and on timescales larger than a few milliseconds the human is capable of large changes of position. We therefore use the approach presented by Petti and Fraichard [41] of verifying partial plans which end in a safe state. The partial plans consist of the immediately subsequent part of the global trajectory, followed by a failsafe manoeuvre bringing the robot to a safe state. The safety of the plans need only be verified for as long as they are not in the safe state.

We first discuss definitions of safety in the next section (Sec. 2.1). We then detail the framework for generating and verifying safe trajectories given a global desired trajectory (Secs. 2.2 and 2.2.1) and finally show how to integrate global path planning within our framework (Sec. 2.3), such that the robot trajectory can be replanned on-the-fly. An experimental implementation is shown in Sec. 2.4.

2.1 Defining safety

As a first step, we need to define what a safe state is. We then show how the robot can verify and guarantee these criteria of safety.

In [42], the authors identify 5 undesired contact scenarios which could be injurious to the human: 1) unconstrained impact (between the robot and a human body part), 2) clamping (of a human body part between 2 parts of the robot), 3) constrained impact (clamping of a human body part between the robot and another object), 4) partially constrained impact (impact of the robot with a human body part, where free motion of the human is constrained by contact with an object elsewhere on the human) and 5) secondary impact (with another object following the first impact with the robot). In all these cases, injury can be avoided by avoiding collision altogether with the robot.

Even a stationary robot is not safe, however–humans can still trip over, fall against and bump into a robot that is not moving. A robot that confuses a human as to what the robot's intended movement is and abruptly stops, leading the human to collide with it, can also not be said to be safe. In such cases, however, the impact energy only comes from the human and is likely to be far lower than that imparted by a moving robot. One possible practicable criterion of safety is that the robot is *safe only when it is stationary*. We define first the reachable set of a system, and the reachable occupancy of both human and robot.

Definition 2.1 (Reachable Set): Given a system with state $\boldsymbol{x}(t)$, input $\boldsymbol{u}(t)$ and the dynamics $\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t))$, where t is time. The possible initial states $\boldsymbol{x}(0)$ and inputs $\boldsymbol{u}(t)$ are bounded by sets, $\boldsymbol{x}(0) \in \mathcal{X}_0, \forall t : \boldsymbol{u}(t) \in \mathcal{U}(t)$. The *exact reachable set* of states $\boldsymbol{x}$ at $t = t_f$ is:

$$\mathcal{R}^e_{\boldsymbol{x}}(t_f) := \ \left\{\boldsymbol{x}(t_f) = \boldsymbol{x}(0) + \int_0^{t_f} f(\boldsymbol{x}(t), \boldsymbol{u}(t))\mathrm{d}t \,\middle|\, \boldsymbol{x}(0) \in \mathcal{X}_0, \forall t : \boldsymbol{u}(t) \in \mathcal{U}(t)\right\}.$$

Furthermore, the *exact reachable set* of states over a time interval $[t_a, t_b]$ is:

$$\mathcal{R}^e_{\boldsymbol{x}}([t_a, t_b]) := \bigcup_{t=t_a}^{t_b} \mathcal{R}_{\boldsymbol{x}}(t)$$

The *reachable set* $\mathcal{R}_{\boldsymbol{x}}$ is understood to mean an overapproximation to the exact reachable set:

$$\mathcal{R}_{\boldsymbol{x}}([t_a, t_b]) \supseteq \mathcal{R}^e_{\boldsymbol{x}}([t_a, t_b])$$

Definition 2.2 (Reachable Occupancy (RO)): Let $\texttt{map}_{\text{occupancy}}$ be a mapping from the state of a system $\boldsymbol{x} \in \mathcal{X}$ to a set in Cartesian space:

$$\texttt{map}_{\text{occupancy}} : \mathcal{X} \to \mathcal{P}(\mathbb{R}^3),$$

where $\mathcal{P}$ represents the powerset. Then the *reachable occupancy (RO)* for a reachable set $\mathcal{R}_{\boldsymbol{x}}$ is:

$$\mathcal{G} \supseteq \{\mathtt{map}_{\mathrm{occupancy}}(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{R}_{\boldsymbol{x}}\}$$

In plain English, we consider the reachable set of a system to mean the set of *system states* a system could possibly reach starting from an initial set of states, and a reachable occupancy to mean an *overapproximation to its occupancy* in space (at a certain reachable set of states).

Definition 2.3 (Human Reachable Occupancy (HRO)): Consider the human body as a dynamical system with an unknown, internal state $\boldsymbol{z}(t)$, and dynamics $\dot{\boldsymbol{z}} = g(\boldsymbol{z}, \boldsymbol{u})$ where $\boldsymbol{u}(t) \in \mathcal{U}$ is some input to the system, and $\mathcal{U}$ is the set of all inputs, possibly subject to certain assumptions on human motion. Let $\mathcal{Z}_0$ be the *initial set* of possible body states $\boldsymbol{z}(t_0)$ given readings from the sensors at time t_0, allowing for measurement uncertainty. Let $\mathcal{F}(\boldsymbol{z}) \subset \mathbb{R}^3$ be the spatial occupancy of the human at a particular state, i.e. the $\mathtt{map}_{\mathrm{occupancy}}$ function in Def. 2.2. The system's *reachable set* as defined in Def. 2.1 is:

$$\mathcal{R}_{\boldsymbol{z}}([t_i, t_j]) = \{\boldsymbol{z}(t_0) + \int_{t_0}^{t} g(\boldsymbol{z}, \boldsymbol{u}) dt \mid \boldsymbol{z}(t_0) \in \mathcal{Z}_0, t \in [t_i, t_j], \boldsymbol{u}(t) \in \mathcal{U}\}.$$

We define the *human reachable occupancy (HRO)* as:

$$\Gamma([t_i, t_j]) \supseteq \{\mathcal{F}(\boldsymbol{z}) | \boldsymbol{z} \in \mathcal{R}_{\boldsymbol{z}}([t_i, t_j])\}. \qquad \square$$

Definition 2.4 (Robot Reachable Occupancy (RRO)): Let $\mathcal{R}_{\boldsymbol{q}}(t)$ be the reachable set of the joint position vector $\boldsymbol{q}$ of a robot at time t, and $\mathcal{W}(\boldsymbol{q}) \subset \mathbb{R}^3$ be the set of Cartesian space it occupies at a certain joint position, i.e. the $\mathtt{map}_{\mathrm{occupancy}}$ function in Def. 2.2. Then the *robot reachable occupancy (RRO)* is defined:

$$\digamma([t_i, t_j]) \supseteq \{\mathcal{W}(\boldsymbol{q}) | t \in [t_i, t_j], \boldsymbol{q} \in \mathcal{R}_{\boldsymbol{q}}(t)\}. \qquad \square$$

Methods for calculating the HRO will be presented and evaluated in Chapter 4, where we will also discuss and evaluate the assumptions made on the human's motion. Methods for calculating the RRO will be presented in Chapter 3.

We can now define the stationary criterion:

Definition 2.5 (Stationary Criterion): Let $\boldsymbol{q}(t)$ be the joint positions of a robot at time t, and $\digamma(t)$ and $\Gamma(t)$ the RRO and HRO respectively. The stationary criterion is fulfilled if, for any nearby humans, the following holds:

$$\forall t : \big(\digamma(t) \cap \Gamma(t) = \emptyset\big) \vee \big(\dot{\boldsymbol{q}}(t) = \boldsymbol{0}\big) \qquad \square$$

When humans and robots work in close proximity, however, such a restrictive safety criterion may impede the robot performance. Especially where the robot and the human collaborate on a task, this is an impractical requirement. Technical Standard 15066 [1] defines permissible impact forces and pressures for various parts of the human body. Rossi et al. [8] define a method using quadratic programming (QP) to keep impact energy under a certain level. In ISO standard 10218-1 [2] the "safety-rated" speed of $0.25\frac{m}{s}$ for the robot end-effector is stipulated for robots in areas where humans may reach them, since this "allows sufficient time for people either to withdraw from the hazardous area or to stop the robot". However, since it is not known which part of the robot the human may collide with, we must guarantee this for the whole robot. We therefore define a second safety criterion for use in this thesis:

Definition 2.6 (Reduced-speed Criterion): Let $\boldsymbol{q}(t)$ be the joint positions of a robot at time t, let $F(t)$ and $\Gamma(t)$ be the RRO and HRO respectively, and let $v(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$ be the magnitude of the maximum speed of any point on the robot's surface in Cartesian space, and $v_{\max}$ is a velocity limit specified in advance. The reduced-speed criterion is fulfilled if, for any nearby humans, the following holds:

$$\forall t : \big(F(t) \cap \Gamma(t) = \emptyset\big) \vee \big(v(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)) \leq v_{\max}\big) \qquad \square$$

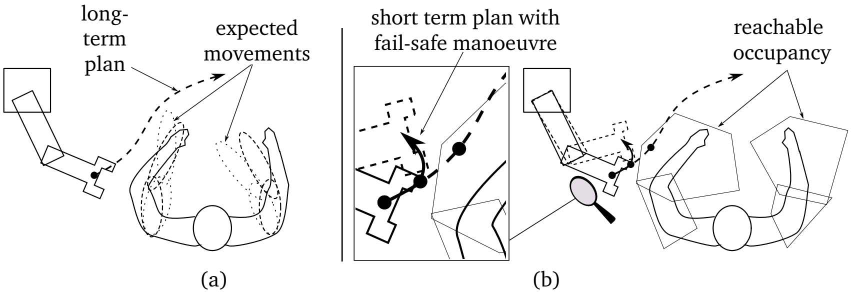


Figure 2.1: (a) Long-term plan, possibly accounting for expected human movement; (b) short-term plan with fail-safe manoeuvre accounting for unexpected movement, using ROs enclosing all possible motion. The section of the long-term plan currently executed has a fail-safe manoeuvre available to it at all times. Should the next section of the long term trajectory not be verified safe, the robot executes this fail-safe manoeuvre.

2.2 Approach

Guaranteed safe robot motion when robots and humans work at close quarters seems like a paradox. The human could perform an unexpected movement at any time, such as a reflex movement away from a hot or sharp object, or trying to catch a falling object. Such movement is difficult to predict even half a second into the future, so how should one plan robot movements that are guaranteed to be safe for humans, with respect to the categories of safety presented in the previous section?

Our approach works on two different timescales, as shown in Fig. 2.1, and has been tested in simulation in [178] and on a robot in [184, 180]. We plan the desired global path around

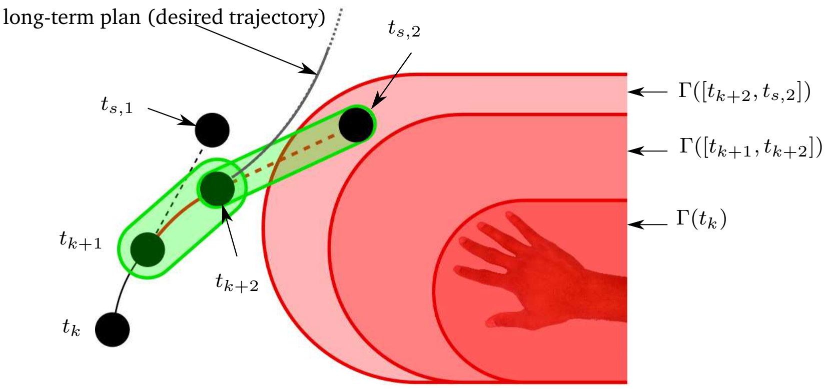

long-term plan (desired trajectory)
$t_{s,2}$
$t_{s,1}$
t_{k+1}
t_{k+2}
t_k
$\Gamma([t_{k+2}, t_{s,2}])$
$\Gamma([t_{k+1}, t_{k+2}])$
$\Gamma(t_k)$

Figure 2.2: Verification of a short-term plan. The desired trajectory during time interval $[t_{k+1}, t_{k+2}]$ would be unsafe to execute, since the RRO during the subsequent failsafe manoeuvre (shown in green) intersects the HRO (red).

the expected movement of the human, and call this the *long-term plan*. To guarantee safety, we also have a *short-term plan* consisting of the immediately subsequent section of the long-term plan, followed by a *failsafe manoeuvre* which brings the robot to a safe state, before the human can reach it. This short-term plan is verified safe before it is executed. If the next short-term plan is not safe, the failsafe manoeuvre from the previous short-term plan is executed. Hence, the robot always has a safe trajectory available to it.

The approach for verification of the short-term plan is shown in more detail in Fig. 2.2, a simple example with a disc robot and a single safety criterion. A short-term plan consisting of the portion of the long-term plan from time t_k to t_{k+1} (bold black line), followed by a failsafe manoeuvre (dashed black line), has been verified safe prior to time t_k. From time t_k to t_{k+1}, the short-term plan is executed by the robot, and simultaneously a new short-term plan consisting of the portion of the long-term plan from time t_{k+1} to t_{k+2} (bold red line), followed by a failsafe manoeuvre (dashed red line) reaching a safe state at $t_{s,2}$ is assembled. The volume in space which would be occupied by the robot during this plan from t_{k+1} to $t_{s,2}$, i.e. the RRO, is calculated. We also calculate the HRO during the same time. If the RRO intersects the HRO, this means the human might touch the robot before it is in the safe state, violating the safety criterium. In this case, at time t_{k+1}, the robot stays on the old short-term plan and starts to perform the failsafe manoeuvre verified prior to t_k. Otherwise, the robot executes the (now verified safe) new short-term plan, and starts to plan and verify the next short-term plan which starts at t_{k+2}.

Note that, although the failsafe manoeuvres shown in Fig. 2.2 are shown off-path for illustration purposes, they are actually planned path-consistent with the long-term plan. During normal operation, when the human is far from the robot, the failsafe manoeuvre is never executed, and the robot performs only the long-term plan.

2.2.1 Planning the Short-Term Plan

We can modify paths both spatially and temporally. It is difficult–or, in the case of fixed-base robots, sometimes even impossible–to modify the path in space quickly enough to avoid the human; it is almost invariably quicker and more practical to bring the robot to a stop rather than bring the robot out of the range of the human. The failsafe manoeuvres we consider, therefore, consist of a path-consistent stop.

Not only should the failsafe manoeuvre be path-consistent, but the robot motion should also be continuous in acceleration to avoid vibration or damage to the drives. This means that the failsafe manoeuvre should be planned as a limited-jerk, limited-acceleration trajectory. Furthermore, when the robot is executing the failsafe manoeuvre, and the human moves away from its workspace, it cannot simply resume the desired trajectory at nominal speed, but must accelerate back to the nominal speed with limited jerk and acceleration. We call this a *recovery manoeuvre*. When the robot is executing the failsafe manoeuvre, therefore, the short-term plan should consist of the first part of a recovery manoeuvre, followed by a failsafe manoeuvre.

Trajectory scaling means scaling the speed of a trajectory while keeping the path constant. It is useful for guaranteeing that trajectories conform to torque limits [43, 44] or derivatives thereof [45], or keep estimated impact energy under a certain level [8]. We consider the long-term plan $\boldsymbol{\xi}$ as a trajectory parametrised by s:

Definition 2.7 (Trajectory): A trajectory is a continuous mapping from a time parameter $s \in [s_0, s_f]$ to a joint position $\boldsymbol{q} \in \mathcal{Q}$,

$$\boldsymbol{\xi} : [s_0, s_f] \to \mathcal{Q}$$ □

By varying $\dot{s} = \frac{\mathrm{d}s}{\mathrm{d}t}$ (N.B. a dot to indicate derivative means the derivative with respect to real time throughout this thesis), we can generate motion on the spatial path of $\boldsymbol{\xi}$ as fast or slow as necessary. The short-term plan therefore only needs to be described in terms of the time parameter s. We define it as a mapping from time to the time parameter: $\Psi_{[t_k]} : [t_k, \infty] \to [s_k, \infty]$. Thus the joint position at time t is $\boldsymbol{\xi}(\Psi_{[t_k]}(t))$.

We formally define a *time-scaling manoeuvre*, and the *failsafe manoeuvre* and *recovery manoeuvre*.

Definition 2.8 (Time-scaling manoeuvre): Let s_a, $\dot{s}_a$ and $\ddot{s}_a$ be the values of s, $\dot{s}$ and $\ddot{s}$ at time t_a, and $\eta \in \{0, 1\}$. Then a time-scaling manoeuvre starting at t_a is a monotone function $\psi^{\eta}_{[t_a]} : [t_a, \infty] \to [0, \infty]$ where $\psi^{\eta}_{[t_a]}(t_a) = s_a$, $\dot{\psi}^{\eta}_{[t_a]}(t_a) = \dot{s}_a$ and $\ddot{\psi}^{\eta}_{[t_a]}(t_a) = \ddot{s}_a$, and for all $t > t_b$, $\dot{\psi}^{\eta}_{[t_a]}(t) = \eta$, $\ddot{\psi}^{\eta}_{[t_a]}(t) = 0$ for some finite $t_b \geq t_a$. We call t_a the start and t_b the end of the time-scaling manoeuvre. □

Definition 2.9 (Failsafe and recovery manoeuvre): A failsafe manoeuvre $\psi^0_{t_a}$ starting at t_a is a time-scaling manoeuvre where $\eta = 0$. A recovery manoeuvre $\psi^1_{t_a}$ starting at t_a is a time-scaling manoeuvre where $\eta = 1$. □

An overview of the verified trajectory planning and the verification algorithm is given in Alg. 1 and Alg. 2 respectively. In Alg. 1, the variables are initialised in lines 1–3. The variable "trajectory_state" consists of the time parameter and its time derivatives $[s, \dot{s}, \ddot{s}]^\top$; these, along with the long-term plan and the current timestep are required as parameters to plan a time-scaling manoeuvre. At every timestep, we execute the previously verified control command in line 5 (we start the verified trajectory planning when the robot is stationary, so that the first control command, generated in line 3, does not move the robot and is *a priori* safe).

Fig. 2.3 is a sketch-plot of a short-term plan in the variables $\dot{s}$ and time t. If the robot is moving at nominal speed because the human is too far away to be verified unsafe, we consider the recovery manoeuvre to be trivial, i.e. in Def. 2.8, $t_b = t_a$, and $\dot{\psi}^1_{[t_a]}(t) = 1$ for the entire trajectory. The planning of a short-term plan starting at time t_k is described in Alg. 3. Given a desired time-scaling manoeuvre (in Alg. 1, a recovery manoeuvre and in Alg. 4, a generic time-scaling manoeuvre), we obtain the trajectory state ($[s, \dot{s}, \ddot{s}]^\top$) at time t_{k+1}. From this state, we plan a failsafe manoeuvre. The concatenation of the desired time-scaling manoeuvre from t_k to t_{k+1} and the failsafe manoeuvre is the short-term plan.

Algorithm 1 Formally Verified Trajectory Planning

```
Input: long-term plan $\boldsymbol{\xi}$, safety criteria $\mathcal{C}$, human sensor data including timestamp $\mathcal{H}_{t_h}$
Output: Safe trajectory
 1: $k \leftarrow 0$
 2: safe$_k \leftarrow 0$
 3: $\Psi_{current} \leftarrow$ plan_failsafe(trajectory_state, $k, \boldsymbol{\xi}$)
 4: while not at goal state do
 5:     execute_control_on_robot($\Psi_{current}, k$)
 6:     if $\neg$ safe$_k$ then        // If it wasn't on the recovery manoeuvre, plan a new one
 7:         $\psi^1_{[t_k]} \leftarrow$ plan_recovery(trajectory_state, $k, \boldsymbol{\xi}$)
 8:     $\Psi_{proposed} \leftarrow$ plan_short_term_plan($k, \psi^1_{[t_k]}, \boldsymbol{\xi}$)
 9:     safe$_{k+1} \leftarrow$ verify($k, \mathcal{C}, \Psi_{proposed}, \mathcal{H}, \boldsymbol{\xi}$)
10:     if safe$_{k+1}$ then
11:         $\Psi_{current} \leftarrow \Psi_{proposed}$
12:     $k \leftarrow k + 1$
```

We next show how to generate the failsafe and the recovery manoeuvres ψ^0 and ψ^1 subject to joint acceleration and jerk limits. We denote the Euclidean norm by $\|\boldsymbol{z}\|$, and $|\boldsymbol{z}|$ represents a vector of the absolute values of the elements of $\boldsymbol{z}$.

2.2.2 Planning manoeuvres subject to limited acceleration and jerk

The method of [46] finds a time-optimal trajectory of joint values $\boldsymbol{q}$ in time, subject to limits on the derivatives of $\boldsymbol{q}$ and given the value of $\boldsymbol{q}$ and some of its derivatives at the start and the end of the trajectory. We adapt this method, using $\dot{s}$ instead of $\boldsymbol{q}$. Recall that the time-scaling manoeuvre in Def. 2.8 requires defining $\dot{s}$ and $\ddot{s}$ at the start and the end of the trajectory

Algorithm 2 `verify`

```
Input: $k$, safety criteria $\mathcal{C} = \{C_1, \dots C_K\}$, short-term plan $\Psi$, human sensor data including timestamp $\mathcal{H}_{t_h}$, long-term plan $\boldsymbol{\xi}$
Output: $is\_safe \in \{0, 1\}$
for $\kappa = 1 : K$ do
    Find time $t_{e,C_\kappa}$ from which robot is safe while following $\Psi$ according to $C_\kappa$
    if $t_{e,C_\kappa} > t_k$ then        // If the robot is not already safe
        RRO $\leftarrow$ find_robot_reachable_occupancy($\Psi, t_k, t_{e,C_\kappa}, \boldsymbol{\xi}$)
        HRO $\leftarrow$ find_human_reachable_occupancy($\mathcal{H}_{t_h}, t_k, t_{e,C_\kappa}$)
        if RRO $\cap$ HRO $\neq \emptyset$ then
            return 0
return 1
```

Algorithm 3 `plan_short_term_plan`

```
Input: $k$, time-scaling manoeuvre $\psi^{\eta}_{[t_a]}$ where $t_a \leq t_k$, long-term plan $\boldsymbol{\xi}$
Output: short-term plan $\Psi_{[t_k]}$
trajectory_state $\leftarrow [\psi^{\eta}_{[t_a]}(t_{k+1}), \dot{\psi}^{\eta}_{[t_a]}(t_{k+1}), \ddot{\psi}^{\eta}_{[t_a]}(t_{k+1})]^\top$
$\psi^{0}_{[t_{k+1}]} \leftarrow$ plan_failsafe(trajectory_state, $k+1, \boldsymbol{\xi}$)
$\Psi_{[t_k]}(t) \leftarrow \begin{cases} \psi^{\eta}_{[t_a]}(t) & t_k \leq t < t_{k+1} \\ \psi^{0}_{[t_{k+1}]}(t) & t \geq t_{k+1} \end{cases}$
```

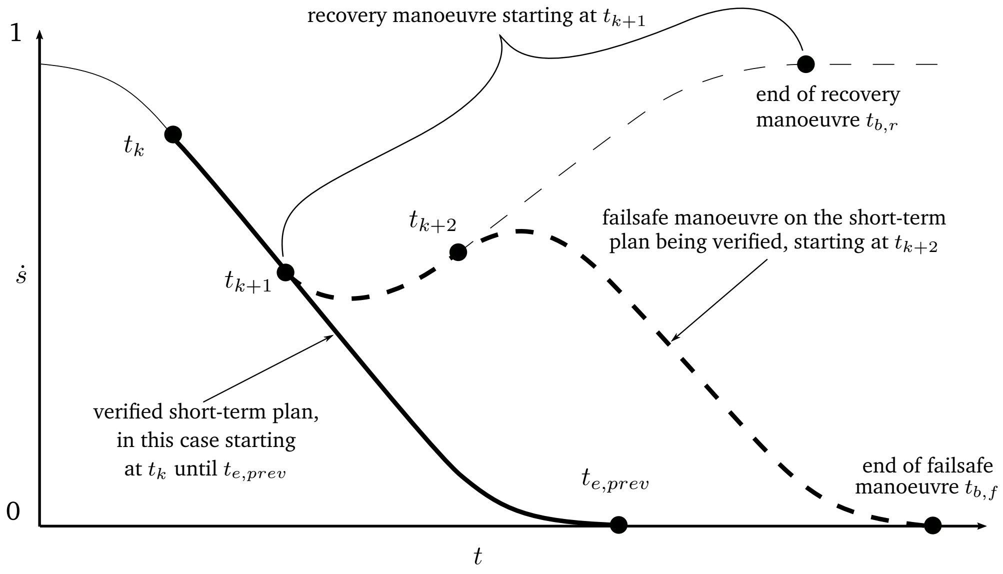


Figure 2.3: Planning of short-term plan. As the robot executes the verified short-term plan during time t_k to t_{k+1} (shown in bold line), it plans the next short term plan (bold dashed line), consisting of one step along a recovery manoeuvre (dashed line) until t_{k+2}, followed by a new failsafe manoeuvre starting from t_{k+2}.

(s is only defined at the start of the trajectory), and we require limits on the maximum joint accelerations and jerks, i.e. we require:

$$\begin{aligned} |\ddot{\boldsymbol{\xi}}(s)| &\leq \boldsymbol{a}_m, \\ |\dddot{\boldsymbol{\xi}}(s)| &\leq \boldsymbol{j}_m, \end{aligned} \tag{2.1}$$

for some maximum joint accelerations and jerks $\boldsymbol{a}_m$ and $\boldsymbol{j}_m$. We show that we can satisfy these limits by calculating bounds for $\ddot{s}$ and $\dddot{s}$ over the length of the manoeuvre such that (2.1) is satisfied over the manoeuvre; we can then use the method of [46] to plan a time-optimal trajectory in $\dot{s}$. We call the values of s at the beginning and end of the time scaling manoeuvre s_a and s_b respectively, as in Def. 2.8. Note that $\boldsymbol{\xi}, \boldsymbol{a}_m, \boldsymbol{j}_m \in \mathbb{R}^m$, where m is the number of joints of the robot.

Lemma 2.1: Given is a long-term plan $\boldsymbol{\xi}(s)$ which is thrice differentiable in s. Let $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ be vectors whose elements are the maximum magnitude of $\frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}s^2}$ and $\frac{\mathrm{d}^3\boldsymbol{\xi}}{\mathrm{d}s^3}$ over the long-term plan between s_a and s_b, for each joint i. That is, where θ_i, λ_i and ξ_i are the i^{th} element of $\boldsymbol{\theta}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$, then $\theta_i = \max_{s \in [0,s_f]} \left(\left|\frac{\mathrm{d}^2\xi_i}{\mathrm{d}s^2}\right|\right)$ and $\lambda_i = \max_{s \in [0,s_f]} \left(\left|\frac{\mathrm{d}^3\xi_i}{\mathrm{d}s^3}\right|\right)$. It is guaranteed that $\ddot{\boldsymbol{\xi}}(s) \leq \boldsymbol{a}_m$ during a time scaling manoeuvre during which $s \leq s_b$, if $\ddot{s}_m$, the limit of $|\ddot{s}|$ over the manoeuvre, is set to:

$$\ddot{s}_m = \max(\min(\boldsymbol{c}), 0), \quad \boldsymbol{c} = \frac{\boldsymbol{a}_m - \boldsymbol{\theta}}{\left|\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}|_{s_a}\right| + \boldsymbol{\theta}(s_b - s_a)}, \tag{2.2}$$

where, for the preceding and subsequent equation $\min(\boldsymbol{z})$ returns the minimum element of the vector $\boldsymbol{z}$, and division is elementwise, Furthermore, it is guaranteed that $\dddot{\boldsymbol{\xi}}(s) \leq \boldsymbol{j}_m$ if $\dddot{s}_m$, the limit of $|\dddot{s}|$ over the manoeuvre, is set to:

$$\dddot{s}_m = \max(\min(\boldsymbol{d}), 0), \quad \boldsymbol{d} = \frac{\boldsymbol{j}_m - \boldsymbol{\lambda} - 3\boldsymbol{\theta}\ddot{s}_m}{\left|\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}|_{s_a}\right| + \boldsymbol{\theta}(s_b - s_a)}. \tag{2.3}$$

□

Proof: We obtain the inequalities which must be satisfied by our choice of $\ddot{s}$ and $\dddot{s}$ using the chain rule:

$$\ddot{\boldsymbol{\xi}}(s) = \frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}s^2}\dot{s}^2 + \frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}\ddot{s} \leq \boldsymbol{a}_m; \tag{2.4a}$$

$$\dddot{\boldsymbol{\xi}}(s) = \frac{\mathrm{d}^3\boldsymbol{\xi}}{\mathrm{d}s^3}\dot{s}^3 + 3\frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}s^2}\ddot{s}\dot{s} + \frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}\dddot{s} \leq \boldsymbol{j}_m. \tag{2.4b}$$

The following are stricter criteria than (2.4a) and (2.4b):

$$\left|\frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}s^2}\dot{s}^2\right| + \left|\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}\ddot{s}\right| \leq \boldsymbol{a}_m; \tag{2.5a}$$

$$\left|\frac{\mathrm{d}^3\boldsymbol{\xi}}{\mathrm{d}s^3}\dot{s}^3\right| + \left|3\frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}s^2}\ddot{s}\dot{s}\right| + \left|\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}\dddot{s}\right| \leq \boldsymbol{j}_m. \tag{2.5b}$$

Rearranging (2.5a) and (2.5b), we obtain the requirements:

$$|\ddot{s}\cdot\mathbf{1}| \leq \frac{\boldsymbol{a}_m - \left|\frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}s^2}\dot{s}^2\right|}{\left|\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}\right|}; \tag{2.6a}$$

$$|\dddot{s}\cdot\mathbf{1}| \leq \frac{\boldsymbol{j}_m - \left|\frac{\mathrm{d}^3\boldsymbol{\xi}}{\mathrm{d}s^3}\dot{s}^3\right| - \left|3\frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}s^2}\ddot{s}\dot{s}\right|}{\left|\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}\right|}. \tag{2.6b}$$

Here, $\mathbf{1} \in \mathbb{R}^n$ is a vector of ones, and division is elementwise. We require the inequalities of (2.6a) and (2.6b) to hold for $s \in [s_a, s_b]$. We consider firstly (2.6a). As we know the global trajectory, we can calculate $\boldsymbol{\theta}$, the maximum of $\frac{\mathrm{d}^2\boldsymbol{\xi}(s)}{\mathrm{d}s^2}$ between $[s_a, s_b]$, and substitute this into (2.6a)[1]. Since $\dot{s} \in [0,1]$, the numerator of the right-hand side attains a minimum at $\boldsymbol{a}_m - \boldsymbol{\theta}$. The global trajectory $\boldsymbol{\xi}$ is at least twice differentiable in s, so in the denominator we use the Lagrange Remainder Theorem, which states $\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s} = \frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}|_{s_a} + \frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}s^2}|_{s^*}(s^* - s_a)$ for some $s^* \in [s_a, s_b]$. This is upper bounded by $\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}|_{s_a} + \boldsymbol{\theta}(s_b - s_a)$. A lower bound of the right-hand side of (2.6a) is therefore:

$$\frac{\boldsymbol{a}_m - \boldsymbol{\theta}}{\left|\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}|_{s_a}\right| + \boldsymbol{\theta}(s_b - s_a)} = \boldsymbol{c}, \tag{2.7}$$

and we take $\ddot{s}_m = \max(\min(\boldsymbol{c}), 0)$. Consider now (2.6b). By additionally using $\boldsymbol{\lambda}$, the maximum of $\frac{\mathrm{d}^3\boldsymbol{\xi}(s)}{\mathrm{d}s^3}$ between $[s_a, s_b]$, and the result $\ddot{s}_m$, we lower-bound[2] the numerator of the right-hand side with $\boldsymbol{j}_m - \boldsymbol{\lambda} - 3\boldsymbol{\theta}\ddot{s}_m$. The denominator is the same as (2.6a). The right-hand side of (2.6b) has a lower bound:

$$\frac{\boldsymbol{j}_m - \boldsymbol{\lambda} - 3\boldsymbol{\theta}\ddot{s}_m}{\left|\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}|_{s_a}\right| + \boldsymbol{\theta}(s_b - s_a)} = \boldsymbol{d}, \tag{2.8}$$

and we choose $\dddot{s}_m = \max(\min(\boldsymbol{d}), 0)$. □

Since s_b is not known *a priori*, we use a conservative estimate s_b' which we expect to be longer than the actual s_b; if it happens that $s_b > s_b'$, we recalculate with an even more conservative estimate of s_b, until $s_b' \geq s_b$. To keep this real-time, we may need to limit the number of iterations.

A number of modifications can be made to these calculations. In a trajectory where the extrema of $\frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}s^2}$ and $\frac{\mathrm{d}^3\boldsymbol{\xi}}{\mathrm{d}s^3}$ can be easily calculated such as a joint-space point-to-point trajectory that is polynomial in time, finding $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ is not a difficult task. For more complex trajectories where it is difficult to determine the maximum jerks and accelerations online, these may be calculated piecewise over the trajectory in advance.

Drawbacks of this method are the following:

- It is not strictly guaranteed that the manoeuvres generated maintain $0 \leq \dot{s} \leq 1$. In practice, this is only a problem when $\dddot{s}_m$ or $\ddot{s}_m$ change rapidly from one timestep to

[1] this is feasible as long as $\boldsymbol{\theta} << \boldsymbol{a}_m$, otherwise the numerator of (2.7) is too small, or even negative, and hence $\ddot{s}_m$ is too restricted, or even zero.

[2] again, only feasible if $\boldsymbol{\lambda} + 3\boldsymbol{\theta}\ddot{s}_m << \boldsymbol{j}_m$ for the same reason as above.

the next, causing the trajectory of $\ddot{s}$ to "overshoot" 0 or 1. Since at all times we have a failsafe trajectory available, verified in advance, which brings the robot to a safe state, if a proposed short-term plan were to overshoot at any point during the plan, this short-term plan is verified as unsafe and the failsafe trajectory is chosen.

- Again, when $\dddot{s}_m$ or $\ddot{s}_m$ change rapidly between timesteps, it may occur that $|\ddot{s}| > \ddot{s}_m$ at the start of a proposed short-term plan. Again, if this occurs, that particular short-term plan is verified as unsafe and the failsafe trajectory is chosen.

- If it is not the case that $\boldsymbol{\theta} << \boldsymbol{a}_m$ or $\boldsymbol{\lambda} + 3\boldsymbol{\theta}\ddot{s}_m << \boldsymbol{j}_m$, then the stopping trajectory may be very long, meaning that we need to recalculate s'_b several times, or even impossible to calculate (if e.g. $\ddot{s}_m$ or $\dddot{s}_m$ are zero)

The last item can be a problem in industrial robotics, where the demanded joint accelerations and torques are often close to the maximum, and we can modify the planning algorithm (Alg. 1) as follows:

Algorithm 4 Formally Verified Trajectory Planning

```
Input: long-term plan $\boldsymbol{\xi}$, safety criteria $\mathcal{C}$, human sensor data including timestamp $\mathcal{H}_{t_h}$
Output: Safe trajectory
 1: $k \leftarrow 0$
 2: safe$_k \leftarrow 0$
 3: $\Psi_{current} \leftarrow$ plan_failsafe(trajectory_state, $k, \boldsymbol{\xi}$)
 4: while not at goal state do
 5:     execute_control_on_robot($\Psi_{current}, k$)
 6:     safe$_k \leftarrow 0$
 7:     for $\eta \in \{\eta_1, \eta_2, ...1\}$ do
 8:         $\psi^{\eta}_{[t_k]} \leftarrow$ plan_time_scaling_manoeuvre(trajectory_state, $k, \boldsymbol{\xi}, \eta$)
 9:         $\Psi^{\eta}_{proposed} \leftarrow$ plan_short_term_plan($k, \psi^{\eta}_{[t_k]}, \boldsymbol{\xi}$)
10:         safe$_{k+1} \leftarrow$ verify($k, \mathcal{C}, \Psi_{proposed}, \mathcal{H}_{t_h}$)
11:         if safe$_{k+1}$ then
12:             $\Psi_{current} \leftarrow \Psi_{proposed}$
13:     $k \leftarrow k + 1$
```

Here in lines 7–12 we check in ascending order for each η, whether we can plan a recovery manoeuvre recovering only to η and not to 1. Again, we make the assumption that $\dot{s} \in [0, \eta]$ throughout the manoeuvre, which we must check this analytically after planning the manoeuvre. In this case equations (2.6) yield:

$$\ddot{s}_m = \max(\min(\boldsymbol{c}), 0), \quad \boldsymbol{c} = \frac{\boldsymbol{a}_m - \eta^2\boldsymbol{\theta}}{\left|\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}|_{s_a}\right| + \boldsymbol{\theta}(s_b - s_a)},$$
$$\dddot{s}_m = \max(\min(\boldsymbol{d}), 0), \quad \boldsymbol{d} = \frac{\boldsymbol{j}_m - \eta^3\boldsymbol{\lambda} - 3\eta\boldsymbol{\theta}\ddot{s}_m}{\left|\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}|_{s_a}\right| + \boldsymbol{\theta}(s_b - s_a)}.$$

Having shown how to obtain the bounds on $|\dddot{s}|$ and $|\ddot{s}|$, we are now able to use the method of [46] to plan a failsafe or recovery manoeuvre in $\dot{s}$ subject to the constraints that $\ddot{s} = 0$ and $\dot{s} = \eta$ at the end of the manoeuvre, and the maximum second and third derivatives of the trajectory parameter are bounded, $|\ddot{s}| \leq \ddot{s}_m$ and $|\dddot{s}| \leq \dddot{s}_m$. This method is based on decision trees and is fast. Details are given in Appendix A.1

2.2.3 Implementation with different criteria for safety

So far, we have introduced two criteria for safety in human-robot co-existence (HRCoex) and shown how to plan short-term plans by scaling the robot's velocity, which bring the robot to a safe state according to Def. 2.5 while satisfying joint accelerations and jerk limits.

We must first consider when the second criterion for safety should be used. Recall that ISO standard 10218-1 [2] states that the robot can be slowed to the "Safety-rated" speed of $0.25\frac{m}{s}$ since this "allows sufficient time for people either to withdraw from the hazardous area or to stop the robot". This suggests, as common sense would also dictate, that the slowing of the robot is a precursor to eventually stopping the robot if the human approaches further, and that the robot should not simply be allowed to operate at the reduced speed without further safety precautions, since this could lead to undesirable contact scenarios.

Such further precautions could be, for example, a collision-detection algorithm based on joint-torque sensing which switches the robot to gravity compensation mode with impedance control when a collision is detected [7]. Such control strategies are a promising branch of collaborative robotics, though outside the scope of this thesis.

We propose in [180] a framework whereby the two criteria of safety are applied in parallel. The robot behaviour is verified against two criteria: The robot must stop before collision, according to the "stopping criterion" C_{stop} in Def. 2.5, if the human is moving according to assumptions on normal human movement from ISO standard 13855 [4]. If the human is moving unexpectedly, the robot must be moving at no more than $v_{\max} = 0.25\frac{m}{s}$, as per the reduced-speed criterion C_{red} in Def. 2.6.

In Sec. 4.2.2, we show how to generate HROs based on the assumptions from [4], i.e., that the maximum speed of the human does not exceed $2\frac{m}{s}$. In Sec. 4.3 and Sec. 4.4, we present methods for calculating HROs which enclose all human movement in an human-robot interaction (HRI) scenario. The reader is referred to these sections for further information; in the following, we denote the former HRO with Γ_{ISO} and the latter with Γ^{e}.

As shown in Alg. 2, we use the same short-term planning for both occupancies, only the verification differs. Let $\Psi_{[t_k]}$ be the short-term plan which starts at the timestep t_k. According to the stopping criterion, the robot is safe when it is stationary; this occurs when $\dot{s} = 0$; call this time $t_{e,C_{stop}}$. In lines 2–5 in Alg. 2, we calculate the HRO $\Gamma_{ISO}([t_k, t_{e,C_{stop}}])$ and the RRO $F([t_k, t_{e,C_{stop}}])$, and check for intersection in line 6. The time when the reduced-speed criterion is first satisfied $t_{e,C_{red}}$. Again, in lines 2–5 in Alg. 2, we calculate the reachable occupancies $\Gamma^e([t_k, t_{e,C_{red}}])$—note this time Γ^e instead of Γ_{ISO}—and $\mathcal{F}([t_k, t_{e,C_{red}}])$, and check their intersection in line 6. We detail how to find $t_{e,C_{red}}$ in the next section.

2.2.4 Calculating Time at which the Robot Becomes Safe

This section is concerned with line 2 in Alg. 2. With respect to all criteria that are verified, the time along the short-term plan until the robot becomes and stays safe according to a specific criterion of safety must be calculated, to calculate the RRO and the HRO during that time and ensure that they do not intersect.

Finding $t_{e,\text{stop}}$ is easy, as it is simply the time on the failsafe manoeuvre $\psi^0_{[t_k+1]}$ attached to the end of $\Psi_{[t_k]}$ where $\dot{s} = 0$. In contrast, $t_{e,\text{red}}$ is more tricky to find. It cannot be assumed that the speed at all points of the robot monotonically decreases over the entire trajectory. We therefore iterate backwards from $t_{e,\text{stop}}$ to t_k to find the point at which the maximum speed on the robot exceeds $v_{\max}$, and then take the previous point. This is shown in Alg. 5.

Algorithm 5 Check when robot is safe with reduced-speed criterion

```
Input: short-term plan $\Psi_{[t_k]}$, long-term plan $\boldsymbol{\xi}$
Output: $t_{e,\text{red}}$
$t \leftarrow t_{e,\text{stop}}$
while $t > t_k$ do
    $v \leftarrow$ find_fastest_point_on_robot($\boldsymbol{\xi}(\Psi_{[t_k]}(t)), \dot{\boldsymbol{\xi}}(\Psi_{[t_k]}(t))$)
    if $v > v_{\max}$ then
        return $t \leftarrow t + \Delta t$
    $t \leftarrow t - \Delta t$
return $t_k$
```

We show how to find the fastest point on the robot in App. A.2. To ensure deterministic latencies, we limit the number of iterations; past a certain number, the criterion is simply verified unsafe.

2.3 Long-term Planning

In cases where the global plan is blocked, e.g. if the human stands in the way of the robot, it may be useful for efficiency reasons to plan a new global path. This brings with it some new challenges. Firstly, all the challenges of autonomous path-planning, including environment modelling and obstacle avoidance, must be addressed. Secondly, continuity with the old path in terms of respecting acceleration and jerk limits is important, as is ensuring the new path is more likely to be verified safe than the old path. Thirdly, a predictive model of the human may be needed, to decide when a blockage of the global path is temporary and when it is likely to persist.

A final consideration is that spatial path planning with collision avoidance can be computationally expensive. Alg. 1 requires that the control input to be applied from time t_{k+1} to t_{k+2} be known at time t_k, and, if the failsafe manoeuvre is to be path-consistent, the path should be known until a sufficient future time so that the robot can plan a path-consistent stop. This requires either highly adaptable and fast methods for path-planning, or new methods to integrate the longer cycle time of the path planner with the shorter verifier cycle.

In [179] we address this problem by proposing a model-predictive control (MPC)-based framework based on Minimum-Intervention Control, where the trajectory is represented not as a mapping from time to position, as in Def. 2.7, but as a distribution of trajectories. If a collision is predicted at some future time, collision-free *via points* are generated and the trajectory is modulated to go through these points using the method of [47]. Since this replanning is fast, the trajectory could be planned up to 50 control cycles in the future, and the verification could be performed, within one control cycle.

However, this method was tested only on the trajectory of a point and not for an entire fixed-base robot; choosing via points for serial-link manipulators may prove to be a trickier task. This approach has also not addressed collisions with the environment or formally bounding acceleration and jerk limits. Furthermore, trajectories are not often represented as distributions in industrial scenarios. We therefore develop more general techniques which allow trajectory replanners with longer cycle times to coexist with a verifier on a shorter cycle time.

2.3.1 Prediction of the Human

We first consider prediction of the medium-term motion of the human. Unlike what will be discussed in Chapter 4, our predictions are not safety-critical and hence we can use probabilistic predictions rather than needing to account for all human movement.

Several models for predicting human motion to inform robot motion have been proposed. For mobile robots, a wide range of probabilistic predictions are used [48]. Rohrmüller *et al.* [49] use reachability analysis, but with a probabilistic approach, to predict humans' probable occupancy and modulate robot trajectories to avoid humans. An approach based on the environmental context is presented by [23], who use the affordances of nearby objects to infer the intended goal of the human, for an assistive robot.

Focussing on robots in industrial environments, prediction of human reaching motions are more relevant. Hoffmann and Breazeal [50] use Markov Processes to predict human movement at a workbench, and adapt robot movement to this. User studies indicated increased fluency with the adaptive robot motion. Using initial arm motion to infer future movement from a Gaussian Mixture Model trained on recorded data, [21], Mainprice and Berenson compute a voxel map of the probability of occupancy by a human, using this as a cost function to guide path planning [51]. Ding *et al.* [22] use HMM, first using the approach of [52] to compute a *topological map* of states, which are then trained on recorded data and used for prediction. Other medium-term prediction algorithms worth noting are those of Perez D'Arpino [53] using classification on early motion, and Zanchettin [54] using Bayesian inference to predict the target of a reaching motion. Pelegrinelli *et al.*[55] present a framework which uses a Partially Observable Markov Decision Process (POMDP) to infer a *probability distribution* over the goals of the human (possibly varying over time), which is used to plan robot motion. If the goal is known, Mainprice *et al.*[56] use an inverse-optimal control method to learn a cost function describing human reaching motions, which can be used to predict human motion.

In our own work [182, 183], the approach of [22] was implemented and extended. In [183] it was found that the effectiveness of the approach was limited by the sensing modalities. Different sensors were tested including a camera with fiducial markers, a depth sensor with reflective markers (both on the X-Box Kinect), and a 2-camera motion-tracking system and reflective markers. However, it was also found that a joint hidden Markov model (HMM) of the entire human arm did not perform better than individual HMM of each marker on the arm, while being computationally more expensive; this implies that the prediction can be parallelised. In [182] the approach was trialled using both a 6-Camera infrared motion capture system, and the X-Box Kinect v2 with the skeleton-tracking software available as part of the SDK [57]. To increase speed of prediction, oct-trees were used to grid the state space, so that regions of space that were occupied more often had higher resolution.

2.3.2 Replanning on Different Timescales

We show how replanning trajectories around humans can be made consistent with the formally verified approach, and discuss factors which affect its effectiveness. For simplicity, our desired and replan trajectories are designed in joint space, though any trajectory that can be parameterised in terms of s can be used. Our desired trajectory is a quintic polynomial starting and ending at zero acceleration. The replan trajectories are quartic-cubic-quartic splines [58], starting at a point on the desired trajectory, going through 2 heuristically-determined via points, and ending at the goal position of the desired trajectory. These quartic-cubic-quartic splines can be designed such that they adhere to joint acceleration and jerk limits. Their derivation is explained in detail in Appendix. A.3.

2.3.2.1 Reconciling Different Cycle Times

Assume firstly that an upper bound for the time the replanner requires to compute a replan path is n times the duration of the verifier control cycle; we then set the duration of the replanner control cycle at n times the verifier control cycle. Replanning starts at time t_k and the replanner starts planning a manoeuvre which would start at time t_{k+n+2}. The reason for the extra 2 control cycles is elucidated below; the concept is illustrated in Fig. 2.4.

Since the replanning takes longer than the control cycle of the verifier, it is necessary for the replanner to know the state of the robot at the time t_{k+n+2} at which the failsafe manoeuvre should start, so that the replanner can plan a trajectory starting from that state, continuous in position, velocity and acceleration with the motion of the robot at that time. However, the only

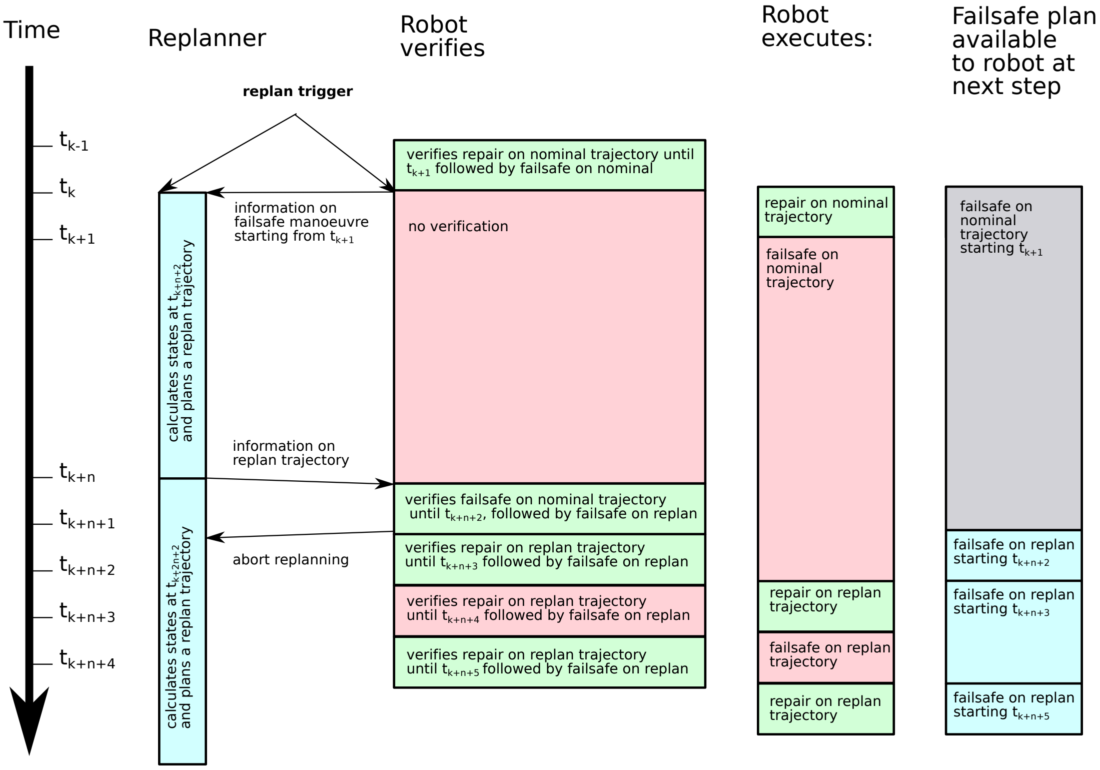

Time
Replanner
Robot verifies
Robot executes:
Failsafe plan available to robot at next step
replan trigger
t_{k-1}
t_k
t_{k+1}
t_{k+n}
t_{k+n+1}
t_{k+n+2}
t_{k+n+3}
t_{k+n+4}
calculates states at t_{k+n+2} and plans a replan trajectory
calculates states at t_{k+2n+2} and plans a replan trajectory
information on failsafe manoeuvre starting from t_{k+1}
information on replan trajectory
abort replanning
verifies repair on nominal trajectory until t_{k+1} followed by failsafe on nominal
no verification
verifies failsafe on nominal trajectory until t_{k+n+2}, followed by failsafe on replan
verifies repair on replan trajectory until t_{k+n+3} followed by failsafe on replan
verifies repair on replan trajectory until t_{k+n+4} followed by failsafe on replan
verifies repair on replan trajectory until t_{k+n+5} followed by failsafe on replan
repair on nominal trajectory
failsafe on nominal trajectory
repair on replan trajectory
failsafe on replan trajectory
repair on replan trajectory
failsafe on nominal trajectory starting t_{k+1}
failsafe on replan starting t_{k+n+2}
failsafe on replan starting t_{k+n+3}
failsafe on replan starting t_{k+n+5}

Figure 2.4: Integration of replanner with cycle time $n\Delta t$ with verifier and control of cycle time Δt. Here we see that the failsafe manoeuvre on the replan trajectory is verified safe at time t_{k+n+1}. If this were not the case, the robot would continue on the failsafe manoeuvre on the nominal trajectory while the replanner replans a new trajectory. This is then verified from t_{k+2n} to t_{k+2n+1}. If this is not verified, then the robot again continues on the failsafe manoeuvre on the nominal trajectory and keeps trying to replan the trajectory.

robot state the replanner knows for certain is the state at t_{k+1}, because the control input for timestep $[t_k, t_{k+1}]$ has been verified safe prior to t_k, see Alg. 1, but control inputs after t_{k+1}have not yet been verified. Hence the only information available to the replanner is the state at t_{k+1}, the nominal trajectory $\boldsymbol{\xi}(s)$, and the failsafe plan $\psi^0_{[t_\alpha]}(t)$ (where $\alpha = k+1$ if the robot was on the repair manoeuvre from $[t_k, t_{k+1}]$; if the robot was already on the failsafe manoeuvre, then $\alpha < k+1$).

Since it is not known whether the robot will be verified safe at any of those future control cycles from t_{k+1} to t_{k+n+2}, the only safe option is for the robot to execute the failsafe manoeuvre for the entirety of the replanning control cycle until t_{k+n+2}. The replanner can then, at time t_k, obtain the robot's joint position, velocity and acceleration at time t_{k+n+2} from:

$$s_a = \psi^0_{[t_\alpha]}(t_{k+n+2}), \quad \dot{s}_a = \dot{\psi}^0_{[t_\alpha]}(t_{k+n+2}), \quad \ddot{s}_a = \ddot{\psi}^0_{[t_\alpha]}(t_{k+n+2}) \tag{2.9}$$

$$\begin{aligned}
\boldsymbol{q} &= \boldsymbol{\xi}(s_a) \\
\dot{\boldsymbol{q}} &= \left.\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}\right|_{s_a} \dot{s}_a \\
\ddot{\boldsymbol{q}} &= \left.\frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}s^2}\right|_{s_a} \dot{s}_a^2 + \left.\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}\right|_{s_a} \ddot{s}_a
\end{aligned}$$

The replanner starts replanning the trajectory at t_k and sends it to the verifier at, or before, time t_{k+n}. During the timestep $[t_{k+n}, t_{k+n+1}]$, instead of constructing a short-term plan consisting of a repair trajectory from t_{k+n+1} to t_{k+n+2} followed by a failsafe manoeuvre path-consistent with the nominal trajectory (as in Alg. 1), the verifier constructs a short-term plan consisting of the trajectory of the robot following the failsafe manoeuvre $\psi^0_{[t_\alpha]}$ from t_{k+n+1} to t_{k+n+2} on the nominal trajectory, followed by a failsafe manoeuvre *on the replan trajectory* starting at t_{k+n+2}.

If this short-term plan is verified safe, the short-term plan is adopted and hence the replan trajectory is adopted: from time t_{k+n+2}, the robot is fully on the replan trajectory, whether the first step of the replan trajectory is on the repair manoeuvre (if verified safe in $[t_{k+n+1}, t_{k+n+2}]$) or the failsafe manoeuvre on the replan trajectory, verified in the previous timestep.

If not, the robot continues on the failsafe manoeuvre of the nominal trajectory. The replanner discards the replan trajectory and plans a new one starting at the position of the robot on the failsafe manoeuvre $\psi^0_{[t_\alpha]}$ (on the nominal trajectory) at t_{k+2n+2}. In fact, the replanner can start planning this new replan trajectory at t_n, as soon as it has sent the old replan trajectory to the robot control, since the robot has not left the failsafe manoeuvre $\psi^0_{[t_\alpha]}$ since t_{k+1}; the new replan trajectory can then be ignored or replanning aborted, if the old replan trajectory is adopted.

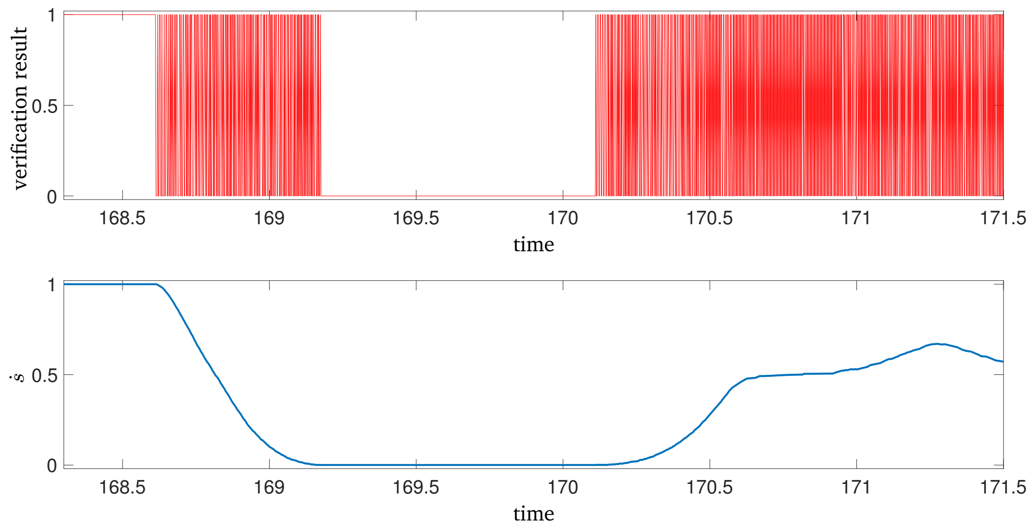


Figure 2.5: Reaction of the robot to a human. The verification result (top plot) and $\dot{s}$ (the change in trajectory parameter, i.e. the proportion of actual speed to nominal speed) are plotted against time. In the left of the figure, the verification result starts out as 1 (safe) as does $\dot{s}$. As the robot approaches the human, the verification result alternates between 1 and 0 (safe and unsafe), eventually settling at 0 as the human gets too close for any repair trajectory to be verified safe. The failsafe manoeuvre is completed and the robot is stationary. Around time 170.1, the human moves slightly out of the way, so that the robot is able to move slowly further on its trajectory.

2.3.2.2 Replanning Condition

When should the robot give up on its current path and try replanning? One could use a probabilistic prediction of the human as in Sec. 2.3.1 to predict whether the human is likely to move out of the way in the next t_{wait} seconds, where t_{wait} is lower the more "impatient" the robot should be. One could also wait until the robot comes to a standstill, although this is likely to be so close to the human that even movement away from the human will be often verified unsafe.

It is useful to consider what happens when the human gets in the way of the robot. Such a situation is shown in Fig. 2.5. Since the failsafe manoeuvre is planned for the worst-case situation, i.e. when the human moves towards the robot as fast as they can, the failsafe manoeuvre is only *fully* executed in such a situation. In most cases, the human is not moving as fast as possible. Therefore, it is often the case that the robot starts the failsafe manoeuvre and registers that the human has not moved so extremely, in which case it is often able to verify subsequent steps safe. The robot then enters a period of alternately verifying safe and unsafe as shown on the left of Fig. 2.5. If the human continues to stand in the way of the robot, and the robot does not replan the trajectory, the robot eventually comes to a complete stop, verifying itself unsafe all the time.

A feasible replanning condition is therefore based on setting a threshold for $\dot{s}$. When $\dot{s}$ drops below $\dot{s}_{th}$ for at least t_{wait} seconds, replanning is triggered. Again, setting $\dot{s}_{th}$ higher results in a more “impatient” replanner. We set $\dot{s}_{th}$ to 0.7 and t_{wait} to 1s, since this was found experimentally not to replan when the human was only temporarily in the way, yet not to wait too long before replanning.

2.3.2.3 Setting Accelerations and Jerks for Replan Trajectory

The robot replans its trajectory because its original trajectory is, or is expected to be, verified unsafe too often. But how should the robot maximise the probability of the *replan* trajectory being verified safe, when the original trajectory is not likely to be?

One important factor is the length of the failsafe manoeuvre and hence the short-term plan. The growth in volume of the RRO over time is dwarfed by the growth of the HRO, and if the short-term plan following the replan trajectory is longer than the remaining failsafe trajectory, the HRO to be verified against the short-term plan on the replan trajectory will be large and there is a chance that the replan trajectory will not be verified safe.

Recall the planning of a failsafe manoeuvre described in Sec. 2.2.1. The maximum allowed second and third derivative of the trajectory parameter s with respect to time, $\ddot{s}_m$ and $\dddot{s}_m$, are given by (2.2) and (2.3). These are influenced by the $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$, which are the vectors of the maximum magnitudes of accelerations and jerks, i.e. $\|\frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}s^2}\|$ and $\|\frac{\mathrm{d}^3\boldsymbol{\xi}}{\mathrm{d}s^3}\|$, over the failsafe manoeuvre. The larger $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ are, the smaller $\ddot{s}_m$ and $\dddot{s}_m$ become (as can be seen in equations (2.2) and (2.3)) and hence the longer the failsafe manoeuvre planned. This is intuitive – smaller higher derivatives lead to a slower-changing trajectory, hence it takes longer to bring $\dot{s}$ to zero.

Using the method presented in Appendix A.3, we can plan a polynomial replan trajectory where the maximum joint jerks and accelerations are limited over the entire trajectory, and hence the values of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ are limited to these values. Setting the jerks and accelerations low results in a replan trajectory of longer duration, however, it will lead to lower $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ along the replan path, which will result in larger $\ddot{s}_m$ and $\dddot{s}_m$, a shorter failsafe trajectory along the replan path, and hence a smaller HRO, increasing the chance that the replan will be verified safe.

A good starting guess for the joint acceleration and jerk limits on the replan trajectory are the values of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ on the short-term plan of the nominal trajectory. Since the values of $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ on the short-term plan of the replan trajectory cannot be greater than these values, $\ddot{s}_m$ and $\dddot{s}_m$ for the replan trajectory would be at least as large as those on the nominal trajectory, and the failsafe manoeuvre would be at least as short as the failsafe manoeuvre on the nominal trajectory.

One drawback of using $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ as the maximal accelerations and jerks at those points is that when $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ are low, the limitations on the allowed joint accelerations and velocities make the replan trajectory impractically long (even if the failsafe manoeuvre on the replan trajectory is short). We therefore use an augmented $\boldsymbol{\theta}'$ and $\boldsymbol{\lambda}'$ both in the planning of the failsafe manoeuvre on the nominal trajectory, and also for setting the maximum accelerations and jerks in the replan trajectory. We can either augment $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ using a threshold:

$$\theta_i' = \begin{cases} \theta_i & \theta_i \geq \theta_{i,th} \\ \theta_{i,th} & \theta_i < \theta_{i,th} \end{cases} , \tag{2.10}$$

where θ_i is the i^{th} element of $\boldsymbol{\theta}$; $\boldsymbol{\lambda}'$ is similarly defined. Alternatively, we can augment it so:

$$\theta_i' = \begin{cases} \frac{\theta_i}{\dot{s}} & \dot{s}_{th} \leq \dot{s} \leq 1 \\ \frac{\theta_i}{\dot{s}_{th}} & \dot{s} < \dot{s}_{th} \end{cases} , \tag{2.11}$$

noting that $\dot{s} \in [0, 1]$. $\dot{s}_{th}$ is a threshold value of $\dot{s}$, so as to avoid dividing by 0. Experimentally, it was found that (2.10) gave a feasible trajectory more often, so this was adopted for the experiments in the next section.

2.4 Experiments

We tested our method in experiments, to verify that replanning of the long-term trajectory is indeed possible alongside formal verification. Our replanning—using polynomial splines and two via points—is deliberately very simple; having demonstrated the principle with such simple long-term trajectory planning, further work can focus on optimising this. The experimental setup is as in Sec. 1.3 and details of the experiment are given subsequently.

The via points of the replanning trajectory were chosen as follows. The positions of the 1st , 4th and 6th joints at the first via point were set as the values of these joints at the start of the replan trajectory (i.e. the position of the robot at time t_{k+n+2} in Fig. A.1), and the positions of these joints for the second via point were set at their values at the end of the replan trajectory, i.e. the goal position of the original trajectory. The other joints were set such that the robot lifts its arm vertically upwards and retracts the end-effector towards the base. The prediction of the HRO used here is the ISO model described in Sec. 4.2.2.

During the experiments, the robot proceeded along the desired trajectory when the human was not obstructing the robot, or only temporarily in the way. This is shown in Figs. 2.6 and 2.8: the human is close enough to the robot around the middle of its trajectory that it slows slightly, but not enough to trigger replanning. However, if the human is continually obstructing the robot, a replan is triggered. This is shown in Figs. 2.7 and 2.9. In the V-REP visualisation on the right of Fig. 2.9, the replan trajectory is marked in red, while the desired trajectory is shown in blue. Had the robot not replanned, the robot would never have reached the goal.

Summary

We show the principles of planning robot trajectories that are guaranteed safe and show that we can simultaneously guarantee two different safety criteria for any desired trajectory, given an overapproximative prediction of the human. We also show how to guarantee that the trajectory will fulfil limits on joint acceleration and jerk, which can be useful for preserving the life of the joints and ensuring the computed trajectories are feasible for the robot. Replanning trajectories on the fly may be useful in case the desired trajectory is blocked, we demonstrate how this can be performed while guaranteeing that our safety criteria are still guaranteed.

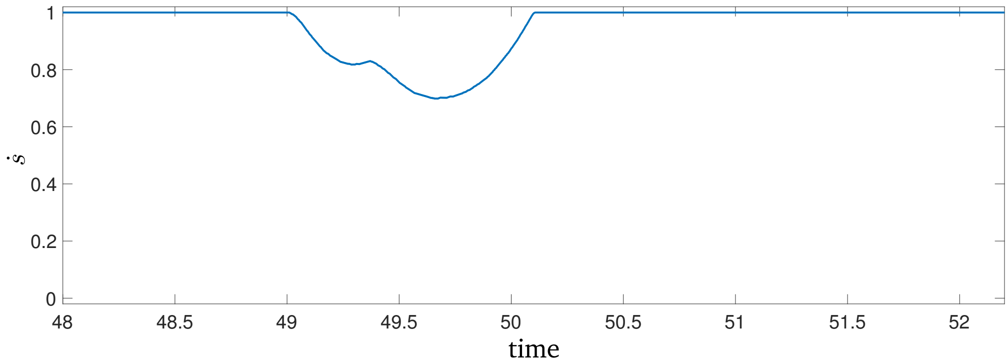


Figure 2.6: Plot of $\dot{s}$ over time, where the human presence causes robot to slow but not to replan

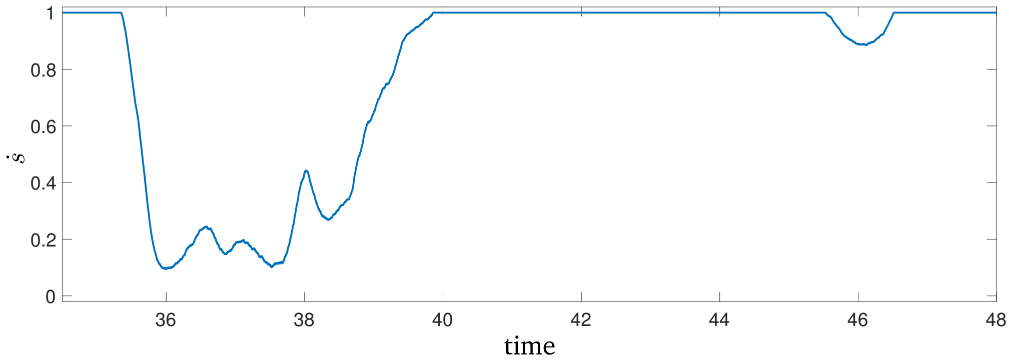


Figure 2.7: Plot of $\dot{s}$ over time, where the human presence causes robot to replan at time 36.5

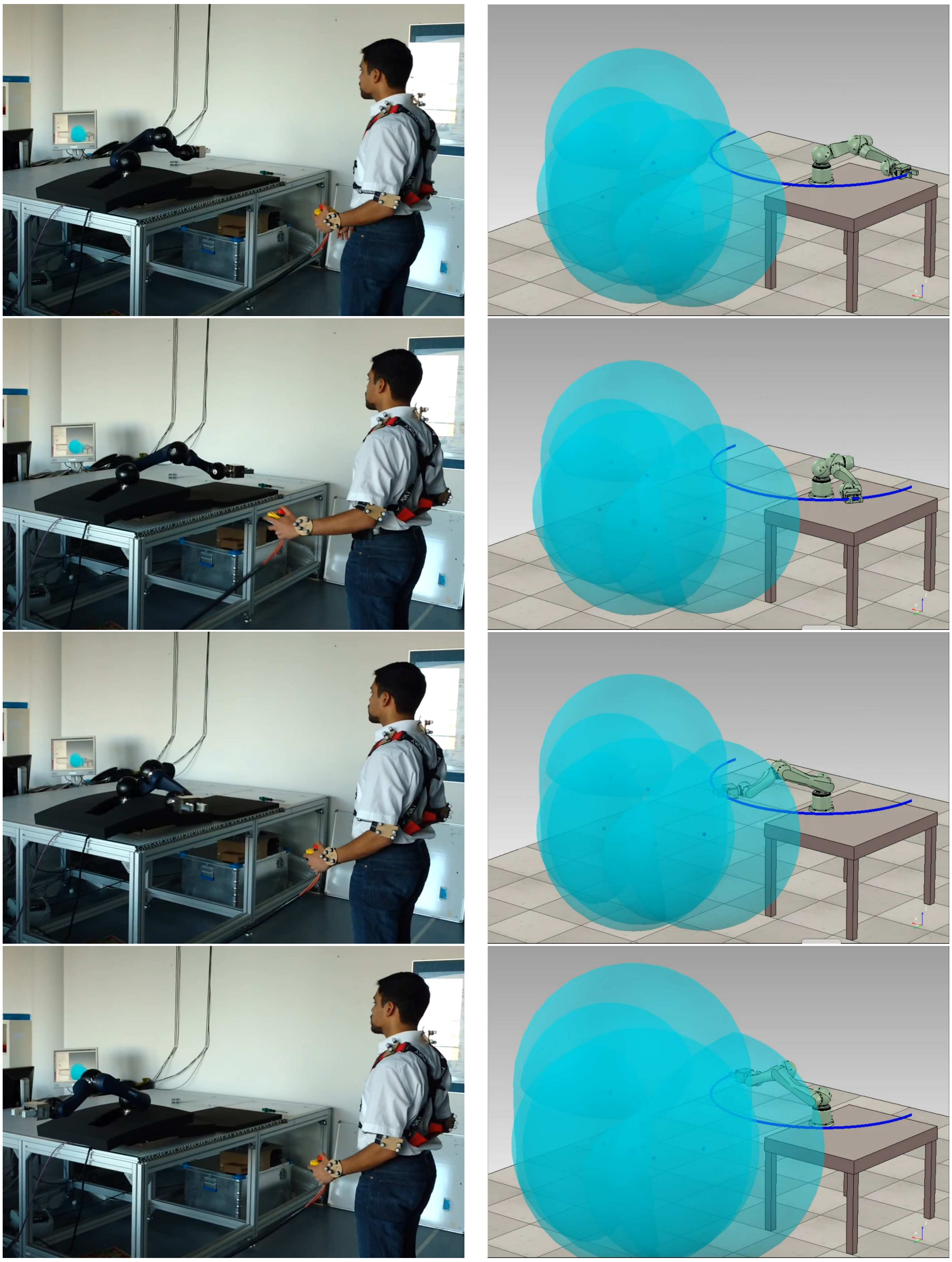


Figure 2.8: Human causes robot to slow but not to replan; speed profile shown in Fig. 2.6. Left: freeze-frames from recording, right: freeze-frames from simultaneous visualisation; HRO in cyan.

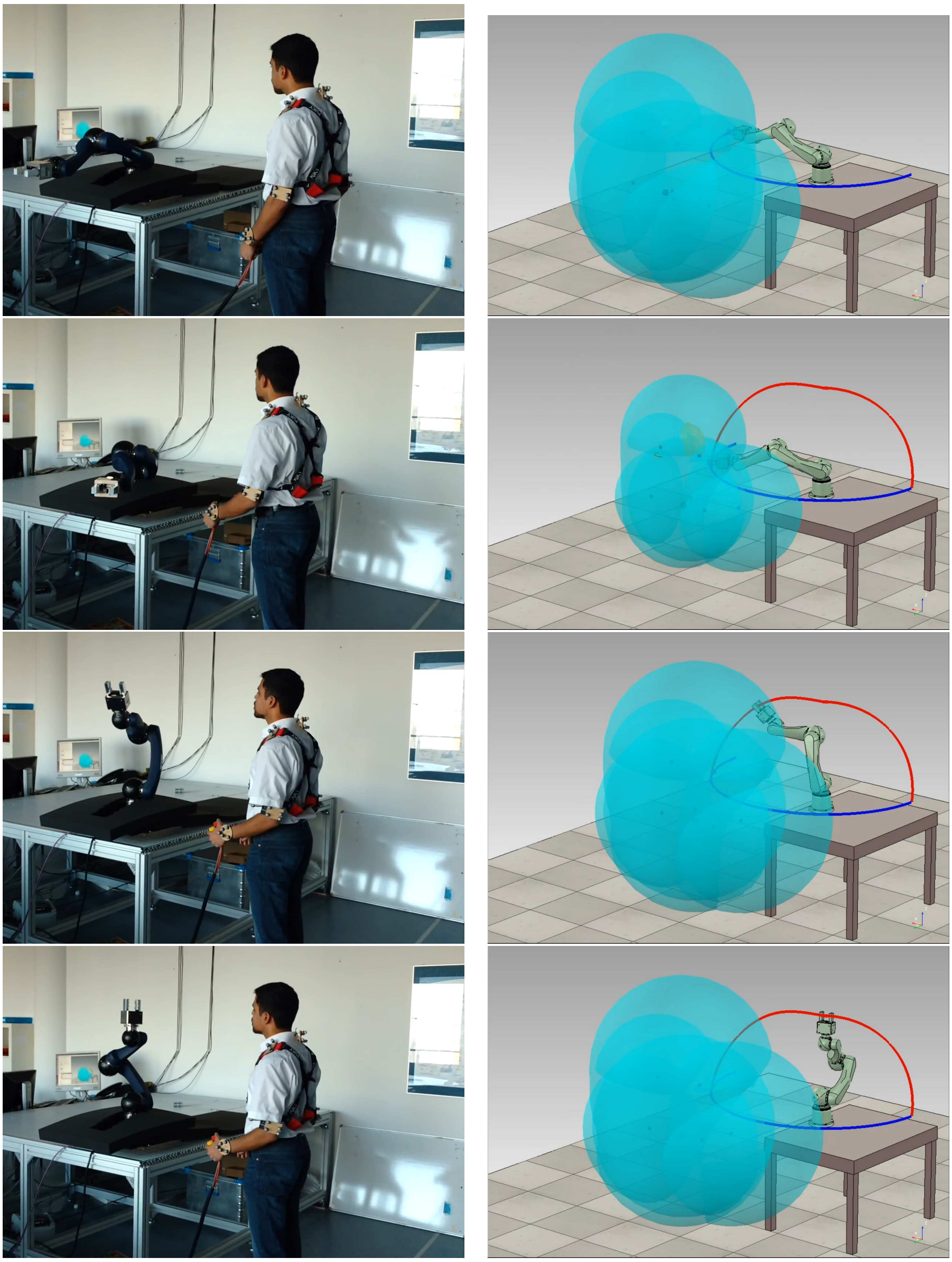


Figure 2.9: Human presence causes robot to replan; speed profile shown in Fig. 2.7. Left: freeze-frames from recording, right: freeze-frames from simultaneous visualisation; HRO in cyan.

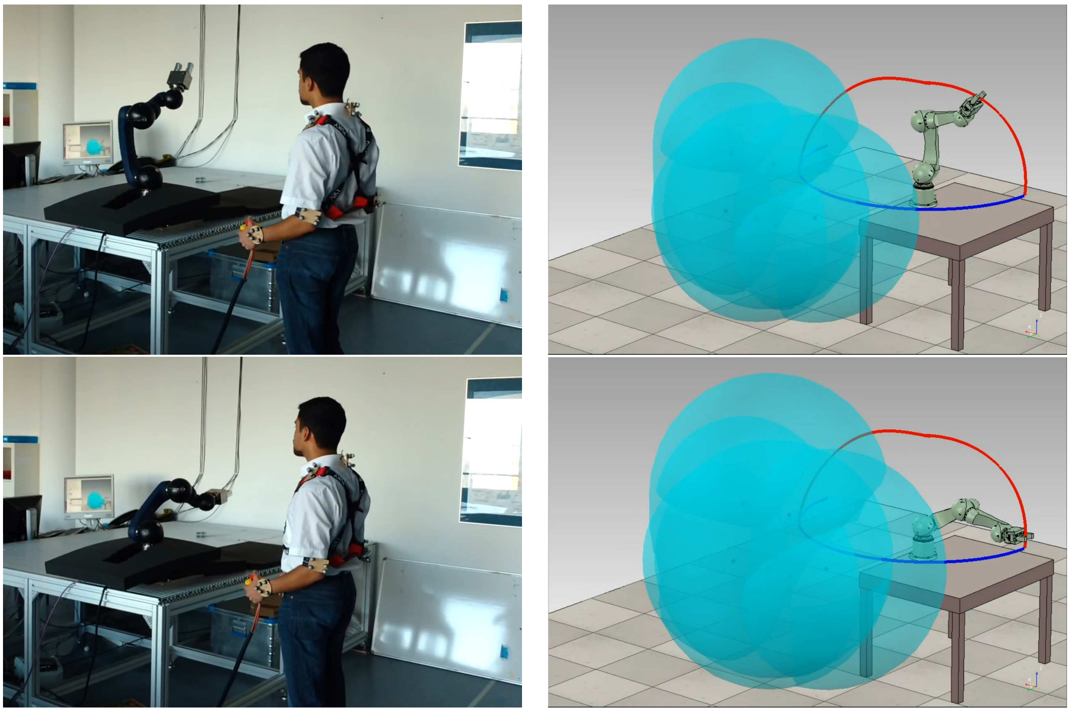


Figure 2.10: (continued) Human presence causes robot to replan; speed profile shown in Fig. 2.7. Left: freeze-frames from recording, right: freeze-frames from simultaneous visualisation; HRO in cyan.

Chapter 3

Robot Occupancy

Future movement of the robot and the human must be predicted to implement the approach presented in the previous chapter. The robot continually verifies safety over the immediately subsequent portion of trajectory, hence it needs to know the regions in space occupied by both human and itself during this time, and check for intersection. We call these regions the human reachable occupancy (HRO) *and the* robot reachable occupancy (RRO)*, defined in Sec. 2.1. This section focusses on how to calculate the latter.*

The robot may have several degrees of freedom; it is therefore important that these methods for predicting the RRO are scalable. We present two novel, scalable methods, which we compare to the state of the art.

The robot reachable occupancy (RRO) is the set of space which the robot occupies during a certain time interval. In the framework of the continuous verifier presented in the previous section in Alg. 1, the time interval is the time from the start of the short-term plan until the robot reaches a safe state. We must compute this *overapproximatively*, and *tightly*. That is, the RRO must not exclude space that the robot could occupy during the time interval (overapproximative), but should also – as far as possible – not include space that is unreachable by the robot (tight).

Methods have been proposed to compute the occupancy of a kinematic chain based on sampling, e.g. [59], however, these are non-overapproximative, and do not scale well with dimension. To the author's best knowledge, the only conservative method that computes an overapproximative RRO is that of Täubig et al. [60], which has exponential complexity in the number of degrees of freedom (DOFs). This is recapitulated in Sec. 3.2. We also present two novel methods for computing the RRO, that of *interval matrices* and *occupancy capsules*, which have cubic and linear complexity in the number of degree of freedoms (DOFs) respectively. The three methods presented below return a conservative and tight occupancy in space, given a description of the robot kinematics (e.g. Denavit-Hartenberg (D-H) parameters), the link geometry and a set of joint positions on the path of the robot. These methods can be extended to modular robots following a method similar to [61].

The state of the art method uses *sphere-swept volumes* and produces a tight but overapproximative volume defined by a set of points. The number of points increases exponentially with the number of degrees of freedom of the kinematic chain. The method of *interval matrices* essentially calculates the set of all possible transformation matrices for the forward kinematics in an overapproximative way, and then applies them to the robot geometry to obtain overapproximative sets of the robot's occupancy. Both these approaches can implicitly account for uncertainty in the joint position. The method of *occupancy capsules* calculates the RRO over a section of the robot's trajectory and then

We first introduce the concept of interval arithmetic (IA), along with definitions of interval vectors, interval matrices and interval tensors.

3.1 Set-based Arithmetic

In this thesis, sets are used extensively. We often require arithmetic operations and functions to be defined on sets, and for all the purposes of this thesis, we require the set-based operations or functions to be overapproximative, i.e. to return a conservative approximation. This means, the set returned by the set-based function or operator should enclose the exact set of values that the function or operation would yield, given any values of the variables or operands in their respective sets. Formally, given a multivariate function or N-ary operator $*(a_1, ..., a_N)$, its set-based equivalent $\circledast$ is defined on intervals $[a_{1,\min}, a_{1,\max}], ..., [a_{N,\min}, a_{N,\max}]$ as follows:

$$\begin{aligned}\circledast([a_{1,\min}, a_{1,\max}], ..., &[a_{N,\min}, a_{N,\max}]) \\ &\supseteq \{*(a_1, ...a_N) \mid a_1 \in [a_{1,\min}, a_{1,\max}], ..., a_N \in [a_{N,\min}, a_{N,\max}]\}\end{aligned}$$

The most common set-based operator used in this thesis is set-based addition, also called the *Minkowski-sum* ($\oplus$).

Interval arithmetic (IA) is a technique for performing set-based calculations with real variables lying within an interval [62, 63, 64]. Many (well-conditioned) arithmetic operations (e.g. addition, subtraction, multiplication, division, power operations) and multivariate functions (e.g. trigonometric, hyperbolic, and exponential functions) have IA implementations. A MATLAB toolbox which is used extensively for the results in this thesis is the INTLAB [65], although the CORA toolbox [66] now also provides equivalent functionality for most relevant applications. In IA, the sets used in the set-based operations are intervals, or interval vectors, matrices or tensors, which we define below. We denote the supremum and infimum of the interval $[a]$ with inf($[a]$) and sup($[a]$), respectively.

Definition 3.1 (Interval tensor, matrix and vector): An N^{th} order interval tensor $[A] \subset \mathbb{R}^{n_1 \times \ldots \times n_N}$ is a tensor, where each element is an interval. An *interval matrix* is a 2nd order tensor and an *interval vector* is a 1st order tensor. □

In the literature, an interval vector is often also called a Cartesian product of intervals, or an axis-aligned bounding box (AABB)—mostly in the context of collision-detection, e.g. [67]—or simply a "box". In this thesis, for consistency, we use the term "interval vector". We also present the definition of the *absolute value* of an interval, or of an interval tensor, used in this thesis.

Definition 3.2 (Absolute value of an interval): The absolute value of interval $[a]$ is denoted $|[a]|$ and is defined by:

$$|[a]| = \max(|\text{inf}([a])|, |\text{sup}([a])|)$$

The absolute value of an N^{th} order interval tensor $[A]$ is defined for each element $[a_{i_1 \ldots i_N}]$ as:

$$|[a_{i_1 \ldots i_N}]| = \max(|\text{inf}([a_{i_1 \ldots i_N}])|, |\text{sup}([a_{i_1 \ldots i_N}])|)$$ □

We next introduce three results which will be useful in the following sections.

Observation 3.1: For all $b \in [a]$, it holds that $|b| \leq |[a]|$. □

Lemma 3.1: let $[T]$ be an interval matrix and $\mathcal{S}$ be a set enclosed in the interval vector $[\boldsymbol{a}]$, i.e. $\mathcal{S} \subseteq [\boldsymbol{a}]$. Then an upper bound of $\{\left|\boldsymbol{s}^\top [T] \boldsymbol{s}\right| | \boldsymbol{s} \in \mathcal{S}\}$ is $\left|[\boldsymbol{a}]^\top\right| |[T]| |[\boldsymbol{a}]|$. □

Proof: The absolute value of the quadratic form, $|\boldsymbol{s}^\top [T] \boldsymbol{s}|$ can also be expressed as the sum:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} s_i s_j T_{ij}$$

We observe that:

$$\left|\sum_{i=1}^{n}\sum_{j=1}^{n} s_i s_j T_{ij}\right| \leq \sum_{i=1}^{n}\sum_{j=1}^{n} |s_i s_j T_{ij}| \leq \sum_{i=1}^{n}\sum_{j=1}^{n} |s_i||s_j||T_{ij}|$$

where n is the length of $\boldsymbol{a}$. Then since $\boldsymbol{s} \in \mathcal{S} \subseteq [\boldsymbol{a}]$, the i^{th} element of $\boldsymbol{s}$ is contained in the i^{th} element of $[\boldsymbol{a}]$: $s_i \in [a_i]$. Hence by Obs. 3.1, $|s_i| \leq |[a_i]|$. So:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} |s_i||s_j||T_{ij}| \leq \sum_{i=1}^{n}\sum_{j=1}^{n} |[a_i]||[a_j]||T_{ij}| = |[\boldsymbol{a}]^\top||[T]||[\boldsymbol{a}]| \qquad \square$$

Observation 3.2: When $\mathcal{S} = [\boldsymbol{a}]$ in the above lemma, we also observe the weaker result that $|\boldsymbol{a}^\top||[T]||\boldsymbol{a}|$ is an upper bound of $|\boldsymbol{a}^\top[T]\boldsymbol{a}|$. □

3.2 Method of Sphere-Swept Volumes

In [60] the authors propose a method to conservatively bound the occupancy of a link on a kinematic chain under the effect of a reachable set of joint positions represented as an interval vector (Def. 3.1). Rigid links on the kinematic chain are first enclosed in a sphere-swept volume (SSV) in their own coordinate system. The SSV is successively transformed and augmented to account for the joint position interval of each degree of freedom until finally an SSV is obtained in the base coordinate system, which encloses the occupancy of the link accounting for all joint positions within the reachable set. We recapitulate their method below and show that the complexity is exponential in the number of degrees of freedom. We first define the Convex Hull, the Closed Euclidean ball, and hence the sphere-swept volume:

Definition 3.3 (Convex Hull): For points $\boldsymbol{p}_1, \boldsymbol{p}_2, ...\boldsymbol{p}_l$, the convex hull $\texttt{CH}(\boldsymbol{p}_1, \boldsymbol{p}_2, ...\boldsymbol{p}_l)$ is defined:

$$\texttt{CH}(\mathcal{P}) = \left\{ \sum_{i=1}^{l} \alpha_i \boldsymbol{p}_i \;\middle|\; \forall i : \alpha_i \geq 0, \sum_{i=1}^{l} \alpha_i = 1 \right\}$$

For sets $\mathcal{P}_1, \mathcal{P}_2, ...\mathcal{P}_l$, the convex hull $\texttt{CH}(\mathcal{P}_1, \mathcal{P}_2, ...\mathcal{P}_l)$ is defined as the intersection of all convex supersets of $\bigcup_{i=1}^{i=l} \mathcal{P}_i$. □

Definition 3.4 (Closed Euclidean ball (ball)): The closed Euclidean ball centred at $\boldsymbol{p} \in \mathbb{R}^3$ with radius r (henceforth just "ball", for brevity) is defined:

$$\mathcal{B}(\boldsymbol{p}\,; r) = \{\boldsymbol{x} \mid \|\boldsymbol{x} - \boldsymbol{p}\|_2 \leq r\} \qquad \square$$

Definition 3.5 (Sphere-swept volume and capsule): A *sphere-swept volume (SSV)* (see Fig. 3.1) defined by a set of points $\mathcal{P} = \{P_1, P_2, ...P_l\}$ and a radius r is a set:

$$\mathcal{V}_r^{\mathcal{P}} = \texttt{CH}(\mathcal{P}) \oplus \mathcal{B}(\boldsymbol{0}\,; r)$$

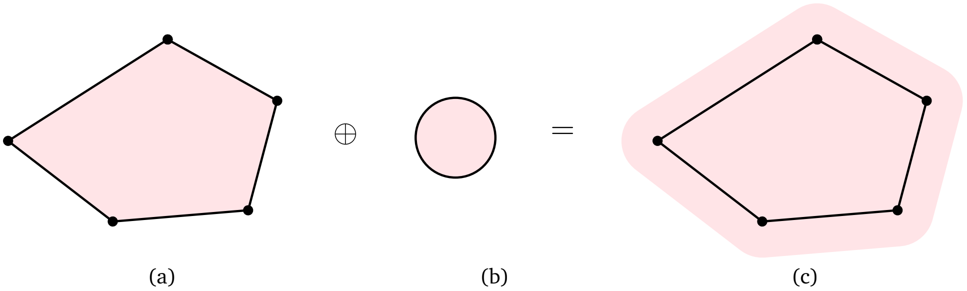


Figure 3.1: A sphere-swept volume (SSV) is the Minkowski sum of the convex hull of a set of defining points, and a closed ball.

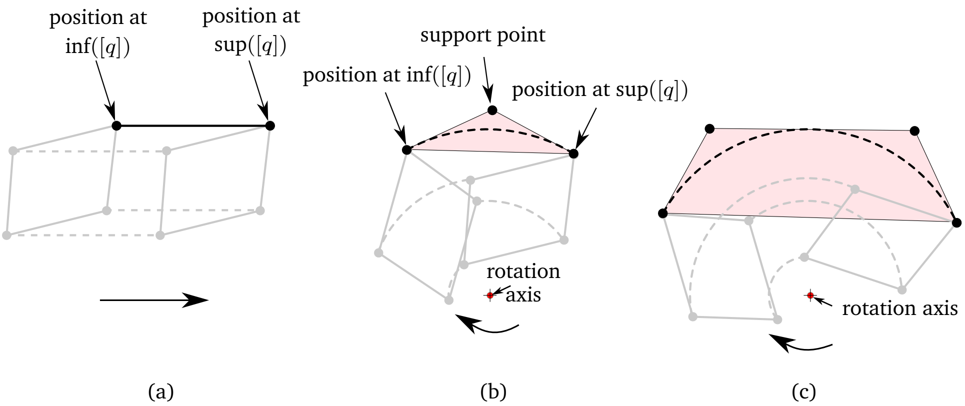


Figure 3.2: (a) a point on the SSV under the action of a prismatic joint is enclosed in a line segment; (b) the occupancy of a point on the SSV under the action of a revolute joint (dashed line) is enclosed by 3 points in the new SSV (solid line) or (c) if the angle subtended is larger, we use more points. The previous SSV had 4 points, the new one would have (a) 8 points, (b) 12 points or (c) 16 points (not all shown, for clarity of illustration).

A *capsule* is a special case of an SSV where $|\mathcal{P}| = 2$, i.e. $\mathtt{CH}(\mathcal{P})$ is a closed line segment. □

Observation 3.3 (reachable occupancy (RO) of SSV from RO of defining points): Given a rigid SSV $\mathcal{V}_r^{\mathcal{P}}$ with defining points $\mathcal{P}{=}\{\boldsymbol{p}_1, ..., \boldsymbol{p}_N\}$, let $\mathcal{G}_{\boldsymbol{p}_1}, ..., \mathcal{G}_{\boldsymbol{p}_N}$ be the ROs of each of its defining points. Then a RO of $\mathcal{V}_r^{\mathcal{P}}$ is:

$$\begin{aligned} & \mathtt{CH}(\mathcal{G}_{\boldsymbol{p}_1}, ..., \mathcal{G}_{\boldsymbol{p}_N}) \oplus \mathcal{B}(\mathbf{0}\,;r) \\ = \quad & \mathtt{CH}(\mathcal{G}_{\boldsymbol{p}_1} \oplus \mathcal{B}(\mathbf{0}\,;r), ..., \mathcal{G}_{\boldsymbol{p}_N} \oplus \mathcal{B}(\mathbf{0}\,;r)) \end{aligned}$$

This follows directly from the SSV's convexity.

The method of sphere-swept volumes is outlined in Alg. 6. We wish to find the set that a rigid link of a kinematic chain could possibly occupy, in world coordinates, for a set of joint positions contained within a set. In this case, this set is defined as, or enclosed in, an interval vector, $[\boldsymbol{q}]$. The link itself is enclosed in an SSV defined in its own (i^{th}) coordinate system: ${}^{i}\mathcal{V}_{r}^{\mathcal{P}_i}$. The left subscript indicates the coordinate frame of reference; we call the coordinate system of the i^{th} link, the i^{th} coordinate system; the 0^{th} coordinate system is understood to mean the world frame.

First, the position in the $i-1^{\text{th}}$ coordinate system is found. See Fig. 3.2(a), and consider a defining point of the SSV. If the joint is prismatic, the motion of this defining point is along a straight line in the direction of the joint axis, and all the possible positions of this defining point for any $q_i \in [q_i]$ lie on the line segment between the position of the point at inf$([q_i])$ and the position at sup$([q_i])$. This line segment can also be described as the convex hull of the position of the point at the two extrema.

Hence, for any $q_i \in [q_i]$, each defining point of the SSV in the $i-1^{\text{th}}$ coordinate system lies in the convex hull of the defining points of the SSV at $q_i =$ inf$([q_i])$ and $q_i =$ sup$([q_i])$. The defining points of the new SSV at this iteration are the union of the points of the old SSV, translated into the $i-1^{\text{th}}$ coordinate system both at $q_i =$ inf$([q_i])$, and at $q_i =$ sup$([q_i])$. The radius is the same as the old SSV. The algorithm for generating the defining points in the $i-1^{\text{th}}$ coordinate system from those in the i^{th} coordinate system is `applyPrismaticTransformation` (see Alg. 7).

If the i^{th} joint is revolute, we follow a slightly different procedure. Again, we construct for each defining point in the old SSV the set of points in the $i-1^{\text{th}}$ coordinate system whose convex hull contains the defining point for all $q_i \in [q_i]$. However, the locus of a point under the motion of a revolute joint is an arc and not a line, and hence the set of positions of each defining point cannot be defined exactly by the convex hull of of a set of points, but instead we create a tight overapproximation using three or more points, as shown in Fig. 3.2. The more points, the tighter the overapproximation, but the more complex the volume. The function `applyRevoluteTransformation` (Alg. 8) returns the defining points of the SSV in the $i-1^{\text{th}}$ coordinate system from the defining points in the i^{th} coordinate system.

The defining points of the SSV are computed iteratively for each joint down to the base, and the algorithm terminates returning the SSV for the link in the world frame. This is done for all links of the robot. Since at each iteration of the algorithm, all the points of the SSV are matrix multiplied by a transformation matrix n_{points} times (for revolute joints) or 2 times (for prismatic joints), and the number of points increases by a factor of n_{points} for revolute or 2 for prismatic joints, the number of points increases exponentially with the number of joints.

See Alg. 7. The function `DHXform` yields the transformation ${}_{i-1}^{\;\;i}T$ from one link coordinate frame to the next in the D-H convention of describing kinematic chains [68]:

$$\texttt{DHXform}(a, d, \alpha, \theta) := \begin{bmatrix} \cos(\theta) & -\sin(\theta)\cos(\alpha) & \sin(\theta)\sin(\alpha) & a\cos(\theta) \\ \sin(\theta) & \cos(\theta)\cos(\alpha) & -\cos(\theta)\sin(\alpha) & a\sin(\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

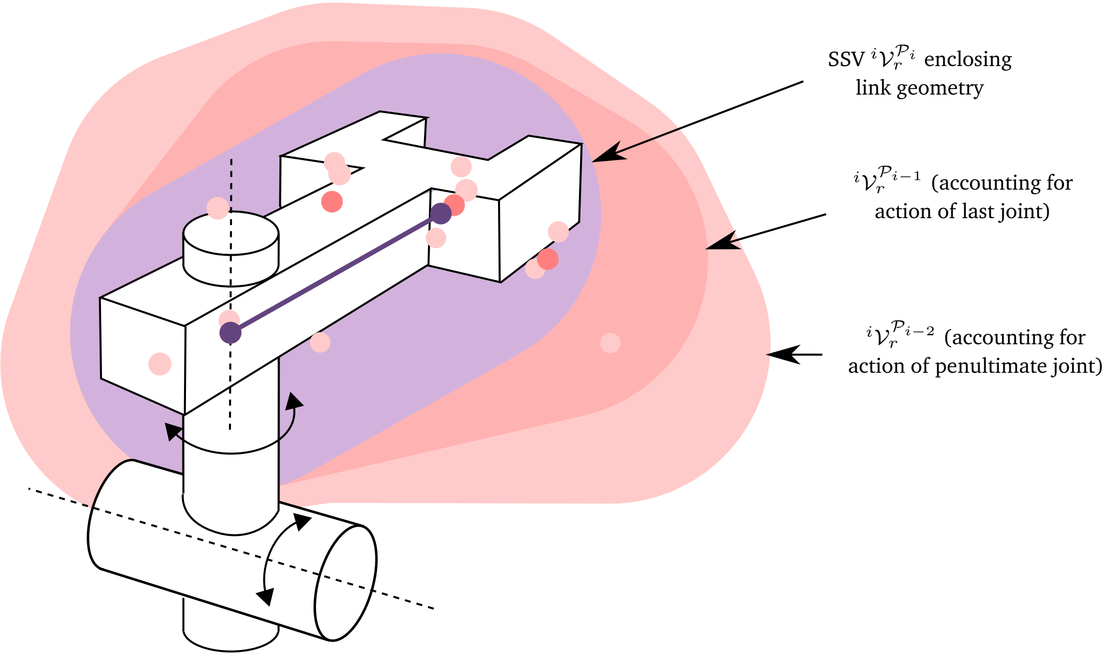

SSV ${}^{i}\mathcal{V}_r^{\mathcal{P}_i}$ enclosing link geometry
${}^{i}\mathcal{V}_r^{\mathcal{P}_{i-1}}$ (accounting for action of last joint)
${}^{i}\mathcal{V}_r^{\mathcal{P}_{i-2}}$ (accounting for action of penultimate joint)

Figure 3.3: Two links of a robot and the SSV of the outermost link generated stepwise by generating a new SSV at each degree of freedom, starting from the body geometry in its own coordinate system. The link is enclosed in an SSV, which is the purple capsule. The volume reachable under all joint positions of the last joint (with vertical axis) is found from this capsule, overapproximatively, shown in pink. From this, the volume reachable under all positions of the second-to-last joint is found, in lighter pink. This is then continued iteratively until the action of all joints of the robot has been considered, we then have the reachable occupancy of the link in the world coordinate system. The defining points are also shown in the respective colours.

Algorithm 6 Enclosing link of kinematic chain [60]

```
Input: DH parameters $\{a_1, ..., a_m\}$, $\{d_1, ..., d_m\}$, $\{\alpha_1, ..., \alpha_m\}$, $\{\theta_1, ..., \theta_m\}$, interval vector of
    joint positions $[\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max}]$ and SSV $\mathcal{V}_r^{\mathcal{P}_m}$ defined by set of points $\mathcal{P}_m$ in link coordinates
    and radius $r$
Output: SSV $\mathcal{V}_r^{\mathcal{P}_0}$ in world coordinates
for $i \in \{m, m-1, ..., 1\}$ do
    if joint $i$ is revolute then
        $\mathcal{P}_{i-1}$ = applyRevoluteTransformation$(a_i, d_i, \alpha_i, [q_i], \mathcal{P}_i)$
    else
        $\mathcal{P}_{i-1}$ = applyPrismaticTransformation$(a_i, [q_i], \alpha, \theta, \mathcal{P}_i)$
return $\mathcal{V}_r^{\mathcal{P}_0}$
```

Algorithm 7 `applyPrismaticTransformation`$(a, [q], \alpha, \theta, \mathcal{P})$ [60]

```
Input: scalars $a$, $\theta$, $\alpha$, interval $[q]$ and set of points $\mathcal{P}_i$ in prismatic joint coordinate system
Output: new set of points $\mathcal{P}_{i-1}$ in previous coordinate system
$\mathcal{P}_1 = \texttt{DHXform}(a, \inf([q]), \alpha, \theta) \cdot \mathcal{P}_i$
$\mathcal{P}_2 = \texttt{DHXform}(a, \sup([q]), \alpha, \theta) \cdot \mathcal{P}_i$
return $\mathcal{P}_{i-1} \leftarrow \mathcal{P}_1 \cup \mathcal{P}_2$
```

Algorithm 8 `applyRevoluteTransformation`$(a, d, \alpha, [q], \mathcal{P})$ [60]

```
Input: scalars $a$, $d$, $\alpha$, interval $[q]$ and set of points $\mathcal{P}_i$ in rotary joint coordinates system
Output: new set of points $\mathcal{P}_{i-1}$ in previous coordinate system
$n_{points} = \texttt{getRequiredPrecisionFactor}(\sup([q]) - \inf([q]))$    // how many support points to use in the overapproximation?
$q_{int} = \frac{q_{\max} - q_{\min}}{2n_{points} - 4}$
$T = \texttt{diag}(\frac{1}{cos(q_{int})}, \frac{1}{cos(q_{int})}, 1, 1)$    // diagonal matrix
$\mathcal{P}_1 = \texttt{DHXform}(a, d, \alpha, \inf([q])) \cdot \mathcal{P}_i$
$\mathcal{P}_{n_{points}} = \texttt{DHXform}(a, d, \alpha, \sup([q])) \cdot \mathcal{P}_i$
for $j \in \{2, ..., n_{points} - 1\}$ do
    $\mathcal{P}_j = T \cdot \texttt{DHXform}(a, d, \alpha, \inf([q]) + (2i - 3)q_{int}, q_{int}) \cdot \mathcal{P}_i$
return $\mathcal{P}_{i-1} \leftarrow \bigcup_{j \in \{1,...,n_{points}\}} \mathcal{P}_j$
```

3.3 Method of Interval Matrices

We present a method in [178] to obtain the RRO using IA and *Zonotopes*:

Definition 3.6 (Zonotope): A zonotope $\mathcal{Z} \subset \mathbb{R}^n$ with centre $\boldsymbol{c} \in \mathbb{R}^n$ and generators $\mathcal{G} = \{\boldsymbol{g}^{(1)}, ..., \boldsymbol{g}^{(p)}\}, \boldsymbol{g}^{(i)} \in \mathbb{R}^n$ is defined:

$$\mathcal{Z} = \left\{ \boldsymbol{c} + \sum_{i=1}^{p} \beta_i \boldsymbol{g}^{(i)} \;\middle|\; \forall i \in \{1, ..., p\} : \beta_i \in [-1, 1] \right\} \tag{3.2}$$

To increase legibility, the superscripts and subscripts of zonotopes are common to their centres and generators, e.g. $\mathcal{Z}_a^b$ has centre $\boldsymbol{c}_a^b$ and generators $\mathcal{G}_a^b = \{\boldsymbol{g}_a^{b,(1)}, ..., \boldsymbol{g}_a^{b,(p)}\}$. □

Zonotopes have numerous computational advantages as a set representation. In particular, they are concise, multiplication and addition operations are very scalable with dimension, and there exists an efficient way to overapproximate the product of an interval matrix and a zonotope [69]. The latter property is specifically useful in our case, since we linearise and overapproximate the transformations of the forward kinematic mapping over the entire joint space reachable set with interval matrices, and apply these sets of transformations to the links of the robot, which are overapproximated by zonotopes. The resulting RRO is a union of zonotopes in Cartesian space. We explain the method below; the method is illustrated in Fig. 3.4 for a single link.

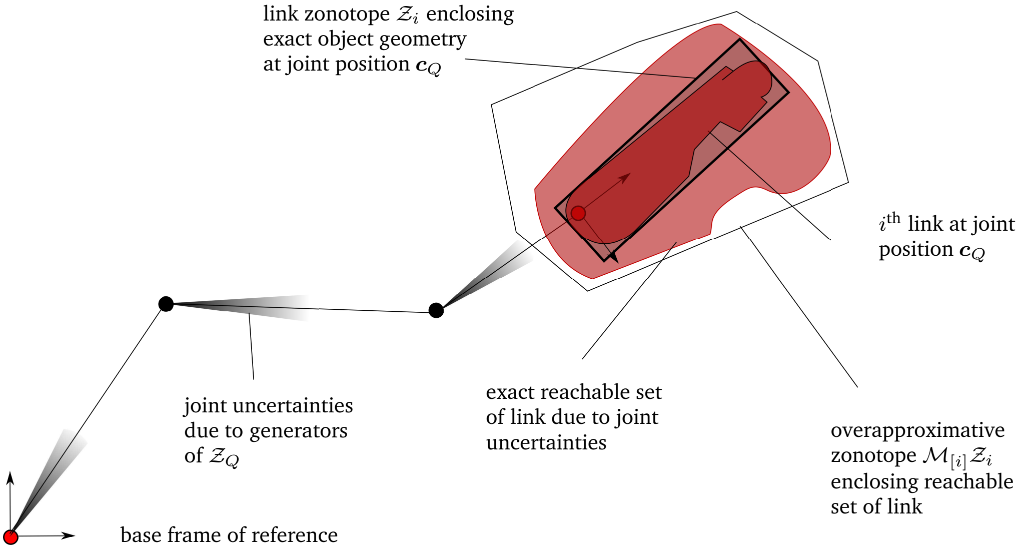

link zonotope $\mathcal{Z}_i$ enclosing exact object geometry at joint position $\boldsymbol{c}_Q$
i^{th} link at joint position $\boldsymbol{c}_Q$
joint uncertainties due to generators of $\mathcal{Z}_Q$
exact reachable set of link due to joint uncertainties
overapproximative zonotope $\mathcal{M}_{[i]}\mathcal{Z}_i$ enclosing reachable set of link
base frame of reference

Figure 3.4: A set of joint positions lead to uncertainty in the position of the link in the world coordinate system; the set of positions can be overapproximated by a zonotope.

While zonotopes map exactly in a linear mapping, the mapping from joint space to an occupancy in Cartesian space is highly nonlinear. The forward kinematics function $F_{[1,\ldots,m]}(\boldsymbol{q})$ is a nonlinear function of the joint positions, which returns transformations for each rigid body of the robot. Here, $F_{[i]}(\boldsymbol{q}) \in SE(3)$ [1] refers to the forward kinematics of the i^{th} , i.e. the transformation from the coordinate system of the i^{th} link to the world frame. When applied to the occupancy set of the i^{th} link represented in its own coordinate system, it yields the occupancy in the world frame. It can be represented as a homogeneous transformation matrix:

$$F_{[i]}(\boldsymbol{q}) = \begin{bmatrix} \multicolumn{3}{c}{R_{[i]}(\boldsymbol{q})} & T_{[i]}(\boldsymbol{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where $R(\boldsymbol{q}) \in SO(3)$ is a 3×3 matrix representing the rotation[2] and $T(\boldsymbol{q}) \in \mathbb{R}^3$ is a translation between the reference frames.

Again, we refer to the coordinate system of the i^{th} link as the i^{th} coordinate system; the 0^{th} coordinate system is the world frame. The exact occupancy of the i^{th} link in its own coordinate system, ${}^{i}\mathcal{O}_i$, is overapproximated by a zonotope ${}^{i}\mathcal{Z}_i \subset \mathbb{R}^3$ (the left superscript denotes the reference frame). We can express this occupancy in homogeneous coordinates as the zonotope ${}^{i}\mathcal{Z}_i' \subset \mathbb{R}^4$, where $\boldsymbol{c}_i' := [\boldsymbol{c}_i^\top, 1]^\top$ and $\mathcal{G}_i' := \{[\boldsymbol{g}_i^{(1)\top}, 0]^\top, \ldots [\boldsymbol{g}_i^{(p)\top\top}, 0]^\top\}$. In this case, the zonotope enclosing the i^{th} link in the world frame is calculate thus: ${}^{0}\mathcal{Z}_i' = F(\boldsymbol{q})_{[i]}\ {}^{i}\mathcal{Z}_i'$ and the RRO at the

[1] $SE(3)$ is the special Euclidean group of translations and rotations in $\mathbb{R}^3$
[2] $SO(3)$ is the special orthogonal group of rotations in $\mathbb{R}^3$

joint position $\boldsymbol{q}$ is:

$$\bigcup_{i=1}^{m} F(\boldsymbol{q})^{i}_{[i]}\mathcal{Z}'_i$$

Instead of the single joint position $\boldsymbol{q}$, we want to account for a set of joint positions along part of the robot's trajectory, represented as (or enclosed in) a zonotope $\mathcal{Z}_Q \subset \mathcal{Q}$. To do this, we create a matrix $\mathcal{M}_{[i]} \supseteq \{F(\boldsymbol{q})_{[i]} \mid \boldsymbol{q} \in \mathcal{Z}_Q\}$. Henceforth, we drop the dependency on the i^{th} link for clarity, i.e., $F(\boldsymbol{q})$ refers to the transformation from own frame to world frame, when it is clear which link is under consideration. We first linearise $F(\boldsymbol{q})$ using a Taylor expansion about the joint position $\boldsymbol{c}_Q$, and express higher order terms as a Lagrange remainder. This will allow us to apply the set of transformations to the object geometry.

$$\begin{aligned} F(\boldsymbol{q}) = F(\boldsymbol{c}_Q) + &\underbrace{\left.\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{q}}\right|_{\boldsymbol{c}_Q}(\boldsymbol{c}_Q - \boldsymbol{q})}_{A(\boldsymbol{c}_Q,\boldsymbol{q})} + \\ &\underbrace{(\boldsymbol{c}_Q - \boldsymbol{q})^{\top}\left.\frac{\mathrm{d}^2F}{\mathrm{d}\boldsymbol{q}^2}\right|_{\boldsymbol{q}^*}(\boldsymbol{c}_Q - \boldsymbol{q}),}_{L(\boldsymbol{c}_Q,\boldsymbol{q})} \end{aligned} \tag{3.3}$$

Here, $\left.\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{q}}\right|_{\boldsymbol{c}_Q} \in \mathbb{R}^{4\times4\times m}$ and $\left.\frac{\mathrm{d}^2F}{\mathrm{d}\boldsymbol{q}^2}\right|_{\boldsymbol{q}^*} \in \mathbb{R}^{4\times4\times m\times m}$ are the tensors of derivatives and second derivatives of the change of each element of the forward kinematics transformation matrix with respect to joint angles (not to be confused with the Robot Jacobian). They are evaluated at $\boldsymbol{c}_Q$ and $\boldsymbol{q}^*$ respectively, where $\boldsymbol{q}^* = \alpha\boldsymbol{c}_Q + (1-\alpha)\boldsymbol{q}$, $\alpha \in [0,1]$ for some α; this follows from the mean value theorem [70].

Evaluating (3.3) over $\mathcal{Z}_Q$ using interval arithmetic would be extremely time-consuming. For example, when considering the m^{th} joint, $\left.\frac{\mathrm{d}^2F_{[m]}}{\mathrm{d}\boldsymbol{q}^2}\right|_{\boldsymbol{q}^*}$ has up to $12m^2$ nonzero values[3] which are nonlinear functions of $\boldsymbol{q}$ to be evaluated in interval arithmetic. This takes a considerable amount of time, hence we bound $A(\boldsymbol{c}_Q,\boldsymbol{q})$ and $L(\boldsymbol{c}_Q,\boldsymbol{q})$ more efficiently.

The term A can be easily bounded over the joint space zonotope. Since $\boldsymbol{q} \in \mathcal{Z}_Q$, $\boldsymbol{q} - \boldsymbol{c}_Q = \sum_{l=1}^{p}\beta_l\boldsymbol{g}_Q^{(l)}$, so:

$$\left.\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{q}}\right|_{\boldsymbol{c}_Q}(\boldsymbol{c}_Q - \boldsymbol{q}) = \sum_{l=1}^{p}\beta_l\left.\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{q}}\right|_{\boldsymbol{c}_Q}\boldsymbol{g}_Q^{(l)}, \text{for some} -1 \le \beta_1, ..., \beta_p \le 1$$

Since,

$$\forall l \in \{1,...,p\}, \qquad -\left|\left.\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{q}}\right|_{\boldsymbol{c}_Q}\boldsymbol{g}_Q^{(l)}\right| \quad \le \quad \beta_l\left.\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{q}}\right|_{\boldsymbol{c}_Q}\boldsymbol{g}_Q^{(l)} \quad \le \quad \left|\left.\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{q}}\right|_{\boldsymbol{c}_Q}\boldsymbol{g}_Q^{(l)}\right|,$$

[3] 9 values in $R_{[i]}$ and 3 values in $T_{[i]}$, all of which may have non-zero derivatives with respect to all m joints, twice.

we can bound the term A over the zonotope:

$$\{A(\boldsymbol{c}_Q,\boldsymbol{q})|\boldsymbol{q}\in\mathcal{Z}_Q\}\subseteq[-B,B],\qquad B=\sum_{l=1}^{p}\left|\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{q}}\bigg|_{\boldsymbol{c}_Q}\boldsymbol{g}^{(l)}\right| \tag{3.4}$$

In the Lagrange remainder L, it is not known *a priori* what the value of α is, hence $\boldsymbol{q}^*$ is also *a priori* unknown; furthermore, $\boldsymbol{q}^*$ varies for different $\boldsymbol{q}$. Since the term L is typically small for small $\mathcal{Z}_Q$, we evaluate *offline* the set of values of $\frac{\mathrm{d}^2F}{\mathrm{d}\boldsymbol{q}^2}$ over the entire joint space using interval arithmetic. We have two options here: 1) partition the state space and evaluate $\frac{\mathrm{d}^2F}{\mathrm{d}\boldsymbol{q}^2}$ over each cell of the partition, storing the values in a lookup table, or 2) evaluate $\frac{\mathrm{d}^2F}{\mathrm{d}\boldsymbol{q}^2}$ over the entire state space. In either case, we obtain an interval tensor $[L]\supseteq\{\frac{\mathrm{d}^2F(\boldsymbol{q})}{\mathrm{d}\boldsymbol{q}^2}\mid\boldsymbol{q}\in\mathcal{Z}_Q\}$. We now require to be found:

$$\{(\boldsymbol{q}-\boldsymbol{c}_Q)^\top[L](\boldsymbol{q}-\boldsymbol{c}_Q)\mid\boldsymbol{q}\in\mathcal{Z}_Q\}=\underbrace{\{\boldsymbol{q}^\top[L]\boldsymbol{q},\mid\boldsymbol{q}\in\mathcal{Z}_Q-\boldsymbol{c}_Q\}}_{\text{occupancy due to Lagrange remainder}}. \tag{3.5}$$

Now, note that an overapproximative enclosure of the set $\mathcal{Z}_Q-\boldsymbol{c}_Q$ is the interval vector $[-\boldsymbol{d},\boldsymbol{d}]$ where $\boldsymbol{d}=\sum_{l=1}^{p}|\boldsymbol{g}^{(l)}|$. Let $f_{ij}(\boldsymbol{q})$ be the element in the i^{th} row and j^{th} column of $F(\boldsymbol{q})$; in this case, $\frac{\mathrm{d}^2f_{ij}(\boldsymbol{q})}{\mathrm{d}\boldsymbol{q}^2}$ is a matrix (2-dimensional); we calculate $D_{ij}=\left|\frac{\mathrm{d}^2f_{ij}(\boldsymbol{q})}{\mathrm{d}\boldsymbol{q}^2}\right|$ offline, for all $i,j\in\{1,...,m\}$. By Lem. 3.1, we can now calculate an overapproximation to the element in the i^{th} row and j^{th} column of the occupancy due to the Lagrange remainder in (3.5) as:

$$[-C_{ij},C_{ij}]=[-\boldsymbol{d}^\top D_{ij}\boldsymbol{d},\boldsymbol{d}^\top D_{ij}\boldsymbol{d}]. \tag{3.6}$$

In such a way, i.e. elementwise, we populate the interval matrix $[-C,C]$, which is an overapproximation to the occupancy due to the Lagrange remainder. We therefore overapproximate the set of all matrices for the forward kinematic transformation for the link in question as:

$$\mathcal{M}=F(\boldsymbol{c}_Q)\oplus[-B,B]\oplus[-C,C]\supseteq\{F(\boldsymbol{q})\mid\boldsymbol{q}\in\mathcal{Z}_Q\}. \tag{3.7}$$

Finally, we reintroduce the subscript i to reference the i^{th} link. From (3.7) we see that $\mathcal{M}_{[i]}$ is the sum of a (non-interval) matrix $F_{[i]}(\boldsymbol{c}_Q)$ and a symmetric interval matrix (i.e. where the infimum is the negative value of the supremum) $[-(B_{[i]}+C_{[i]}),B_{[i]}+C_{[i]}]$. In [69, Thm. 3.3], it is shown that we can overapproximate the occupancy of the i^{th} link, $\mathcal{M}_{[i]}\mathcal{Z}_i'$, by the zonotope with centre $F_{[i]}(\boldsymbol{c}_Q)\boldsymbol{c}_i'$ and generators: $\{F_{[i]}(\boldsymbol{c}_Q)\boldsymbol{g}_i^{(1)'},...,F_{[i]}(\boldsymbol{c}_Q)\boldsymbol{g}_i^{(p)'},\boldsymbol{h}^{(1)},...,\boldsymbol{h}^{(4)}\}$, where each of $\boldsymbol{h}^{(j)}$ is defined such that the j^{th} element is the j^{th} element of the vector:

$$\boldsymbol{d}=(B_{[i]}+C_{[i]})\Big(|\boldsymbol{c}_i'|+\sum_{j=1}^{p}|\boldsymbol{g}_i^{(j)}|\Big), \tag{3.8}$$

and the other elements are 0.

An advantage of this approach is that the reachable sets do not have to be interval vectors but can be obtained as zonotopes, for example using the methods described in [69]. However, to perform collision-checking with the HRO, either the vertices or the facets of each zonotope of the RRO must be enumerated. Where p is the number of generators, vertex enumeration of a zonotope can be solved using reverse search algorithms with time complexity $O(p^2)$ [71, Thm 3.2]; facet enumeration is of complexity $O(p \cdot \binom{p}{n-1})$ as shown in [69].

The method has the additional disadvantage that the overapproximation of the Lagrange remainder term becomes very large if $\mathcal{Z}_Q$ is large. A comparison is presented in Sec. 3.6.

3.4 Method of Occupancy Capsules

Drawbacks of the previous two methods are the complex occupancies which are generated. To check for collisions, the zonotope in the previous method would be converted into a polytope. Both polytopes and zonotopes can be collision-checked fairly efficiently using a modified GJK algorithm [72], which has the drawback of being iterative and hence nondeterministic in computation time. Much quicker, however, is collision checking of capsules.

Furthermore, both of the previous algorithms have more than linear complexity in the number of the robot's degrees of freedom, as shown later in Sec. 3.5. Below we present a third method for calculating the RRO over a section of the robot's trajectory, which has linear complexity in the number of degrees of freedom of the robot. Since the prediction is a union of capsules, very fast collision-checking with deterministic computation times can be used, as opposed to collision checking for SSVs.

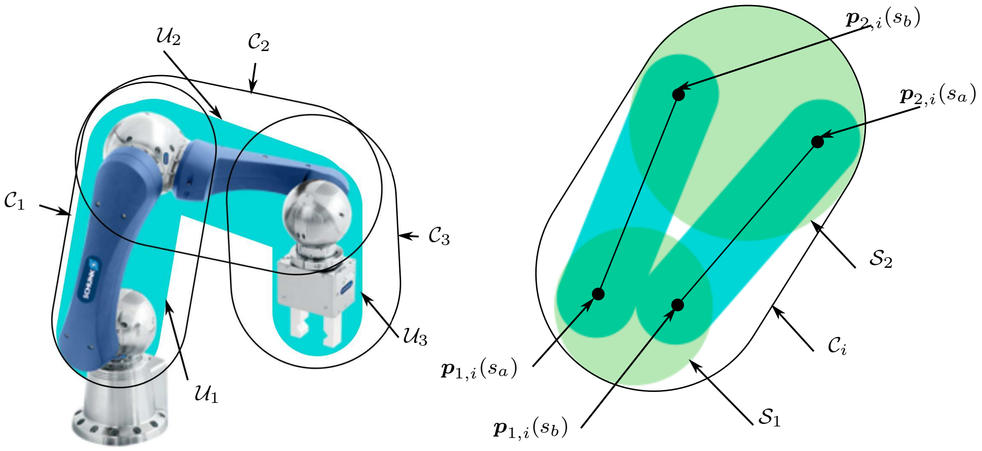
This method differs slightly from the previous methods, in that the set of joint positions is not an interval vector or a zonotope, but a segment of a parametrised curve in the joint space, i.e. a section of the robot's path $\boldsymbol{\xi}$ parameterised by a time parameter s, as defined in Def. 2.8.

If the RRO during the trajectory from $s = s_a$ to $s = s_b$ is to be found, we assume the positions at the start and end of the trajectory, i.e. $\boldsymbol{\xi}(s_a)$ and $\boldsymbol{\xi}(s_b)$, to be known, as well as the forward kinematics and the maximum value of the second derivative with respect to s of the Cartesian position of all the defining points of the capsules which define the robot occupancy, $\max_{s\in[s_a,s_b]}(\frac{\mathrm{d}^2\hat{F}(\boldsymbol{\xi})}{\mathrm{d}s^2})$. For the j^{th} defining point of capsule i, we call this value $\alpha_{j,i}$

Firstly, we enclose the geometry of each link i of the robot in a capsule $\mathcal{U}_i$, fixed in the link's own coordinate system, which are shown as the blue shaded capsules in Fig. 3.5. We obtain from the algorithm a capsule $\mathcal{C}_i$ enclosing the link geometry over the section of path from s_a to s_b, in the global coordinate system; these are the black outlined capsules in Fig. 3.5. Fig. 3.5(a) shows how the RRO is the union of these capsules, while Fig. 3.5(b) shows how each capsule $\mathcal{C}_i$ is calculated.

3.4.1 Ball and Capsule enclosure

We next introduce some necessary terminology and operators (all norms are Euclidean), before detailing the algorithm. In the models in this section, we often enclose two balls in a capsule or another ball (recall the definition of a ball from Def. 3.4); we call the operators for this CE and



(a) (b)

Figure 3.5: (a) the occupancy of each link of the robot is enclosed in capsules $\mathcal{U}_i$ (b) a capsule $\mathcal{C}_i$ is generated, which encloses the robot's occupancy during a section of its trajectory.

BE respectively. In order to define CE and BE on the balls $\mathcal{B}(\boldsymbol{p}_1; r_1)$ and $\mathcal{B}(\boldsymbol{p}_2; r_2)$, we first define the following terms:

$$\begin{aligned} i &= \texttt{indmax}(r_1, r_2), & j &= \texttt{indmin}(r_1, r_2) \\ \boldsymbol{x} &= \boldsymbol{p}_i - \boldsymbol{p}_j, & \alpha &= \max(r_i - r_j, \|\boldsymbol{x}\|) \\ \beta &= \min(r_i - r_j, \|\boldsymbol{x}\|) & \boldsymbol{p}_k &= \boldsymbol{p}_j + \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|} \cdot \beta \end{aligned} \tag{3.9}$$

The operators indmax and indmin output the indices of the maximum and minimum of their arguments. Let $\mathbf{0} \in \mathbb{R}^3$ be the zero vector. We are then able to define the operators:

$$\begin{aligned} \texttt{CE}(\mathcal{B}(\boldsymbol{p}_1; r_1), \mathcal{B}(\boldsymbol{p}_2; r_2)) &:= \sup([\boldsymbol{p}_i, \boldsymbol{p}_k]) \oplus \mathcal{B}(\mathbf{0}; r_i), \\ \texttt{BE}(\mathcal{B}(\boldsymbol{p}_1; r_1), \mathcal{B}(\boldsymbol{p}_2; r_2)) &:= B\Big(\frac{\boldsymbol{p}_i + \boldsymbol{p}_k}{2}; \frac{r_i + r_j + \alpha}{2}\Big), \end{aligned} \tag{3.10}$$

where $\sup([\boldsymbol{p}_i, \boldsymbol{p}_k])$ denotes the line segment between $\boldsymbol{p}_i$ and $\boldsymbol{p}_k$.

We prove that BE and CE yield the enclosing ball and capsule with the smallest volume, in Appendix A.4.

3.4.2 Detailed Algorithm

The positions of the defining points of $\mathcal{U}_i$ in the global coordinate frame, $\boldsymbol{p}_{1,i}$ and $\boldsymbol{p}_{2,i}$, can be found from the forward kinematics, i.e. $\boldsymbol{p}_{1,i}$ and $\boldsymbol{p}_{2,i}$ are functions of the joint position. Since the joint position in turn is a function of the path parameter s, for simplicity we express the Cartesian positions of the defining points of capsule i as $\boldsymbol{p}_{1,i}(s)$ and $\boldsymbol{p}_{2,i}(s)$.

As mentioned previously, we assume an upper bound for $\left\|\frac{\mathrm{d}^2\boldsymbol{p}_{1,i}(s)}{\mathrm{d}s^2}\right\|$ and $\left\|\frac{\mathrm{d}^2\boldsymbol{p}_{2,i}(s)}{\mathrm{d}s^2}\right\|$ for

$s \in [s_a, s_b]$ to be known. We call the value of these upper bounds $\alpha_{1,i}$ and $\alpha_{2,i}$ respectively; these can be precomputed over the entire trajectory, to obtain a conservative bound.

To obtain $\mathcal{C}_i$, we first calculate the two balls $\mathcal{S}_{1,i}$ and $\mathcal{S}_{2,i}$ which enclose the spatial path of the defining points $\boldsymbol{p}_{1,i}$ and $\boldsymbol{p}_{2,i}$ of the link capsule from s_a to s_b, enlarged by the radius $r_{\mathcal{U}_i}$ of $\mathcal{U}_i$; i.e.

$$\mathcal{S}_{1,i} \supseteq \{\boldsymbol{p}_{1,i}(s) | s \in [s_a, s_b]\} \oplus \mathcal{B}(0; r_{\mathcal{U}_i}),$$
$$\mathcal{S}_{2,i} \supseteq \{\boldsymbol{p}_{2,i}(s) | s \in [s_a, s_b]\} \oplus \mathcal{B}(0; r_{\mathcal{U}_i}).$$

These balls are then enclosed in a capsule to obtain $\mathcal{C}_i$:

$$\mathcal{C}_i = \texttt{CE}(\mathcal{S}_{1,i}, \mathcal{S}_{2,i}).$$

We now show how to generate $\mathcal{S}_{1,i}$ and $\mathcal{S}_{2,i}$. The path traced by $\boldsymbol{p}_{1,i}$ and $\boldsymbol{p}_{2,i}$ during the short-term plan cannot be assumed to be a straight line. However, we approximate it by a straight line with a known amount of deviation. Without loss of generality, consider the point $\boldsymbol{p}_{1,i}$. We bound the maximum deviation from the line segment between $\boldsymbol{p}_{1,i}(s_{k+1})$ and $\boldsymbol{p}_{1,i}(s_e)$, when the maximum value of $\left\|\frac{\mathrm{d}^2\boldsymbol{p}_{1,i}}{\mathrm{d}s^2}\right\| = \alpha_{1,i}$ is known.

Lemma 3.2: Let $\boldsymbol{x}(\varsigma) \in \mathbb{R}^3$ be the position of a point at time parameter ς and suppose $\forall \varsigma :$ $\|\ddot{\boldsymbol{x}}(\varsigma)\| \leq \alpha_i$. Let $\boldsymbol{x}_0$ and $\boldsymbol{x}_f$ be the known positions at time parameters $s = 0$ and $s = s_f$ and L be the line segment between them. For all $\varsigma \in [0, s_f]$, the distance from $\boldsymbol{x}(\varsigma)$ to L is no greater than $\alpha_i \frac{s_f^2}{8}$. □

Proof: From the equations of motion we have:

$$\boldsymbol{x}(s) = \boldsymbol{x}_0 + \dot{\boldsymbol{x}}(0)s + \int_0^s \int_0^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' \tag{3.11}$$

Choosing $s = s^*$, we have the position of $\boldsymbol{x}(s^*)$ at an arbitrary time $s^* \in [0, s_f]$. The line segment L can be expressed as the set $L = \{\boldsymbol{x}_0 + \lambda(\boldsymbol{x}_f - \boldsymbol{x}_0) | \lambda \in [0, 1]\}$. Substituting $s = s_f$ in (3.11), we obtain an expression for $\boldsymbol{x}_f = \boldsymbol{x}(s_f)$, and choosing $\lambda = \frac{s^*}{s_f}$ in the expression for the set L, we see that the point $\boldsymbol{x}' = \boldsymbol{x}_0 + \dot{\boldsymbol{x}}(0)s^* + \frac{s^*}{s_f} \int_0^{s_f} \int_0^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma'$ lies on L.

The difference $\boldsymbol{y}$ between $\boldsymbol{x}(s^*)$ and $\boldsymbol{x}'$ is:

$$\boldsymbol{y} = \frac{s^*}{s_f} \int_0^{s_f} \int_0^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' - \int_0^{s^*} \int_0^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' \tag{3.12}$$

Observing that $\int_0^s \int_0^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' = \int_0^{s^*} \int_0^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' + \int_{s^*}^{s_f} \int_0^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma'$, we obtain:

$$\boldsymbol{y} = \frac{s^*}{s_f} \underbrace{\int_{s^*}^{s_f} \int_0^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma'}_{b} - \frac{s_f - s^*}{s_f} \underbrace{\int_0^{s^*} \int_0^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma'}_{c} \tag{3.13}$$

By splitting the inner integral of b into two parts, we observe that:

$$b = \int_{s^*}^{s_f} \int_0^{s^*} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' + \int_{s^*}^{s_f} \int_{s^*}^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma'$$

And in the first term, the inner integral does not depend on ς', hence the outer integral integrates a constant. We can therefore rewrite b as:

$$b = (s_f - s^*) \int_0^{s^*} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma + \int_{s^*}^{s_f} \int_{s^*}^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' \tag{3.14}$$

Following a similar argument, c can be rewritten

$$\begin{aligned} c &= \int_0^{s^*} \int_0^{s^*} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' - \int_0^{s_f} \int_{\varsigma'}^{s^*} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' \\ &= s^* \int_0^{s^*} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma - \int_0^{s_f} \int_{\varsigma'}^{s^*} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' \end{aligned} \tag{3.15}$$

Substituting (3.14) and (3.15) back into (3.13) we obtain:

$$\boldsymbol{y} = \frac{s^*}{s_f} \underbrace{\int_{s^*}^{s_f} \int_{s^*}^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma'}_{d} + \frac{s_f - s^*}{s} \underbrace{\int_0^{s^*} \int_{\varsigma'}^{s^*} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma'}_{e} \tag{3.16}$$

We observe that d is the double-integral of acceleration from $\varsigma = s^*$ until s_f, which is upper-bounded by $\frac{\alpha_i (s_f - s^*)^2}{2}$. Furthermore, the expression e, which can be rewritten:

$$\int_0^{s^*} \int_{\varsigma'}^{s^*} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' = \int_0^{s^*} - \int_{s^*}^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma' = \int_{s^*}^{0} \int_{s^*}^{\varsigma'} \ddot{\boldsymbol{x}}(\varsigma) d\varsigma d\varsigma', \tag{3.17}$$

is the acceleration double-integrated "backward" from $\varsigma = s^*$ until 0. This is upper-bounded by $\frac{\alpha_i \varsigma^{*2}}{2}$. Since both s^* and $s_f - s^*$ are positive, we have:

$$\|\boldsymbol{y}\| \leq \frac{s^*}{s_f} \frac{\alpha_i (s_f - s^*)^2}{2} + \frac{s_f - s^*}{s_f} \frac{\alpha_i (s^*)^2}{2}. \tag{3.18}$$

The right hand side attains a maximum of $\alpha_i \frac{s_f^2}{8}$ at $s^* = \frac{s_f}{2}$ (this can be seen by setting to zero the differential of the right hand side with respect to s^*). Hence the shortest distance between $\boldsymbol{x}(s^*)$ and L is no more than $\alpha_i \frac{s_f^2}{8}$. □

We now calculate $\mathcal{S}_{1,i}$ and $\mathcal{S}_{2,i}$ as follows.

$$\begin{aligned}
r_{1,i} &= \alpha_{1,i}\frac{(s_a - s_b)^2}{8} + r_{\mathcal{U}_i} \\
r_{2,i} &= \alpha_{2,i}\frac{(s_a - s_b)^2}{8} + r_{\mathcal{U}_i} \\
\mathcal{S}_{1,i} &= \mathtt{BE}(\mathcal{B}(\boldsymbol{p}_{1,i}(s_{k+1}); r_{1,i}) \ , \ \mathcal{B}(\boldsymbol{p}_{1,i}(s_e); r_{z,i})) \\
\mathcal{S}_{2,i} &= \mathtt{BE}(\mathcal{B}(\boldsymbol{p}_{2,i}(s_{k+1}); r_{2,i}) \ , \ \mathcal{B}(\boldsymbol{p}_{2,i}(s_e); r_{z,i})) \\
\mathcal{C}_i &= \mathtt{CE}(\mathcal{S}_{1,i}, \mathcal{S}_{2,i})
\end{aligned} \tag{3.19}$$

The error $\alpha_{1,i}\frac{(s_a-s_b)^2}{8}$ and $\alpha_{2,i}\frac{(s_a-s_b)^2}{8}$ reflects the deviation of the points $\boldsymbol{p}_{1,i}$ and $\boldsymbol{p}_{2,i}$ from the straight line between them during this portion of the trajectory. Currently, $\alpha_{1,i}$ and $\alpha_{2,i}$ must be calculated during the planning of the path, and recalculated if the path is replanned. The method in its current implementation does not account for deviations from the robot's spatial path caused by, for example, trajectory tracking errors. In the case of robots with rigid links and good trajectory tracking, this is of limited importance, however, should be considered as scope for future work.

If the section of path over which the RRO is to be predicted is long, the overapproximation grows large, see comparison in Sec. 3.6. However, dividing the prediction horizon into smaller timescales, computing the prediction over these timescales and using the union of these predictions can produce a tighter RRO, at the expense of more collision-checks.

3.5 Note on Complexity

We examine complexity with respect to m, the number of DOFs.

Sphere-Swept Volumes In the method of [60] in Sec. 3.2, the defining points of the SSV are multiplied by either 3 (revolute DOFs) or 2 (prismatic DOFs) 4×4 transformation matrices at each successive DOF, obtaining an SSV with either 3 or 2 times as many defining points as the one at the previous DOF. The number of defining points in the SSV, and hence the number of multiplication operations to be performed, grows exponentially with the DOFs of the robot.

Collision-checking would be performed as distance checks between polytopes using a modification of the Gilbert-Johnson-Keerthi (GJK) algorithm [72], for which computation time has been found approximately equal to the number of points in the volume [73].

Interval Matrices The method of *interval matrices* in Sec. 3.3 requires the calculation of the kinematic transformations for each link coordinate system. For each additional link, this involves multiplying the forward kinematic transformation for the previous link, ${}^{0}_{i-1}\hat{F}(\boldsymbol{q})$ by the D-H transformation matrix ${}^{i-1}_{i}T$ in (3.1) from the $i-1^{\text{th}}$ coordinate system to the i^{th} ; the forward kinematics is therefore linear in the number of joints m.

It also requires the derivative of the forward kinematic transformations. Here, complexity is $O(m^2)$, since for each successive joint $i \in \{1, ..., m\}$, the derivative of the forward kinematics with respect to joint position of the j^{th} joint, $\frac{\mathrm{d}\,{}^{0}_{i}\hat{F}(\boldsymbol{q})}{\mathrm{d}q_j}$, is found for $j \in \{1, ..., i-1\}$ by multiplying $\frac{\mathrm{d}\,{}^{\;\,0}_{i-1}\hat{F}(\boldsymbol{q})}{\mathrm{d}q_j}$ by the D-H transformation matrix in (3.1) from the $i - 1^{\text{th}}$ coordinate system to the i^{th} , and for $j = i$, by multiplying the forward kinematic transform until $i - 1$, ${}^{\;\,0}_{i-1}\hat{F}\boldsymbol{q}$, by the derivative of (3.1) with respect to q_i. This is then $\frac{m(m-1)}{2}$ multiplication operations in total.

Next, we consider the calculation of B and C in (3.4) and (3.6). In (3.4), B is the summation over all p generators of $\left|\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{q}}\right|_{\boldsymbol{c}_Q} \in \mathbb{R}^{4\times4\times m}$ multiplied by by a generator $\boldsymbol{g}^{(l)} \in \mathbb{R}^{m\times1}$, requiring $16mp$ operations *for each of the m joints*, leading to $O(m^2)$ complexity (since p does not depend on m). In (3.6), D_{ij} is calculated by evaluating $\boldsymbol{d}^{\top}D_{ij}\boldsymbol{d}$ for each DOF; since $D_{ij} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{d} \in m \times 1$, and this must be done for each of m DOFs, this is of complexity $O(m^3)$. Finally, (3.8) is simply summation and is independent of the DOFs, but must be done for each link, hence is linear in m. Hence, the overall complexity of the algorithm is $O(m^3)$.

Occupancy Capsules The method of *occupancy capsules* requires the calculation of the kinematic transformations for each link coordinate system, which is linear in m as explained previously. The rest of the online calculations for each joint are independent of m: the calculation of $\mathcal{S}_{j,i}$ and $\mathcal{C}_i$ in (3.19) does not depend in m. Since all these calculations must be done for each joint $i \in \{1, ..., m\}$, obtaining the RRO by this method is therefore also linear in m.

3.6 Comparison of Methods

We compare these three methods in terms of their *computation time* and the volume of the RRO. We use a Unimation PUMA 560 6-DOF arm for comparison, since this is a familiar, well-studied robot and a CAD model is available freely online[4]. We calculated occupancies on a polynomial trajectory (i.e., joint position was a quintic function of time), where all joint positions started at $0°$ and ended at $90°$, joint speeds and accelerations were zero at both ends; the robot reached a top speed of $135°/s$ and the trajectory duration was $1.25s$.

We tested the methods from Sec. 3.2 (SSVs), Sec. 3.3 (*interval matrices*) and Sec. 3.4 (*occupancy capsules*) on this trajectory in three conditions. The trajectory was divided into 10, 20, 50 and 100 equal time intervals, of duration $125ms$, $62.5ms$ $25ms$ and $12.5ms$ respectively, and the RRO was calculated over these time intervals. Each link of the robot occupancy was enclosed in SSVs (for the method of SSVs), zonotopes (*interval matrices*) and capsules (*occupancy capsules*). For the latter, the second link was enclosed in 2 capsules, which resulted in a tighter enclosure than using a single capsule, since the link was long and flat. These are shown in Fig. 3.6.

We present the average, standard deviation, and maximum volume of the RRO over all time intervals in the trajectory in Tab. 3.1. The occupancies using each method are also visualised in Fig. 3.6 for $\boldsymbol{q} = \boldsymbol{45}°$ (top row) the portion of trajectory where $\boldsymbol{43}° \leq \boldsymbol{q} \leq \boldsymbol{47}°$, corresponding

[4] `grabcad.com/library/robot-puma-560`, retrieved 26.12.2017

Table 3.1: Volume comparison of the RRO over a sample of data (volumes in m^3). The trajectory is split into equal time intervals, of 4 different durations, and the average, standard deviation, and maximum volumes over the trajectory are calculated.

Size of time intervals		$12.5ms$	$25ms$	$62.5ms$	$125ms$
SSVs	μ	0.040	0.042	0.047	0.057
	σ	0.002	0.003	0.007	0.016
	$\max$	0.043	0.048	0.056	0.079
Interval Matrices	μ	0.043	0.056	0.149	1.478
	σ	0.200	0.018	0.104	1.925
	$\max$	0.053	0.082	0.314	4.918
Occupancy Capsules	μ	0.056	0.060	0.073	0.102
	σ	0.004	0.007	0.015	0.034
	$\max$	0.062	0.069	0.094	0.149

Table 3.2: Mean computation time over a section of data executed on a non-real-time machine

Method	Mean computation time (ms)
SSVs	0.52
Interval Matrices	1.03
Occupancy Capsules	0.27

to $s \in [0.610, 0.640]ms$ (middle row) and the portion of trajectory where $\mathbf{40}^\circ \leq \boldsymbol{q} \leq \mathbf{50}^\circ$, corresponding to $s \in [0.588, 0.662]ms$ (bottom row).

The methods of *interval matrices* and *SSVs* are tighter at smaller time horizons than the method of *occupancy capsules*; this is because the enclosures of the robot geometry in zonotopes and SSVs are tighter than the enclosure in capsules (or sets of capsules). However, the RRO found from *interval matrices* grows much more quickly than the other two, as a longer proportion of trajectory is enclosed. In Fig. 3.6, for example, even where the subtended angles of each joint is 10°, the method of *interval matrices* has a large overapproximation.

The tightest volume at longer time intervals comes from the method of SSVs. Note, however, that the RROs from the method of *SSVs* are complex–the SSV of the final link has $2 \times 3^6 = 1458$ defining points. This makes collision checking time-consuming during verification. By contrast, all volumes enclosing each link in the method of *occupancy capsules* are single capsules, with the exception of the second link, which is enclosed in 2 capsules.

We also present the average computation time to calculate the RRO in Tab. 3.2 (as can be seen from the algorithms presented in the previous sections, the computation time is independent of the length of the trajectory enclosed in the RRO). The fastest to compute in our tests is the method of *occupancy capsules*. Furthermore, since this is linear in the number of DOFs, while the method of *SSVs* is exponential and *interval matrices* is of cubic time complexity, it is likely that the method of *occupancy capsules* will excel in robots with additional DOFs, while the method of *SSVs* will become very slow.

One drawback of the method of *occupancy capsules* is that the set of positions accounted for is a one-dimensional subspace of the joint space, i.e. it is a portion of the trajectory, but does not account for uncertainty. A further disadvantage is that the maximum accelerations of the defining points of the capsules which enclose the link must be found in advance over the entire trajectory. Beyond the scope of this thesis, but an interesting direction for future work, could be to try to derive these automatically. For example, for point-to-point joint-space trajectories, the state space could be gridded and the Hessian matrix of the forward kinematics pre-computed offline over each segment of the grid as an interval tensor; the deviation from the line segment in Cartesian space between start and end position could be calculated.

Summary

The challenge of bounding the robot occupancy in a conservative way for a set of joint positions lies in the tradeoff between accuracy and computation time as number of degrees of freedom increase. State-of-the-art methods, although accurate, do not scale well with dimension. Here, two alternatives are presented. When the set of joint values is small, these methods yield tight occupancies, and are computationally more scalable than the state of the art. Furthermore, they generate the RRO in representations more suited to collision-checking against the HRO during verification.

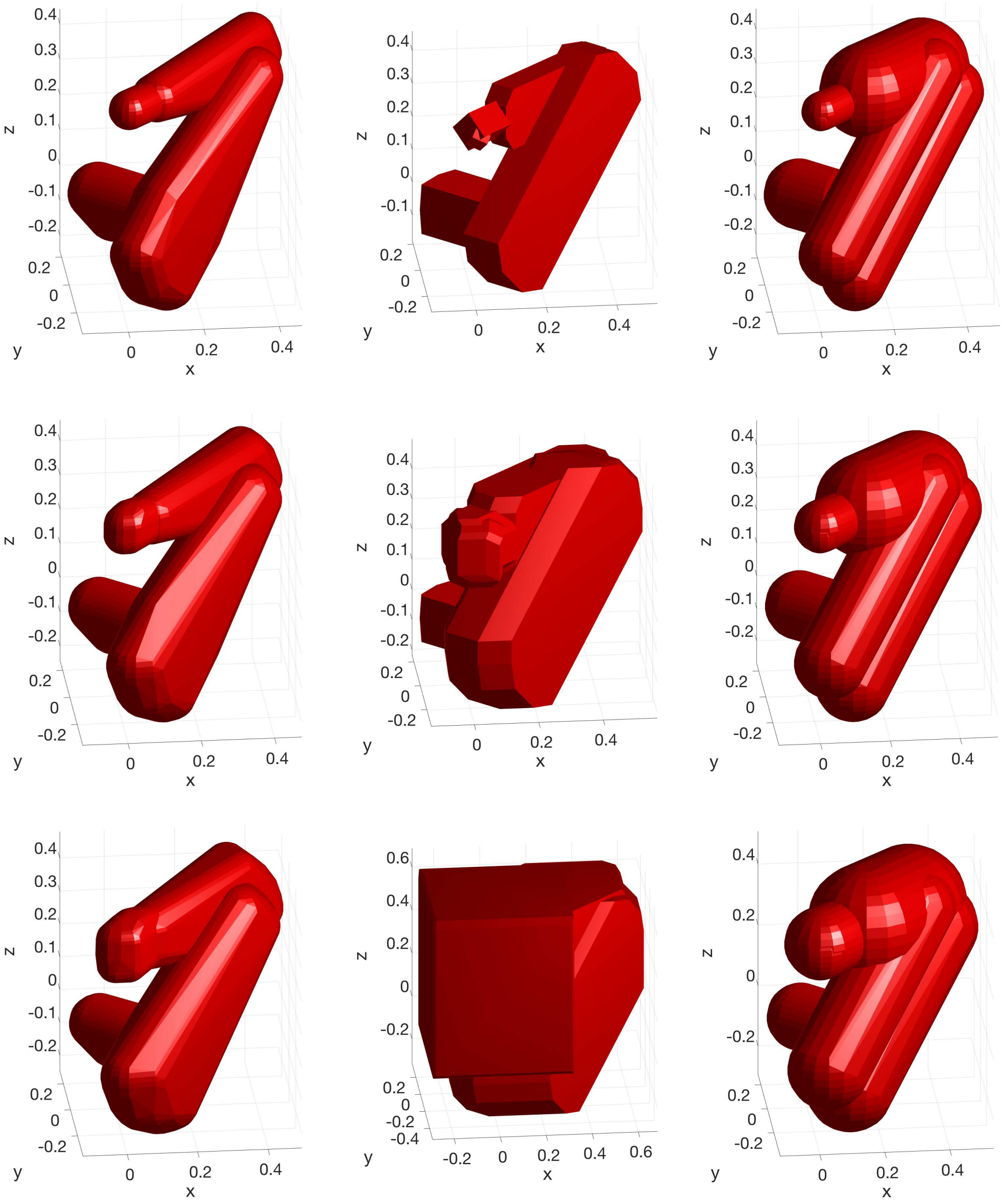


Figure 3.6: The methods of *SSVs* (left), *interval matrices* (middle), and *Occupancy capsules* (right), with all joint angles $45°$ (top), on the section of trajectory between all joint angles $43°$ and $47°$ (middle) and on the section of trajectory between all joint angles $40°$ and $50°$ (bottom).

Chapter 4

Human Occupancy

Is it possible to predict all areas the human could occupy while the robot is moving? This chapter tackles conservative prediction of human movement, which involves different challenges from the robot reachable occupancy (RRO) in the previous chapter. The robot trajectory and body geometry are known, whereas neither the intended movement, the dynamics, the geometry nor the physical capabilities of the human are known. In fact, we found that humans exceeded speeds used in the current standards for safety, even when performing everyday movements expected in a human-robot co-existence (HRCoex) scenario.

There is the added challenge of sensing the human's pose: humans, unhelpfully, do not have encoders fitted to their joints, and also have variable body geometry. Since the human's position is only available from sensor information, the model of human movement used to predict future occupancy depends on the type of sensor.

The following section discusses simple methods of predicting the human occupancy conservatively with simple but cheap and widely-used sensors. We then describe and compare novel methods for more complex occupancy predictions, where the sensors provide more accurate information about the human's pose, in Sec. 4.3 and Sec. 4.4.

4.1 Approaches without Skeletal Pose

Light curtains and alarmed doors are typical sensors currently used in factories to ascertain the position of the human, so we start by applying our approach with these sensors. The problem of determining whether the human is inside the cage or on the robot's side of the light curtain requires either an assumption that the human is never in the dangerous area when the system starts (e.g. in the case of cages, having doors that cannot shut from the inside) or additional sensors, such as a low-resolution camera or laser scanner, or floor-mounted pressure sensors. A risk-assessment must still be performed to determine that the human has no other means of approach to the robot and hence their presence is always detected.

When the light sensor is breached or the cage door is opened, we know that the human is somewhere in the doorway or the field of view of the light sensor. The entirety of this area is thus the human reachable occupancy (HRO). Having no other information, we calculate the HROs at future times using the maximum speed of the human from ISO 13855 [4], which is used to determine distances for mechanical safeguards. This puts the maximum speed of human movement at $v_{\max,human} = 2.0\frac{m}{s}$, or $v_{\max,human} = 1.6\frac{m}{s}$ when restricted to non-ambulatory upper-body movement. As we demonstrate later, humans can move much faster than this, however, this may be an acceptable estimate for our purposes. The reachable occupancy grows as shown in Fig. 4.1. Given l light curtains or alarmed doors, where the i-th door or curtain lies in the plane defined by $\{\boldsymbol{x} \in \mathbb{R}^3 \,|\, \boldsymbol{\nu}_i^\top \boldsymbol{x} \geq b_i\}$ with the robot in the halfspace $\{\boldsymbol{x} \in \mathbb{R}^3 \,|\, \boldsymbol{\nu}_i^\top \boldsymbol{x} = b_i\}$, and where $t_{resp,i}$ and $t_{tr,i}$ are the response time of the sensor and the time since it was triggered, respectively, the HRO can expressed as the union:

$$\Gamma([0,t]) = \bigcup_{i\in\{1,\dots,l\}} \left\{\boldsymbol{x} \in \mathbb{R}^3 \,\middle|\, \texttt{min\_dist}(\boldsymbol{x}, \text{sensor}_i) \leq v_{\max}(t_{resp,i} + t + t_{tr,i})\right\}$$
$$\bigoplus \text{space outside robot cage,}$$

where `min_dist` is the minimum distance between a point and a body in space, $t_{tr,i}$ is the duration for which the i^{th} sensor has been triggered (if it has not yet been triggered, $t_{tr,i} = 0$), and $t_{resp,i}$ is the response time for the i^{th} sensor, i.e. its worst-case latency. This HRO is illustrated in Fig. 4.1. For ease of collision-checking, it can be overapproximated as the union of halfspaces:

$$\Gamma([0,t]) = \bigcup_{i\in\{1,\dots,l\}} \left\{\boldsymbol{x} \in \mathbb{R}^3 \,\middle|\, \boldsymbol{\nu}_i^\top \boldsymbol{x} \geq b_i - v_{\max}(t_{resp,i} + t + t_{tr,i})\right\},$$
$$\bigoplus \Gamma([0,t]) = \bigcup_{j\in\{1,\dots,h\}} \left\{\boldsymbol{x} \in \mathbb{R}^3 \,\middle|\, \boldsymbol{\nu}_j^\top \boldsymbol{x} \geq b_j\right\},$$

where the robot cage has h planar walls, and the j^{th} wall is defined by the plane $\{\boldsymbol{x} \in \mathbb{R}^3 \mid \boldsymbol{\nu}_j^\top \boldsymbol{x} = b_j\}$, where the normal vector $\boldsymbol{\nu}_j$ points outward, away from the robot.

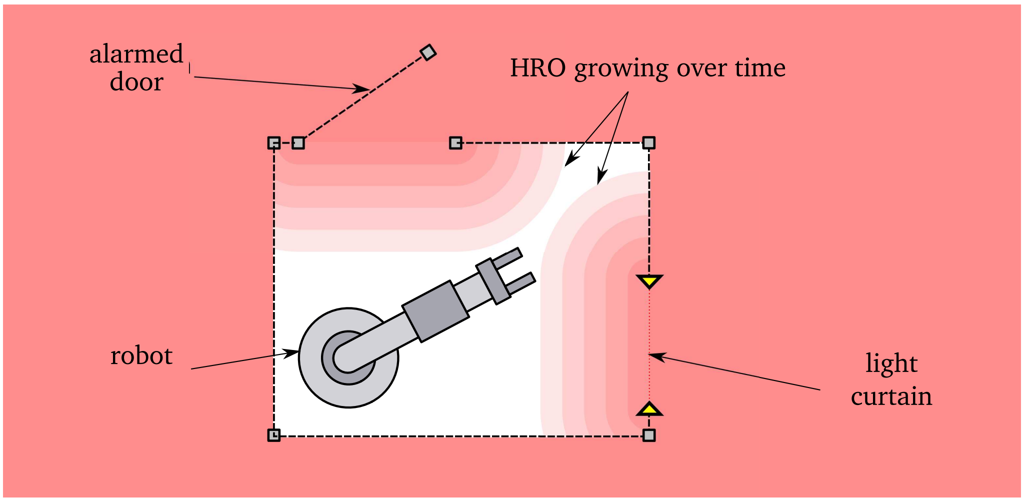


Figure 4.1: Illustration of human reachable sets of a human growing over time, with a robot in a cage equipped with a sensorised door and a light curtain.

Alarmed doors and light curtains are currently widely-used methods of detecting an incursion into the robot workspace. When the door is opened or the light curtain breached, the robot executes a Category 0 or Category 1 stop according to IEC 60204 [37] (see Footnote 5 in Sec. 1.1) meaning that power is removed from the motors and the operator must manually restart the robot to resume operation. In the formal approach of [178], the robot would perform a Category 2 stop, meaning that the robot does not remove power to the motors and can restart as soon as the human is verified safe.

A light sensor might be acceptable for an application where cost is an issue, for example in small and medium-sized enterprises (SMEs) in the food industry. Such simple sensing methods, however, do not exploit the whole potential of the continuously verified approach.

4.2 Approaches with Skeletal Pose

Cameras with depth sensors and motion-capture systems enable us to obtain the pose of the human or humans in the robot workspace, and thus use more complex models to predict human motion. As mentioned in the introduction, we are not interested in predicting the most probable behaviour or a probability distribution of behaviours but rather the entire set of behaviours possible by the human. Because of the quick reactiveness required in the formally guaranteed trajectory planner, the fast possible movement of the human and the latencies of typical pose-detection algorithms, this is a difficult problem. In the rest of this chapter, we address this challenge.

From the pose obtained from sensors it is possible to reconstruct an occupancy in space which encloses the human at an instantaneous point in time, see Fig. 4.2. The fidelity of the

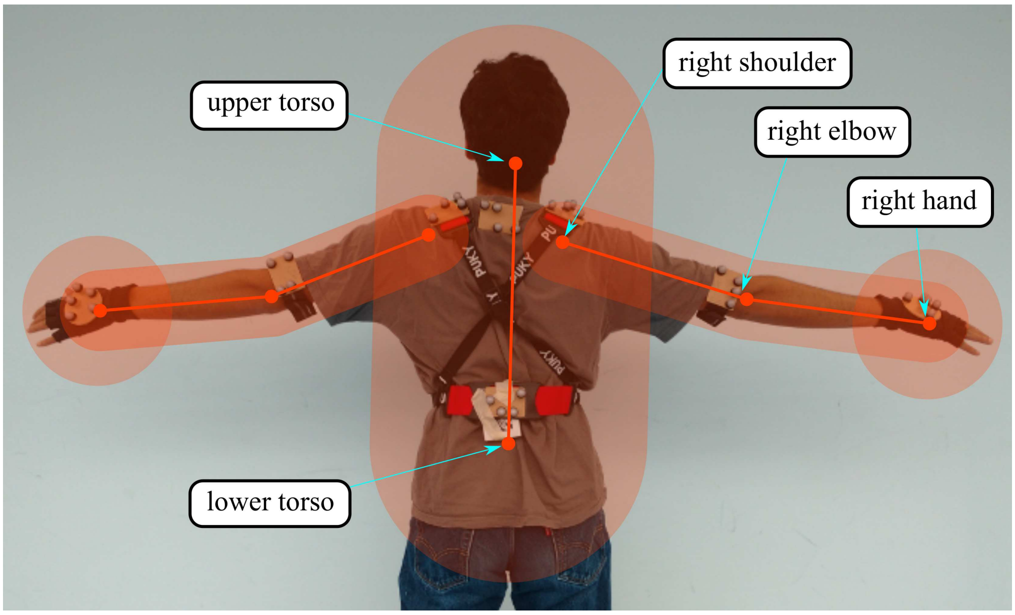


Figure 4.2: Position of defining points obtained from infrared retroreflective markers on the human, and the volumes enclosing the human which are defined using them. The marker clusters on the human are associated with coordinate systems in the tracking software. The origins of these coordinate systems are the red dots in the picture, which define 5 capsules (torso, right and left upper arms and forearms) and two spheres on the hand.

occupancy depends on the application area. In large spaces, ambulatory motion is the most relevant motion to consider, whereas at closer range to the robot, for example at a production line staffed by both robots and humans, upper-body movement is most relevant.

Ambulatory motion prediction has been previously studied in the field of safe mobile robots. Rohrmüller *et al.* [49] use a velocity and acceleration-limited model of human walking to compute probabilistic reachable sets, which inform a graph-search-based motion planner. Liu *et al.* [74] propose the same method of partial motion plans and online verification to control a mobile, human-following robot and show that this method improves upon state of the art methods with static safety zones around the robot.

Although the verification methods presented in this thesis could be applied equally well to mobile robots as to fixed-base robots, we focus on the latter. In such a situation, the lesser-tackled problem of predicting upper-body motion comes into play. We must first determine how to build a conservative approximation to the spatial occupancy of the human from sensor data.

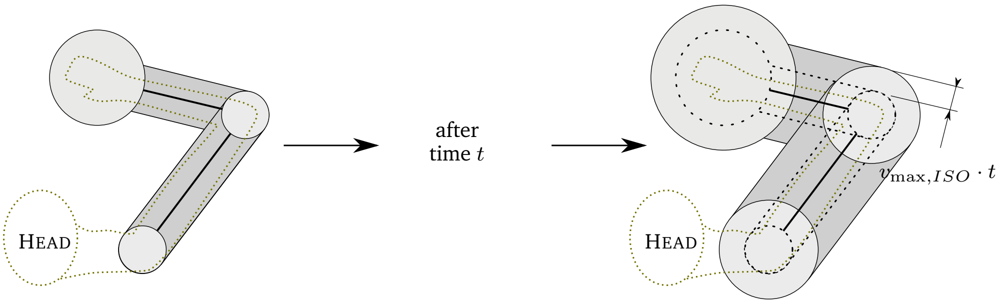


Figure 4.3: Prediction $\Gamma_{ISO}(t)$ using maximum speed of the human from [4].

4.2.1 Simplified Upper Body Occupancy

Having obtained the pose of the human from sensors, we wish to conservatively estimate the *occupancy*, i.e. the subset of Cartesian space occupied by a human given a certain pose.

The human is a high-degree of freedom structure occupying a nonconvex volume. In contrast to industrial robots, the surface of the human is deformable. However, the human skeleton is rigid, and several of the degrees of freedom of the body either have a limited range of motion, or their motion has a limited effect in the change of spatial occupancy. For example, there are 4 joints in the shoulder complex [75], but only the glenohumeral (GH) joint between the humerus and the shoulder blade has a range of motion more than a few degrees. Furthermore, the wrist has two degrees of freedom, but the movement of the hand occasioned through these degrees of freedom is rather small.

We use an occupancy as shown Fig. 4.2. This occupancy consists of 5 capsules and 2 spheres, whose defining points on the human may be obtained from motion-tracking markers. Since the spine is fairly limited in movement (at least, relative to the pelvis), we enclose the entire trunk and head in a capsule defined by points on the upper and lower torso.

We enclose the occupancy of the upper arm in a capsule defined at the shoulder and elbow, and the forearm in a capsule defined at the wrist and elbow. Since the wrist movement and the movement of the fingers do not change the occupancy greatly (compared to, for example, the elbow or the GH joint at the shoulder), we enclose the hand in a sphere centred at the wrist point. The radius of the sphere at the hand is chosen as $0.205m$, since this is the length from the wrist to the middle fingertip of a 95th percentile British male [76]. The forearm and upper arm radii are chosen to be $0.1m$, to account for clothing, and the radius of the capsule enclosing trunk and head is chosen as $0.35m$.

Note that the capsules enclosing the arm are only defined by points on the arm, which removes the connection between the arm and the torso and thus the need to model the sternoclavicular, acromioclavicular and scapulothoracic joints in the shoulder complex.

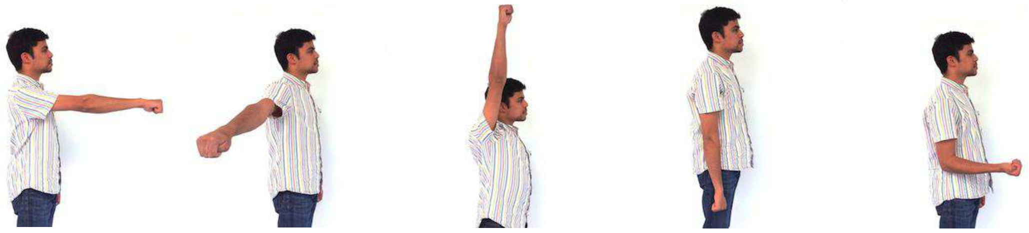


Figure 4.4: From left to right, positions A, B, C, D and E.

4.2.2 ISO-based Movement Model

When estimating the maximum speed of the human to calculate safety distances, the values to be used as specified by ISO 13855 [4] are $v_{\max,ISO} = 2.0\frac{m}{s}$ or $v_{\max,ISO} = 1.6\frac{m}{s}$ depending on whether whole-body motion or just upper-body motion should be considered. A conservative way to predict the reachable occupancy given these assumptions is to extend the radii of the capsules in Sec. 4.2.1 by the distance d that the human could have moved until the prediction horizon, given by:

$$d(t_e - t_0) = v_{\max,ISO}(t_e - t_0 + t_{lat}), \tag{4.1}$$

where t_{lat} is the worst-case latency of the sensor and the processing software, t_e is the prediction horizon and t_0 is the time that the control receives the pose estimate.

4.2.3 Investigating the Human's Dynamic Parameters

A drawback of the models introduced in the previous section is that they are based on some assumed maximum human velocities which do not reflect the true capabilities of human movement. In fact, humans can (and do) move faster than the assumed maximum speeds of $1.6\frac{m}{s}$ for the upper body or $2.0\frac{m}{s}$ for the whole body, for example when performing a reflex movement such as drawing the hand away from something hot or sharp, or catching a falling object. Since the scope of safety measures in the ISO standards [4, 2, 37] requires provision for unexpected use and expected misuse when working alongside a robot in a factory environment, one must also account for the such unexpected movements when claiming formal guarantees of safety for a trajectory planner: *if the assumptions of the model are invalid, then the guarantees of safety no longer hold.*

A study of human extreme movement was carried out with the dual aim of 1) testing the validity of the assumptions in [4] and 2) developing new, conservative models which are parameterised by this extreme movement and hence account for all human movement.

The kinematic and dynamic parameters of the models are obtained from collecting movement data from 38 healthy subjects (26 male, 12 female), aged between 18 and 49 years (median: 24 years), doing between 0 and 7 hours of sport a week (median: 3 hours). While their movements were tracked with a 6-Camera Vicon motion capture system, subjects were instructed to perform the following movements, designed to cover as much of the arm workspace as possible, as fast as they were able (see Fig. 4.4):

1. Punch[1] to the front, ending in position A.
2. Punch to the front then recover to position E.
3. Position C to position D, via position B.
4. Position C to position D, via position A.
5. Position A to position B, elbow not allowed to bend.
6. Position A to position B, elbow allowed to bend.

All movements were used to test the approach in Sec. 4.5.1 whereas only movements 1, 5 and 6 were used in Sec. 4.5.2. It was found that all test subjects were faster than the assumed maximum upper-body speed of $1.6\frac{m}{s}$ from [4]. The maximum speed found over all test subjects was $20\frac{m}{s}$. Using this in the model of Sec. 4.2.2 would lead to reachable occupancies grow extremely fast at larger time horizons, meaning that the robot movement would be verified unsafe unnecessarily often and performance would be limited. In practice, however, humans can only achieve these top speeds over short periods of time, and are limited by other constraints such as bounds on acceleration and a fixed skeletal structure (i.e., the distance between wrist and elbow, and between elbow and shoulder, do not change).

4.2.4 Joint Space and Cartesian Space Modelling Approaches

We can therefore propose more complex abstractions of human motion which account for extreme, unexpected movement, which are nevertheless fast enough to calculate a prediction in real time within the framework of Alg. 1. Our abstractions are grouped into those which model the arm as a kinematic chain (*Joint Space* approaches), and those which do not (*Cartesian Space* approaches). We present these in the following sections 4.3 and 4.4.

The difference between these two approaches can be seen in Fig. 4.5. In the joint-space approaches, the human pose found by the sensors is converted into a vector of joint positions in joint space, which is then enlarged by an error to account for measurement uncertainty, to obtain an initial set of joint positions. Reachability analysis [77] is then performed in the joint space to obtain the reachable set of joint positions during the prediction time interval (in the context of verified trajectory planning described in Chapter 2, the time from the start of the short-term plan until the robot is safe). This is then converted to a reachable occupancy Γ_{JS} in

[1] Punching movements were performed both against a punchbag and without one, in case subjects moved faster when using a punchbag for psychological reasons

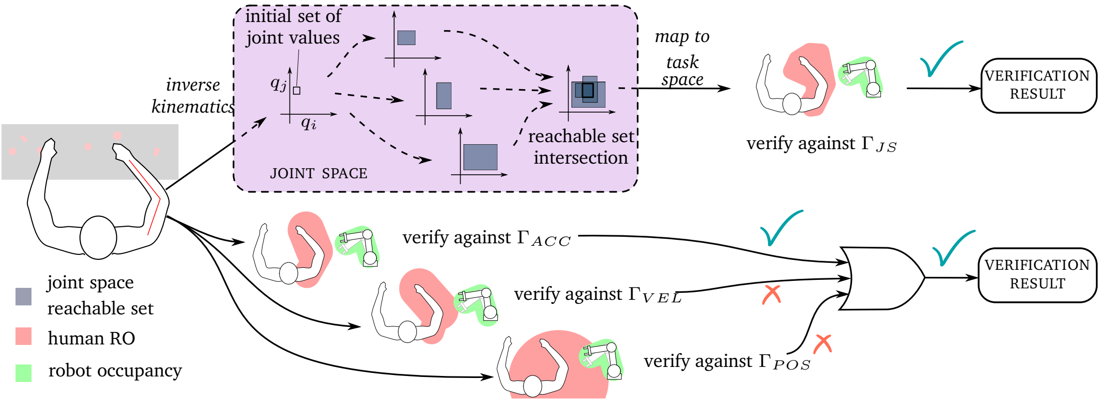

initial set of joint values
inverse kinematics
q_j
q_i
JOINT SPACE
reachable set intersection
map to task space
verify against Γ_{JS}
VERIFICATION RESULT
verify against Γ_{ACC}
verify against Γ_{VEL}
verify against Γ_{POS}
VERIFICATION RESULT
joint space reachable set
human RO
robot occupancy

Figure 4.5: An illustration of the joint space approaches described in Sec. 4.3 (above) and the Cartesian space approach described in Sec. 4.4 (below)

Cartesian space using a variation on the methods presented in [60], and checked against the reachable occupancy of the robot to verify safety.

In the Cartesian space approach, the reachable sets of the points used to define the occupancy described in 4.2.1 are found first, and the RO of the capsules enclosing forearm, upper arm, torso and the sphere enclosing the hand are found in Cartesian space directly. However, in order to account for constraints on acceleration, velocity and the fact that the rigid body lengths do not change, we require making three predictions of the HRO, Γ_{ACC}, Γ_{VEL}, and Γ_{POS}. Each of these is a union of spheres and capsules. However, we do not intersect these occupancies, since the intersection of spheres and capsules is not necessarily itself a sphere, capsule or sphere-swept volume (SSV), and is therefore hard to collision-check. Instead, we verify the short-term plan against each HRO individually; if any HRO is verified safe, we know that the exact HRO is also safe, and hence we consider the short-term plan verified.

The advantage of the joint-space approaches is that the kinematic structure of the arm – the fact that rigid bodies do not deform – is automatically taken into account, and that only one HRO need be checked against the RRO. As will be discussed in Sec. 4.3, the orientation of the shoulder with respect to the torso cannot be neglected and adds to computational complexity as well as enlarging the reachable occupancy due to overapproximation. The occupancy is also in a more complex representation–a set of sphere-swept volumes–compared to the Cartesian space method, where the occupancy is a set of capsules. This means that less-complex collision-checking algorithms can be used for the Cartesian Space approach.

The advantage of the Cartesian Space approach is that one does not have to perform expensive inverse kinematics, which significantly improves computational efficiency as shown later in Sec. 4.5. A disadvantage, however, is that the HROs found from the three models presented above are not explicitly intersected, possibly giving rise to situations where the robot is verified unsafe against each model individually, though the intersection of Γ_{POS}, Γ_{VEL}, and Γ_{ACC}

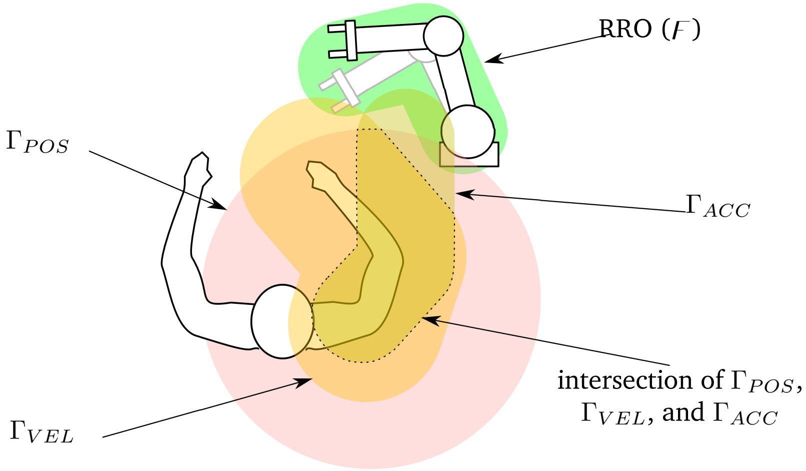

RRO ($\mathcal{F}$)
Γ_{POS}
Γ_{ACC}
intersection of Γ_{POS}, Γ_{VEL}, and Γ_{ACC}
Γ_{VEL}

Figure 4.6: Although the HRO from the individual models all intersect with the RRO, their intersection does not. Hence in this case, the robot would be verified unsafe when in fact it could have been verified safe.

would be intersection-free (see Fig. 4.6). This is a trade-off we make against speed of computation. In the next sections, we describe both approaches.

4.3 Joint Space Approaches

The approximation of the human arm by a kinematic chain of revolute joints has a history in the literature, e.g. [81, 82, 79]. The arm from shoulder to wrist has often been considered to have seven degrees of freedom, see Fig. 4.7(a). The shoulder complex consists of several joints (see Fig. 4.8), however, most have a limited range of movement, and we can consider the movement of all joints but the GH joint as part of torso movement; the arm is then considered to start at the GH joint. This is a ball-and-socket joint, and can be modelled by 3 revolute joints in series, the elbow joint can be modelled by a revolute joint with axis approximately normal to the forearm and upper arm, pronation/supination of the hand is a revolute joint with axis along the forearm and flexion/extension and lateral movement of the hand is modelled by two orthogonal revolute joints at the wrist. In the following models, the position of the i^{th} joint is denoted q_i.

4.3.1 4-DOF model

Since the latter three joints affect mostly the position of the hand and fingers, and hence do not contribute to large changes in the occupancy of the human, we ignore these; as mentioned in Sec. 4.2.1, we account for all motion of the hand within a sphere centred at the wrist, so we are able to neglect these latter three degrees of freedom, leaving us with a 4-DOF model. Such a simplified model has been used in e.g. [80, 81, 83].

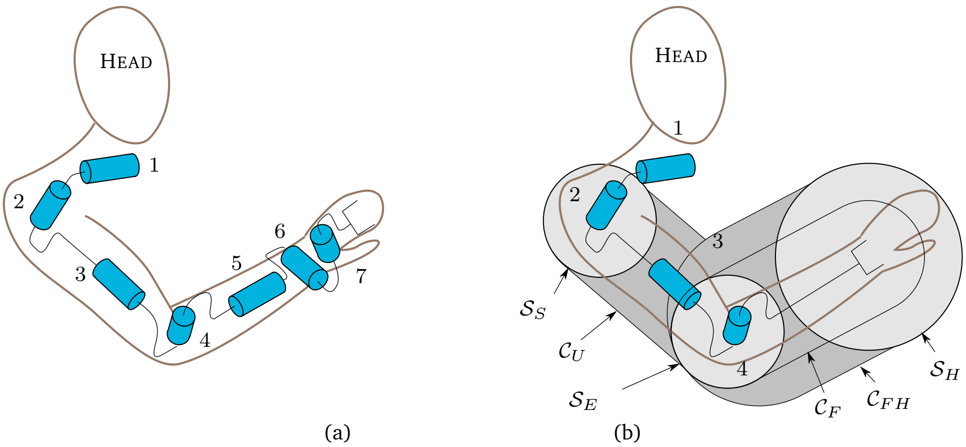


Figure 4.7: (a) classical 7-DOF human arm, (b) 4-DOF simplification excluding wrist and hand movement, similar to [78, 79, 80]. q_1, q_2 and q_3 are the joint angles at the shoulder, with q_3 being the rotation around the upper arm. q_4 is flexion and extension of the shoulder.

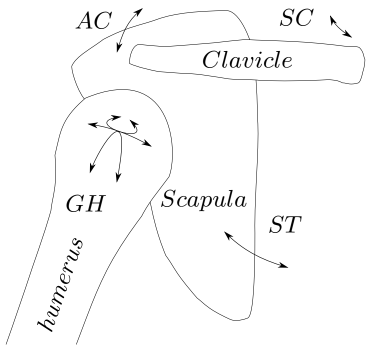


Figure 4.8: The shoulder complex. SC: Sternoclavicular joint, AC: Acromioclavicular joint, ST: Scapulothoracic joint, GH: Glenohumeral joint. All but the latter have limited motion; we focus attention on the GH joint, which is a ball-and-socket (spherical) joint.

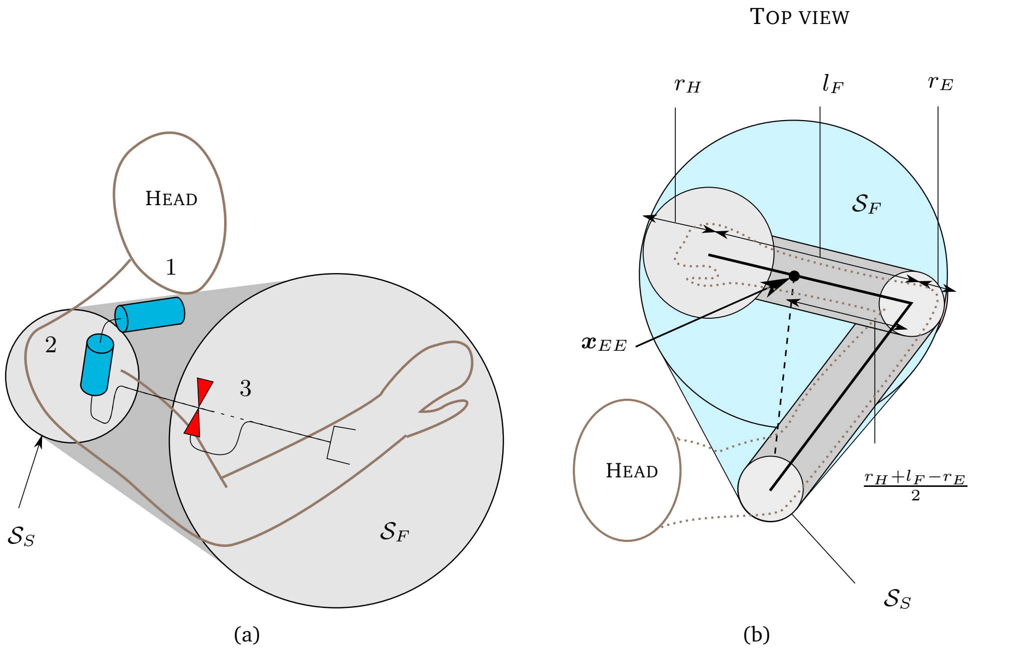


Figure 4.9: (a) The 3-DOF model kinematics. q_1 and q_2 are the joint angles of the rotational joints and q_3 is the position of the prismatic joint, with 0 position at the shoulder. (b) The calculation of r_F, the radius of $\mathcal{S}_F$

The occupancy predicted by this 4-DOF model consists of the occupancy of the upper arm, enclosed in a capsule $\mathcal{C}_U$, the forearm, enclosed in a capsule $\mathcal{C}_F$ and the hand, enclosed in a sphere $\mathcal{S}_H$. We may also make a further simplification to reduce the number of volumes in the HRO to collision-check with the RRO, by enclosing $\mathcal{C}_F$ and $\mathcal{S}_H$ in a capsule $\mathcal{C}_{FH}$, see Fig. 4.7(b).

Recall from Sec. 4.2.3, that we perform inverse kinematics on the arm to obtain parameters of the maximum velocities and accelerations. In this 4-DOF model, the first three joints are orthogonal to each other, leading to a coordinate singularity when the upper arm is near alignment with the first joint axis (similarly to Gimbal lock). Also, another singularity occurs at the elbow: when the elbow is fully outstretched, the position of the third joint is indeterminate from inverse kinematics. We show how we solve the coordinate singularity in Sec. 4.3.4. To solve the latter problem, we replace the third and fourth revolute joints with a prismatic joint as detailed in the following section.

4.3.2 Simplified 3-DOF arm model

To simplify the 4-DOF arm to a 3-DOF abstraction, we also modify the occupancy of the arm from the two capsules and a sphere described in Sec. 4.2.1, to create another occupancy which encloses that of Sec. 4.2.1. See Fig. 4.9: The capsule $\mathcal{C}_F$ and the sphere at the hand $\mathcal{S}_H$ enclosing the hand in the occupancy of Sec. 4.2.1 is now enclosed in a sphere $\mathcal{S}_{FH}$. We also enclose the shoulder in a sphere $\mathcal{S}_S$ with radius r_S, which is the same that of as $\mathcal{C}_U$ in the occupancy of Sec. 4.2.1.

Consider now a sphere $\mathcal{S}_E$ around the elbow with the same radius r_S as the capsules $\mathcal{C}_F$ and $\mathcal{C}_U$. This is a subset of $\mathcal{C}_F$ and is therefore enclosed in $\mathcal{S}_{FH}$. Notice that the capsule $\mathcal{C}_U$, being defined by the shoulder and elbow points–which are the centres of the spheres $\mathcal{S}_S$ and $\mathcal{S}_E$–and having a radius equal to these spheres, is in fact the convex hull of spheres $\mathcal{S}_S$ and $\mathcal{S}_E$. In that case, we deduce that the capsule $\mathcal{C}_U$ is included in the convex hull of $\mathcal{S}_{FH}$ and $\mathcal{S}_S$. We take this to be our occupancy, as shown in Fig. 4.9(a).

The kinematics of the 3-DOF abstraction are shown in Fig. 4.9(a). The first two rotary joints at the shoulder are kept, but are followed by a prismatic joint with an axis from the shoulder to the end-effector. The end-effector is the centre of the sphere $\mathcal{S}_{FH}$; contraction and extension of the prismatic joint corresponds approximately to flexion and extension of the elbow.

The end-effector lies on the line segment from the wrist to the elbow, and its position $\boldsymbol{x}_{EE}$ is calculated from the positions of wrist $\boldsymbol{x}_W$ and elbow $\boldsymbol{x}_E$, and radii r_S and r_H of $\mathcal{S}_S$ and $\mathcal{S}_H$, as follows (see Fig. 4.9(b)):

$$\lambda = \frac{r_H - r_S + \|\boldsymbol{x}_W - \boldsymbol{x}_E\|}{2\|\boldsymbol{x}_W - \boldsymbol{x}_E\|}$$
$$\boldsymbol{x}_{EE} = (1-\lambda)\boldsymbol{x}_E + \lambda \boldsymbol{x}_W$$

4.3.3 Placement of Shoulder Coordinate System from Markers on Arm

Fig. 4.10 shows the positions of the markers on the arm, and motion capture data was fit to the kinematic model as follows. The base coordinate system has the same orientation as that of the clavicle in the ISB recommendations [84], and is defined by the markers T10 on the mid-spine, C7 on the upper spine, and STRN and CLAV on the lower and upper ends of the breastbone respectively. The origin is located at the centre of the GH spherical joint, and is estimated at $40mm$ below (i.e. in the negative y-direction from) the marker RSHO or LSHO (on the markers, the prefix L/R refers to left/right arm respectively), and referred to as the *shoulder*. The joint angles, however, are defined differently from [84], as motivated in Sec. 4.3.4. The *elbow* position is taken to be the marker RELB/LELB. The *wrist* position is taken to be the midpoint of the two markers on the wrist RWRA and RWRB/LWRA and LWRB.

4.3.4 Placement of First Joint Axis

In both the 3-DOF and the 4-DOF models, the first two joints in the kinematic chain are both revolute and are orthogonal. When the elbow (in the 4-DOF model) or the end-effector (in the 3-DOF model) lies on the first joint axis, this occasions a coordinate singularity, meaning that at this point, q_1 cannot be calculated from inverse kinematics, and near this point, the value of q_1 varies a lot for small variations in the task space, see Fig. 4.11.

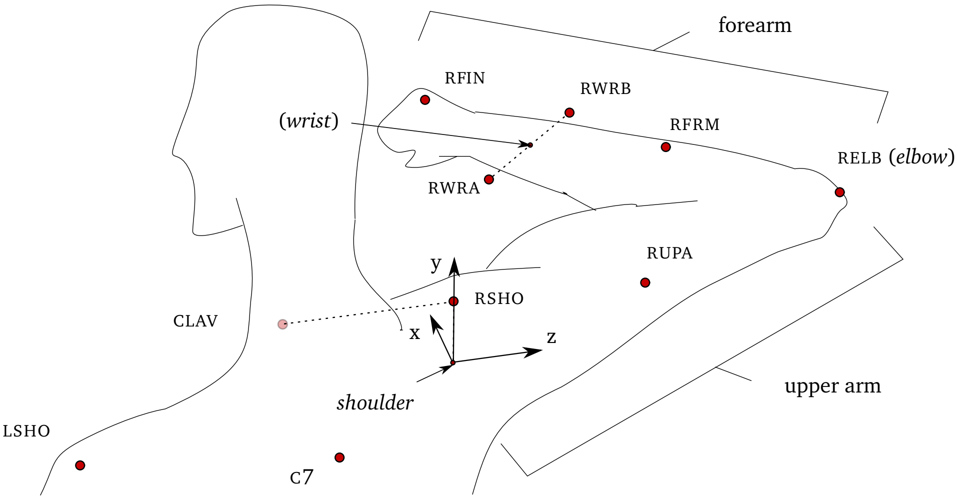


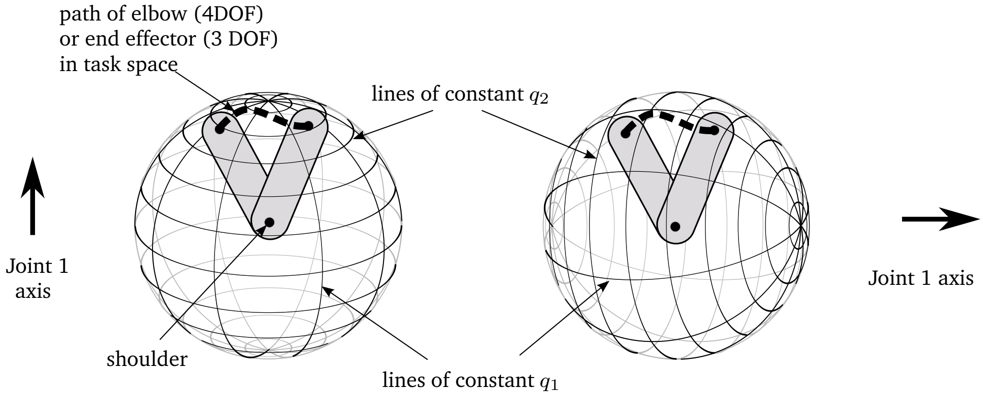
Figure 4.10: Arm base coordinate system with markers. Marker CLAV is at the top end of the sternum on the torso anterior; STRN (not depicted) is at the base of the sternum also on the torso anterior; T10 (not depicted) is on the back, on the mid-spine. The base coordinate system origin is 40mm below RSHO (in the negative y direction) and its orientation is defined by T10, STRN, CLAV and C7. The z-axis is parallel to the line through CLAV and RSHO.

Since we can freely choose the orientation of the first joint axis, we can avoid this singularity. In [83], the authors choose the first joint axis to point horizontally away from the body; Zacharias [85] notes that this moves the singularities away from the positions encountered in normal human movement. In order to find an axis through the origin, i.e. the shoulder, that the elbow (4-DOF model) or end effector (3-DOF model) does not intersect during any movement, we plot all marker data, as shown in Fig. 4.12. From observation of the sampled workspace, we were able to see that one possible axis for the 3-DOF model is the axis along the long axis of the clavicle (i.e. the z-axis in Fig. 4.10 of Appendix 4.3.3) rotated around the y-axis by $5°$ and subsequently around the x-axis by $5°$, depicted in Fig. 4.12(a). In Fig. 4.12(b) is shown a possible axis for the 4-DOF model; the long axis of the clavicle rotated around the y-axis by $20°$. The reader can verify on their own arm that it is exceedingly difficult to align the humerus with the clavicle, since this position is blocked by the Acromial Process.

4.3.5 Accounting for Measurement Uncertainty

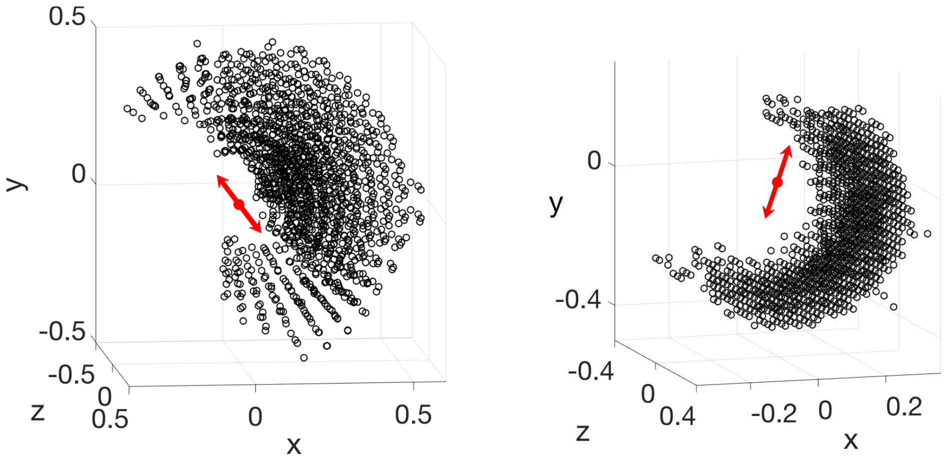
The sensors used to obtain the pose of the human are subject to uncertainty. In order to account for this, we enlarge the initial pose in the joint space by predetermined uncertainty to obtain an initial set in joint space. However, this is not sufficient to guarantee that the initial reachable occupancy accounts for the measurement uncertainty, in the case of the 4-DOF model.

In the 3-DOF model, the forearm is enclosed, if $\mathcal{S}_{FH}$ encloses the forearm plus measurement error; the upper arm is then again automatically enclosed as both $\mathcal{S}_E$ and $\mathcal{S}_S$ are inside the convex hull of $\mathcal{S}_S$ and $\mathcal{S}_{FH}$ plus measurement error.



(a) (b)

Figure 4.11: Choosing the joint 1 axis as in (a) means that the inverse kinematic solution shows rapidly changing q_1 during the trajectory shown; if the end effector aligns with the joint 1 axis the inverse kinematics is undefined. A more appropriate choice of axis to avoid singular values of q_1 and minimise $\dot{q}_1$ and $\ddot{q}_1$ is shown in (b).



(a) (b)

Figure 4.12: The sampled workspaces of (a) the end-effector of the 3-DOF model and (b) the elbow of the 4-DOF model, along with their respective axes (red lines). The workspace does not intersect the axes, hence the workspace should be free of singular positions. Scale in metres; coordinate system is that of the shoulder.

In the 4-DOF model, this is not the case; see Fig. 4.13. If we choose to enclose $\mathcal{C}_F$ and $\mathcal{S}_H$ in a capsule $\mathcal{C}_{FH}$, only uncertainty in the upper arm length may cause the arm occupancy to lie outside the predicted reachable occupancy. Otherwise, uncertainty in both the length of the forearm and upper arm cause this problem. In these cases, we add one or two extra prismatic joints to account for the measurement uncertainty in the upper arm and forearm length. This is shown in Fig. 4.14.

4.3.6 Dynamic Model

As previously mentioned, we use reachability analysis to determine the reachable occupancies (ROs) of the human. We calculate the reachable sets of the system (Def. 2.1), which we then convert to reachable occupancys (ROs) (Def. 2.2). We discuss here two possible models to determine the dynamics of the presented kinematic chain from the motion data gathered in Sec. 4.2.3: a torque-based model (Sec. 4.3.7) and an acceleration-based model (Sec. 4.3.8). We motivate the use of the acceleration-based model. We define the kinematic chain's *state* $\boldsymbol{x}$ as $[\boldsymbol{q}^\top, \dot{\boldsymbol{q}}^\top]^\top$, where $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are the vectors of joint positions and velocities respectively.

4.3.7 Torque Model

The human body is actuated by muscles contracting and hence exerting forces on bones, transmitted through tendons. The torque effected on the joints of the skeleton depends on the moment arm of the muscles about the joint, their length and various other state variables of the muscles, as well as intra-muscular force transmission [86].

The field of human biomechanical modelling has a rich literature and experimental datasets which could be used to find the range of torques applicable to actuate each joint of the arm. For example, Holzbaur *et al.* [25] present a mechanical model of the human arm where the maximum muscle force for each muscle in the human arm is taken from a range of datasets. The relationship between the moment arm and the bone dimensions is further presented in [87]. An intuitive approach would be to find the range of torques from these muscles and use these as the inputs to the dynamical model. As an alternative, Otis *et al.*[88] measure the range of abduction, flexion and rotation torques for a range of shoulder joint positions and angular velocities, and Amis *et al.*[89] measure the forces in the elbow during extreme elbow movements, allowing us to estimate the range of torques for all DOFs of the 4-DOF model in Sec. 4.3.1. The dynamics of the torques in our simplified models of the human arm are identical to those of a kinematic chain and can be written as ([90]):

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{q} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{f}(\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_{ext} \tag{4.2}$$

With $\boldsymbol{\tau}$ the total torques on the system, $\boldsymbol{\tau}_{ext}$ the external torques, $\boldsymbol{M}(\boldsymbol{q})$ the mass and inertia matrix, $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ the vector of Coriolis and centrifugal terms, $\boldsymbol{f}(\dot{\boldsymbol{q}})$ the friction terms, and $\boldsymbol{g}(\boldsymbol{q})$ the gravity terms.

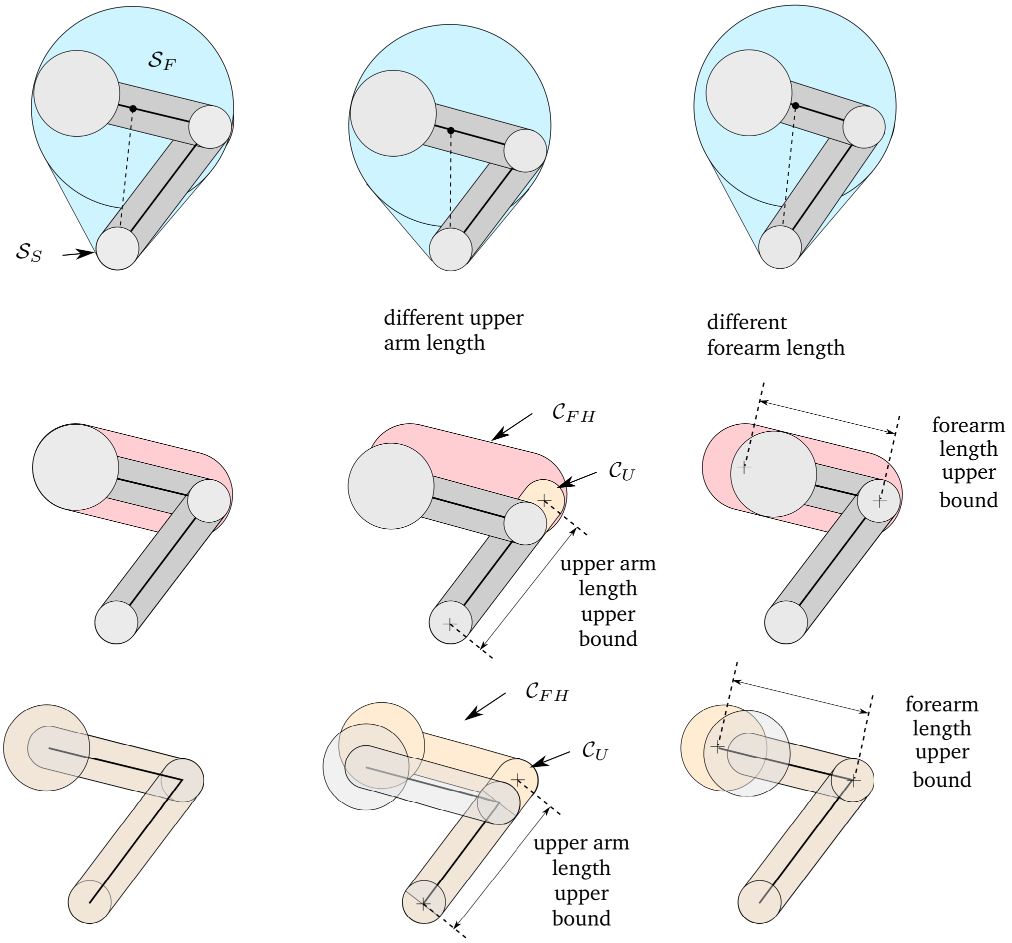

$\mathcal{S}_F$
$\mathcal{S}_S$
different upper arm length
different forearm length
$\mathcal{C}_{FH}$
$\mathcal{C}_U$
upper arm length upper bound
forearm length upper bound
$\mathcal{C}_{FH}$
$\mathcal{C}_U$
upper arm length upper bound
forearm length upper bound

Figure 4.13: The 3-DOF model (above) and 4-DOF with $\mathcal{C}_F$ and $\mathcal{S}_H$ enclosed in capsule $\mathcal{C}_{FH}$ (middle) and without enclosure (below), coping with measurement uncertainty in arm length. The 4-DOF model cannot account for variations in humerus length, as the variable upper arm length means the forearm may not be enclosed by $\mathcal{C}_{FH}$; if $\mathcal{C}_F$ and $\mathcal{S}_H$ are not enclosed in $\mathcal{C}_{FH}$, the hand may also fall outside the occupancy in case of measurement uncertainty in the forearm.

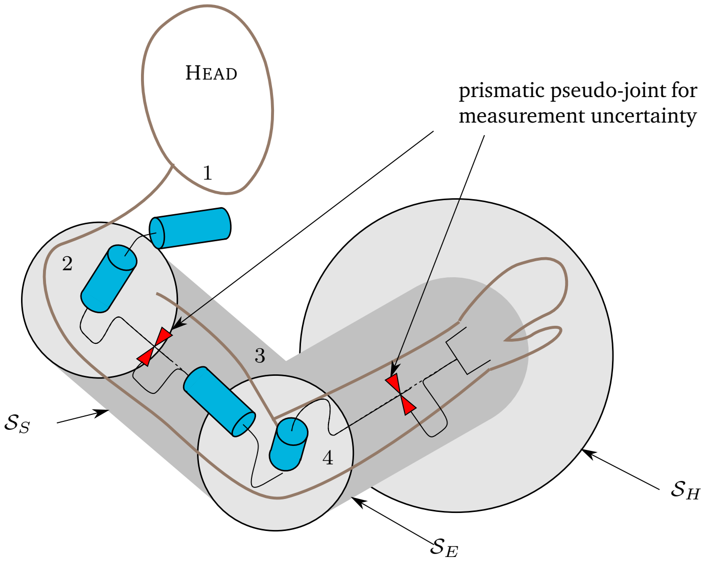


Figure 4.14: Measurement uncertainty in the forearm and upper arm is accounted for by adding prismatic pseudo-joints in the 4-DOF model. In the case that the forearm $\mathcal{C}_F$ and hand $\mathcal{S}_H$ are merged to a capsule $\mathcal{C}_{FH}$, only a prismatic joint in the upper arm is necessary.

It is challenging to calculate the reachable sets of joint positions in such a system, since:

1. The mass, inertia and friction parameters for calculating $\boldsymbol{M}$, $\boldsymbol{n}$, $\boldsymbol{f}$ and $\boldsymbol{g}$ are uncertain and vary greatly from one human subject to the next; accounting for such uncertainty in the reachability analysis can be done (e.g. [91]) but introduces overapproximation.

2. Gravity $\boldsymbol{g}(\boldsymbol{q})$ depends on the orientation of the base coordinate system, and $\boldsymbol{\tau}_{ext}$ depends on the dynamics of the base coordinate system and on external contacts with the arm. Hence the movement of the entire human body must also be measured and accounted for.

3. The whole system dynamics in (4.2) are highly non-linear. While the reachable sets of nonlinear systems of equations can be calculated [27, 91], computation times are currently not real-time-capable.

These difficulties, particularly the last one, motivate the use of linear dynamic models which are described in the following sections.

4.3.8 Linear Model

We present here a model where joint acceleration $\ddot{\boldsymbol{q}}$ is a linear function of the state $\boldsymbol{x} = [\boldsymbol{q}^\top, \dot{\boldsymbol{q}}^\top]^\top$. We calculate reachable sets for such a model starting from *differential inclusions* [92] describing the robots dynamics. A comprehensive guide on how to calculate reachable

Table 4.1: Correlations between joint positions and velocities, and acceleration, over entire dataset (highest values for each joint acceleration in **bold**).

(a) 4-DOF System

r^2	q_1	q_2	q_3	q_4	$\dot{q}_1$	$\dot{q}_2$	$\dot{q}_3$	$\dot{q}_4$
$\ddot{q}_1$	**-0.36**	-0.12	-0.10	**0.20**	0.05	0.14	-0.07	**0.18**
$\ddot{q}_2$	**-0.11**	**-0.32**	0.02	0.08	-0.07	0.05	-0.05	**0.15**
$\ddot{q}_3$	**-0.09**	0.04	**-0.17**	-0.02	0.03	0.01	0.04	**-0.08**
$\ddot{q}_4$	0.07	0.01	-0.02	**-0.25**	**-0.16**	**-0.15**	0.06	0.07

(b) 3-DOF system

r^2	q_1	q_2	q_3	$\dot{q}_1$	$\dot{q}_2$	$\dot{q}_3$
$\ddot{q}_1$	**-0.39**	-0.06	**-0.15**	0.04	0.12	**-0.22**
$\ddot{q}_2$	-0.06	**-0.33**	-0.03	**-0.06**	0.05	**-0.19**
$\ddot{q}_3$	-0.04	0.03	**-0.23**	**0.15**	**0.16**	0.07

sets for linear systems is presented in [69, Chapter 3], and implemented in the CORA software [66], which we used.

Despite the wealth of literature on maximum arm torques and muscle forces, no complete study of the maximum human arm joint accelerations has yet been carried out prior to our work, to the best knowledge of the author. Studies of extreme motion of individual joints such as the elbow [89] and of typical (but not extreme) motion of arm joints [93]. We therefore show how to use the movement data gathered in Sec. 4.2.3 to generate these differential inclusions.

Let k be the number of joints in the kinematic model (3 or 4), C a matrix of size $k \times 2k$ of coefficients to be determined, $\boldsymbol{I}_k$ an identity matrix of dimension k, $\boldsymbol{0}_{k\times 1}$ a vector of zeros of length k, $\boldsymbol{0}_{k\times k}$ a matrix of zeros of size $k \times k$, and $\boldsymbol{u}_{\min}$ and $\boldsymbol{u}_{\max}$ a vector of uncertain inputs to the model. The differential inclusions describing the dynamics are of the form:

$$\begin{bmatrix} \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} \end{bmatrix} = \dot{\boldsymbol{x}} \in \begin{bmatrix} \boldsymbol{0}_{k\times k} & \boldsymbol{I}_k \\ \multicolumn{2}{c}{\boldsymbol{C}} \end{bmatrix} \boldsymbol{x} \oplus \begin{bmatrix} \boldsymbol{0}_{k\times 1} \\ [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}] \end{bmatrix}. \tag{4.3}$$

Reachable sets of linear, first-order systems of differential inclusions such as (4.3) can be calculated rapidly for any time interval, when given an initial set. In order to avoid the wrapping effect, the reachable sets are obtained as zonotopes, and overapproximative reachable sets are obtained as detailed in [94, 69].

Note that the coefficients of $\boldsymbol{C}$ can be freely chosen, as long as the entire differential inclusion accounts for all the data. If $\boldsymbol{C}$ were to be a matrix of zeros, this means that the acceleration does not depend on the state. In order that (4.3) account for all of our observed data, $\boldsymbol{u}_{\min}$ and $\boldsymbol{u}_{\max}$ would be the minimum and maximum accelerations observed over all the data. Instead, if we choose the coefficients of $\boldsymbol{C}$ such that the range of inputs $[\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}]$ is minimised, we

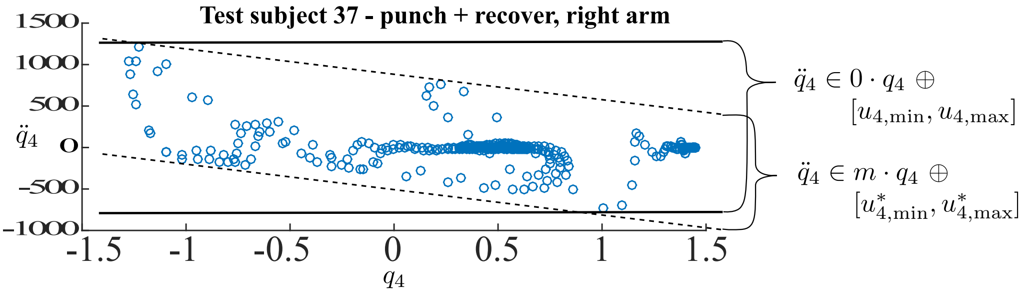


Figure 4.15: By accounting for the dependency of $\ddot{q}_4$ on q_4 (dashed lines), the range of uncertain inputs $[u^*_{\min}, u^*_{\max}]$ in the bottom model ($\ddot{q}_4 \in m \cdot q_4 \ \oplus [u^*_{4,\min}, u^*_{4,\max}]$) can be chosen to be smaller than $[u_{\min}, u_{\max}]$, as in the top model ($\ddot{q}_4 \in [u_{4,\min}, u_{4,\max}]$).

can limit the growth of the reachable sets. Observation of our data indicates some correlations between joint positions, velocities and accelerations, shown in Tab. 4.1, indicating that the range of accelerations could be expressed as a linear function of the joint positions and velocities. Indeed, dependence of the joint torque on the joint angles is well-known and has been studied for single-joint and multi-joint cases. In a single joint, the dependency is due to muscle physiology, as described in 4.6.1. In the multi-joint case, muscle-on-muscle impingement and intramuscular force transmission could cause dependencies.

Fig. 4.15 illustrates the dependency of the range of $\ddot{q}_4$ on q_4 during a punching movement. This dependency is intuitive, since only nonpositive acceleration is possible at the positive joint limit, and only nonnegative acceleration at the negative joint limit. Taking this into account reduces the range of $[\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}]$.

Our problem can be stated thus: given a set of training data consisting of time series of joint positions, velocities and accelerations, how can we find the coefficients of C in (4.3) such that $[\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}]$ is minimised, and all training data is consistent with (4.3)?

We consider each joint acceleration independently (since joint accelerations do not depend on other joint accelerations). We rearrange (4.3) to obtain:

$$\ddot{q}_i(t) \in \sum_{j=1}^{j=2k} C_{i,j} x_j(t) \oplus [u_{\min,i}, u_{\max,i}]. \tag{4.4}$$

Let us construct k datasets $\mathcal{S}_i, i \in \{1, ...k\}$, where the data points in each are the values of $[\boldsymbol{x}^\top_{corr_i}, \ddot{q}_i]^\top$ in the motion capture data. From these data sets, we wish to find $C_{i,1,...,k}$ such that $u_{i,\max} - u_{i,\min}$ is minimised[2], for each joint i. Equation (4.4) needs to hold for all elements $\boldsymbol{s}_i \in S_i$, so we rearrange it to show:

$$\begin{aligned} u_{\max,i} &\geq \{ \big[-C_{i,1...k} \ , \ 1 \big] \ \boldsymbol{s}_i | \boldsymbol{s}_i \in \mathcal{S}_i \}, \\ u_{\min,i} &\leq \{ \big[-C_{i,1...k} \ , \ 1 \big] \ \boldsymbol{s}_i | \boldsymbol{s}_i \in \mathcal{S}_i \}. \end{aligned} \tag{4.5}$$

[2] The notation $C_{i,1,...,k}$ is short for $[C_{i,1}, C_{i,2}, ..., C_{i,k}]$

We can easily find the minimal $u_{i,\max} - u_{i,\min}$ by reformulating the constraints in (4.5) as a linear programming problem, see [95, Ch. 2.17]. We made one modification, in that we only made the acceleration of each joint dependent on the three most highly-correlated state variables, shown in bold in Tab. 4.1.

4.3.9 A Combination of Linear Models

In addition to a prediction using acceleration as a function of the model state, we wish to account for limits on joint position, velocity and acceleration. In [96, Prop. 1], it is shown that, since the reachable sets $\mathcal{R}^{(1)}, ...\mathcal{R}^{(n)}$ from n overapproximative models $\mathcal{M}_1, ...\mathcal{M}_n$ all enclose the exact reachable set $\mathcal{R}^e$, then $\mathcal{R}^e \subseteq \bigcap_{i=1}^{n} \mathcal{R}^{(i)}$. Indeed, finding the intersection of 4 simple predictions is quicker and computationally more efficient than adapting the model in Sec. 4.3.8 to account for these limits in the prediction.

We compute the reachable sets from the following models:

1. a 0^{th} order model of maximum joint position:
   $\mathcal{R}_{\boldsymbol{q}}^{(1)}([0,t]) = [\boldsymbol{q_{inf}}\ ,\ \boldsymbol{q_{sup}}]$,

2. a 1st order model of maximum joint velocity:
   $\mathcal{R}_{\boldsymbol{q}}^{(2)}([0,t]) = \mathcal{Q}(0) \oplus [\boldsymbol{\dot{q}_{inf}}\ ,\ \boldsymbol{\dot{q}_{sup}}]t$,

3. a 2nd order model of maximum joint accelerations:
   $\mathcal{R}_{\boldsymbol{q}}^{(3)}([0,t]) = \mathtt{CH}(\mathcal{Q}(0)\, , \mathcal{Q}(0) \oplus \dot{\mathcal{Q}}(0)t \oplus \left[\boldsymbol{\ddot{q}_{inf}}\, , \boldsymbol{\ddot{q}_{sup}}\right]\frac{t^2}{2})$,

4. the 2nd order model $\mathcal{R}_{\boldsymbol{q}}^{(4)}([0,t])$ based on acceleration as a function of state, described in the previous section.

In the 1st and 2nd models, the reachable set of the time interval $[0,t]$ is identical to the reachable set of the final time, since reachable sets of future times enclose those of previous times: $\mathcal{R}_{\boldsymbol{q}}^{(1)}([0,t]) = \mathcal{R}_{\boldsymbol{q}}^{(1)}(t)$ and $\mathcal{R}_{\boldsymbol{q}}^{(2)}([0,t]) = \mathcal{R}_{\boldsymbol{q}}^{(2)}(t)$. This is not the case for the 3rd model; we must compute the convex hull (see Def. 3.3), indicated by the operator CH, of the initial set $\mathcal{Q}(0)$ and the reachable set at time t found using second-order motion dynamics. This encloses the reachable set of the interval, since it is a special case of [69, Prop. 3.3]). Since interval vectors can be quickly and easily intersected, we enclose this convex hull in a Cartesian product of intervals. Sets $\mathcal{R}_{\boldsymbol{q}}^{(1)}$ and $\mathcal{R}_{\boldsymbol{q}}^{(2)}$ are obtained as Cartesian products of intervals, and the set $\mathcal{R}_{\boldsymbol{q}}^{(4)}([0,t])$ is obtained as a zonotope, and is also enclosed in a Cartesian product of intervals. The intersection $\mathcal{R}_{\boldsymbol{q}}([0,t]) = \bigcap_{i=1}^{4} \mathcal{R}_{\boldsymbol{q}}^{(i)}([0,t])$ yields a tighter overapproximative interval of joint positions than any model on its own.

4.3.10 Accounting for the Elbow Singularity in the 4-DOF Model

Although the singularity at the shoulder can be avoided by clever placement of the first joint axis, there still exists a singularity at the elbow. This occurs when the elbow joint in the 4-DOF model is fully extended or fully flexed, i.e. $q_4 \approx -\frac{\pi}{2}$ or $q_4 \approx \frac{\pi}{2}$, meaning that q_3 (rotation around the long axis of the upper arm) is undefined [97]. Near this singularity, the inverse kinematics is affected: very small changes in Cartesian position lead to large changes in q_3, hence extremely high values of $\ddot{q}_3$ can be observed.

We account for these erroneously high values when building the dynamic model by ignoring the acceleration and velocity of the 3rd joint when $q_4 < -\frac{\pi}{2} + \epsilon$ or $q_4 > \frac{\pi}{2} - \epsilon$; we choose $\epsilon = 0.225 rad$.

4.3.11 Representation in Space

We use the method of Sec. 3.2 to convert the reachable set in the joint space to an HRO in Cartesian space. We showed in Sec. 3.6 that this is the tightest overapproximation when the joint space reachable set is large, and due to the unpredictable intention of the human as well as the fast dynamics that the human is capable of, the joint space reachable sets grow rather large even after a few milliseconds.

4.3.12 Accounting for Moving Shoulder Coordinate System

The RO is obtained in the shoulder coordinate system, however, this is not fixed: it rotates and translates as the human torso and shoulder complex moves. Motion studies on torso movement may determine whether the complex but limited movement may be best approximated by the enlargement of uncertainties in the simpler model, or whether additional degrees of freedom may be necessary, as in [79]. We take the former approach, and model the shoulder as a transforming coordinate system with limits on its translational and rotational acceleration.

The coordinate system of the shoulder is $T(t)$ at time t as shown in Fig. 4.16. Let the translational velocity of the coordinate system $T(t_0)$ be $\boldsymbol{b}$ and the angular velocity be ω around normalised axis $\boldsymbol{n}$. The maximum magnitude of the translational acceleration of the coordinate system is a and of the angular acceleration is α. Note that a maximum angular acceleration of magnitude α leads to a maximum linear acceleration of a point at radius r from the centre of rotation of magnitude αr.

We account for the change in coordinate system by adding, between T_0 and the first joint axis, an extra prismatic joint with axis in the direction of $\boldsymbol{b}$, followed by an extra rotational joint with axis $\boldsymbol{n}$. The joint positions at time t are then $\|\boldsymbol{b}\|(t - t_0)$ and $\omega(t - t_0)$ respectively. The kinematic chain is simply extended by two joints to account for the movement of the coordinate system, and the enlargement of the reachable occupancy is found using the method of Sec. 3.2, in the same way as for the other joints.

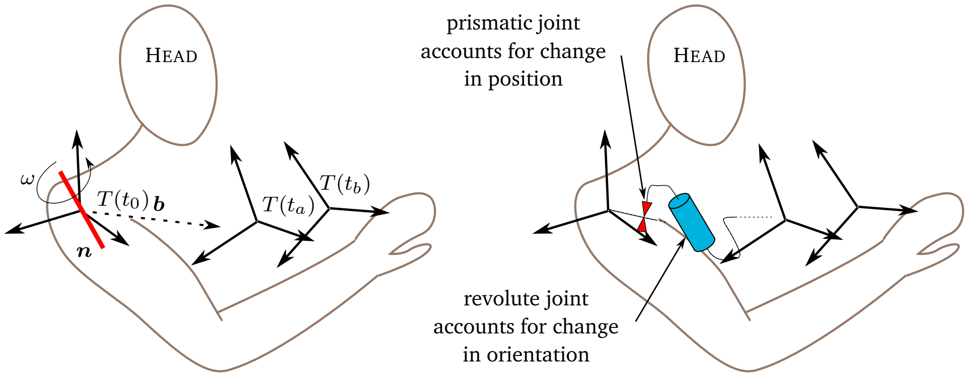

HEAD
ω
$T(t_0)$
$\boldsymbol{b}$
$\boldsymbol{n}$
$T(t_a)$
$T(t_b)$
prismatic joint accounts for change in position
HEAD
revolute joint accounts for change in orientation

Figure 4.16: Movement of the shoulder coordinate system is accounted for by adding a prismatic and revolute joint. Not shown: we account for rotational and translational accelerationsby augmenting the radii of the SSVs of the HRO.

To account for the fact that the translational and angular velocities of the coordinate system are not constant, but can change with translational and angular acceleration of maximum magnitude a and α respectively, we enlarge the radius of the SSVs at time t by the error term $(a+\alpha r_{\max})t^2$, where $r_{\max}$ is the farthest distance on each SSV to the shoulder. Over the interval $[t_a, t_b]$, the SSVs are enlarged by $(a + \alpha r_{\max})t_b{}^2$.

In this way, we have obtained a RO of the human arm as a union of SSVs by modelling the human arm as a kinematic chain of revolute joints on a floating base. We now present a method which does not use a kinematic chain.

4.4 Cartesian Space Approach

There are two main drawbacks to the joint-space approach presented in Sec. 4.3. Firstly, inverse kinematics is time-consuming, and during the training phase, one must account for singularities of the model in the inverse kinematics. Secondly, the HROs generated are complex SSVs with at least $2^2 \cdot 3^3 = 108$ points (3-DOF model) or $2^3 \cdot 3^5 = 1944$ points (4-DOF model). This makes collision-checking time-consuming, especially when there are multiple humans in the robot's workspace.

Contrasting with this, the Cartesian space approach calculates 3 HROs directly in the task space as unions of capsules and spheres, using 3 models of human movement. Since all 3 HROs are overapproximative, their intersection is also overapproximative [96, Prop. 1]. Each HRO is then verified against the RRO during the short-term plan.

Recall from Def. 3.4 the definition of the closed ball of radius r centred at $\boldsymbol{p}$ as $\mathcal{B}(\boldsymbol{p}; r) = \{\boldsymbol{x} \mid \|\boldsymbol{x} - \boldsymbol{p}\| \leq r\}$, and from Sec. 3.4.1 the operators $\texttt{BE}(\mathcal{B}_1, \mathcal{B}_2)$ and $\texttt{CE}(\mathcal{B}_1, \mathcal{B}_2)$, which output a ball and a capsule tightly enclosing the balls $\mathcal{B}_1$ and $\mathcal{B}_2$. The subscripts *S, E, W, U, F* and *H* refer to *shoulder, elbow, wrist, upper arm, forearm* and *hand*.

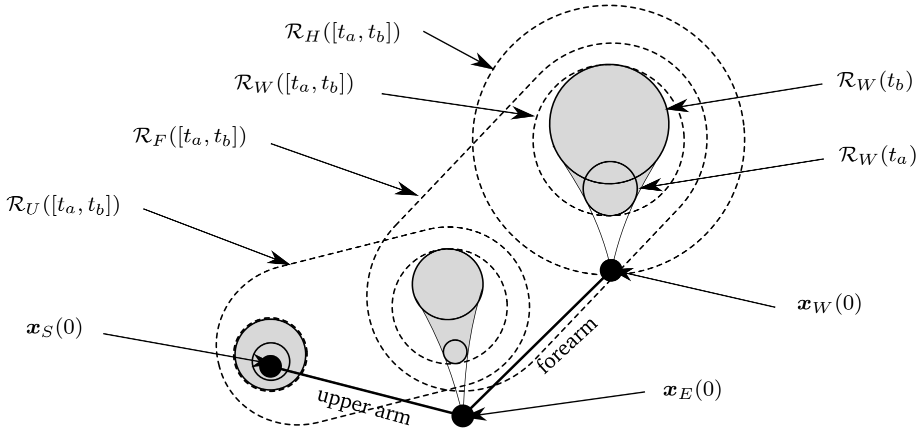


Figure 4.17: Occupancy Γ_{ACC} of the human arm over time interval $[t_a, t_b]$.

4.4.1 Model Using Acceleration Limits

The first model predicts human movement using a second order model. Consider that a point $\boldsymbol{y}$ with initial speed $\dot{\boldsymbol{y}}(0)$ and position $\boldsymbol{y}(0)$ can accelerate at maximally $a_{\boldsymbol{y},\max}$ in any direction, i.e. $\|\ddot{\boldsymbol{y}}\| \leq a_{\boldsymbol{y},\max}$. The position of $\boldsymbol{y}$ after time t is:

$$\boldsymbol{y}(t) = \boldsymbol{y}(0) + \dot{\boldsymbol{y}}(0) \cdot t + \int_0^t \int_0^{\tau'} \ddot{\boldsymbol{y}}(\tau) d\tau d\tau',$$

which one can bound by:

$$\mathcal{R}_{\boldsymbol{y}}(t) = B(\boldsymbol{y}(0); \delta y) \oplus B(\dot{\boldsymbol{y}}(0) \cdot t; \delta \dot{y} \cdot t) \oplus B(\mathbf{0}; \frac{a_{y,\max}}{2} \cdot t^2), \tag{4.6}$$

where δy and $\delta \dot{y} \in \mathbb{R}$ are the measurement uncertainties of position and velocity. $\mathcal{R}_{\boldsymbol{y}}(t)$ is the RO of $\boldsymbol{y}$ at time t. The RO over a time interval, $\mathcal{R}_{\boldsymbol{y}}([t_a, t_b])$, is enclosed in $\texttt{BE}(\mathcal{R}_{\boldsymbol{y}}(t_a), \mathcal{R}_{\boldsymbol{y}}(t_b))$. In this way the occupancy of the interval $\cup_{t=t_a}^{t_b} \mathcal{R}_{\boldsymbol{y}}(t)$ is enclosed; the proof follows from [98, Prop. V.2].

We calculate the RO of shoulder, elbow and wrist, $\mathcal{R}_S([t_a, t_b])$, $\mathcal{R}_E([t_a, t_b])$ and $\mathcal{R}_W([t_a, t_b])$, as in (4.6). The maximum accelerations $a_{\boldsymbol{S},\max}$, $a_{\boldsymbol{E},\max}$ and $a_{\boldsymbol{W},\max}$ are found from the parameterisation data in Sec. 4.2.3. Fig. 4.17 shows how the RO is built; we omit time dependency in the remaining derivation for clarity. The RO of the forearm $\mathcal{R}_F$ and of the upper arm $\mathcal{R}_U$ are capsules which enclose $\mathcal{R}_W$ and $\mathcal{R}_E$, and $\mathcal{R}_E$ and $\mathcal{R}_S$, respectively; $\mathcal{R}_F$, $\mathcal{R}_U$ are extended by the radius r_S of the capsule enclosing the upper arm and forearm in the occupancy of Sec. 4.2.1 (taken as $0.1m$), and $\mathcal{R}_W$ is extended by the radius of the sphere around the hand, r_H (taken

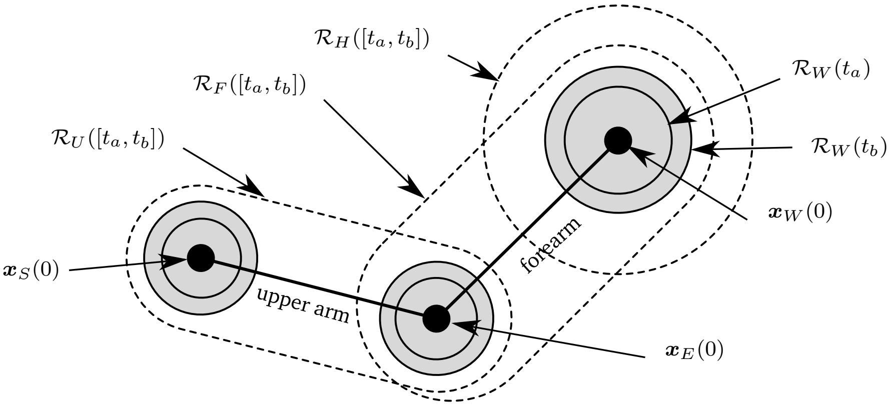


Figure 4.18: Occupancy Γ_{VEL} of the human arm over time interval $[t_a, t_b]$.

as $0.205m$).

$$\begin{aligned} \mathcal{R}_U &= \texttt{CE}(\mathcal{R}_S, \mathcal{R}_E \oplus B(\mathbf{0}; r_S)), \\ \mathcal{R}_F &= \texttt{CE}(\mathcal{R}_E, \mathcal{R}_W \oplus B(\mathbf{0}; r_S)), \\ \mathcal{R}_H &= \mathcal{R}_W \oplus B(\mathbf{0}, r_H). \end{aligned} \tag{4.7}$$

The HRO found by the first model, Γ_{ACC}, is the union of the capsules enclosing the forearm and upper arm and the sphere enclosing the hand:

$$\Gamma_{ACC} = \mathcal{R}_F \cup \mathcal{R}_U \cup \mathcal{R}_H. \tag{4.8}$$

4.4.2 Model Using Velocity Limits

The acceleration model of Sec. 4.4.1 does not account for bounds on velocity. We present a second Cartesian space model which uses only the position $\boldsymbol{y}(0)$ and the velocity bound $\|\dot{\boldsymbol{y}}\| \leq v_{\boldsymbol{y},\max}$. The equation of motion can be written as:

$$\boldsymbol{y}(t) = \boldsymbol{y}(0) + \int_0^t \dot{\boldsymbol{y}}(\tau) d\tau,$$

from which we obtain:

$$\mathcal{R}_{\boldsymbol{y}}(t) = B(\boldsymbol{y}(0); \delta y + v_{\boldsymbol{y},\max} \cdot t), \tag{4.9}$$

The parameters $v_{\boldsymbol{y},\max}$ are again found from the data in Sec. 4.2.3. As seen in Fig. 4.18, the RO at the future time t_b strictly encloses RO at previous times, therefore $\Gamma_{VEL}([t_a, t_b]) = \Gamma_{VEL}(t_b)$. Having calculated the RO of the shoulder, wrist and elbow from (4.9), Γ_{VEL} is then found from (4.7) and (4.8), only substituting Γ_{VEL} for Γ_{ACC} in (4.8).

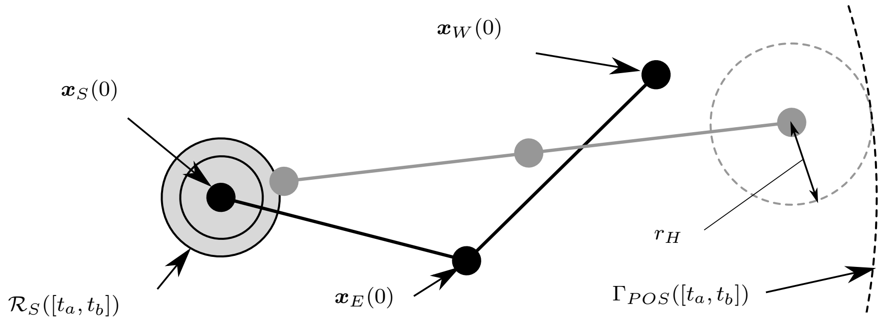


Figure 4.19: Occupancy Γ_{POS} of the human arm over time interval $[t_a, t_b]$.

4.4.3 Model Using Rigid Arm Structure

The third model, shown in Fig. 4.19, avails of the fact that the rigid bodies of the arm have constant length. The RO cannot extend from the shoulder more than the length of the upper arm (distance from shoulder to elbow) plus the length of the forearm (elbow to wrist), plus the length of the hand, r_H. The movement of the shoulder can be accounted for using the velocity model described previously. We therefore have:

$$\begin{aligned}\Gamma_{POS}([t_a,t_b]) = B(\boldsymbol{x}_S(0); v_{S,\max}\cdot t + \|\boldsymbol{x}_S(0) - \boldsymbol{x}_E(0)\| \\ + \|\boldsymbol{x}_E(0) - \boldsymbol{x}_W(0)\| + \delta y + r_H)\end{aligned}$$

4.5 Evaluation of Models

We evaluate the predictions of the human in a variety of ways. The requirements of the predictions is that they be *conservative, tight* and *quick to compute*. We test the conservativeness in two ways: we perform extensive conformance checking [40] on human movements which might occur in human-robot interaction (HRI) situations, and we explore the space of physically possible movements using a biomechanical model. The tightness we evaluate by comparing the volume of the ROs over a range of prediction horizons. The computation time we evaluate on a real-time machine and we also compare the computational complexity of the algorithms. We report the results from [185], which compare the 3-DOF model with the 4-DOF model; and from [184], comparing the joint space (JS) approach using the 4-DOF model with the Cartesian space (CS) approach.

4.5.1 Comparison of Joint-Space Models

The data we use for validation is from the publicly available Motion-Capture Database from Carnegie Mellon University Graphics Lab [3]. The movements we use are grouped into four categories:

[3] Available `mocap.cs.cmu.edu`, accessed 11.08.15.

1. *Everyday* motions e.g. construction work, machining work, manipulating objects, stumbling. (96 files)
2. *Sports-related* motions e.g boxing, throwing/shooting balls and batting balls. (67 files)
3. *Dance-related* motions e.g swing dance, Indian dance and modern dance. (58 files)
4. *Acrobatic* motions, i.e. any motion where both feet are in the air simultaneously, including jumps, cartwheels, backflips and swings from a trapeze. (68 files)

The *Everyday* motions are expected to be similar to usual movement in a HRCoex scenario; we add *Sports-related*, *Dance-related* and *Acrobatic* movements to test the prediction's limits.

The computations were run on MATLAB R2016b using a 2.8GHz Intel i7 processor with 16GB RAM. The toolbox CORA [66] was used for the reachability analysis, the toolbox INTLAB [65] for interval computations, the Robotic Toolbox [99] to generate the robot kinematics and dynamics, and the MPT toolbox [100] to visualise and compute the volume of SSVs. We estimate joint position uncertainties at $\pm 0.04 rad$ for revolute joints and $\pm 0.001 m$ for both prismatic joints, and the length uncertainty of the forearm and upper arm; we also took joint velocity uncertainty to be $\pm 0 \frac{rad}{s}$ and $\pm 0 \frac{m}{s}$ for revolute and prismatic joints.

We also make one simplification in [185], which is to enclose the sphere of the hand $\mathcal{S}_H$ and the capsule of the forearm in one larger capsule, as described in Sec. 4.3.1. This reduces the number of volumes to collision-check and speeds up verification time.

Since we also wish to show that the prediction is a *tight* over-approximation, i.e., that they enclose all human movement relevant to HRCoex scenarios and that smaller occupancies would not, we evaluate the following predictions:

1. 3-DOF model, henceforth denoted Γ_{3DOF}
2. 4-DOF model, denoted Γ_{4DOF}
3. ISO-based prediction as in Sec. 4.2.2, denoted Γ_{ISO}
4. 3-DOF model where the range of velocities and accelerations is reduced by half, denoted $\Gamma_{3DOF,50\%}$
5. 4-DOF model where the range of velocities and accelerations is reduced by half, denoted $\Gamma_{4DOF,50\%}$

4.5.1.1 Conformance Checking

We performed conformance checking [40] on our model against a publicly available database of human movement, in order to test whether the predictions from our models, parameterised on extreme movement data from 38 subjects, also accounts for unseen subjects and movements.

We filter the joint positions with a Kalman filter with the state $\boldsymbol{x} = [\boldsymbol{q}^\top, \dot{\boldsymbol{q}}^\top]^\top$, error covariance a diagonal matrix of $0.001 rad$ for rotational DOFs and $0.0005 m$ for the 3rd (prismatic)

Table 4.2: Number of movement files conformant, i.e. all markers are inside occupancy at all times during movement (**bold** indicates that all movement in the category are conformant)

Movements	*Everyday*	*Sport*	*Dance*	*Acrobatics*
$\Gamma_{3DOF}([0, 16.7]ms)$	**96**	55	66	50
$\Gamma_{4DOF}([0, 16.7]ms)$	**96**	57	66	57
$\Gamma_{ISO}([0, 16.7]ms)$	**96**	43	62	23
$\Gamma_{3DOF,50\%}([0, 16.7]ms)$	95	55	66	48
$\Gamma_{4DOF,50\%}([0, 16.7]ms)$	95	56	66	53
$\Gamma_{3DOF}([0, 33.3]ms)$	**96**	53	**67**	49
$\Gamma_{4DOF}([0, 33.3]ms)$	**96**	57	**67**	57
$\Gamma_{ISO}([0, 33.3]ms)$	90	15	40	13
$\Gamma_{3DOF,50\%}([0, 33.3]ms)$	95	52	63	40
$\Gamma_{4DOF,50\%}([0, 33.3]ms)$	93	50	**67**	46
$\Gamma_{3DOF}([0, 50.0]ms)$	**96**	53	**67**	49
$\Gamma_{4DOF}([0, 50.0]ms)$	**96**	57	**67**	59
$\Gamma_{ISO}([0, 50.0]ms)$	81	7	23	13
$\Gamma_{3DOF,50\%}([0, 50.0]ms)$	95	51	63	37
$\Gamma_{4DOF,50\%}([0, 50.0]ms)$	93	49	65	43

joint in the 3-DOF model; process covariance a diagonal matrix of zeros for the position part of the state and $0.1\frac{rad}{s}$ for the velocity part for the rotational DOFs ($0.05\frac{m}{s}$ for the prismatic DOF).

We then calculate the reachable sets during a time interval $[0, t_f]$ in the future and evaluate the reachable occupancies. We check whether the markers at the arm at time t_f in the future are all contained in the reachable occupancy, for each timestep of the motion capture file, and for both arms. Using simple distance checks between markers on rigid bodies, we determine when markers are incorrectly tracked and exclude these markers in these frames from our analysis. The values of t_f we test are $t_f = 16.7ms$, $33.3ms$, and $50ms$. Since the test data is sampled at $60Hz$ and $120Hz$, t_f corresponds to an integral number of timesteps in all data files.

The results are shown in Tab. 4.2. This table shows that Γ_{3DOF} and Γ_{4DOF} correctly enclose all movement in the *Everyday* dataset. In the other datasets, some movements fall outside the prediction volume. The arm dynamics of *Sports-related* or *Acrobatic* movements are faster than those of the movements used in the parametrisation (Sec. 4.2.3). Furthermore, reducing the range of velocities and accelerations by half, as in $\Gamma_{3DOF,50\%}$ and $\Gamma_{4DOF,50\%}$, does not include all movements even in the *Everyday* dataset. Fig. 4.20 shows a movement from the everyday data which falls outside $\Gamma_{3DOF,50\%}$ and $\Gamma_{4DOF,50\%}$ but is contained in Γ_{3DOF} and Γ_{4DOF}.

The ROs Γ_{3DOF} and Γ_{4DOF} are therefore tight in the sense that the parametrisation from the archetypal movements suffices to enclose relevant movement but is not excessive, since

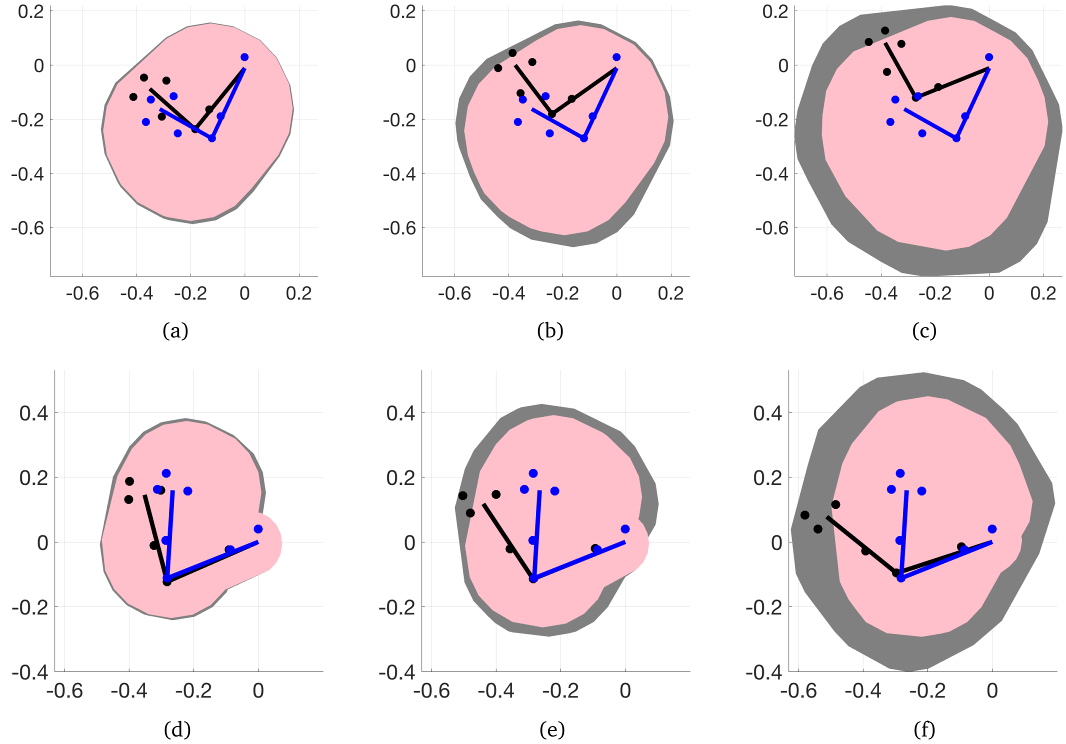


Figure 4.20: Predictions from: (a–c) the 3-DOF model and (d–f) the 4-DOF model, for movements from the *Everyday* dataset. The predictions are for the time interval $[0, t_f]$ where $t_f = 16.7ms$ (a, d), $t_f = 33.3ms$ (b, e) and $t_f = 50.0ms$ (c, f), from left to right. Filled circles represent markers on the arm and lines represent the forearm and upper arm. Blue circles and lines are the positions at $t = 0$; black is at $t = t_f$. All markers lie within the grey ROs (calculated with parameters from Sec. 4.2.3), but do not always lie within the pink ROs (calculated with parameters from Sec. 4.2.3 with the range reduced by 50%). This illustrates that the parameters we obtain from the extreme movement data are not overly conservative. Origin at shoulder; scale in metres.

Table 4.3: Volume comparison over a sample of data. Mean μ; standard deviation σ; number of samples $= 20$ (volumes in m^3)

		3-DOF	3-DOF (from [186])	4-DOF	4-DOF (from [186])	ISO
$\Gamma([0, 16.7]ms)$	μ	0.199	0.201	0.107	0.110	0.086
	σ	0.024	0.024	0.009	0.009	0.003
$\Gamma([0, 33.3]ms)$	μ	0.273	0.278	0.223	0.239	0.116
	σ	0.041	0.041	0.026	0.028	0.004
$\Gamma([0, 50.0]ms)$	μ	0.480	0.488	0.537	0.555	0.151
	σ	0.086	0.087	0.075	0.080	0.006

reducing the range of velocities and accelerations causes the prediction to fail. Finally, the ISO-based prediction Γ_{ISO}, while simple and based on the currently accepted standards, does not estimate human motion conservatively in any category.

If we were to include *Acrobatic* motions in the archetypal motions used in the parameterisation, our predictions might also account for such movements. The RO would be larger in this case, and the verifier would be more cautious, at the expense of robot efficiency. Since acrobatic-like movements are generally prohibited by guidelines for workshop behaviour, we argue that this range of motion can be safely ignored (while unexpected but everyday behaviour such as reflex movements must still be taken into account). Determining exact set of movements used in parameterisation of the models is a matter for legislators and standards committees.

4.5.1.2 Volume of Predictions

In Tab. 4.3, we evaluate the volume of the arm ROs over a subset of the data, and compare them to each other, and also to the approach from [186], where the reachable set of joint values is calculated with only the first 3 models in Sec. 4.3.9 and not the reachability analysis of the linear system of acceleration as a function of state. This results in a less tight reachable occupancy, but adds additional computation. Fig. 4.21 illustrates the growth of the RO with t_f.

It can be seen that the volume of Γ_{ISO} is much smaller than the other ROs. The prediction Γ_{4DOF} is a tighter overapproximation at shorter prediction horizons than Γ_{3DOF}, although this advantage decreases with increased prediction horizon t_f. This can be partially explained by the fact that the enclosure of revolute joints as in [60] (Sec. 3.2) introduces overapproximation, and there are 2 more revolute joints in the 4-DOF model as the 3-DOF model. This overapproximation can be reduced by using more points to overapproximate the arc of movement of the revolute joint as suggested in [60, eq. 26], however, this increases the number of defining points of the SSV and hence adds to computation time for generating the RO and collision-checking with the RRO.

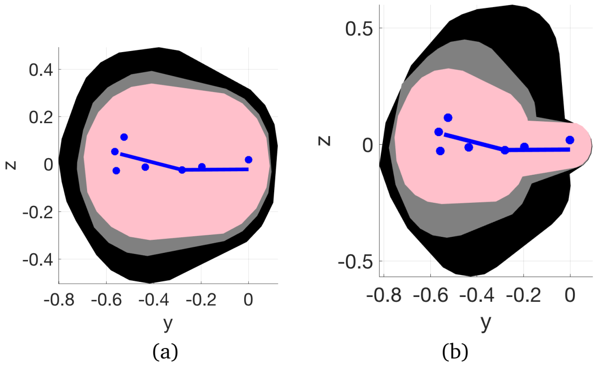


Figure 4.21: The growth of the ROs in (a) the 3-DOF and (b) the 4-DOF model. The time intervals for the prediction are $[0, 16.7]ms$ (pink), $[0, 33.3]ms$ (grey) and $[0, 50.0]ms$ (black). Filled circles represent markers on the arm and lines represent forearm and upper arm, at time $t = 0$. Origin at shoulder; scale in metres.

The 3-DOF model is better able to account for uncertainty in joint lengths as mentioned in Sec. 4.2.3; the 4-DOF model requires extra degrees of freedom to account for parametric uncertainty in the rigid bodies of the arm.

We also see in Tab. 4.3, that availing of the extra reachable set prediction using acceleration as a function of state improves the tightness. Of course, a less tight overapproximation means more false-positive collisions are detected and the efficiency of the robot is reduced.

4.5.1.3 Computation Time on Non-Real-Time Machine

We compared the computation times on the non-real time machine on which the conformance checking was evaluated, a Macbook Pro running MATLAB R2016b on a 2.8GHz Intel i7 processor with 16GB RAM. The calculation of the reachable set using the model of acceleration as a function of state takes longer than the other three models. One could consider an anytime algorithm, where several reachable sets from different models are calculated in parallel, and the intersection of only those predictions calculated in time are used to compute the reachable set. Hence, if the reachability analysis in the 4th model takes too long, only the first 3 models are used; if the 4th model is computed in time, it is also used to tighten the overapproximation of the reachable set.

The dependency of the computation of $\Gamma(t_f)$ on the prediction horizon in Tab. 4.4 is due to the fact that the number of support points in the conversion to the reachable occupancy using the method of Sec. 3.2 is set dynamically from the size of the interval in the reachable set. An interval of less than $\frac{\pi}{2}$ has one support point, less than π has two, less than $\frac{3\pi}{2}$ and 6 otherwise. At longer time horizons, the reachable set is larger and therefore more support points are used, increasing the computation time. In principle, if the number of support points is fixed, the

Table 4.4: Computation times on non-real-time machine (ms)

Model	obtain $\mathcal{X}_0$	obtain $\mathcal{R}_X(t_f)$	obtain $\Gamma(t_f)$	Total
$\Gamma_{3DOF}([0, 16.7]ms)$	0.27	0.76	0.37	1.41
$\Gamma_{3DOF}([0, 33.3]ms)$	0.28	0.77	0.38	1.42
$\Gamma_{3DOF}([0, 50.0]ms)$	0.27	0.77	0.38	1.42
$\Gamma_{4DOF}([0, 16.7]ms)$	0.33	0.81	1.68	2.82
$\Gamma_{4DOF}([0, 33.3]ms)$	0.33	0.81	1.93	3.07
$\Gamma_{4DOF}([0, 50.0]ms)$	0.33	0.81	2.93	4.08

number of computations is independent of the time horizon and predictions to several time horizons could be computed in parallel.

Finally, upper-body movement and movement in the shoulder complex contribute to overall motion of the arm. Although we only calculated the ROs in the shoulder coordinate system, the conversion to the world coordinate system should be evaluated. The next comparison uses the 4-DOF arm prediction in world space.

4.5.2 Comparison of Joint-Space versus Cartesian-Space Methods

Here we compare the CS methods with the JS method using the 4-DOF arm model. In this case, we compute the prediction in the world space including the overapproximation in Sec. 4.3.12. To distinguish this from Γ_{4DOF} in the previous section we call the prediction Γ_{JS}.

For a fairer comparison, we did not filter the time series of the joint values after obtaining them through inverse kinematics from the time series of the marker positions. Instead, we first filtered the time series of the marker positions, velocities and accelerations (the velocities and accelerations were found using numerical differentiation) with a Kalman filter. We took the error covariance matrix to be a diagonal matrix of $0.0005m$ and process covariance to be a diagonal matrix of zeros for the position and velocity parts of the state, and $1000\frac{m}{s^2}$ for the acceleration part of the state. We found that these parameters eliminated noise, without unduly attenuating the accelerations and velocities. The parameters for the CS approach, $v_{\boldsymbol{S},\max}$, $v_{\boldsymbol{E},\max}$, $v_{\boldsymbol{W},\max}$, $a_{\boldsymbol{S},\max}$, $a_{\boldsymbol{E},\max}$ and $a_{\boldsymbol{W},\max}$, were then obtained as the maxima and minima of these time series. The method for obtaining the joint position, velocity and acceleration limits is described in Appendix. A.5.

We used a database which was a subset of that of the study in Sec. 4.5.1, in order to focus on and examine individual, HRI-relevant sets of movements. From the database we selected 8 subjects performing the following movements:

1. subject 62: construction work/random (25 motions);
2. subject 70: carrying a suitcase; (10 motions);

3. subject 80: selected everyday motions (44 motions);
4. subject 82: jumping, pushing, banging (10 motions);
5. subject 76: swatting at bug (1 motion);
6. subject 94, Indian dance (16 motions);
7. subject 102, basketball (32 motions);
8. subject 124, sport-related motions (13 motions)

The first 5 subjects performed everyday motions, which is expected to be similar to motion experienced in a factory setting. The latter 3 subjects performed motion not expected in a factory setting, in order to test the limits of the prediction.

4.5.2.1 Conformance Checking

For each arm and for every timestep of the movement data file, we predict the RO at the time intervals $[t_i, t_f] = [8.3, 16.7]; [25.0, 33.3]; [41.7, 50.0]$ and $[58.3, 66.7]$ milliseconds in the future, where t_i and t_f are counted from the current timestep. We then check that all markers on the arm at time t_f are contained within the RO (again, the evaluation data was captured at $60Hz$ and $120Hz$, therefore t_f corresponds to a whole number of timesteps in all data files, meaning that markers at t_f are can be evaluated for inclusion in the reachable occupancy. If that is the case for the entire movement file, we say that the conformance checking has passed in Tab. 4.5. In the CS approach, we consider the arm contained only if it is contained in all of Γ_{ACC}, Γ_{VEL}, and Γ_{POS}. Measurement uncertainties of the position were taken to be $0.002m$ and $0.01rad$, and those of the velocity were taken to be $0.02\frac{m}{s}$ and $0.1\frac{rad}{s}$.

Tab. 4.5 shows the conformance-checking for both approaches, as well as the model Γ_{ISO} from Sec. 4.2.2. We see that the Γ_{ISO} does not account for several of the everyday movements, whereas both JS and CS approaches do.

4.5.2.2 Volume of Predictions

We compare the average volumes of the ROs over a sample of the data set in Tab. 4.6. With our parameterisation, Γ_{ACC} is tighter than Γ_{VEL} or Γ_{POS}. This does not mean, however, that the latter ROs are redundant. Cases may arise, during high-speed movements, where Γ_{VEL} is verified safe and Γ_{ACC} is not, since the acceleration model does not account for the fact that the arm cannot accelerate beyond its velocity limits. Furthermore, at prediction horizons greater than around 50, both Γ_{ACC} and Γ_{VEL} become extremely large, extending far away from the shoulder. The prediction Γ_{POS}, which incorporates the constraint of a fixed total arm length, can help verify the trajectory at longer prediction horizons, for example if the sensors are temporarily occluded and historic sensor data needs to be used for the prediction.

Table 4.5: Number of movement files conformant, i.e. all markers are inside occupancy at all times during movement (**bold** indicates that all movement in the category are conformant)

Dataset (total movement files)	JS approach	CS approach	ISO approach
1 (25 files)	**25**	**25**	**25**
2 (10 files)	**10**	**10**	9
3 (44 files)	**44**	**44**	35
4 (10 files)	**10**	**10**	7
5 (1 files)	**1**	**1**	0
6 (16 files)	**16**	**16**	4
7 (32 files)	3	25	0
8 (13 files)	8	0	0

Table 4.6: Volume comparison over a sample of data. Mean μ; number of samples $= 20$ (volumes in m^3)

Prediction interval (ms)	Γ_{JS}	Γ_{ACC}	Γ_{VEL}	Γ_{POS}	Γ_{ISO}
$[8.3, 16.7]$	0.08	0.08	0.44	2.05	0.09
$[25.0, 33.3]$	0.24	0.21	1.51	2.39	0.11
$[41.7, 50.0]$	0.69	0.64	3.87	2.76	0.15
$[58.3, 66.7]$	1.88	2.17	7.98	3.17	0.19

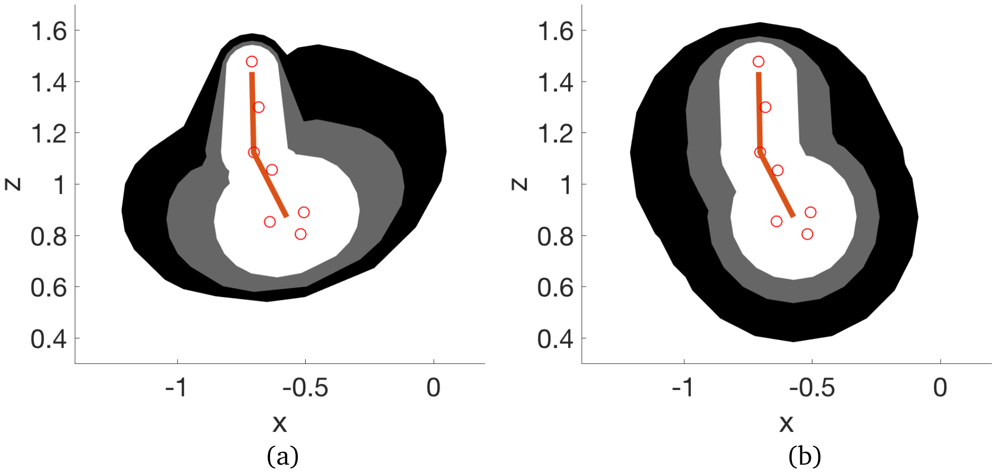


Figure 4.22: The growth of the arm occupancy for (a) Γ_{JS} and (b) Γ_{ACC}. The shoulder is at the top and the hand is at the bottom, the projections of $\Gamma([8.3, 16.7])$ in white, $\Gamma([25.0, 33.3])$ in grey and $\Gamma([41.7, 50.0])$ in black are underlaid. Left: Γ_{JS}, right: Γ_{ACC}.

At prediction interval $[58.3, 66.7]ms$, the JS prediction starts to outperform the others in terms of tightness, since it inherently accounts both for the constant size of the body parts as well as the maximum joint velocities, positions and accelerations. For prediction horizons under $50ms$, however, the volume of Γ_{JS} is comparable with that of Γ_{ACC}. We found that accounting for the moving base of the shoulder as in Sec. 4.3.12 had a significant effect on the volume of Γ_{JS}, increasing it sometimes by up to 40% of that of the RO prior to that step (i.e. in the shoulder coordinate system). This strongly suggests that better movement models of the shoulder coordinate system could be developed and used. A kinematic parameterisation of the shoulder complex is presented in [81], for example, and for the spine and torso in [101]. A projection of the ROs Γ_{JS} and Γ_{ACC} is in Fig. 4.22.

4.5.2.3 Computation Time

We compare the computation times of the JS approach with the CS approach on a Speedgoat real-time target machine running real time Simulink 2015b, in Tab. 4.7. While both were faster than $1ms$, the CS approach was better: generation and collision-checking of complex SSVs in the JS approach using the iterative Gilbert-Johnson-Keerthi (GJK) algorithm takes longer than the efficient capsule-capsule collision-checking in the CS approach from [67]. The inverse kinematics and accounting for movement of the shoulder coordinate system also slow calculation.

Table 4.7: Average execution time on a real-time target machine during $720s$ of operation (μs)

Approach used	CS approach	JS approach
Plan and send joint positions	352	352
Compute ROs and verify	4	124
Total	356	476

4.6 Exploration with Biomechanical Model

Validating the predictions against recorded movements gives an indication as to whether the predictions include the movements they were designed to cover, i.e. unexpected use and expected misuse in a HRCoex environment. This should include unexpected movements such as reflex movements and catching falling objects, but, by their very nature, these are hard to replicate and even harder to ensure that all situations are considered. In order to test the predictions more thoroughly, one can use a biomechanical model to explore the movement space, in order to try to find movements which are physically possible which invalidate the prediction. This chapter is based on the work presented in [187].

Digital Human Models (DHMs) such as the Delft Shoulder and Elbow model [24] and the model of Holzbaur *et al.* [25] can accurately model the musculoskeletal system of humans from measured and clinical data and have a wide range of industrial and research applications in the field of ergonomics, orthopaedics, rehabilitation and sport science. In this work, we use the model of [25], shown in Fig. 4.23 since it is implemented on nonproprietary software and is widely used in the biomechanics literature for human motion prediction, e.g. [80, 102]. We next elucidate the dynamics of this model.

4.6.1 Dynamics of Biomechanical Model

The biomechanical model contains 12 rigid bodies, 11 of which are bones of the arm and torso and 1 is the complete hand. The 50 muscles receive an individual input, or *excitation*, between 0 and 1 and are the actuators in the model, exerting forces on the rigid bodies.

The muscle dynamic model is from Schutte [104]. The muscle force is a nonlinear function of its length (itself a function of the kinematics of the arm) and its *activation*. The excitation $u \in [0, 1]$ is the input to the muscle, and influences the activation $a(u, t)$ through the first order dynamics:

$$a = \int \frac{u - a}{\tau(a, u)} dt, \tag{4.10}$$

$$\tau(a, u) = \begin{cases} (0.5 + 1.5a)t_{act}, & \text{for } u > a, \\ (0.5 + 1.5a)^{-1} t_{deact}, & \text{for } u \leq a. \end{cases} \tag{4.11}$$

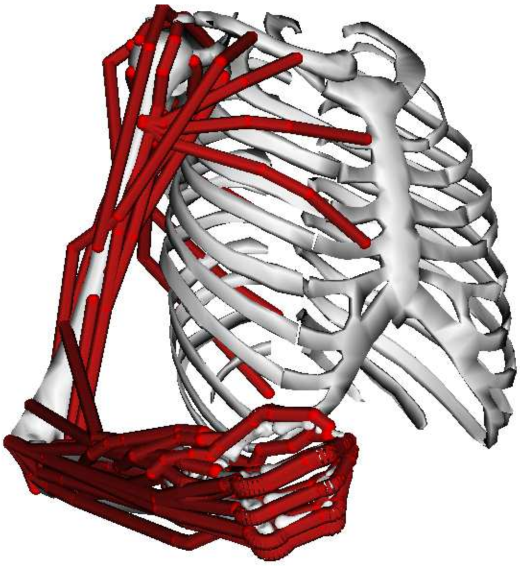


Figure 4.23: Screen capture from OpenSim biomechanical simulation [103]

For this muscle model ([104]), $t_{act} = 7.667\,ms$ and $t_{deact} = 1.460\,ms$; the resulting activation curve is depicted in Fig. 4.24.

These complex dynamics mean that muscles at rest cannot immediately exert their maximum force, nor can fully activated muscles stop pulling at once. Additionally, the muscles and the tendons are elastic and exert a passive contractile force when stretched above their slack length. The active force which the muscles exert is a function of the activation, the length of the muscle and its rate of change of length, the length of the tendon, and the *pennation angle*, i.e. the angle at which the muscle fibres are attached to the tendon. The reader is referred to [104] for a more detailed description of the muscle model and its biological background.

The biomechanical model has 7 rotary DOFs (see Fig. 4.25):

- *elv_angle (1), shoulder_elv (2), shoulder_rot (3)* at the shoulder,
- *elbow_flexion (4)* at the elbow,
- *pro_sup (5)* the pronation/supination of the forearm,
- *deviation (6), flexion (7)* at the wrist,

The orientation of the torso can also be changed, but is not actuated by any muscles, hence we set this such that the torso is vertically aligned (i.e. the human is standing upright and not moving).

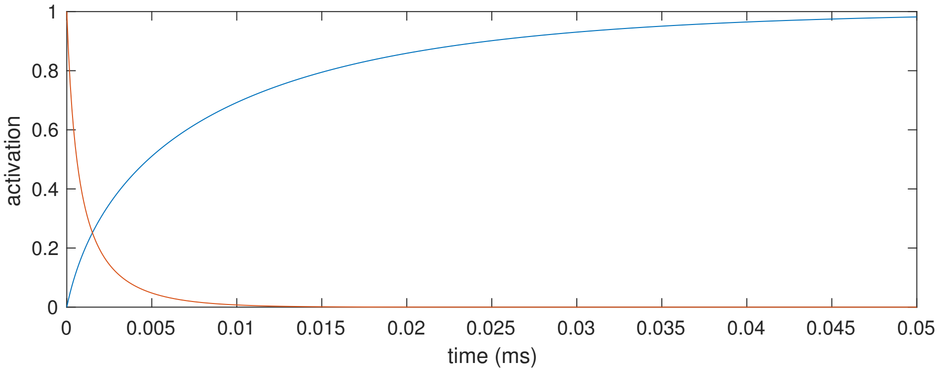


Figure 4.24: Activation profile (blue) and deactivation profile (orange) of a muscle with initial activation at 0 and 1 respectively.

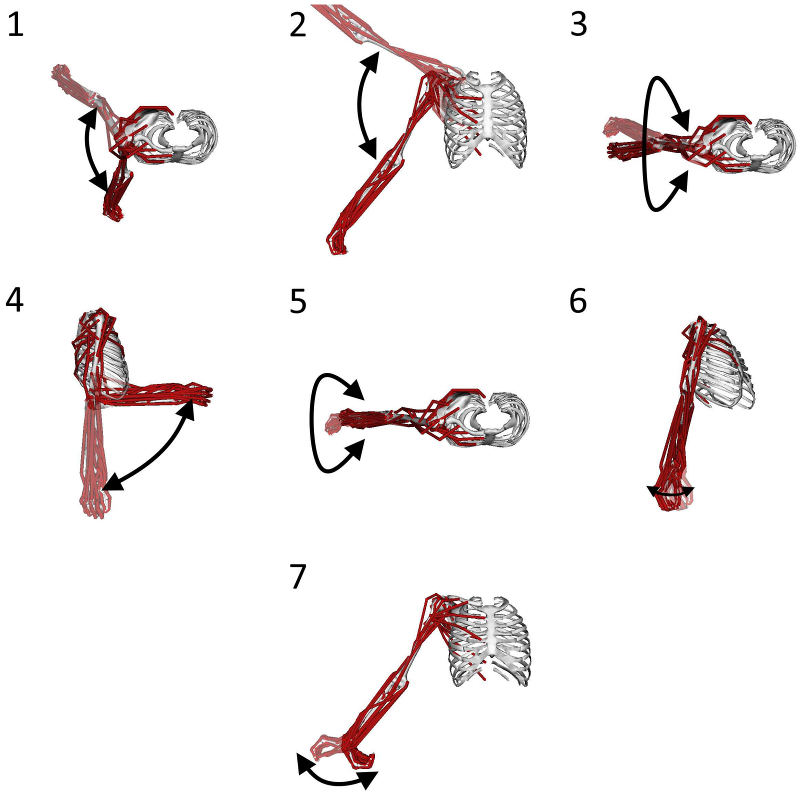


Figure 4.25: Degrees of freedom in the biomechanical model. Images from OpenSim [103]

4.6.2 Exploration Assumptions

The DHM's input space is continuous and high-dimensional. In order to simplify the exploration process, we make some assumptions on the input to the model (i.e. the muscle excitations) and the simulation itself.

(a) We restrict the excitation of individual muscles to either 0 or 1, though the DHM accepts any floating point number in $[0, 1]$.

(b) Only the positions at the beginning and end of any potential simulation are considered when computing the occupancy; potential extrema at other times during the simulation are ignored.

(c) Muscle excitations stay constant during step 2 of our method (see below), until the time horizon t_f.

4.6.3 Method

Challenges to finding extreme movements with biomechanical models are:

1. Prediction is expensive in computation and memory. Exploration for the most extreme movements should be as efficient as possible in terms of the number of simulations performed.

2. The input to the biomechanical model is the excitations of each muscle; there are 50 muscles in the model from [25]. In addition to the large input dimensionality, the force exerted by the muscle is a nonlinear, dynamic function of the muscle state, so finding the input for extreme movements is not a trivial task.

Addressing the first challenge: exploring high-dimensional state spaces efficiently requires clever sampling methods. Deterministic sampling methods such as sampling from a lattice in the input space can be less efficient at higher dimensions as the number of required samples grows exponentially. Random sampling methods can often excel here. An exploration algorithm that has been used widely and successfully is the rapidly-exploring random tree (RRT), which biases the sampling towards unexplored areas of the search space, hence exploring the input space faster. However, random sampling is not superior to deterministic sampling in every case [105]. As well as deterministically sampling from a lattice, samples can be derived from some criteria of optimality, which reward useful samples [106].

In our scenario, we may wish to use information about the biomechanical structure of the arm to accelerate our search for extreme movements. For example, the *long triceps*, *lateral triceps* and *medial triceps* all act to extend the arm, hence exciting/not exciting these three together would make sense when searching for extreme movements. By clustering the muscles into groups which act similarly and exciting each group together, and limiting the excitation of each muscle to 0 (no excitation) and 1 (full excitation), we can exhaustively explore a smaller input space and create a "scaffold" tree for an RRT algorithm to further explore. For example, for 10 groups, we have $2^{10} \approx 10^{3}$ combinations of excitations, which is possible to simulate exhaustively, whereas simulating all combinations of each of the 50 muscles yields $2^{50} \approx 1.1 \cdot 10^{15}$ combinations, which is infeasible to simulate exhaustively.

To address the second challenge of nonlinear dynamics, we adapt the approach of [107], in which a nonlinear, hybrid system is linearised to compute an approximation to the optimal input to reach a desired state. The 3 steps in our method are summarised below:

1. *Reduce input space dimension.* We cluster similar muscles using k-means clustering to produce m_{groups} groups of muscles. This clustering is performed once only.

2. *Initial tree of explored states (scaffold).* For each different starting state, we perform deterministic sampling of the groups obtained in the previous step, to quickly obtain likely extreme movements. Given a prediction horizon t_f, we divide this into n equal timesteps. For each input $u \in \{0,1\}^{m_{groups}}$, we simulate the arm from the initial position to times $\frac{t_f}{n}, 2\frac{t_f}{n}, ..., t_f$, and construct a "scaffold", i.e. a tree of states (see Fig. 4.26)[4].

3. *Explore with RRTs.* We start from the scaffold found in the previous step, and use RRTs to efficiently explore the state space further. This accounts for cases of extreme movement where the inputs change over the simulation time interval.

The combination of the grouping of the muscles to build a scaffold for the exploration tree (detailed in Sec. 4.6.4) and the RRT including linearisation of the dynamics (detailed in Sec. 4.6.6) results in an exploration which surpasses random sampling in efficiency. The next sections elaborate on each of these steps.

4.6.4 Muscle Clustering

We quantified the effects of each muscle acting individually, and grouped muscles with similar effects together. Alg. 9 shows the approach. The number of inputs, i.e. muscles, is m_{inputs}; $\mathbf{0}$ represents a vector of zeros with length m_{inputs}; $\boldsymbol{u}_i \in \mathbb{R}^{m_{inputs}}$ means that the excitation of the i^{th} muscle is 1, and that of the rest is 0. The function $\texttt{simulate}(\boldsymbol{s}, \boldsymbol{u}, t)$ outputs the state after a simulation of length t starting from state $\boldsymbol{s}$ with input $\boldsymbol{u}$. The state space comprises 7 joint angles, muscle fibre lengths and the rate of change of these lengths, muscle activations and the simulation time elapsed.

[4] $\{0,1\}^n$ is shorthand for $\{0,1\} \times ...$ (n times) $... \times \{0,1\}$

Algorithm 9 Muscle clustering

```
$\mathbf{a} \leftarrow simulate(\boldsymbol{s}_{init}, \mathbf{0}, 100ms)$
for $i < m_{inputs}$ do
    $\boldsymbol{c}_i \leftarrow simulate(\boldsymbol{s}_{start}, \boldsymbol{u}_i, 100ms) - \mathbf{a}$
    $\hat{\boldsymbol{c}}_i \leftarrow \frac{\boldsymbol{c}_i}{\|\boldsymbol{c}_i\|}$
perform k-means clustering
```

We performed weighted k-means clustering on the normalised differences of the joint angles to the joint angles obtained when simulating with an input of **0** for each muscle. The distance metric is a weighted Euclidean distance in the joint angle space; the first 4 joints have weight 1 and the last 3 (i.e. supination/pronation and flexion/extension of the hand, rotation of the wrist) are weighted 0.01, since they contribute very minimally to the change in occupancy. We chose the number of clusters as $m_{groups} = 10$.

4.6.5 Construction of Scaffold Tree

The activation dynamics mean that the muscles achieve peak forces when activation is high, meaning that the excitations must be high continuously for several milliseconds (illustrated in Fig. 4.24). The purpose of the scaffold is to try and find extreme cases very fast, by keeping excitations the same until the prediction horizon t_f. This is the rationale for simplification (c) in Sec. 4.6.2.

With only 2^{10} inputs from the 10 groups, it becomes feasible to simulate each input combination. We construct the scaffold to be a starlike tree[5], where the starting state of the arm is the root vertex, and the leaves are the states after t_f, see Fig. 4.26.

4.6.6 Rapidly-Exploring Random Trees

It may be the case that extreme movements occur when excitations change between the start of the prediction and the prediction horizon. To find these cases, which would be ignored by simplification (c) in Sec. 4.6.2, we explore the input space further using RRTs. Fig. 4.26 illustrates the concept. The states $\boldsymbol{s}_{i,j}$ are given by applying the i^{th} combination of excitations as input for j timesteps (the length of the timestep is $\Delta t = \frac{t_f}{n}$, and (in this particular case), $n = 3$). This is the first and second step of the method presented in the preceding subsections, and generates the red volume. Using an RRT to further explore the tree found in the second step ought to enable us to find more corner cases, since we include movements during which the excitations may change.

In contrast to most implementations of RRTs, we have no goal state–we wish simply to explore the state space until the prediction horizon. We therefore adapt the RRT algorithm slightly from [108], so as to account for the time left until the prediction horizon. This is

[5] We call a tree *starlike*, if exactly one of its vertices has degree greater than 2.

Table 4.8: The muscle groups found in Sec. 4.6.4, the bodies to which they are connected and the a description of the movement performed by the muscles together

Group	***1***	***2***	***3***	***4***	***5***	***6***
Muscles	supra-spinatus, infraspinatus	anterior deltoid; pectoralis major (clavicular, sternal, ribs); coraco-brachialis	subscapularis; teres major; latissimus dorsi (thoracic, lumbar, iliac)	teres minor	biceps (short, long); brachialis; brachio-radialis; palmaris longus; pronator teres	triceps (long, lateral, medial); anconeus
Rigid bodies	humerus, scapula, clavicle	humerus, scapula, clavicle, thorax	humerus, scapula, thorax	humerus, scapula	humerus, scapula, radius, ulna	humerus, scapula, ulna
Move-ment	rotate upper arm	pull arm to thorax	rotate arm inward	rotate arm outward	flex elbow	extend elbow

Group	***7***	***8***	***9***	***10***
Muscles	medial deltoid	posterior deltoid	Flexor carpi (radialis, ulnaris), extensor pollicis brevis, abductor pollicis longus; flexors: pollicis longus, digitorum (both superficialis & profundus, all 4 fingers)	supinator; pronator quadratus; extensors: carpi (radialis longus, radialis brevis, ulnaris), digitorum communis (all 4 fingers), digiti minimi, indicis propius, pollicis longus
Rigid bodies	humerus, radius, hand	humerus, radius, ulna, hand	humerus, radius, ulna, hand	humerus, radius, ulna, hand
Movement	lift arm	lift arm	flex hand	extend hand

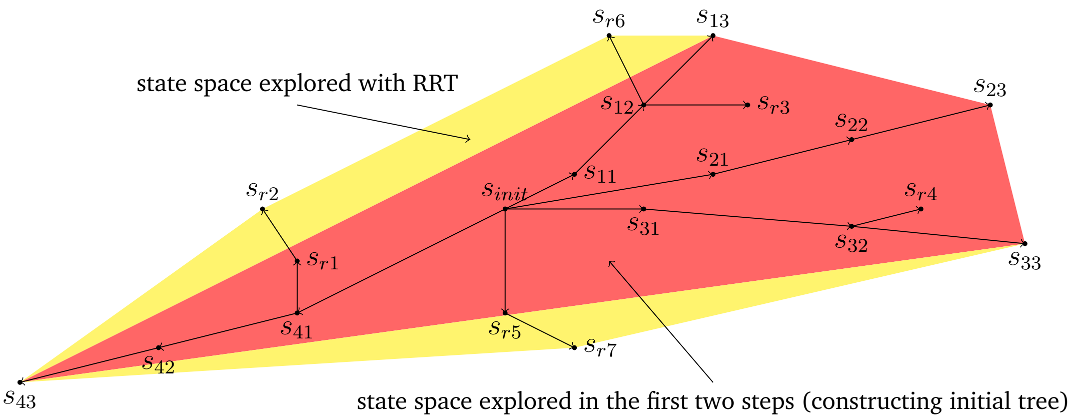


Figure 4.26: The process of exploration. We construct a “scaffold” tree by simulating with each possible grouped input. Then, we use an RRT to account for the cases that the inputs change during the simulation.

important, since it makes no sense to simulate states at the prediction horizon time t_f any further. Therefore, the function $\texttt{find\_closest}(tree, \boldsymbol{s})$ returns the state in $tree$ closest to $\boldsymbol{s}$ in terms of the Euclidean distance between the joint angles (not accounting for the activations, muscle fibre lengths, and the rate of change of muscle fibre length), whose time state variable is less than t_f. The algorithm is shown in Alg. 10.

Algorithm 10 RRT pseudocode

```
1: $tree \leftarrow initialize$
2: **for** $i < \eta$ **do**
3:     $\boldsymbol{s}_{goal} \leftarrow random\_value\_in\_state\_space$
4:     $\boldsymbol{s}_{start} \leftarrow find\_closest(tree, \boldsymbol{s}_{goal})$
5:     $\boldsymbol{u}_{best} \leftarrow \texttt{find\_best\_input}(\boldsymbol{s}_{start}, \boldsymbol{s}_{goal})$
6:     $\boldsymbol{s}_{best} \leftarrow \texttt{simulate}(\boldsymbol{s}_{start}, \boldsymbol{u}_{best}, \Delta t)$
7:     $tree \leftarrow \texttt{add\_to\_tree}(tree, \boldsymbol{s}_{start}, \boldsymbol{s}_{best})$
```

In lines 5 and 6 of Alg. 10, the algorithm attempts to find the state closest to $\boldsymbol{s}_{goal}$ which is reachable from $\boldsymbol{s}_{start}$ in one timestep. The highly nonlinear dynamics mean that it is infeasible to analytically determine the optimal input. We test two methods, one using on random sampling and the other, adapted from [107], based on linearising the system dynamics.

4.6.6.1 Random Sampling for RRT Input

We attempt to find the state $\boldsymbol{s}_{best}$ in Alg. 10 by generating k random inputs $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_k$, and evaluating $\boldsymbol{s}_j \leftarrow simulate(\boldsymbol{s}_{start}, \boldsymbol{u}_j, \Delta t)$ for each of them. The state $\boldsymbol{s}_j$ which minimises $\|\boldsymbol{s}_j - \boldsymbol{s}_{goal}\|$ is then assigned as $\boldsymbol{s}_{best}$. One can either sample randomly from the 10 input groups, or randomly sample from all the 50 inputs. The latter allows excitations where not all muscles within one group act together, which can potentially find additional corner cases.

4.6.6.2 RRTs with Linearized Dynamics

Even the 10-dimensional input space proves hard to sample–we must choose a very high k to adequately sample the space. An interesting solution for finding $\boldsymbol{s}_{best}$ in Alg. 10 is presented by [107] for nonlinear hybrid dynamics, where the dynamics are linearised around the point $\boldsymbol{s}_{start}$; we adapt their approach here. In order to linearise the dynamics, we first perform $m_{groups}+1 = 11$ simulations of duration Δt starting at $\boldsymbol{s}_{start}$; in the first case with all excitations zero $\boldsymbol{u} = \boldsymbol{0}$, and then with excitations $\boldsymbol{u} = \hat{\boldsymbol{u}}_i, i \in \{1, 2, ...10\}$, where $\hat{\boldsymbol{u}}_i$ represents the excitations of the muscles in the i^{th} *group* being set to 1, and the rest are set to 0. We show the approach in Alg. 11.

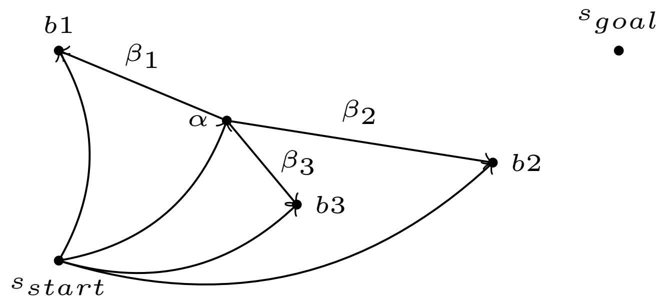


Figure 4.27: The linearisation of the dynamics, described in Sec. 4.6.6.2.

Algorithm 11 Linearisation procedure

```
$\boldsymbol{\alpha} \leftarrow simulate(\boldsymbol{s}_{start}, \mathbf{0}, \Delta t)$
for $i < m_{groups}$ do
    $\boldsymbol{b}_i \leftarrow simulate(\boldsymbol{s}_{start}, \hat{\boldsymbol{u}}_i, \Delta t)$
    $\boldsymbol{\beta}_i \leftarrow \boldsymbol{b}_i - \boldsymbol{\alpha}$
$\boldsymbol{B} \leftarrow [\boldsymbol{\beta}_1^\top \boldsymbol{\beta}_2^\top \cdots \boldsymbol{\beta}_{m_{groups}}^\top]$
```

The linearised system approximates $simulate(\boldsymbol{s}_{start}, \boldsymbol{u}, \Delta t)$ by $\boldsymbol{\alpha} + \boldsymbol{B}\boldsymbol{u}$. Finding the optimal input is then equivalent to minimising the distance to the goal: $\| \boldsymbol{s}_{goal} - (\boldsymbol{\alpha} + \boldsymbol{B}\boldsymbol{u}) \|$; this can be expressed as a quadratic programming (QP) problem and solved for $\boldsymbol{u}_{best}$:

$$\begin{aligned} &\text{minimize: } (\boldsymbol{s}_{goal} - (\boldsymbol{\alpha} + \boldsymbol{B}\boldsymbol{u}))^\top (\boldsymbol{s}_{goal} - (\boldsymbol{\alpha} + \boldsymbol{B}\boldsymbol{u})), \\ &\text{subject to } \mathbf{0} \leq \boldsymbol{u} \leq \mathbf{1} \end{aligned}$$

Finally, we simulate the system with the input $\boldsymbol{u}_{best}$, in line 6 of Alg. 10.

4.6.7 Evaluation of Exploration Methods

We tested the effectiveness of the 3-step method presented in the previous paragraphs. For comparison, we tested the following algorithms:

1. *random* – simulating the arm until t_f; inputs for each simulation chosen randomly from $\{0, 1\}^{50}$,
2. *enumeration* – steps 1 and 2 of the method (as described in Sec. 4.3.1) only,
3. *RRT random* – step 3 only, without the previous steps, using the random input method as in Sec. 4.6.6.1. We choose $k = 12$ for a fair comparison with the linearisation approach.
4. *RRT linearised* – step 3 only, without the previous steps, using the linearisation to find the input as in Sec. 4.6.6.2.

5. *enum-RRT* – the 3-step method: *RRT linearised* starting from the scaffold obtained by *enumeration*.

Simulations were run on a computer with Intel i5-4460, 3.20GHz processor and 8GB of RAM. The version of OpenSim used was 3.3, with the exploration program written in C++ and MATLAB 2016b used for the solution of the QP and visualisation.

In our algorithm, the most computationally expensive step is the biomechanical simulation– other calculations and write operations are comparatively negligible. However, the numerical solution of the dynamical equations in the simulation has a variable step size, and the number of steps required is not known *a priori*. We observed empirically that the duration we simulate was approximately proportional to the computation time, even when performing computations on a machine with nondeterministic execution times. Therefore, we compare computation time by comparing the *simulation time* $t_{simulated}$: this is the total duration over which the biomechanical model is simulated (i.e., running 50 simulations up to a prediction horizon of $20ms$ is $t_{simulated} = 1sec$). Similarly, to compare the extent of exploration of the state space, we evaluate the volume of the convex hull of the occupancies of the extreme arm positions found. We call this the *explored volume*.

In these comparisons, each algorithm simulates an overall simulation time of $t_{simulated} = 200$ seconds, unless they terminate earlier. We take the number of timesteps to be $n = 3$, and a time horizon of $t_f = 50ms$. In Figs. 4.28 and 4.29, the x-axis is $t_{simulated}$, the y-axis is the size of the explored volume reached after x seconds. Fig. 4.28 plots explored volume against the simulation times of *RRT linearised* and *RRT random* the two variations on the RRTs presented in Secs. 4.6.6.1 and 4.6.6.2. The linearization appears to allow the RRT to explore more effectively.

Fig. 4.29 plots explored volume against simulation time, for the methods *random*, *enumeration*, and our method (i.e. including the the additional RRT step). The growth in size of the volume during the simulation is plotted using solid lines; dashed lines are added for ease of comparison of the volumes' final sizes, if the algorithm terminates prior to $t_{simulated} = 200s$. The first and second steps of the method in Sec. 4.6.3, i.e. reducing the input space by clustering muscles and constructing a scaffold tree by deterministic sampling the state space, dramatically improve the efficiency of exploration.

4.6.8 Evaluation of Reachable Occupancies

For a variety of starting states of the arm, it was tested if the arm occupancies found using exploration of the biomechanical model are in fact contained in the overapproximative ROs we predict. We tested against the ROs of the 4-DOF model, without accounting for the shoulder movement (since we are simulating a stationary upright-standing human, see Sec. 4.6.1, we do not account for movement of the shoulder coordinate system in the overapproximative prediction).

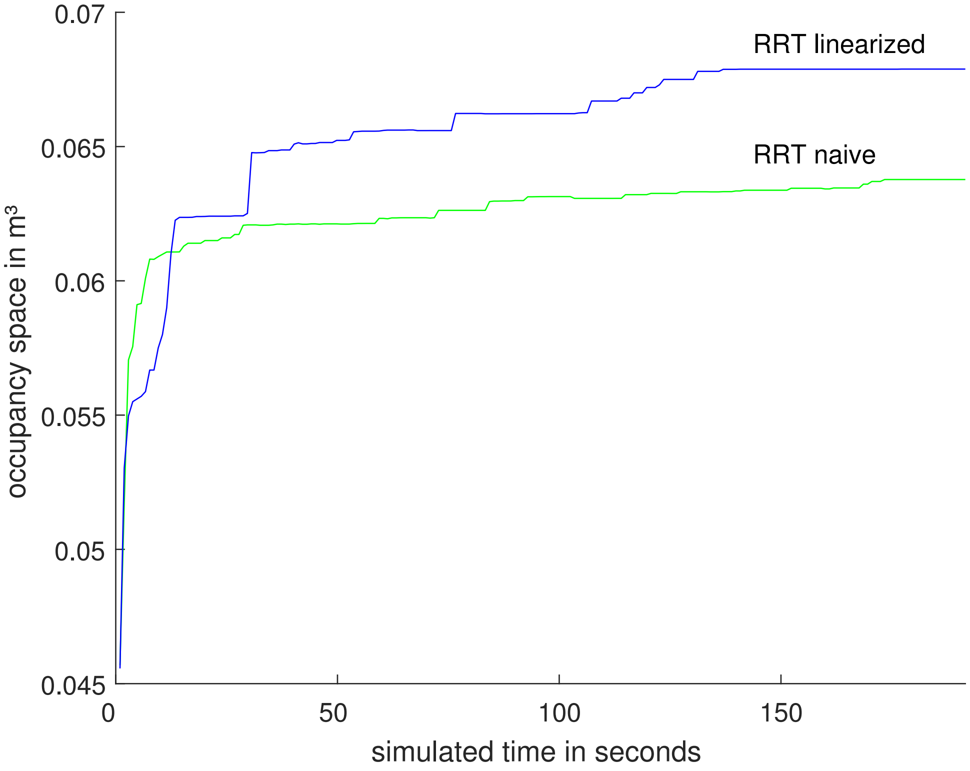


Figure 4.28: Explored volume versus simulation time, using RRTs only, either using random excitations to find the best input (*RRT random*), or by linearising the dynamics (*RRT linearized*).

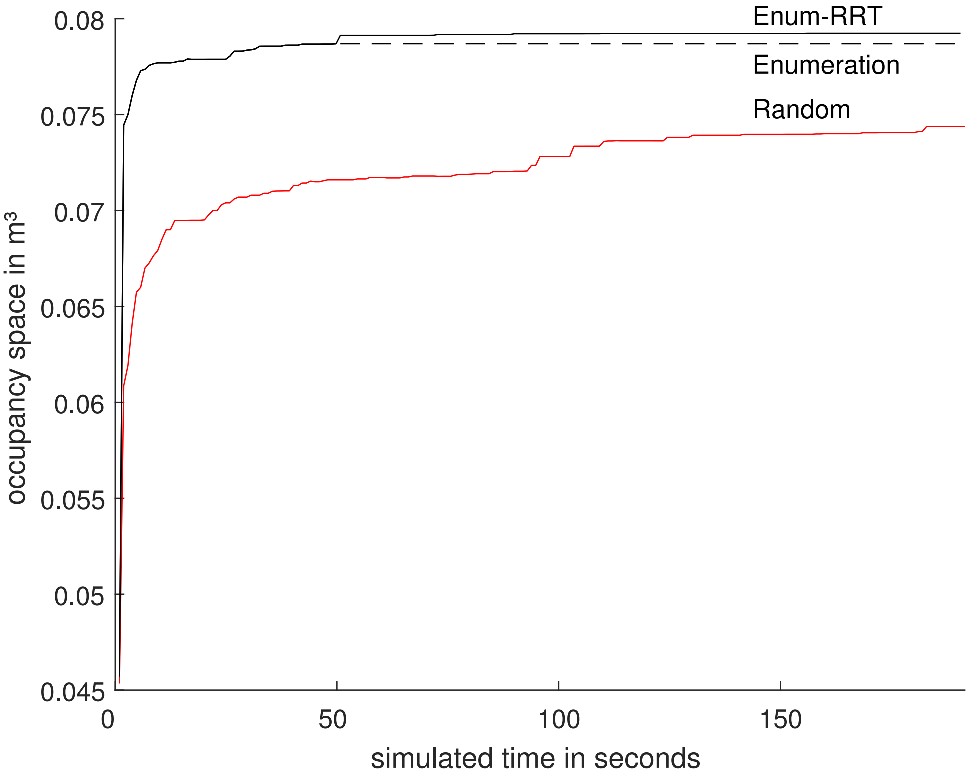


Figure 4.29: Comparison of our 3-step method (*Enum-RRT*), against the methods *enumeration* and *random*.

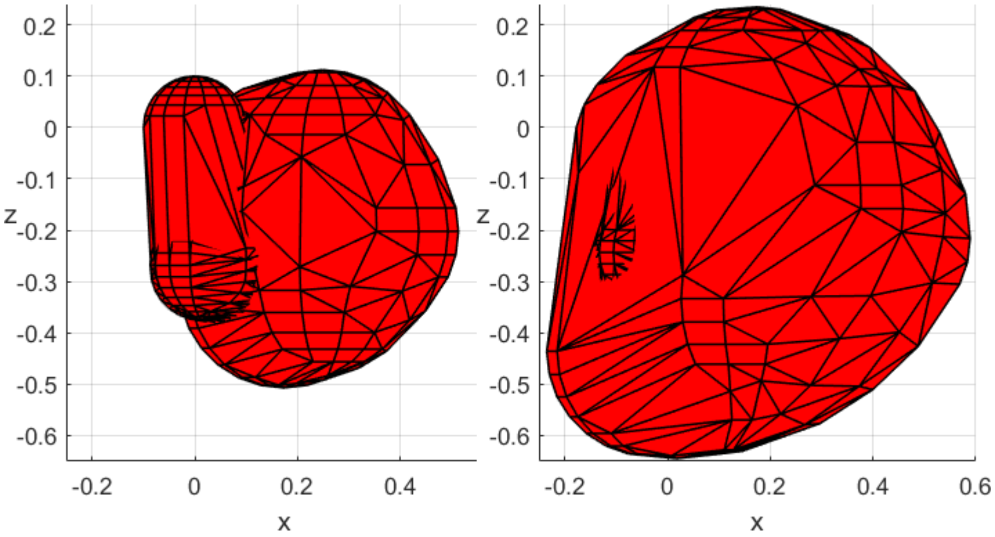


Figure 4.30: The reachable occupancies (axes in metres) calculated with the overapproximative model from [185] presented in Sec. 4.3.1, intervals $33\,ms$ (left) and $50\,ms$ (right).

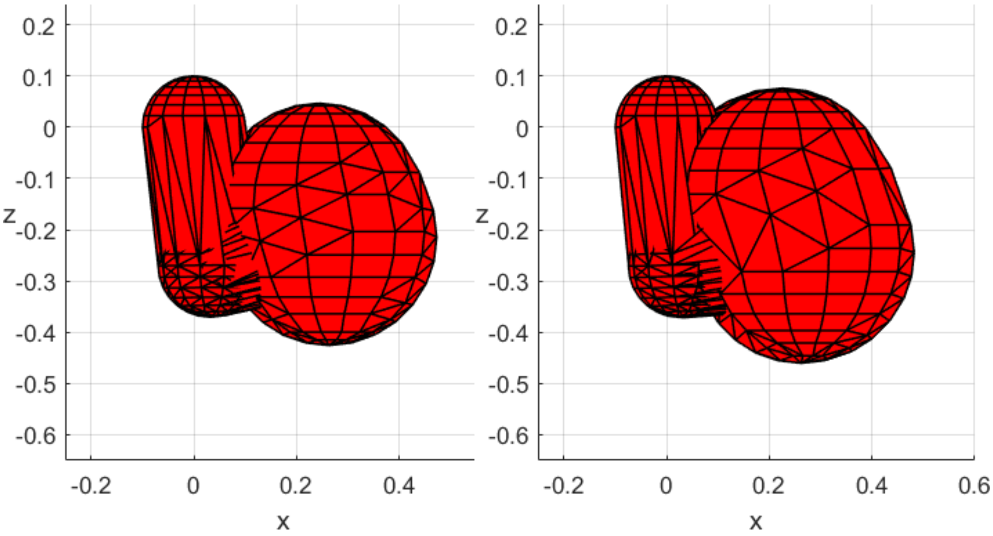


Figure 4.31: The explored volumes (axes in metres) calculated using our method, intervals $33\,ms$ (left) and $50\,ms$ (right).

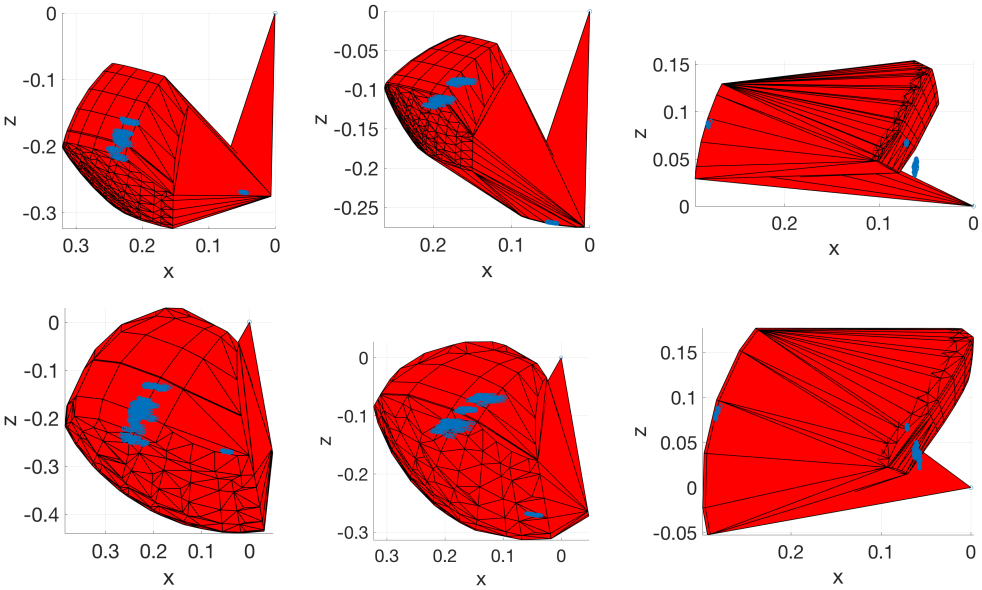


Figure 4.32: Shoulder, elbow and wrist points end positions of movements found from the exploration, overlaid on the convex hull of the overapproximative sets of shoulder and elbow positions, and the convex hull of the overapproximative sets of elbow and wrist positions, found using the method of [60], Sec. 3.2, from the reachable sets obtained with the method of Sec. 4.3.1. Origin at the shoulder; top row has $t_f = 33ms$ and bottom row has $t_f = 50ms$, columns are different starting states.

Tests from 3 starting positions are shown in Fig. 4.32, to time horizons of $t_f = 33.3ms$ and $t_f = 50.0ms$. We obtain the sets of elbow positions E and wrist positions W found from the markers in each of the movements found during the exploration; these are shown as blue dots. We also obtain the overapproximative, continuous sets of elbow and wrist positions using the reachability analysis from [185], as detailed in Sec. 4.3.1; these are red volumes.

In one of these situations, arm positions are found to fall outside those predicted by the overapproximation. However, on closer inspection, these occur when the elbow joint is flexed beyond realistically achievable angles, sometimes over $175°$. This, as the reader will be able to verify, cannot occur in a human arm (see e.g. [109], the maximum elbow flexion of an healthy male subject is $146°$), but occurs in the biomechanical model because this does not check collisions in soft tissue—the muscles of the arm cannot intersect in real life, hence the elbow cannot flex arbitrarily much.

4.6.9 Note on Computational Complexity

We examine complexity of the exploration algorithm, both with respect to η, and also with respect to m_{groups}, using the simulation time as defined in Sec. 4.6.7 The first step of the 3-step method presented in Sec. 4.6.3 requires m_{inputs} simulations for a duration of Δt. In the second step, computing the scaffold tree in requires $2^{m_{groups}}$ simulations, for a total duration of $n\Delta t$. Finally, in the third step, the RRT with the method of linearising the system (Sec. 4.6.6.2) requires $m_{groups} + 2$ simulations [6], simulating a duration of Δt per iteration, for η iterations. The total simulation time required by the algorithm is therefore $m_{inputs}\Delta t + 2^{m_{groups}}n\Delta t + \eta(m_{groups} + 2)\Delta t$. The time complexity with respect to η is $\mathcal{O}(\eta)$, and with respect to m_{groups} is $\mathcal{O}(2^{m_{groups}})$.

Summary

We have shown that the human reachable occupancy (HRO) is possible to compute conservatively, tightly and fast. The subset of human behaviour which must be accounted for in a HRCoex scenario can be modelled using simple models and parameterised with user data from a range of humans. The models used to compute the HRO depend on the sensing modalities available. Where sensors are available which can output the skeletal pose of the human, models may be used which take body geometry into account yet are possible to compute in real time. Two different approaches are presented which use the skeletal pose, either availing of a kinematic model of the human arm, or doing without. These approaches are evaluated using conformance checking, and a biomechanical model, to validate their conservativeness, and are tested on a robotic system.

[6] One simulation in line 1 of Alg. 11, m_{groups} simulations in lines 2–4 of the same algorithm, and one simulation in line 6 of Alg. 10

Chapter 5

The Human Aspect

Robotic systems intended for co-working with people must function well alongside their human colleagues—this thesis would be incomplete without investigating how our proposed approach fares in real human-robot co-existence (HRCoex). While guaranteeing safety is the top priority, the system should also work as efficiently as possible. Furthermore, the human co-workers should also feel safe around the robot. Making automated systems behave in a way that is comprehensible to humans while making them feel safe and engendering the correct amount of trust leads to more efficient co-working from both parties and greater acceptance, and hence greater uptake, of the technology.

Neglecting to engage with the human aspect now as HRCoex is in the public eye would not only limit successful uptake of HRCoex, but could lead to a cultural antipathy and mistrust towards robots which would be very hard to shift when robots become more ubiquitous.

Besides safety, it is important for successful human-robot co-existence (HRCoex) that the human both trusts the robot and comprehends its robot's behaviour, and that the robot can work efficiently with the human often entering its workspace. The three aspects of trust, comprehension and efficiency are presented in the following sections; in Sec. 5.4 and Sec. 5.5 we present experiments investigating these aspects in HRCoex scenarios.

5.1 Trust

What does it mean to trust a robot? Can robots even be trusted? Trust in the context of automation has had various definitions, one of the more applicable being *"the attitude that an agent will (a) help achieve an individual's goals in a situation characterized by (b) uncertainty and (c) vulnerability"* [110]. In a HRCoex scenario, (a) the robot can only help the human by not getting in the way, (b) it is uncertain what the robot's intention is, and (c) the robot has the potential to physically harm the human.

In human-robot interaction (HRI), trust contributes to how much humans rely on the robot [110], and trust is used as an indicator of humans' acceptance of the robot's computational skills (functional acceptance) and general intelligence (social acceptance), see e.g. [111, 112]. The correct amount of trust is important for harmonious co-working: over-reliance can lead to potentially dangerous mistakes–e.g. depending on the robot for safety-critical operations over and above its functionality. On the other hand, lack of trust can lead to misuse or disuse of the system, and may also be dangerous [110].

It is helpful to break down and understand the elements that compose trust of an agent. McAllister [113] identifies *affective* and *cognitive* trust; the former is based on the co-dependent, helping relationship between trusted and trustee, and the latter derives from the trustee's confidence in the reliability of the trusted. We do not consider social, humanoid robots, hence psychology of human-human relationships is not so applicable in our case.

A more relevant distinction for our case is made in [114], between an individual's *propensity to trust* autonomous systems, and their *historic trust* engendered by familiarity with the system. An interesting finding was that both the propensity to trust and the characteristics of the machine–whether its performance at a task was good or poor–affect the historic trust in the system. That the machine's performance affects historic trust is unsurprising (an unreliable machine will not fill a user with confidence, see e.g. [115]), but that the preconceived attitudes towards automation should affect historic trust is worthy of note for practitioners. In Sec. 5.5, we measure propensity to trust at the beginning of the experiment, since this may affect trust in the robot after acquaintance with the robot and task.

In addition, several factors affect the amount of trust humans have in automation. In the meta-analysis in [116] the authors classify factors as human-related factors, subdividing into factors to do with the human's learned abilities, and their personality; robot-related factors,

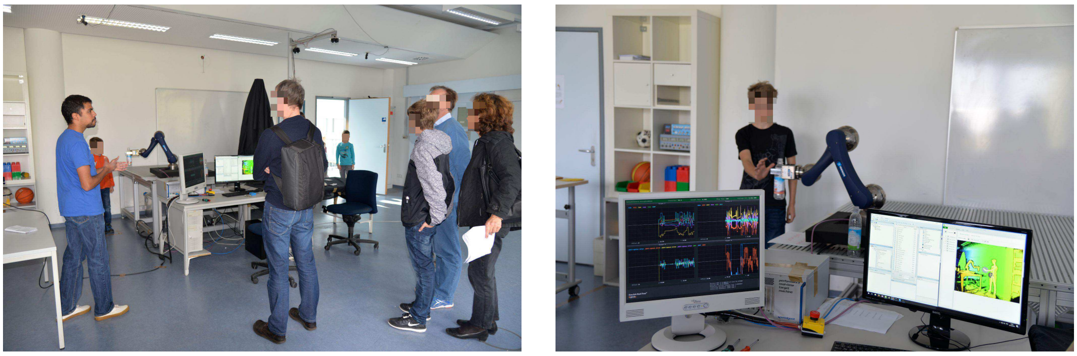


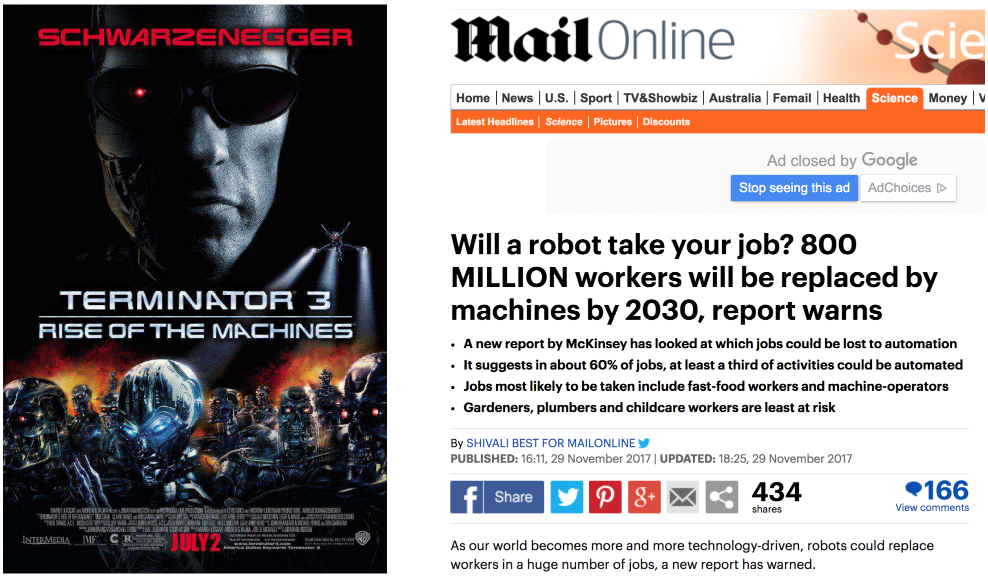
Figure 5.1: Children interacting with the robot during TU München open day, tracked by a Microsoft Kinect camera and depth sensor. Images courtesy of Doris Herrmann, TU München

subdividing into robot-performance-based, and attribute-based (e.g. anthropomorphism, proximity of the robot); and environmental factors, subdividing into factors based on the robot's task, and factors based on the cohesion of the human-robot team. Factors based on robot performance and attributes were found to have a large and moderate effect on trust, respectively; human factors are shown to have a smaller effect on trust. However, the authors also identify a lack of research in human-related aspects in the trust relationship, including operator experience, attitudes towards robots, and propensity to trust.

It is worth mentioning, finally, that robots are a novel phenomenon to the vast majority of the population. Unless one works with robots, for example in a factory or in research, one seldom encounters a robot manipulator outside of the television or laptop screen; Fig. 5.1 shows a rare exception at the university open day, where children and adults were able to interact with the robot (using our formally verified trajectory planning). While some visitors were apprehensive and other more trusting, most were simply curious. As robots become globally more ubiquitous in everyday life, humans' trust and perceptions of them are likely to change. These attitudes will also be shaped by our cultural and social context .

5.1.1 Cultural context

Preconceptions of robotics are heavily culturally dependent. As an example, the division in the perception of automata between Japan and the West is well-studied, e.g. [117, 118] and is most simply summarised as: artificial reproduction of life or natural processes is seen in Japan as "true aesthetics" whereas in the Judaeo-Christian-influenced West it is (often implicitly) perceived as "playing God" . Examples can be easily drawn from the rich corpus of western apocalyptic stories of automata: the Prague Golem, Frankenstein's monster, Karol Čapek's R.U.Robots, Schiller's *der Zauberlehrling*, the sci-fi writer Isaac Asimov's *Foundation* universe [119] and the *Terminator* film franchise [120], see Fig. 5.2(a). Japan, on the other hand, has a more forgiving picture of robots in popular culture, with several anime robot characters [118] and a long history with humanoid robots (Honda's *Asimo*, Fujitsu's *Enon*, Osaka University's *Repliee* series



Will a robot take your job? 800 MILLION workers will be replaced by machines by 2030, report warns

- A new report by McKinsey has looked at which jobs could be lost to automation
- It suggests in about 60% of jobs, at least a third of activities could be automated
- Jobs most likely to be taken include fast-food workers and machine-operators
- Gardeners, plumbers and childcare workers are least at risk

By SHIVALI BEST FOR MAILONLINE
PUBLISHED: 16:11, 29 November 2017 | UPDATED: 18:25, 29 November 2017

As our world becomes more and more technology-driven, robots could replace workers in a huge number of jobs, a new report has warned.

(a) (b)

Figure 5.2: (a) film poster for Terminator 3 with scary robots (source: `imdb.com`, ©2003 IMF Internationale Medien und Film GmbH & Co. 3 Produktions KG), (b) Sensationalist news article about robots (source: `dailymail.co.uk`, accessed 1.1.2018)

of robots and *Geminoid*). While antipathy towards automation in human co-workers may be based on the cultural *Weltanschauung*, it may also rooted in the fear of the threat that robots pose to their lifestyle as reported by news media. Sensationalist news articles concerning the change in the job market often play on people's fears of unemployment, see Fig. 5.2(b). Within the industry, in fact, a wide range of perspectives exist on the societal changes that "Industry 4.0" will bring, e.g. [121] argues that the ageing demographic in the West will lead rather to labour shortages.

Two scales, the Negative Attitude towards Robots Scale (NARS) and the Robot Acceptance Scale (RAS), are commonly used to measure the negative attitudes of humans towards robots as well as anxiety and communication avoidance towards the robot [122]. The simplistic assumption that Far-Eastern cultures were generally less concerned by robots has not been borne out experimentally, e.g. in [123], although cultural background did affect attitudes towards robots. Japanese participants were more concerned about societal implications of robotics and the implications of an emotional relationship with robots, compared to US participants, who had the most positive attitude towards robotics of all countries surveyed; the authors postulate that this could be down to more realistic representation of the skills and shortcomings of robots in Japanese media. Furthermore, participants were recruited only from universities and an online robotics community, which may not present an accurate cross-section of the population. In [124] cultural differences were also found. However, rather than religious differences or media portrayal, social hierarchy and family structure was found to have more of an impact on preconceptions of robots.

Conducting experiments at an international university (Technische Universität München), our test subjects came from a wide variety of cultural backgrounds. Although the cultural context is an integral part of the discussion of trust, we chose not to focus on the cultural influence on trust and acceptance of robots in the experimental studies which follow in Sec. 5.4 and Sec. 5.5, though we include it in discussions.

5.2 Comprehending robot intent

Humans working near robots need to know two things: firstly, that the robot is near them and operational, and secondly, what the robot is going to do, i.e. for fluent co-working, the human should understand where the robot will move. The first is a requirement in ISO 10218: robots intended for collaborative operation must be equipped with a visual signal, which shows when the robot is currently in collaborative operation [2]. Augmented reality (AR) presents a wealth of possibilities for communicating the robot's intent visually or auditorily [125] .

With anthropomorphic robots numerous possibilities exist, for example using facial expressions and moving the gaze to indicate the significance of a certain object. Gaze following, an indicator of actor engagement, has been observed when the robot's actions are aligned with its gaze [126], and social cues have been shown to enhance acceptance in terms of intention to use the robot [127].

Attempts have been made to understand and apply behavioural rules governing human-human interactions to HRI. When humans interact, they match or synchronise the timing of behaviours with the interaction partner [128]. By *turn-taking* is meant "*the fundamental way that humans organize interactions with each other*" [129]. Turn-taking does not only depend on the behaviour in question but also social cues such as gaze [130], and norms may vary across human cultures [131].

Verbal as well as non-verbal cues are used in human interaction for guiding turn-taking [132]. Understanding of social cues as well as the social appropriateness of actions are studied for a robot bartender. For the former, speech recognition and identification of body language (body posture and head pose) was successful in identifying human cues, with 91% accuracy for nonverbal bartender attention requests [133]. In terms of social appropriateness, for example, a robotic bartender should aim to execute actions in order for greatest efficiency, but it is perhaps more important to serve customers in the order that they approach the bar [134] – this rule is completely culturally determined. Reacting to human cues must also consider timing of the reaction as well as being interrupted during interaction, which can greatly influence the perceived team-work performance of the robot [135]

In our case, we do not have a humanoid robot, but only a serial-link robotic manipulator; the robot can express itself only through the motion of the robot manipulator. Even if a humanoid robot were available, the aforementioned social signalling of intent may not be appropriate: in close co-working, human co-workers may miss facial expressions or auditory cues when they are not paying attention to the robot. Imagine, for example, trying to set a table with a co-worker in a noisy, low-lit environment such as a restaurant. One body of work has focussed on how to shape end-effector trajectories to achieve effective co-working by avoiding collisions. This often has to do with understanding human movement in order to predict and avoid collision, e.g [56], where inverse optimal control is used to learn a cost function for a human co-worker's motion, and this is used to predict future motion giving a goal region, or [55], which calculates a probability distribution over the human's goal using a Partially Observable Markov Decision Process (POMDP). Such motion often tries to mimic human behaviour, on the premise that this is what the human is used to and would find predictable, hence easier to work alongside [136]. The human (perhaps subconciously) tries to understand what the robot will do and how it will move, and will adapt their behaviour accordingly.

5.2.1 Predictable, Legible and Expressive Motion

Since straight-line motion is typical of human point-to-point movements [137], it is postulated that predictable robot motion is also straight line movements: human well-being and performance is adversely affected by fixed-base robot motion which is not straight-line motion of the tool centre-point (TCP) [138]. Huber *et al.*found that robot-human handovers are more fluent when the robot motion is TCP straight line and the speed profile is minimum jerk, than when the robot motion describes a straight line in joint-space (not Cartesian space) and the speed profile is trapezoidal [139].

How a robot moves can influence the human's perception of the robots intent; the ability of a movement to communicate the intent of the robot is known as its *legibility*. Predictable motion differs from legible motion, in that the former is the most likely motion *given the goal*, whereas the latter is the motion that will lead an observer to infer the goal; in practice this means that while predictable motions are straight lines, legible motions are exaggerated towards the goal position [140], or in the case of human-robot coexistence, where the goal is non-collision, away from the human. Legibility in this case leads to increased perceived safety [141]. On the other hand, robots performing deliberately deceptive reaching motions can be perceived as malicious and untrustworthy [142].

Expressive motion refers to the desire to communicate a *state* rather than simply an intent. Humans are known to ascribe intentions even to movements of non-humanoid automata (we observed this also in the proceeding user studies). For example, in [143], participants ascribe emotions, such as anger and sadness, and intentions, such as exploration of space, to movements of an actuator with only one rigid body. Knight *et al.*[144] show how the shape of the path of a mobile robot (meandering or direct, changing over time) influenced human onlookers' perception of the robot's task (e.g. exploratory or goal-driven) and confidence or hesitancy. This lends credence to the idea that the movements of a (non-humanoid) robotic arm, devoid entirely of human features, can also elicit an socially understandable response. The authors of [145] seek to contextualise expressive robot motion characteristics in terms of the Laban Effort System [146], which provides a rich framework for specifying how social robots should behave.

5.2.2 Movement Cues

By movement cues, we mean expressive motions which are irrelevant to the task, but which are used to communicate intent to the human. In human co-working, a frequently observed movement when two humans are reaching towards a common object or walking towards a common space is a "back-off" motion, where one actor temporarily reverses the direction of motion of their hand during a reaching motion, before continuing the motion or starting a different reaching motion. Such a "back-off" movement cue could be interpreted either as an actor simply acknowledging the human's presence, or showing submissiveness.

5.2.3 Dominance and Submissiveness

Dominance is "the degree to which one actor attempts to regulate the behaviour of the other" [147]. It has been suggested that humans are less trusting of a robot that exhibits dominant behaviour than of one that is submissive [148]; this effect was not observed for dominant humans, which could be due to western cultural perceptions of robots as subservient. In [149], test subjects were found to prefer robots as servants or assistants compared to as friends. However, humans have known biases towards dominant actors. For example, dominance has been shown to increase trustworthiness for children [150]. Submissive robots may therefore encounter challenges building trust with co-working humans.

In human-robot co-working, submissiveness can be expressed as one party waiting for the other, and dominance as one party taking priority with their actions over the other. In previous work, it is suggested that humans prefer the robot to wait for them than the other way around, e.g. [151]. Since our method would be categorised as submissive under this understanding of dominance, it is of interest to see how humans react to it compared with more dominant methods.

5.3 Efficiency

Efficiency and its related property *fluency* in HRI, can be measured in various ways. Several studies measure human or robot idle time [152]. Others measure concurrent motion (time when the both robot and human are moving) [153] and time between the robot and human action, e.g. during a handover [139].

In HRCoex the human and robot do not explicitly interact, so one could conceivably measure the robot idle time, and the human's task-completion time. The human's idle time is hard to measure reliably: how does one tell if a human is waiting for the robot, or simply working slowly? A human, unlike a robot, does not spend the same time on the same task when performing it twice: humans tire, get distracted or become confused. One could conceivably measure the time the human spent concentrating on the task and concentrating on the robot, using gaze tracking. We did not do this due to time and cost constraints. In the following studies, we measured the time the human took for the same task with various robot behaviours (Sec. 5.4) and the time over various trials (Sec. 5.5).

5.4 User study: dominance and movement cues

In our formally-verified approach, the robot yields to the movement of the human, interrupting its own motion until the human moves out of the way. In co-working, it is important that this behaviour is understood, or this could lead to an *impasse*, where the human and the robot are both waiting for each other to move and lose time in the process. Furthermore, while the robot's behaviour may be provably safe, it is not necessarily intuitive, and such unpredictability may incline the human to distrust their robotic co-worker and/or lead to loss of efficiency. We wished to explore in [190] whether humans found such submissive behaviour preferable to dominant robot behaviour in terms of trust and predictability and whether motion cues helped humans to understand what the robot's intention was.

5.4.1 Methodology

We conducted a study of 5 different robot behaviours in a co-working scenario where the human and robot share a workspace but work on disjoint tasks (HRCoex) The human's task was the assembly of sandwiches, which is an easy-to-understand task coming from an industry where automation and human-robot coexistence could become well-established in coming years [154]. Test subjects were 25 volunteers aged between 18-32 years with a median age of 25, 10 female and 15 male. None of the test subjects reported any sensory or motor impairments.

The study was within-subject. Subjects filled out a declaration of consent and a questionnaire of personal information (F0). The human first stood at position A in Fig. 5.3, while the movement of the robot, without reaction to the human, was demonstrated. They then moved to position B, and performed the sandwich assembly task, while the robot remained stationary. The subjects were informed that the robot would not move while they did this task.

The sandwich assembly task consisted of assembling 6 sandwiches at precise positions on a table. The base slices of bread were already in place, and the subject had to place 4 "ingredients" (to avoid mess, these were card cut-outs of tomato (red), lettuce (green), salami (pink) and cheese (yellow) slices) followed by a slice of bread. The exact ingredients and their order of placement were displayed on a graphical instruction sheet, e.g. as in Fig. 5.5. Different instructions were given for each trial and their order was randomised. Subjects were instructed to pick and place one ingredient at a time, and to perform the task "quickly and precisely".

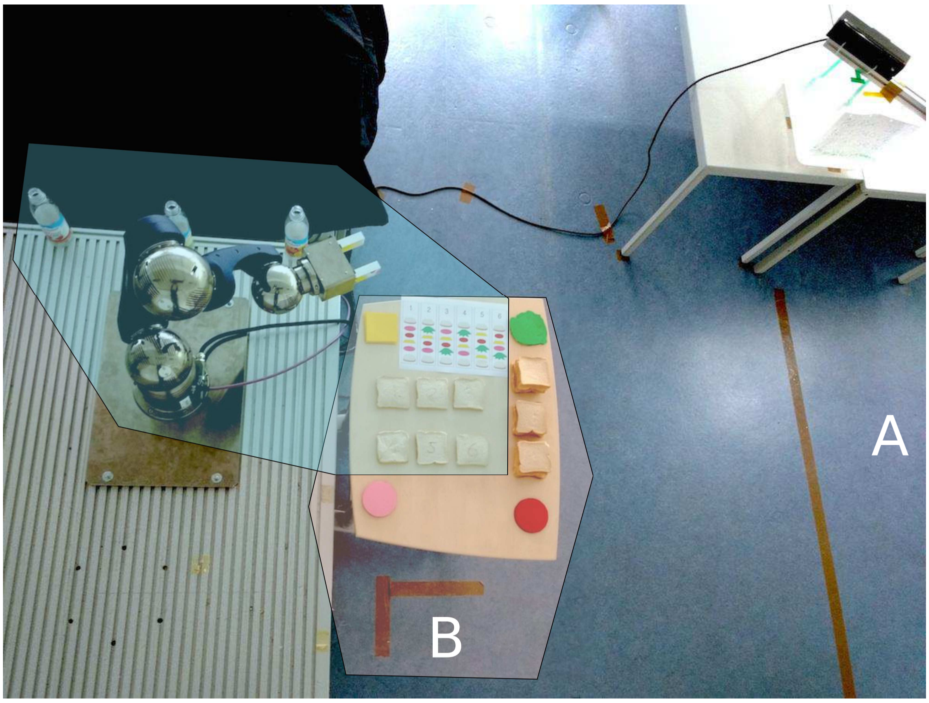


Figure 5.3: Setup of experiment showing human workspace, robot workspace, and overlap. On the table, the piles of ingredients–tomato (red), lettuce (green), salami (pink) and cheese (yellow)–are at the corners; the 6 sandwich bases are in the middle, the instruction sheet is at the top and the piles of bread are on the right.

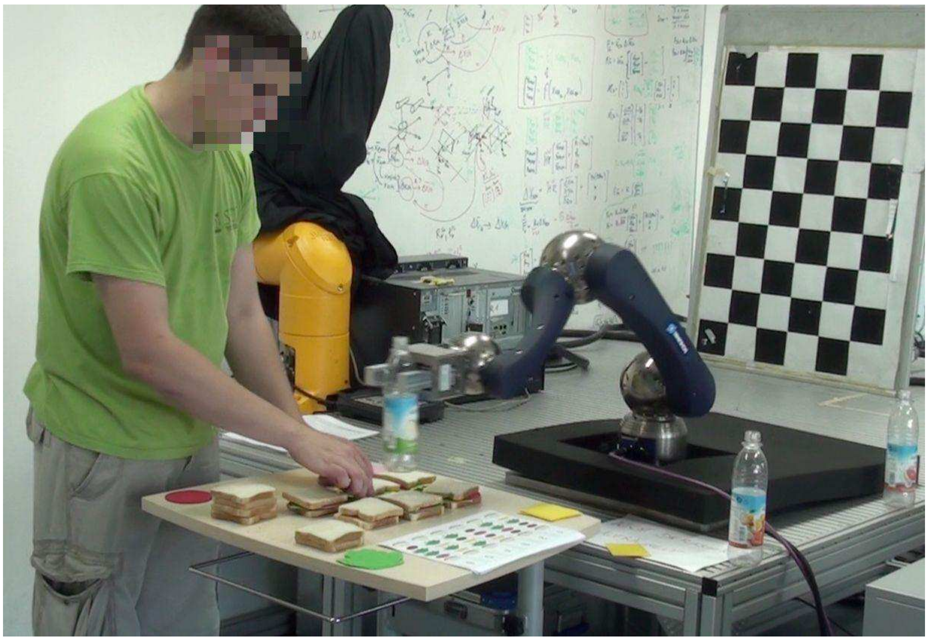


Figure 5.4: Human performing experiment with robot moving in the workspace.

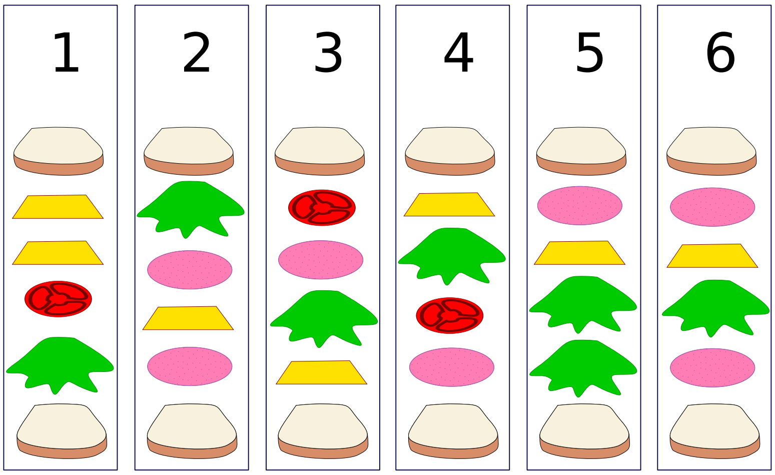


Figure 5.5: Graphic instruction sheet for one trial.

The subjects were then instructed to perform the task again, but informed that the robot would move at the same time. The robot's task was to pick up three "seasoning bottles" (to prevent culinary chaos, these were empty bottles) and pour them on the sandwiches, one after another. A freeze-frame from one experiment is shown in Fig. 5.4. After performing their task alongside the moving robot, the subject completed a questionnaire (F1). The activity followed by the same questionnaire was repeated another 4 times, each time with a different robot behaviour; the behaviours were presented in randomised order and are detailed in the next section (Sec. 5.4.3). During each activity, we measured the human and the robot's time to completion (TTC) .

At the end of the experiment, the subjects completed a second questionnaire (F2). They were then asked for their preferred strategy (that where they felt most comfortable working alongside the robot) and any other feedback they had on the experiment.

5.4.2 Questionnaires

The questionnaires F1 and F2 comprised subscales of the questionnaire *Trust in Automation* from Körber *et al.*[155]. In this work, 6 different aspects of trust were identified and questions were organised into subscales testing each aspect. Subscales *Reliability/Competence*, *Understanding* and *Trust* made up F1, while *Familiarity [with similar automated systems]*, *Intention of Developers* and *Propensity to trust*, made up F2[1]. These questionnaires can be found in Appendix. A.6.

[1] in German, subscales *Kompetenz/Reliabilität*, *Verständlichkeit/Vorhersagbarkeit* and *Vertrauen* made up F1, while *Vertrautheit*, *Intention der Entwickler* and *Neigung zu Vertrauen*, made up F2

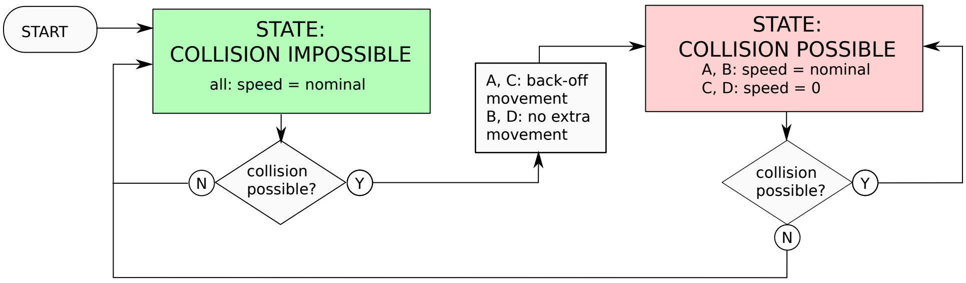


Figure 5.6: Illustration of control for behaviours DB, DN, SB, SN

5.4.3 Robot behaviours

The robot had 3 “seasoning bottles” in its workspace. The movement of the robot was such: the robot moved its end effector above each bottle in turn, lowered it, gripped the bottle, moved up, moved towards the workspace of the human, positioned the end effector over three sandwiches in turn, rotated the end-effector by $90°$ (to mime pouring), and rotated back. After finishing with the three sandwiches, it returned the bottle to its original location: moving the end-effector over the original location, lowering it, ungrasping the bottle, moving it up, and then proceeding to the next of the three bottles.

The spatial path of the robot was the same for all 5 behaviours; the only difference was that the speed along this path was scaled by a factor δ. The 4 behaviours are described in Sec. 5.4.3.1; the fifth behaviour was an implementation of [156], described in Sec. 5.4.3.2.

The nominal behaviour of the robot consisted of straight-line movements, since these have been shown to be better for human performance and well-being in co-working scenarios, possibly due to their better predictability [138]. Huber *et al.*[139] observed that straight lines led to more fluent handovers, in combination with minimum-jerk velocity profiles, compared with joint space point-to-point motions with trapezoidal (non-jerk-limited) velocity profiles. We therefore made our nominal velocity profiles jerk-limited (distance along the trajectory was a quintic polynomial in time).

5.4.3.1 Behaviours 1–4

The first four robot behaviours are dominant-back-off (DB), dominant-non-back-off (DN), submissive-back-off (SB), and submissive-back-off (SN). They are described in Alg. 12, which is a modification of Alg. 1 and illustrated in Fig. 5.6. At each control cycle, the robot plans a short-term plan similar to Sec. 2.2.1, but only with limited acceleration (not limited-jerk). The duration of the plan was found from the maximum joint accelerations $\boldsymbol{a}_{\max}$ (set at 1.75π for the first 5 joints and at 3.5π for the last joint) and the joint speed at the subsequent timestep $\dot{\boldsymbol{\xi}}(t_{k+1})$ so:

$$t_{stp} = \max\Big(\frac{|\dot{\boldsymbol{\xi}}(t_{k+1})|}{\boldsymbol{a}_{\max}}\Big), \tag{5.1}$$

where division is elementwise and the $\max$ operator returns the maximum of the vector's elements. The speed is scaled by a scaling factor δ defined by:

$$\delta(t) = \begin{cases} 1 & t_k \leq t \leq t_{k+1} \\ \frac{t - t_{k+1}}{t_{stp}} & t_{k+1} \leq t \leq t_{k+1} + t_{stp} \\ 0 & t > t_{k+1} + t_{stp} \end{cases} \tag{5.2}$$

It then estimates the robot reachable occupancy (RRO) during the short-term plan[2], checking it against the human reachable occupancy (HRO) calculated according to Sec. 4.2.2 using $v_{\max,human,ISO} = 1.6\frac{m}{s}$. If the HRO and RRO intersect, the flag `is_safe` is set to 0.

In the case that `is_safe` $= 0$, in the back-off trajectories (DB and SB), a back-off motion is now performed. This consists of the robot stopping, and reversing back along its path for 0.5 seconds. Acceleration was limited so that the movement was not too abrupt.

In the dominant strategies (DB and DN), the robot then continues on its trajectory, i.e. δ ramps back to 1 (DB) or stays at 1 (DN) whereas in the submissive strategies (SB and SN), the robot stops on the failsafe manoeuvre (SN) or stays stationary (SB).

The robot continues to plan and verify a short-term plan. If the HRO and RRO do not intersect, the flag `is_safe` is set to 1. To avoid `is_safe` chattering between 0 and 1, hysteresis is introduced by augmenting the RRO by a small margin when `is_safe` $= 0$.

Algorithm 12 Robot behaviours

```
Input: long-term plan $\boldsymbol{\xi}$, behaviour $\in$ { DN, DB; SN, SB} , human sensor data $\mathcal{H}$,
Output: Trajectory
$\delta \leftarrow 1$
is_safe $\leftarrow 1$
while not at goal state do
    if is_safe then
        increment trajectory with speed scaled at $\delta$
        is_safe $\leftarrow$ plan and verify short-term plan
        if $\neg$is_safe then
            if behaviour = DB $\vee$ behaviour = SB then
                perform back-off manoeuvre
            if behaviour = SB $\vee$ behaviour = SN then
                perform fail-safe manoeuvre
                $\delta \leftarrow 0$
    else
        increment trajectory with speed scaled at $\delta$
        is_safe $\leftarrow$ plan and verify short-term plan with hysteresis
        if is_safe $\wedge$ (behaviour = SB $\vee$ behaviour = SN) then
            perform recovery manoeuvre
            $\delta \leftarrow 1$
```

In effect, in DN the robot effectively ignores the human, and SN is a simplified version of the

[2] N.B. not calculated in a conservative way.

formally verified behaviour described in Alg. 1. The behaviour SB backs off and stays stationary, which could be interpreted as very submissive, and the behaviour DB backs off when the robot approaches the human (or vice-versa) but continues on its path, in which context the "back-off" manoeuvre could be interpreted as acknowledgement of the human's presence.

5.4.3.2 Behaviour 5

The fifth robot behaviour, time-scaling (TS), is as described in [156], where the velocity is scaled by a factor $\delta \in [0, 1]$, which is a function of the estimated time to collision of the robot from the human and the braking time of the robot. The braking time of the robot was taken to be $300ms$. In this behaviour, the robot slows when it comes closer to the human and when it is moving fast. It can be seen as a compromise between dominant and submissive behaviour.

The control input is again a scaling factor δ, which scales the trajectory (without placing bounds on the joint accelerations, jerks or torques) based on the principle that the time to collision of the human and the robot should be greater than or equal to the braking time. We do not detail the derivation here but refer the reader to [156, 157].

Assume the robot to be composed of m links represented by line segments $L_i = \overline{\boldsymbol{p_{1,i}}\,\boldsymbol{p_{2,i}}}$ for each $i \in \{1, ..., m\}$. For each link i we calculate:

$$\boldsymbol{E}_i = \begin{bmatrix} (\boldsymbol{p}_{obs} - \boldsymbol{p}_{1,i})^\top \boldsymbol{J}_{1,i} \\ (\boldsymbol{p}_{obs} - \boldsymbol{p}_{1,i})^\top \boldsymbol{J}_{2,i} - (\boldsymbol{r}_{2,i} - \boldsymbol{r}_{1,i})^\top \boldsymbol{J}_{1,i} \end{bmatrix},$$

$$\boldsymbol{f}_i = \min_{\boldsymbol{p} \in L_i} \|\boldsymbol{p}_{obs} - \boldsymbol{p}\|^2 \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

where $\boldsymbol{p}_{obs}$ is the point on the obstacle nearest the link line segment $\overline{\boldsymbol{p_{1,i}}\,\boldsymbol{p_{2,i}}}$ (the term $\boldsymbol{f}_i$ then represents the minimum separation of obstacle and human), and $\boldsymbol{J}_{1,i}$ and $\boldsymbol{J}_{2,i}$ are the Jacobian matrices evaluated at $\boldsymbol{p}_{1,i}$ and $\boldsymbol{p}_{2,i}$ at joint position $\boldsymbol{q}$ (the dependency on $\boldsymbol{q}$ is not shown, for clarity). The scaling factor δ is then obtained so:

$$\delta = \min_{i \in \{1,\ldots,m\}} (\delta_i), \quad \delta_i = \begin{cases} \frac{\boldsymbol{f}_i}{t_{b,i} \boldsymbol{E}_i \dot{\boldsymbol{q}}} & \boldsymbol{E}_i \dot{\boldsymbol{q}} > 0 \\ 1 & \boldsymbol{E}_i \dot{\boldsymbol{q}} \leq 0 \end{cases}.$$

Unlike the dominant strategy, the robot does slow down nearer to the human. However, unlike the submissive strategy, there is no abrupt stopping as soon as a collision is possible, but rather a gradual scaling down of velocity related to the distance between human and robot and their relative velocities. Hence, this behaviour can be considered a middle ground between dominant and submissive.

5.4.4 Hypotheses

We hypothesised:

H1 Robot TTC will be less in dominant strategies (DB, DN) than submissive strategies (SB, SN);

H2 Robot TTC for time-scaling (TS) will be (a) more than in dominant strategies (DB, DN) but (b) less than submissive strategies (SB, SN);

H3 Human time to completion will be less in submissive strategies (SB, SN) than in dominant strategies (DB, DN);

H4 submissiveness and movement cues improve *understanding*;

H5 submissiveness and movement cues improve perceptions of *competence and reliability*;

H6 submissiveness and movement cues improve *trust*.

5.4.5 Results

H1 We found that dominant behaviours were significantly more efficient in terms of robot TTC than submissive strategies, and that the time-scaling (TS) behaviour was between dominant and submissive strategies in terms of efficiency. Human TTC was not significantly different. Strategy DN engendered significantly lower trust than the submissive strategies. Detailed results are presented below.

Tab. 5.1 shows the robot and human TTC. Hypothesis H1 stated that the TTC for the robot will be less in dominant strategies than in submissive strategies. **This was borne out by the evidence**: A repeated measures analysis of variance (ANOVA) showed that the robot took significantly longer in SB than DB ($p < 10^{-6}$) and in SN compared to DN ($p < 10^{-6}$). Furthermore, strategies with a back-off movement took significantly longer than those without: SB was longer than SN ($p = 0.008$) and DB was longer than DN ($p < 10^{-6}$).

H2 As postulated by hypothesis H2, the TS behaviour did indeed take **longer** than both DN and DB (repeated measures ANOVA gave $p < 10^{-6}$ and $p < 10^{-5}$). It was also faster than both SB and SN ($p < 10^{-6}$ for both). **Hypothesis 2 is therefore upheld.**

H3 Hypothesis H3 proposed that human TTC would be less where the robot behaviour was submissive and with movement cues. The human times **did not exhibit any significant difference** when tested by a repeated-means ANOVA. However, despite the instructions to the subjects being to complete the task “as quickly and precisely” as possible, 3 subjects chose to step back away from the workspace and wait for the robot while it finished its task. Since we did not *explicitly* say that the subject should keep working while the robot works, on consideration, we did not exclude these subjects. If one were to exclude these subjects, one observes statistically significant differences in the manner predicted by the hypothesis, between DN and SN ($p = 0.009$), DN and SB ($p = 0.043$), and DB and SN ($p = 0.029$).

Table 5.1: Time to completion for robot and human subjects (seconds)

		DN	DB	SN	SB	TS
Robot:	μ	80.2	85.1	119.6	124.9	98.6
	σ	0.8	2.1	15.6	13.3	10.7
Human:	μ	60.0	60.8	57.4	57.6	61.3
	σ	13.6	16.2	15.3	16.4	14.9

Table 5.2: Human factors scores for human subjects (scale of 1–5)

	DN	DB	SN	SB	TS	
Understanding:	μ	3.21	3.23	3.25	3.17	3.18
	σ	1.28	1.29	1.21	1.24	1.08
Competence/	μ	3.04*	3.08	3.24	3.31*	3.19
Reliability:	σ	0.82	1.05	0.93	0.82	0.86
Trust:	μ	2.62**	3.01	3.31**	3.31**	3.17
	σ	1.11	1.37	1.19	1.15	1.05

* near significance, worthy of further investigation;
** significant difference between DN and the behaviours DB, SB and SN.

In light of this, we designed the experiment for the following study in Sec. 5.5 such that the human could not avoid going into the robot's workspace. In that way, such avoidant behaviour was excluded. It is also worth noting that lower human TTC may be an illusory benefit: if the human is working faster because they feel harried by the robot, rather than because the robot does not hinder them, this is not conducive to good co-working. In the design of our subsequent study (Sec. 5.5), we included pertinent questions to gauge whether this was the case (see Sec. 5.5.4).

The remaining hypotheses were human-factors based. We calculated Cronbach's α [158] to determine the internal consistency of each subscale, i.e., whether the subscales were measuring the same quality. For *Trust*, this came to 0.93; for *Understanding*, 0.90; for *Competence/Reliability*, 0.84. These indicate strongly that the scales are internally consistent.

H4 and H5 Hypotheses H4 and H5 postulated that submissiveness and movement cues improve the perception of the robot's competence/reliability and the human's understanding of its behaviour. Since the data for competence/reliability was found to be significantly non-normally distributed using a Shapiro-Wilk test [159] at the 5% confidence level, we used Friedman's test [160], which is the non-parametric equivalent to a repeated-means ANOVA, with broadly the same assumption on the data except for normality of distribution. Evidence to uphold these hypotheses was **not statistically significant**. However, behaviour SB exhibited higher scores in the subscale of competence/reliability than DN with confidence $p = 0.061$. While this is not significant, this is perhaps worth further investigation with a larger dataset.

H6 Trust was also found not to be normally distributed so we used Friedman's test to test significant differences between robot behaviours. Hypothesis H6 proposed that trust is increased when the robot exhibits submissiveness and movement cues. This was upheld in that there was **statistically significant higher trust scores** with SB ($p = 0.0016$) and SN ($p = 0.0184$), compared to DN. Between DB and the submissive strategies there was no significant difference, nor was there between SB and SN. The mean trust scores and standard deviations are shown in Tab. 5.2.

Comments from the users showed that, despite the robot being demonstrated beforehand, some participants took some time to get used to it.

Subject 20 "After the first two or three runs, I was getting it [i.e. comprehending] more, what it was trying to do, and that made me look upon it more favourably."

Subject 14 "At the beginning I was wondering whether I have to be careful of it... but while assembling [the sandwiches] I never thought about it.'

Others ignored the robot from the beginning, whereas others had not gotten used to it even after several trials.

Subject 22 "At the beginning I didn't really pay attention to the robot and our interaction."

Subject 8 "The last two [trials] were really, really bad. I was actually scared."

These responses show that subjects' perceptions of the robot do indeed change according to experience with the robot. This motivates our longer-term study in Sec. 5.5.

5.4.5.1 Dominance

In the literature it has been claimed that humans prefer making the robot wait while performing their own task, rather than yielding priority to the robot [151, 152, 148]. In the submissive strategies SN and SB, the majority of participants did indeed continue with their task as the robot approached the shared workspace and halted, hence assuming a more dominant role in the interaction. In contrast, as mentioned in Sec. 5.4.5, 3 subjects even stepped away from the robot to allow it continue its movement and waited for the robot to move out of the way, allowing the robot to dominate in the interaction.

User comments supported the diversity of preference here. Typical comments from users who preferred the robot to wait were:

Subject 4 "As long as I didn't have to wait for it, and it didn't get in my way then that's ok. I didn't like when it was blocking my cheese [in the dominant strategies.]"

Subject 8 “I felt more comfortable and safe [when it stopped and waited].”

Subject 22 “[SN] was the best, because I could comprehend it better[3], and because it wasn't in the way.”

Others, however, preferred the robot not to wait, for a variety of reasons:

Subject 7 - “[Strategies SB and SN] were rather bad, because although the robot was very careful, it ended up hindering my working by stopping so that I could not move forward.”

subject 18 “[It is best] when the robot just does its task. I find it bothersome[4] when it suddenly stops and when it waits for me. [When it stops] it also slows me down. Because I was so focussed on what ingredient I could take without disturbing the robot, I neglected the task.”

Subject 23 “I found it best when the robot completely ignored me, because then one can estimate what happens next, to some extent”

The differences in waiting preferences could be due to subjects' perception of the robot's role– as a co-worker or as an unintelligent machine that is in the way; it can also stem from how much the human feels threatened by the robot or expects to be injured by it, since someone apprehensive of the robot may prefer it to wait. Attitudes towards robots is influenced by cultural background (Sec. 5.1.1) and may also change with experience of the robot. In the subsequent study (Sec. 5.5), we investigate human's perceived safety and how it changes with experience of the robot. Since the human's feelings about waiting for the robot are critical to successful co-working, we also specifically asked subjects how they felt about waiting for the robot in the subsequent study.

The user feedback shows that perception/reaction to dominant behaviour (DN and DB) comes down to personal taste. In fact, when asked for their preferred strategy, 10 subjects chose dominant strategies, 10 chose submissive strategies, and 5 chose TS. We suggest that the degree of dominance in the robot's movements could be tailored to the preferences of the human co-worker. In addition to psychometric tests or questionnaires prior to working, preferences could be gauged on the fly using some kind of state estimator. In [161], the autonomous agent (in this particular work, a self-driving car) performs harmless “information gathering actions” in the course of operation; from the human's reaction to these actions it gauges the state of the human. In [161] this is alertness, whereas for us these properties could be the human's preference in terms of dominance and reaction to movement cues.

Perceived dominance has been shown to be related to other features of the robot motion in [162], where the acceleration and curvature of the path of a mobile robot affected dominance, as well as influencing other features of the robot's perceived affect. In this study, the robot nominal trajectory was the same over all trials, but this is an interesting aspect to investigate in future work, for example when replanning trajectories on the fly as in Sec. 2.3.

[3] in German: “... *weil ich es besser einschätzen konnte*”

[4] in German: “*ich finde es lästig*”

5.4.6 Understanding of the Robot Behaviour

Users did not always understand the same movement strategy in the same way. For example, with respect to the DB behaviour, some found the behaviour good because:

Subject 5 "It took the human into account, but nevertheless continued working"

Subject 12 [as justification for it being the best strategy] "the robot kept stopping, but it managed to get more done than the other strategies"

Whereas for others, it seemed like the robot was malfunctioning.

Subject 4 "I thought there was something wrong with it because... it backed off, and then it got in my way again."

In general, many subjects appreciated the backing-off movement. Three subjects said that backing off was "cool", and one said that it "made sense". One elaborated further:

Subject 15 "when someone works together with several people, and they get in each others' way, then one of them draws back. That is what one is used to, and therefore I found it the most natural way to work. This is the most predictable reaction–when one knows that [the robot] always draws back."

However, even SB could be interpreted as unpredictable, because:

Subject 11 "it wasn't always a good thing that it stopped [in strategy SB], because one just didn't know when it was going to move again."

User comments again suggest that increased comprehension of the robot's behaviour leads to positive valency, e.g:

Subject 2 [as justification for their preferred strategy] "I could best understand what was going on".

Subject 14 "when [I] know what it is doing, then I trust [the robot]".

Several subjects also mentioned that they would like to understand better what the robot's task was. Interestingly, some tried to take the robot's capabilities into account, e.g. a subject without a background in robotics or artificial intelligence commented that they used only one arm to do their task, because "I thought the robot would get too confused if I used two arms".

Another source of confusion was the condition that made the robot enter the unsafe mode. Most subjects seemed to understand that the robot would stop/back off if the human came close to its end-effector, but when proximity to the robot's base or other joints elicited the same reaction from the robot, this was not well understood. Our safety criterion considers every part of the the robot as unsafe when it is moving, for example, clamping of body parts could occur at the base when the robot is in motion. Some subjects had their left arm close to the base of the robot in one of the submissive strategies, which led the robot to stop, but the subject did not realise that they were close to the robot.

This could be down to *functional bias*, the idea that human's conception of an object's spatial location is biased towards the location of its functional part, in our case, the robot end-effector [163]. Approaching the end-effector would be seen as approaching the robot whereas approaching the base is not. People's awareness of the "danger zones" on a robot will most likely evolve as robots become more ubiquitous at home and at work, in the same way that humans know which parts of a knife or a gun are dangerous to touch.

Two subjects were not aware that the robot was responding to them at all, despite the camera positioned in front of them. In our longer-term study (Sec. 5.5), the subjects wore a motion-capture suit and were told that this was so that the robot can detect them.

5.5 Longer-term study of Dynamic Safety Zones

Participants in the study of Sec. 5.4 encountered our robot for the first time during their appointment; some had previous experience with robotics, and others had none. In an industrial human-robot coexistence scenario, it can be expected that subjects will not have prior experience working with robots, and that their attitude to robots develops with experience to the robot.

While first impressions are indeed influential, in industry co-working will not be limited to one session. Dautenhahn [149] notes that the novelty of many HRI experiments can produce unrepresentative responses from participants in terms of their trust and reported safety. Prior experience with robots is also a factor influencing trust which has been underinvestigated [116]. For this reason, we wished to see how subjects without prior experience with robots adapt to working with the robot, both in terms of their attitude towards the system as well as the efficiency of both them and the robot.

In the previous study, we had several unanswered questions: do people prefer it if the robot waits for them? Does it make them feel safer? Do subjects understand how the robot reacts to them? And does this change with experience of the robotic system?

5.5.1 The Process of Accustomisation

Long or medium-term studies in HRI have focussed mainly on service robotics in home environments e.g. [164], in particular for elderly care [165, 166], or rehabilitation robotics. [167] In industrial human-robot interaction there appears to have been little to no research in this area. Research into acceptance of technology has focussed a lot on automated driving or teleoperation of unmanned vehicles, e.g. [168, 169], but is recently starting to include human-robot co-working scenarios, e.g. [112].

Studies in HRI are often conducted in one appointment per test subject, e.g. [152, 55], or even without contact to with a physical robot at all, but only through videos e.g. [142, 144].

The less the contact with the robot, the less the study can reveal about human's reactions to working with robots after the initial encounter and the less relevant to the real scenario of HRCoex, where humans work alongside physical robots on a continual basis.

In [170], a 5-week study with 8 appointments was carried out to ascertain subjects habituation to approach distances and directions of 4 designs of robot. Effects of accustomisation were already observed in the 2$^{\text{nd}}$ appointment. A study of first-time operators of a juice pasteurisation unit [171] where 60 trials were distributed over 3 days showed improvement in operator performance and trust even in the first few trials. From user comments in the previous experiment in Sec. 5.4, subjects changed their perceptions of the robot after a few trials. We therefore decided to schedule 3 days, each of 4 HRCoex trials. These were to occur within the space of 8 days, because this was the shortest period over which we could practically engage subjects to commit to an experiment.

As HRI becomes a reality in factories, an increasingly popular and commendable trend in research is the trial of robots with factory workers or in factory environments. A pioneering system in this field is the PaDY (in-time Parts and Deliverables to You) robot developed by the team of Kosuge at Tohoku University [35, 36]. This is a 2-DOF arm which delivers parts to workers on a production line; at the same time it avoids collisions, in [36] using hidden Markov models (HMMs) to predict the motion of the human.

The group of Steil conducted a study of their method for kinaesthetic teaching of redundant robots on 49 industrial manufacturing workers in [172]. The group of Vanderborght [127] conduct a social robotics study of a robot co-worker employing body-language as a communication aid with 11 factory workers of mean age 45.7 who had been working at their current factory for 22.5 years on average. Such studies of the target population in industrial HRCoex are invaluable. It is a fact of experiments at campus universities that most experimental subjects are either students, faculty or administrative staff; all of whom can be expected to have different perceptions and understanding of technical subjects than the average production line worker, as well as a higher level of formal education.

We did not have access to production-line workers. However, we specifically recruited subjects who had no experience working with robotics, and publicised this study further than the university campus to try and reach the widest range of people.

5.5.2 Robot behaviours

Recall from Sec. 1.1 the four modes of operation permitted by ISO 10218-1 [2]: safety-rated monitored stop (SRMS), hand guiding, speed and separation monitoring (SSM), and power and force limiting. Recall also that the methods presented in this thesis fall under the approach SSM.

The simplest mode of operation is the SRMS, which brings the robot to a stop when the human enters its workspace. We implemented this in accordance with ISO 10218-1 [2] and ISO 13855 [4], and compared it with our method. This is described in Sec. 5.5.3.

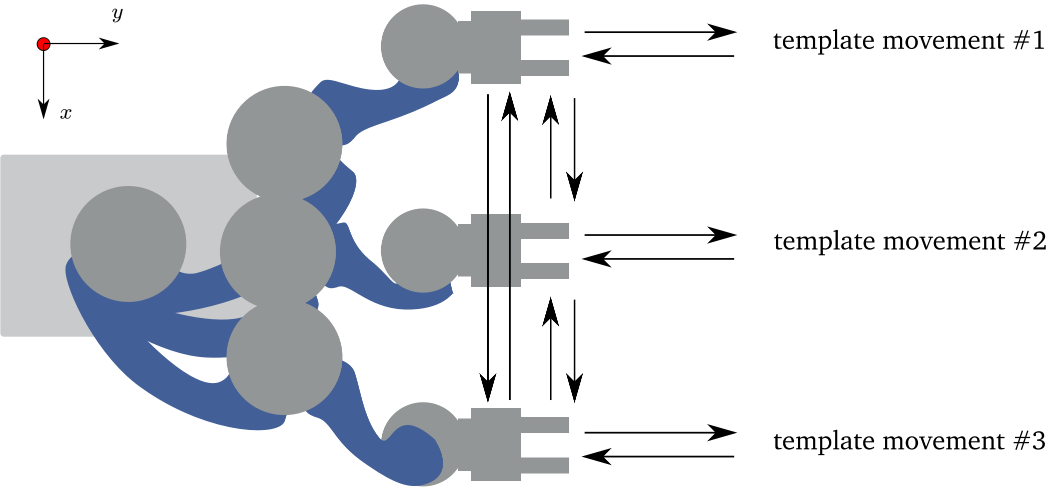


Figure 5.7: Template movements: away from the base in the y-direction for $1.7s$ and back again for $1.7s$, at three different starting positions, between which the robot moves randomly.

The predefined long-term plans of the robot were three template movements where the robot moved outward (in the y-direction) for $1.7s$ and inward again for $1.7s$, at 3 different x-axis locations, as shown in Fig. 5.7. The robot executed these movements in random order, moving between the start positions of the 3 template movements randomly, and continuously until the human had finished their task. The response of the robot to the human entering its workspace was one of the following two methods. The movements were piecewise straight-line-TCP and limited-jerk in the Cartesian space, as motivated in Sec. 5.4.1.

5.5.2.1 Safety-rated monitored stop

The SRMS was implemented as a virtual cage: The workspace of the robot was enclosed in an axis-aligned bounding box as shown in Fig. 5.8. This was extended by the safety distance S as defined in [4, eq. 2], calculated as:

$$S = (K \times T) + C, \tag{5.3}$$

where the terms are defined as follows:

S the safety distance

K the maximum speed of the human, $v_{\max,human,ISO} = 2.0\frac{m}{s}$

T the lag of the entire sensing loop, i.e. the sensor and transmission latency, control cycle Δt and robot stopping time $t_{s,\max}$. We used the maximum stopping time over the entire trajectory, $174ms$; we took latency to be $1ms$ for transmission, the control cycle was $2ms$ and the sensor latency was the frame-rate of the cameras plus the software latency given

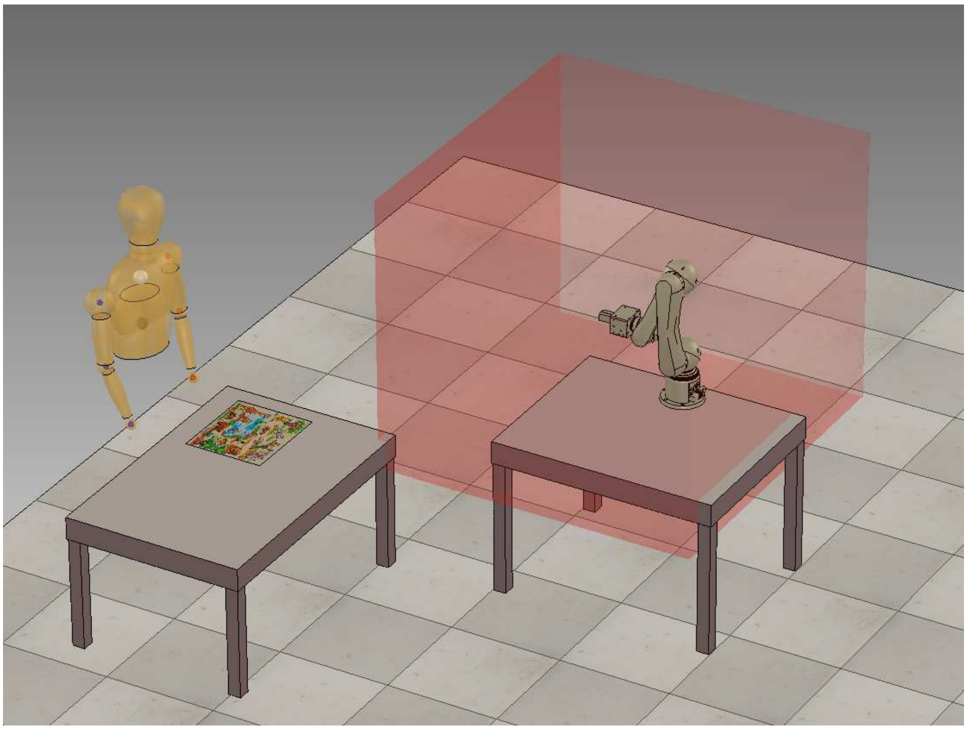


Figure 5.8: Simulation of the workspace with human, workbench, robot and red virtual cage around the robot workspace. Image from [192]

on the Vicon website [5]: $1.5ms$ for 5 objects and $2.8ms$ for 10 objects. As we had 7 objects, we took the latency at a conservative $5ms$, plus the cycle time of the Vicon Tracker, which was $4ms$. Together, this came to $186ms$.

C the penetration distance. This is the amount that the human can penetrate the co-working area without being detected, and is relevant for light curtains which have a beam resolution. The wider the beam resolution, the further the human can penetrate the light curtain curtain without being detected, and if the resolution is more than $40mm$ (i.e. the beams are $> 40mm$ apart), this is taken as the length of the human arm, $850mm$. For us, this is zero, since the cameras give the exact positions of the markers.

When any marker from the human enters the cage, the robot performs a stop, which is identical to the failsafe manoeuvre of the SSM approach described below.

5.5.2.2 Speed and separation monitoring

The SSM behaviour is as described in Chapter 2, with the short-term planning performed as in Sec. 2.2.1 subject to maximum joint accelerations of $10rad/s^2$ and jerks of $200rad/s^3$; these were found to give acceptably smooth failsafe manoeuvres. The human model used for this purpose is the ISO model described in Sec. 4.2.2, using $v_{\max,human,ISO} = 2.0\frac{m}{s}$ for fair comparison with the SRMS approach, since this speed was used for the human in (5.3).

5.5.3 Methodology

Our experiment consisted of a human subject repeatedly entering the robot's workspace while completing a puzzle task. 30 healthy subjects aged between 22 and 30 years old, 14 male and 16 female, were chosen. Two experimenters conducted each experiment; one interacting with the subject and one operating the robot. The subjects were assigned at random to one of two groups for each movement strategy (SSM or SRMS), and the experimenter interacting with the subject was not made aware of which group the subject was in. At no point did the subject see the other movement strategy (between-subject design). The subjects were split into two groups of 15; in the SRMS, for the first two subjects, the control on the robot was implemented incorrectly, leaving 13 subjects in the SRMS group and 15 subjects in the SSM group.

After reviewing and signing a declaration of consent, the subject filled out a questionnaire relating to their personal information and experience of working with robots (F0), followed by a questionnaire about their trust in automated systems (F1). The subject then watched an instruction video of the task they were to perform, and the instructor answered any questions about the task and assisted the subject to attach the markers to their body. One difference from the setup in Sec. 1.3 is the absence of the marker on the lower back, since this was too often occluded from view of the cameras. Instead, the torso was modelled as a vertically aligned

[5] `www.vicon.com/products/software/tracker`, retrieved: 26.12.2017

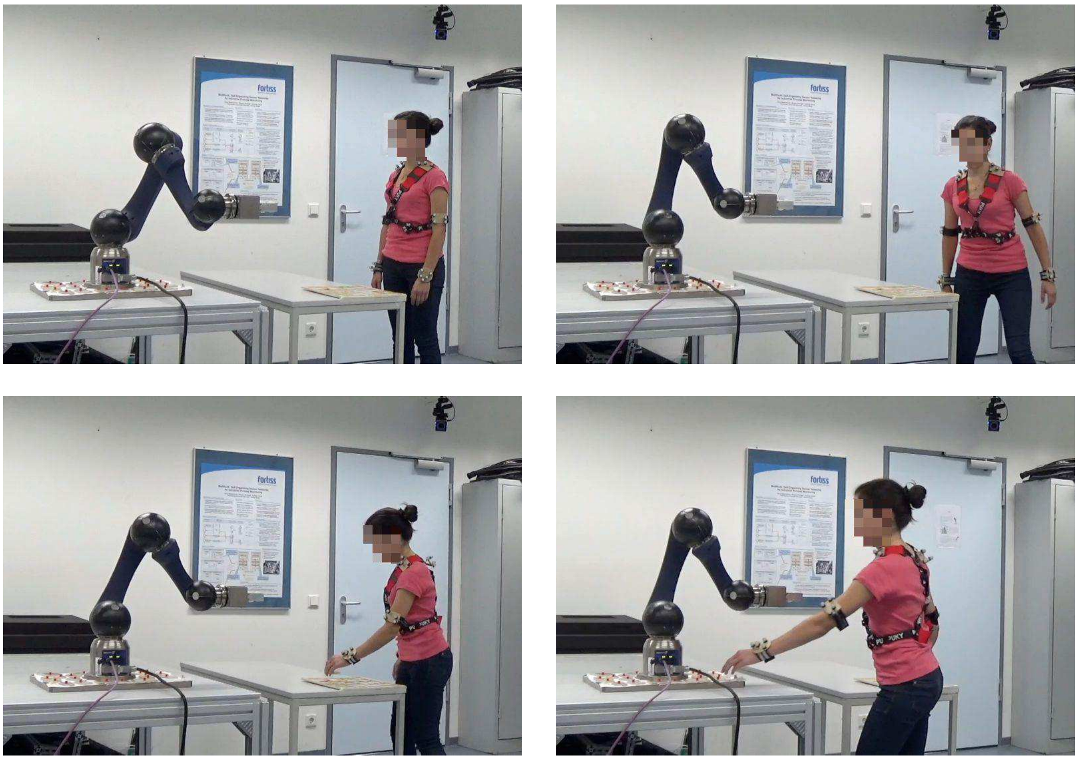
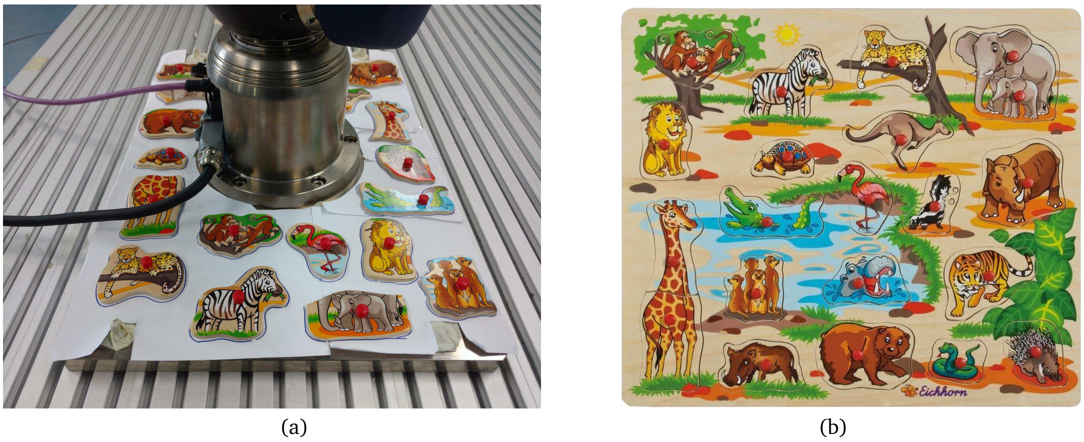


Figure 5.9: The task of the human, clockwise from top left: when instructed by the experimenter, the subject picks a piece from the robot base and places it in the puzzle on the workbench, repeating this one piece at a time until the puzzle is complete.

capsule with one defining point defined by the marker on the neck, and one defined 50cm below this defining point. In effect, the marker on the lower back was replaced by a virtual marker vertically below the marker on the neck.

Since we are interested in coexistence rather than collaboration, the human task was designed to be independent of the robot task. As shown in Fig. 5.9, the human was tasked to assemble a children's jigsaw puzzle, Fig. 5.10(a), on a table outside of the robot workspace. However, the pieces for the puzzle were in the robot workspace (the starting locations of the pieces were kept the same over all trials, shown in Fig. 5.10(b) by the robot base). In order to complete the puzzle, the subject had to enter the workspace of the robot and pick up the pieces, one at a time, return to the table, and fit the piece in the puzzle. The difficulty of the puzzle was chosen such that the human subjects should not have to think much about where to place the pieces and should be able to accustom themselves to the position of the pieces after one practice round.

The subject performed their task once with the robot not moving, in order to accustom themself to the task. This was not timed. The human then performed the task at the same time as the robot performed its task. Following this first trial, the subject filled in a questionnaire


(a)

(b)

Figure 5.10: (a) the position of the pieces at the robot base, (b) the puzzle, completed.

about co-working with the robot (F2). The subject then performed another 3 trials with exactly the same setup and task.

At each of the remaining two appointments, 4 trials were conducted. After the last trial, the subject filled in the questionnaire about co-working with the robot (F3, identical to F2), and a questionnaire about their trust in the system and their feeling of safety with this system (F4).

Human TTC was stopped with a stopwatch during the experiments, and robot idle time was defined as the nominal time of the completed robot movements subtracted from the time the robot actually took for these movements.

Subjects were compensated €5 for each of the first and second trials, and €30 for the final trial. The experiment duration lasted approximately 35 minutes for the first appointment, 20 minutes for the second and 25 minutes for the third, depending on how quickly the subject performed the task and answered the questionnaires.

5.5.4 Questionnaires

The subject filled in 5 questionnaires over the course of the 3 appointments, F0 – F4; the full text of each is documented in Appendix. A.7. F0 was personal information to find the range of ages and genders, and the subjects' prior experience with robots. F1 and F4 were to measure the trust in automation before and after, as well as the feeling of safety. We included the subquestionnaires *Propensity to trust* and *Intentions of the developers* from the "Trust in Automation" questionnaire of Körber *et al.*[155]; in F1 these were asked about automated systems in general, and in F4, about our system in particular. In F4 we also added the statements:

6 I understand how the robot works better than at the start,

8 I need more time to adapt to the behaviour of the robot, to understand how the subject feels about their own adaptation to the robot, and the statements:

7 I feel safer, when the robot waits for me,

9 It is annoying when the robot waits for me,

to gauge how the human feels about the robot waiting for them.

Although the well-known Godspeed questionnaire series [173] has a subquestionnaire on "Perceived Safety", we did not include these questions, since they were not appropriate to our experimental design. The Godspeed questionnaire (Part V) asks participants, on a scale of 1–5, to " *Please rate your emotional state on these scales: 1) Anxious–Relaxed 2) Calm–Agitated 3) Quiescent–Surprised*".

These questions do not mention the robot. Since we are studying human-robot *co-existence* rather than *collaboration*, the robot is less the focus of the human's attention and is often ignored altogether. Feedback from the previous study (Sec. 5.4) and pilot studies indicated that several participants did not take account of the robot at all, or perceived it minimally, and these participants often did not understand questions without reference to the robot as referring to their recent interaction with the robot, but rather understood them to be referring to their emotional state over the day or week in general. We believe that such a scale is confusing to the subject in HRCoex experiments, and will not result in a reliable result. Furthermore, as noted by [149], we were unable to put the humans in a position of real danger, so the chance that the robot can cause the human to be significantly agitated or anxious is minimal. Instead, we asked the human explicitly in question 7 of F4 whether they felt safer, when the robot waits for them, and also used questions from [152] on perceived safety in F2 and F3, as described next.

Questionnaires F2 and F3 were identical and were administered after the first trial with the robot, and after the last, respectively. Seven of the questions were taken from the eight used in [152] to measure the satisfaction with the robot as a team-mate, perceived safety and comfort with the robot. We omitted the question "The robot and I worked well together", because we did not want to mislead the subjects into thinking that they are collaborating with the robot on a task: again, our aim was to test co-existence rather than collaboration. The questions, and the qualities they measure, were:

1 I trusted the robot to do the right thing at the right time. (*satisfaction as team-mate/perceived safety*)

2 The robot did not understand how I wanted to do the task. (*satisfaction as team-mate*)

3 The robot kept getting in my way. (*satisfaction as team-mate*)

X The robot and I worked well together. (*satisfaction as team-mate* – not used)

4 I felt safe when working with the robot. (*perceived safety*)

5 The robot moved too fast for my comfort. (*comfort*)

6 The robot came too close to me for my comfort. (*comfort*)

7 I trusted the robot would not harm me. (*perceived safety*)

The authors of [152] classified items 1–4 as pertaining to the "satisfaction with the robot as a team-mate", and items 5–8 as pertaining to "perceived safety and comfort". We also noticed that item 1 was also concerned with perceived safety in a HRCoex scenario.

We added to these questions 4 of our own, all of which were concerned with how the subject was able to understand how the robot moved.

8 The reaction of the robot to me was easy to comprehend.

9 The robots movement in my presence was confusing.

10 I was distracted by the robots movement.

11 The robots movement surprised me.

The hypotheses to be tested in this user study are presented next.

5.5.5 Hypotheses

Our hypotheses were as follows:

Efficiency hypotheses

H1 SSM leads to less idle time of the robot compared to SRMS;

H2 SSM leads to more time for the human to complete their task than SRMS;

H3 Idle time of the robot decreases with training, for (a) SSM and (b) SRMS;

H4 Time for the human to complete their task decreases with training, for (a) SSM and (b) SRMS;

Human factors hypotheses

H5 Subjects have a greater *propensity to trust* a system they are familiar with than an unfamiliar system, for (a) SSM and (b) SRMS;

H6 Subjects exhibit more positive impressions of the *intention of the developers* in a system they are familiar with than an unfamiliar system, for (a) SSM and (b) SRMS;

H7 Subjects have a greater *understanding* of the robot's movement after training, for (a) SSM and (b) SRMS;

H8 Subjects feel more satisfied with the robot as a co-worker after training, for (a) SSM and (b) SRMS;

H9 Subjects feel more comfortable after experience, for (a) SSM and (b) SRMS;

H10 Subjects perceive the robot as safer after experience, for (a) SSM and (b) SRMS;

H11 Subjects perceive the robot as safer with SRMS than SSM, (a) before experience and (b) after experience with the robot;

H12 Subjects have a greater *understanding* of the robot's movement with SRMS than SSM, (a) before experience and (b) after experience with the robot;

H13 Subjects prefer the robot to wait for them, for (a) SSM and (b) SRMS;

H14 Subjects require more time to adapt to the robot after 12 trials, for (a) SSM and (b) SRMS.

5.5.6 Results

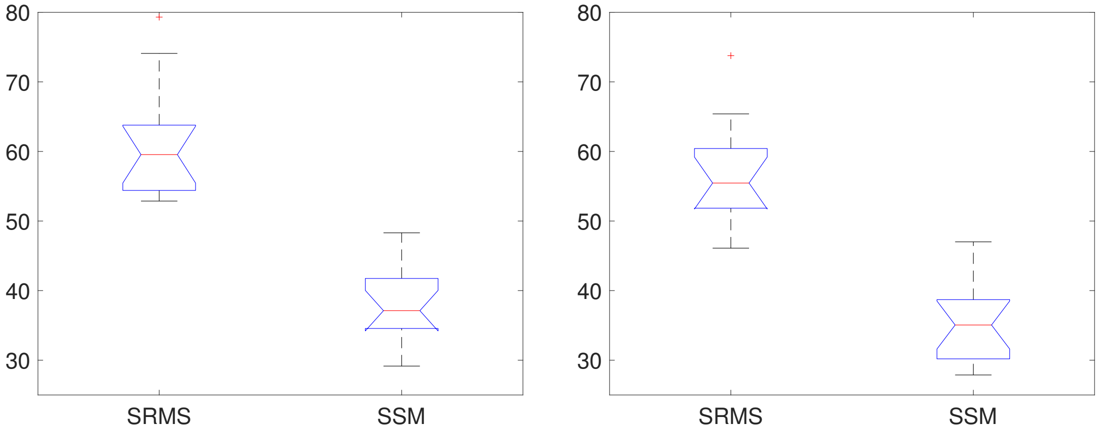
Using SSM with our formally verifying algorithm reduced robot idle time by 36% compared to the SRMS control experiments, averaged over all trials. Both human TTC and robot idle time improved as the human became accustomed to the robot. After 12 trials with the robot, subjects from the SSM group rated their understanding of the robot movement higher than after one trial, and subjects from both groups also rated perceived safety higher. After training, subjects from the SRMS group rated their feeling of safety higher than the SSM group. Below we describe the results in detail.

5.5.6.1 Efficiency hypotheses

We averaged the TTC for the human and the idle time for the robot over the last 4 trials and over the first 4 trials, and tested the data for normality with the Shapiro-Wilk test [159]. The assumption of normality did not apply to the robot times averaged over the last 4 trials ($p < 0.05$ that the data is normally distributed), so to ensure validity of the results we tested Hypotheses H1 and H2 with the Kruskal-Wallis H test [174] [6]. In each case, the robot idle time was significantly lower with SSM than SRMS. The results are shown in Tab. 5.3 and in box plots in Fig. 5.11. The percentage reduction in robot idle time in SRMS compared to SSM is 38% ($p < 10^{-5}$) over the first 4 trials and 37% ($p < 10^{-5}$) over the last 4 trials, showing that the efficiency advantage of SSM persists even after training. Over all trials averaged, the reduction was 36%. **H1 is therefore confirmed**.

For the human TTC, no significant difference was found, neither over the last nor first 4, nor over all trials. **H2 is therefore rejected**.

[6] The Kruskal-Wallis H test is the non-parametric equivalent to ANOVA which does not assume normal distribution.



(a) (b)

Figure 5.11: Box and whisker plots for (a) the average robot idle times in the first 4 trials and (b) the average robot idle times in the last 4 trials.

Hypotheses H3 and H4 concern the effect of training on the human. It is expected that training leads to lower robot idle times (H3) and human TTC (H4). The average of each subject's TTC and robot idle times were not significantly different from normal, so we applied repeated measures ANOVA on the data. For the SSM, we observed a decrease in robot idle times of 14% ($p < 10^{-4}$) and in human TTC of 22% ($p < 10^{-6}$). For SRMS, we observed a decrease in robot idle times of 11% ($p < 10^{-4}$) and in human TTC of 25% ($p < 10^{-6}$). These can be seen in Fig. 5.12 and Tab. 5.3. Hence **both H3 and H4 are confirmed**.

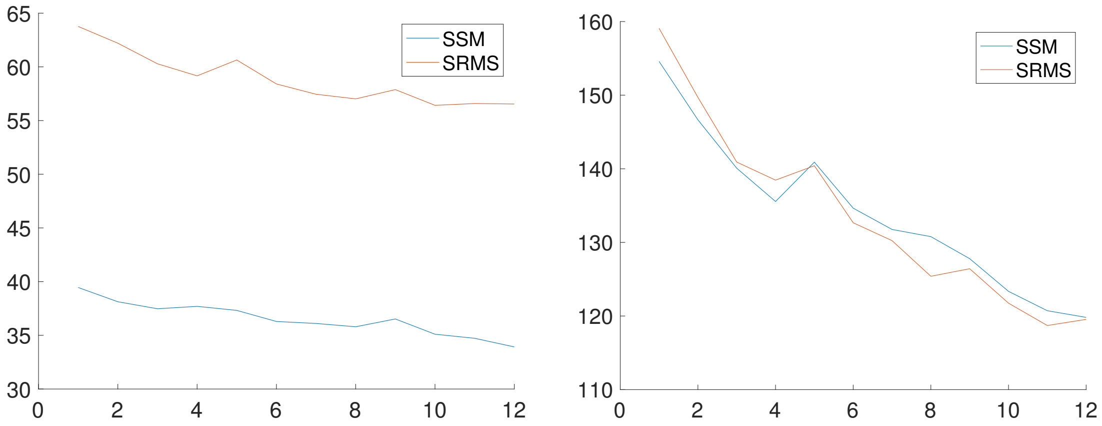
5.5.6.2 Human factors hypotheses

H5 We applied Cronbach's α [158] to test which sets of questions were sufficiently internally consistent. It turned out that the subscale of "propensity to trust" both in F1 and F4 were not internally consistent ($\alpha < 0.6$). **We could therefore not test "propensity to trust" in H5 as a quality**, and therefore looked at the questions individually. Since the data were not normally distributed, we performed Friedman's non-parametric test[7] [160]:

- Responses were significantly lower for "One should be careful with *this* system" in F4, compared to "One should be careful with *unfamiliar autonomous* systems" in F1, in SRMS ($p = 0.0067$) and in SSM ($p = 0.0005$).

- Responses were significantly higher for "I trust *a* system more than I mistrust it" in F4, compared to "I trust *this* system more than I mistrust it" in F1, in SRMS ($p = 0.0114$) and in SSM ($p = 0.0067$).

[7] Friedman's test is the non-parametric equivalent to a repeated measures ANOVA and operates on broadly the same assumptions.



(a) (b)

Figure 5.12: The development over the trials for (a) the robot idle time and (b) the human TTC.

Table 5.3: Idle time for robot and TTC for human subject (seconds)

	Robot idle time				Human TTC			
	SSM		SRMS		SSM		SRMS	
Trial	μ	σ	μ	σ	μ	σ	μ	σ
1	39.5	6.5	63.8	10.1	154.6	23.0	159.1	24.5
2	38.1	6.2	62.2	7.1	146.7	19.8	149.7	23.9
3	37.5	6.1	60.3	9.0	140.1	19.7	140.9	22.0
4	37.7	6.2	59.2	8.3	135.6	16.7	138.5	25.0
1–4 average	**38.2**	**5.5**	**61.4**	**8.2**	**144.2**	**18.9**	**147.0**	**22.9**
5	37.3	4.8	60.6	7.2	140.9	20.4	140.4	24.8
6	36.3	5.1	58.4	6.3	134.7	19.7	132.7	23.2
7	36.1	4.8	57.4	7.6	131.8	20.1	130.2	22.3
8	35.8	5.8	57.0	7.2	130.8	21.4	125.4	20.8
9	36.5	5.0	57.9	7.0	127.8	20.4	126.4	23.5
10	35.1	5.9	56.4	7.6	123.4	21.9	121.7	21.2
11	34.7	6.1	56.6	8.2	120.7	24.0	118.7	20.9
12	33.9	5.2	56.5	7.3	119.8	21.7	119.5	24.0
9–12 average	**35.1**	**5.4**	**56.9**	**7.4**	**122.9**	**21.7**	**121.6**	**22.1**

- Responses were significantly higher for “*This system* generally works well” in F4, compared to “*Automated systems* generally work well’ in F1, in SRMS ($p = 0.0339$) but **not** in SSM ($p = 0.65$).

Note the change in wording: subjects were asked about systems in general, such as the robot, in F1 (subjects were informed that “system” pertained to the robot), and then about this system, which they had then experienced for 12 tasks in total.

H6 The subscale “intention of the developers” was acceptably internally consistent at $\alpha = 0.69$. For H6(b), the data failed the Shapiro-Wilk test for normality ($p < 0.05$ that the data is normally distributed). We therefore used Friedman’s test, which showed that the score for the intention of the developers was **significantly higher**, with a median of 4.25 before (F1), and 5 after (F4). For H6(a), 6 out of 15 answers were “no response preferred”, meaning only 60% of respondents gave an answer. We therefore do not consider the data adequate to make a reliable statement about H6(a).

H7 Items 8–11 of questionnaires F2 and F3 measure subjects’ understanding of the robot. Omitting item 10 led to an α value of 0.82, which is a good level of internal consistency. For H7(a), the distribution of scores of understanding were not significantly different from normal, so we used a repeated-measures ANOVA to ascertain that **subjects rated their understanding of the robot significantly higher (**$p = 0.0207$**) after 12 trials than after 1 trial** for SSM. For H7(b), the Shapiro-Wilk test at 0.05 confidence level indicated that scores were not normally distributed; using the Friedman test we could not ascertain that subjects rated their understanding higher ($p = 0.0833$) after experience with the robot with the SRMS behaviour.

H8 and H9 In [152], the authors divided the questions into two groups, the first supposedly pertaining to user satisfaction and the second pertaining to perceived safety and comfort. However, applying Cronbach’s α to these questions yields values below 0.6, indicating that their internal consistency is low, i.e. they do not measure the same thing. Indeed, in the original study, the authors do not analyse the questions in these groups together as one measure, but rather analyse them individually.

We associated perceived safety, comfort, and satisfaction with the robot as a team-mate with items $\{1, 4, 7\}$, $\{6, 7\}$ and $\{1, 2, 3\}$ respectively (see Sec. 5.5.4). Only perceived safety had an acceptable value of $\alpha = 0.75$, the others had $\alpha < 0.6$. **We were therefore unable to verify hypotheses H8 and H9**. Similarly to H5, we examined the questions individually.

- For SSM, significant improvement was experienced in responses to “I trusted the robot to do the right thing at the right time” (p = 0.0455) and “The robot moved too fast” (p = 0.0455), and marginal significance in the responses to “I felt safe when working near the robot” ($p = 0.0956$) and “The robot came too close to me” ($p = 0.0588$).

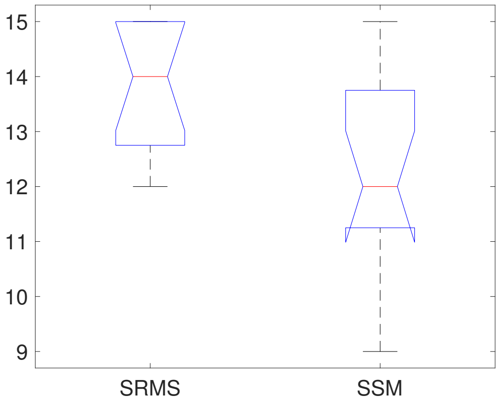


Figure 5.13: Box and whisker plots for perceived safety after 12 trials (significant difference).

- For SRMS, significant improvement was experienced in responses to "I felt safe when working near the robot" ($p = 0.0253$), "I trusted the robot would not harm me" ($p = 0.0455$), and "I was distracted by the robots movement" ($p = 0.0143$).

H10 For hypothesis H10(a), the distribution of the scores of subjects' perceived safety was not significantly different from normal according to the Shapiro-Wilk test; we therefore applied a repeated-measures ANOVA and found that **perceived safety was rated significantly higher after 12 trials compared to after 1 trial for SSM** ($p = 0.0128$). For H10(b), scores were significantly different from normal ($p < 0.05$ in the Shapiro-Wilk test), hence we performed Friedman's non-parametric test to ascertain **significant improvement for perceived safety in SRMS** ($p = 0.0082$).

H11 Hypothesis H11(a) was tested with an ANOVA and found no significant difference ($p = 0.2778$). The data after training, relevant to H11(b), was found significantly non-normally distributed using a Shapiro-Wilk test; a Kruskal-Wallis test showed a **significant improvement in perceived safety for SRMS compared to SSM** ($p = 0.0100$). This is shown in Fig. 5.13.

H12 Hypothesis H12(a) was tested with an ANOVA and no significant difference was found ($p = 0.3358$). For H12(b), we used a Kruskal-Wallis test since the data was found with a Shapiro-Wilk test to be significantly non-normally distributed; here also there was **no significant difference** between groups ($p = 0.1832$).

H13 and H14 For hypotheses H13 and H14, questions 6-9 in F4 were relevant:

6 I understand how the robot works better than at the start.

7 I feel safer, when the robot waits for me.

8 I need more time to adapt to the behaviour of the robot.

9 It is annoying when the robot waits for me.

We analysed them question-by-question. The data were not normally distributed, so we used the Kruskal-Wallis test to look for significant differences between strategies. Only the 8th statement, “I need more time to adapt to the behaviour of the robot” was marginally significantly ($p = 0.0750$) more agreed with in SRMS than SSM, however, this hypothesis should be further investigated to ascertain significance. Bar charts of the responses to each question broken down by group are found in Fig. 5.14

5.5.7 Discussion of results

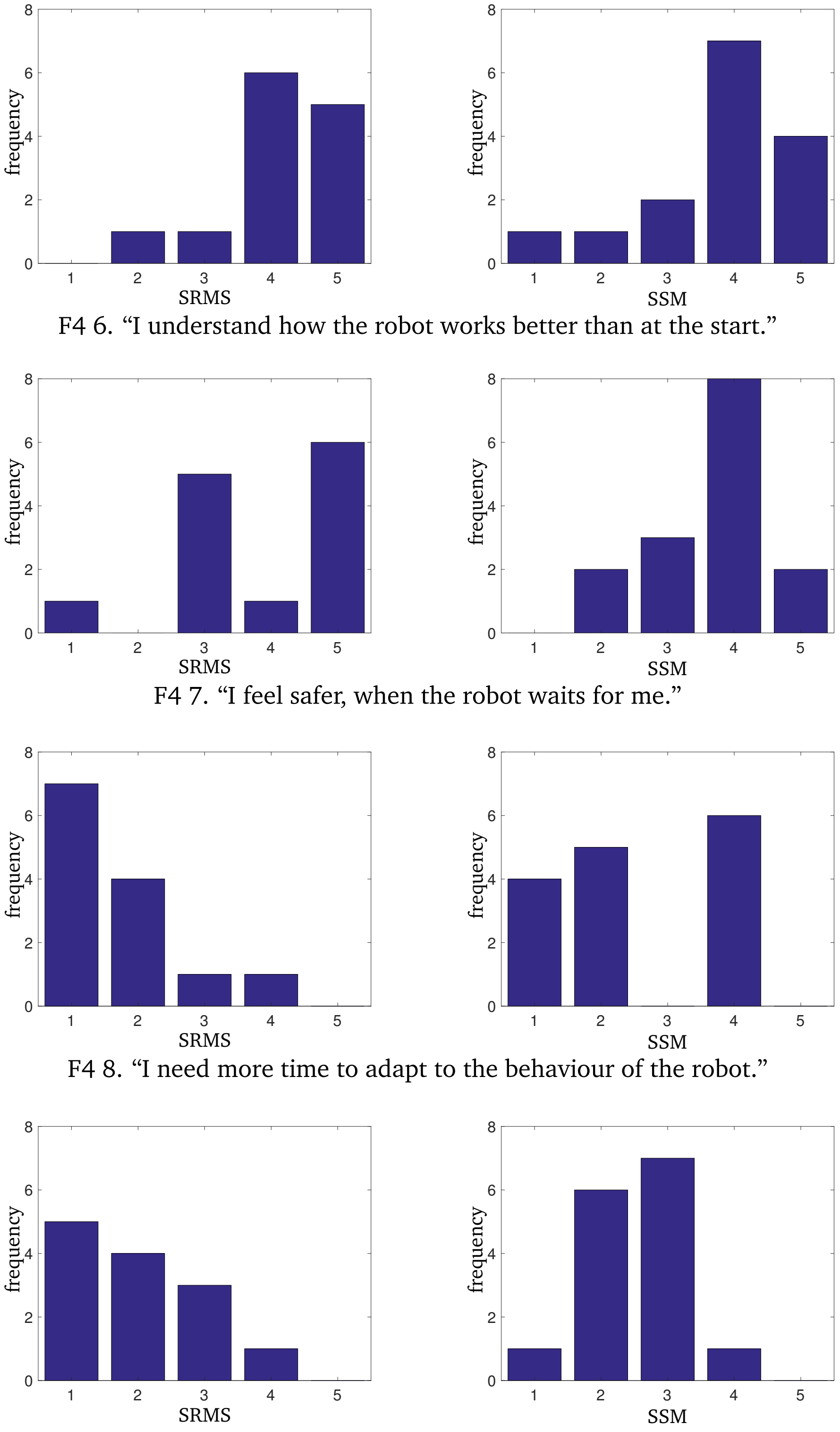
That the robot is more efficient using the formally verified method of SSM is explicable in that the robot need only alter its movement as long as the human is directly in danger of collision, whereas the SRMS behaviour stops the robot as soon as the human appears in its workspace. With practice, the human becomes more efficient and faster at their task, and spends less time in the robot’s workspace as a result, explaining the improvement with increasing number of trials of the the human TTC and the robot idle time.

Perceived safety was significantly higher in SRMS than SSM after experience with the robot, though not at the beginning. The robot stopped, on average, earlier and farther away from the human in SRMS than SSM. Previous work shows that robots should keep a certain distance, often greater than the actual safe distance, to make interacting humans feel safe and comfortable [48], and various factors such as personality traits affect how close humans comfortably interact with robots [175, 176]. In our approach, the latency of the system and the uncertainty of the sensors dictate how close the robot will come to the human, since the more latency and uncertainty, the larger and less tight the HRO and hence the greater the distance the robot will keep for the human. Since we have seen that perceived safety does improve with increased exposure to the robot, it may be that subjects simply require more training to feel completely safe with the formally verified approach (SSM), as suggested by the marginally significant result of H14.

5.5.8 User comments

Similarly to the previous study of Sec. 5.4, humans tended to ascribe behaviours to the robot. Interestingly, one subject commented that they believed the robot adapted to them, and was surprised upon learning that this was not the case. Other users suggested that the robot could be improved by learning to recognise the human’s movements and avoiding them. This could be carried out using a scheme as described in Sec. 2.3.

Another user commented that they used the noise from the robot’s harmonic drives to gauge the robot’s motion, even when they were not looking at it directly. This allowed them to work



F4 6. “I understand how the robot works better than at the start.”

F4 7. “I feel safer, when the robot waits for me.”

F4 8. “I need more time to adapt to the behaviour of the robot.”

F4 9. “It is annoying when the robot waits for me. ”

Figure 5.14: Histograms of answers to questions in F4 (administered after 12 trials with the robot, on the 3rd day of experiments), for SRMS (left) and SSM (right)

more efficiently, but would have been less of an advantage in a noisy factory environment, where humans may even be wearing ear defenders. A further observation was that the robot was moving relatively slowly and that the end-effector was blunt and seemingly non-hazardous; were this not the case, the subject would have been more apprehensive.

5.5.9 Extraneous factors

A number of extraneous factors could have affected the human's perception of safety and trust in the robot. Firstly, the harmonic drives on the robot were often noisy. On the other hand, this could be seen as an advantage of experiments with a real robot, as opposed to with video recordings–in real HRI, the physical characteristics of the robot cannot be ignored.

At one point, the low-level control of the robot malfunctioned and the data for that subject had to be omitted (this subject is not counted as one of the 30 who performed the experiment, since this occurred on the subject's first appointment and we did not schedule further appointments). As a result of this, we had to reschedule appointments; while we did not specifically tell the subjects affected that the robot control was not functioning, this may have impacted their impression of the trustworthiness of the developers.

Summary

We have discussed different human aspects of the approach – trust, predictability, efficiency, and how the human adapts to the robot over time. Our approach allows the robot to work more efficiently than the state of the art, in scenarios where the human is continually intruding into the robot's workspace. Other aspects are harder to determine. The results of the first study suggest that the robot giving way to the human as is done in the formally verified approach, with and without with a movement cue which suggests giving way, can improve trust. The longer term study appears to shown that stopping further away from the human, as in the state of the art, engenders a higher perceived safety than our method, although perceived safety, as well as understanding of the robot, increase with experience of the robot. It may be the case that with continued experience, human co-workers feel safer around their robotic colleagues.

Chapter 6

Conclusion

We have explored the challenges in allowing humans and robots to work together while guaranteeing safety properties of the robot's trajectory. We present the control concept and show how it can be used to guarantee multiple safety properties and in tandem with reactive trajectory replanning. Extreme movement of humans is analysed and modelled, and methods for predicting and bounding this movement, as well as the movement of the robot, are presented.

The current technological trends towards increased computational efficiency and more accurate, faster sensing technologies, as well as trends in production requiring humans to interact more closely with machines, will drive this concept of verified safety forward. Computational power is already sufficient to implement such verified safety, and while the sensing modalities used in this thesis are unusual for an industrial scenario, improvements in sensing technology should make fast, accurate and reliable sensing less of a hurdle to its efficacy. Industrially certifiable sensors such as the Safety Eye (Pilz GmbH & Co. KG), while compatible with this approach, currently have high latencies that limit the effectiveness. It is shown in experiments in this thesis that our method improves robot efficiency by 36% compared to a virtual cage; it is likely that implementing this approach with industrially certified sensors could also improve efficiency, though it is uncertain as to what extent.

Furthermore, trends in the "*Industrie 4.0*" include increased presence of automation in lighter manufacturing industries, for example in small and medium-sized enterprises (SMEs), where human dexterity and intelligence may be needed alongside robot strength and repeatability. Increased customisation of products and the necessary increase in production flexibility means that cages and situation-specific risk-prevention measures become cumbersome; a robot that perceives its surroundings through sensors can autonomously detect human presence and certify itself safe wherever it is introduced on the production line. Such autonomous self-certification has the potential to transform the manufacturing industry as well as any field where humans work alongside machines.

6.1 Further Work

This work is a first step–from here there are several roads forward. More accurate, in-depth, and faster modelling of human movement including improved accounting for the lower body may be investigated. The prediction of the robot reachable occupancy (RRO) using Occupancy Capsules presented in Sec. 3.4 should be extended to account for uncertainty in measurement and in the future trajectory. Long-term trajectory planning using probabilistic predictions of the human has been described in this thesis (Sec. 2.3.1); its implementation, including the question of when to replan an interrupted trajectory, is a multifaceted problem which is worth investigating.

The human aspect is also non-negligible. Well-placed trust is key to the uptake of the system for all stakeholders: the system integrator, the factory manager as well as the worker who works alongside the robot. More research is needed to make sure the robot works well for the very people it exists to serve, especially long-term studies and research in production environments with factory workers. Outreach to certification agencies and integrators can determine what additional research is needed to raise the technology readiness level of this work.

Alternatively, verification can be applied to different behaviours of the robot. We have considered normal operation with controlled stops (Category 2); an interesting and more complex problem is conservatively predicting the behaviour of the robots during braking and emergency stops (Category 0 and 1). Here the movement dynamics is hybrid and nonlinear; the torque exerted on the robot joints by the brakes is dependant on time, temperature, and break wear. A start has been made on this fascinating topic in [193], by modelling the robot as a hybrid automaton with uncertain parameters and dynamics.

Certainly, as robots become ever more a part of the human world, guarantees of safety become increasingly important and research in this field will broaden and deepen. This dissertation only sets the foundations, and demonstrates with hardware available today, that we can guarantee safe robot motion.

Appendices

A.1 Planning Time-Scaling Manoeuvres

The planning method is based on decision trees and is very fast. The authors of [46] present 10 cases, or “trajectory types” according to the boundary constraints and the constraints on the trajectory. Since the variable we are planning in is $\dot{s}$, we have bounds on the first and second derivative (i.e. $\ddot{s}$ and $\dddot{s}$) over the trajectory, $|\ddot{s}| \leq \ddot{s}_m$ and $|\dddot{s}| \leq \dddot{s}_m$ as well as boundary values of $\dot{s} = \dot{s}_0$ and $\ddot{s} = \ddot{s}_0$ at the start, and $\dot{s} = \eta$ and $\ddot{s} = 0$ at the end. This is a “Type IV” trajectory in [46]. The method is presented in Alg. 13 without explanation, since a derivation and comprehensive explanation is found in [46]. The equals sign ($=$) denotes a comparison that the difference is less than an ϵ value, to mitigate floating point errors.

The planning returns 6 values: 3 time values t_{01}, t_{12} and t_{23}, and 3 values of $\dddot{s}$, $\dddot{s}_{01}$, $\dddot{s}_{12}$ and $\dddot{s}_{23}$, to be applied for these times. This is illustrated in Fig. A.1.

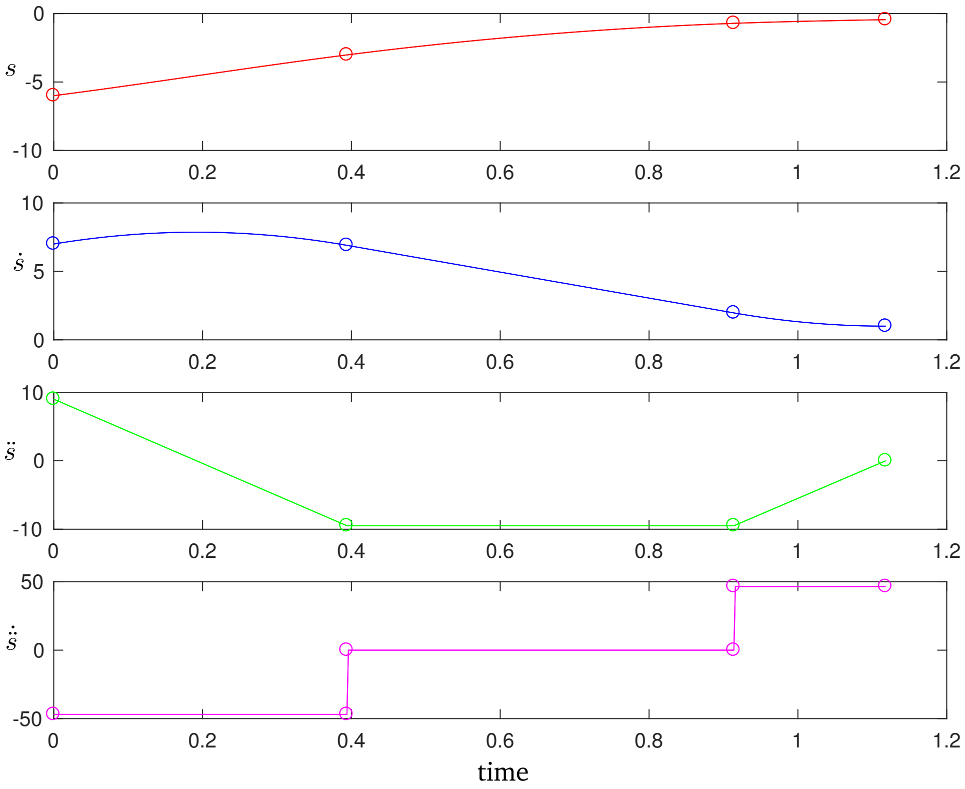


Figure A.1: The time-scaling manoeuvre planned in $\dot{s}$. Only the jerk is discontinuous, and three constant values of jerk are constantly applied for a period of t_{01}, t_{12} and t_{23} respectively. Circles demarcate the boundaries in the jerk regimes.

Algorithm 13 Obtaining a trajectory in $\dot{s}$ subject to constraints on the derivatives and boundary conditions

Input: bounds $\ddot{s}_m$, $\dddot{s}_m$, initial conditions $\dot{s}_0$, $\ddot{s}_0$, final condition η, timestep Δt

Output: $t_{01}, t_{12}, t_{23}, \dddot{s}_{01}, \dddot{s}_{12}, \dddot{s}_{23}$

```
$t_{01}, t_{12}, t_{23}, v_{01}, v_{12}, v_{23}, \dddot{s}_{01}, \dddot{s}_{12}, \dddot{s}_{23} \leftarrow 0$
if $(\dot{s}_0 = \eta) \wedge (\ddot{s}_0 = 0)$ then return
$t^* \leftarrow$ roundToTimestep($\frac{|\ddot{s}_0|}{\dddot{s}_m}, \Delta t$)
$v^* \leftarrow \dot{s}_0 + \frac{\ddot{s}_0 t^*}{2}$
if $v^* = \eta$ then
    $t_{01} \leftarrow t^*$
    $\dddot{s}_{01} \leftarrow -\frac{\ddot{s}_0}{t^*}$
    return
if $\ddot{s} > 0$ then
    if $v^* > \eta$ then
        $d \leftarrow \dot{s}_0 - \frac{2\ddot{s}_m^2 - \ddot{s}_0^2}{2\dddot{s}_m}$
    else if $v^* < \eta$ then
        $d \leftarrow \dot{s}_0 + \frac{2\ddot{s}_m^2 - \ddot{s}_0^2}{2\dddot{s}_m}$
else
    if $v^* > \eta$ then
        $d \leftarrow \dot{s}_0 - \frac{2\ddot{s}_m^2 - \ddot{s}_0^2}{2\dddot{s}_m}$
    else if $v^* < \eta$ then
        $d \leftarrow \dot{s}_0 + \frac{2\ddot{s}_m^2 - \ddot{s}_0^2}{2\dddot{s}_m}$
$t_{01}, t_{12}, t_{23}, v_{01}, v_{12}, v_{23}, a^*, d \leftarrow$ subroutine($\dddot{s}_m, \ddot{s}_m, \eta, \ddot{s}_0, \Delta t, d$)
$v' \leftarrow \eta - \dot{s}_0 - v_{01} - v_{12} - v_{23}$
$a^* \leftarrow a^* + \frac{2v'}{t_{01} + 2t_{12} + t_{23}}$
$j_{01} \leftarrow \frac{a^* - \ddot{s}_0}{t_{01}}$
$j_{23} \leftarrow \frac{-a^*}{t_{23}}$ return
```

Algorithm 14 `subroutine`

```
Input: $\dddot{s}_m, \ddot{s}_m, \eta, \ddot{s}_0, \Delta t, d$
Output: $t_{01}, t_{12}, t_{23}, v_{01}, v_{12}, v_{23}, a^*$
if $d > \eta$ then
    $a^* \leftarrow -\ddot{s}_m$
    $t_{01} \leftarrow$ roundToTimestep($\frac{a^* - \ddot{s}_0}{\dddot{s}_m}, \Delta t$)
    $t_{23} \leftarrow$ roundToTimestep($\frac{a^*}{\dddot{s}_m}, \Delta t$)
    $v_{01} \leftarrow (\ddot{s}_0 + a^*)\frac{t_{01}}{2}$
    $v_{23} \leftarrow a^* \frac{t_{23}}{2}$
    $t_{12} \leftarrow$ roundToTimestep($\frac{\eta - \dot{s}_0 - v_{01} - v_{23}}{a^*}, \Delta t$)
    $v_{12} \leftarrow a^* t_{12}$
else
    $a^* \leftarrow -\sqrt{\left|(\dot{s}_0 - \eta)\dddot{s}_m + \frac{\ddot{s}_0^2}{2}\right|}$
    $t_{01} \leftarrow$ roundToTimestep($\frac{a^* - \ddot{s}_0}{\dddot{s}_m}, \Delta t$)
    $t_{23} \leftarrow$ roundToTimestep($\frac{a^*}{\dddot{s}_m}, \Delta t$)
    $v_{01} \leftarrow (\ddot{s}_0 + a^*)\frac{t_{01}}{2}$
    $v_{23} \leftarrow a^* \frac{t_{23}}{2}$
    $t_{12}, v_{12} \leftarrow 0$
```

Algorithm 15 `roundToTimestep`

```
Input: $t_{unrounded}, \Delta t$
Output: $t_{rounded}$
$t_{rounded} = \left\lceil \frac{|t_{unrounded}|}{\Delta t} \right\rceil \Delta t$
```

A.2 Finding Fastest-Moving Point on Robot

We consider the i^{th} link of the robot to be enclosed in a capsule $\mathcal{U}_i$. We find the velocity of the fastest point on $\mathcal{U}_i$, for each link, then take the maximum of those.

See Fig. A.2. The instantaneous translational velocity $\boldsymbol{v}$ and angular velocity $\boldsymbol{\omega}$ of the capsule can be found at point $\boldsymbol{p}_2$ (without loss of generality) from the geometric Jacobian $J_{\boldsymbol{p}_2}$ and instantaneous joint velocities $\dot{\boldsymbol{q}}$ as $J_{\boldsymbol{p}_2}\dot{\boldsymbol{q}} = [\dot{\boldsymbol{x}}^\top \boldsymbol{\omega}^\top]^\top$.

In pure translation, $\boldsymbol{\omega} = \mathbf{0}$; this is a special case in which case all points of $\mathcal{U}_i$ are travelling with the same speed, i.e. $\|\dot{\boldsymbol{x}}\|$. Otherwise, the translational and rotational velocities can be expressed as a twist relative to a screw axis, defined by a normalised direction $\boldsymbol{n} = \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$ and an offset from the origin $\boldsymbol{o}$. The twist can be decomposed into a translational velocity $v\boldsymbol{n}$ along that axis and an angular velocity $\boldsymbol{\omega} = \omega\boldsymbol{n}$ around the axis. By finding $\boldsymbol{o}$, we can calculate the fastest-moving point on the capsule.

We decompose $\dot{\boldsymbol{x}}$ into its component parallel to the twist axis, $\dot{\boldsymbol{x}}_\parallel = v\boldsymbol{n}$, defined by $(\dot{\boldsymbol{x}}^\top\boldsymbol{n})\boldsymbol{n}$, and the component perpendicular to the twist axis $\dot{\boldsymbol{x}}_\perp = \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_\parallel$. The latter is related to $\boldsymbol{o}$ in the following way:

$$\boldsymbol{v}_\perp = \boldsymbol{\omega} \times (\boldsymbol{p}_2 - \boldsymbol{o})$$

We consider the cross product as multiplication with a skew-symmetric matrix $S(\omega)$; rearranging, (since $\boldsymbol{o}$ is non-unique) we can choose any of the solutions to the system of equations $S(\omega)\boldsymbol{o} = S(\omega)\boldsymbol{p}_2 - \boldsymbol{v}_\perp$.

Now that we have calculated the twist, we can calculate the fastest moving point on $\mathcal{U}_i$. The velocity of a point $\boldsymbol{p} \in \mathcal{U}_i$ is given as:

$$\dot{\boldsymbol{p}} = \underbrace{v\boldsymbol{n}}_{\text{translational term}} + \underbrace{\omega\boldsymbol{n} \times (\boldsymbol{p} - \boldsymbol{o})}_{\text{angular term}}.$$

Since the translational and angular terms are orthogonal, the magnitude of the velocity is

$$\|\dot{\boldsymbol{p}}\| = \sqrt{v^2 + \omega\|\boldsymbol{n} \times (\boldsymbol{p} - \boldsymbol{o})\|^2}. \tag{A.1}$$

Observe that $\|\boldsymbol{n} \times (\boldsymbol{p} - \boldsymbol{o})\|$ is the perpendicular distance between $\boldsymbol{p}$ and the twist axis; this attains a maximum at the farthest point of the capsule from the twist axis. Recall that the capsule is the Minkowski sum of $\mathcal{B}(0\,;r)$ and the line segment $\overline{\boldsymbol{p}_1\,\boldsymbol{p}_2}$; in other words, the set of points that are no greater than a distance r from the line segment $\overline{\boldsymbol{p}_1\,\boldsymbol{p}_2}$. The distance from the twist axis to the farthest point on the capsule will then be the distance to the farthest point on $\overline{\boldsymbol{p}_1\,\boldsymbol{p}_2}$, plus r. By convexity, the farthest point on $\overline{\boldsymbol{p}_1\,\boldsymbol{p}_2}$ from the twist axis is either at $\boldsymbol{p}_1$ or $\boldsymbol{p}_2$. Hence (A.1) attains a maximum over the capsule $\mathcal{U}_i$ at:

$$\max_{\boldsymbol{p}\in\mathcal{U}_i} \|\dot{\boldsymbol{p}}\| = \sqrt{v^2 + \omega(\max(\|\boldsymbol{n} \times \boldsymbol{p}_1 - \boldsymbol{o}\|, \|\boldsymbol{n} \times \boldsymbol{p}_2 - \boldsymbol{o}\|) + r)^2}$$

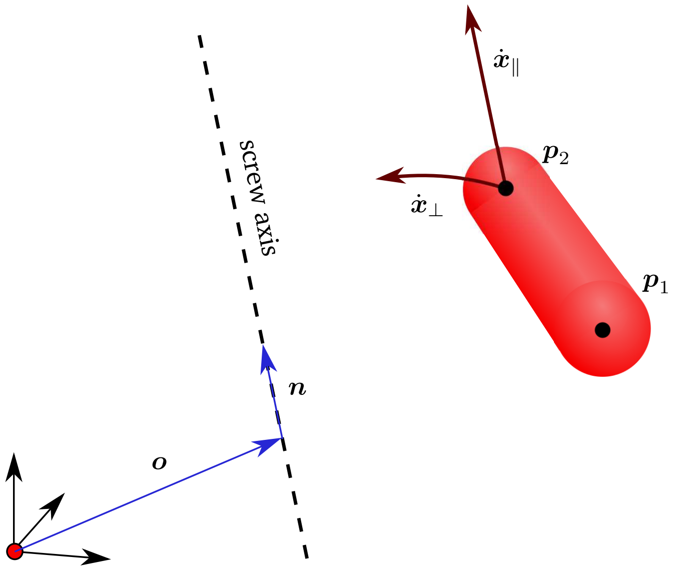


Figure A.2: Calculating the fastest moving point on the robot capsule. The dashed line is the screw axis, with normal vector $\boldsymbol{n}$ and offset $\boldsymbol{o}$. The defining points of the capsule are $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, and $\dot{\boldsymbol{x}}_{\perp}$ and $\dot{\boldsymbol{x}}_{\|}$ are the components of the velocity of $\boldsymbol{p}_2$ which are perpendicular and parallel to the screw axis respectively. The fastest moving point is the furthest one from the axis. .

A.3 Polynomial Trajectories

We show how the replan trajectories are planned, continuous with the current trajectory of the robot and respecting joint acceleration and jerk limits.

The replan trajectory starting at time parameter s_0 at start joint positions, velocities and accelerations $\boldsymbol{q}_{start} \in \mathcal{Q}$, $\dot{\boldsymbol{q}}_{start} \in \mathbb{R}^m$ and $\ddot{\boldsymbol{q}}_{start} \in \mathbb{R}^m$, passing though via points $\boldsymbol{q}_{v,1} \in \mathcal{Q}$ and $\boldsymbol{q}_{v,2} \in \mathcal{Q}$ at time parameters s_1 and s_2 respectively, and ending at goal position $\boldsymbol{q}_{goal} \in \mathcal{Q}$ at time parameter s_3, has the following constraints:

$$\begin{aligned}
\boldsymbol{\xi}(s_0) &= \boldsymbol{q}_{start} \\
\dot{\boldsymbol{\xi}}(s_0) &= \dot{\boldsymbol{q}}_{start} \\
\ddot{\boldsymbol{\xi}}(s_0) &= \ddot{\boldsymbol{q}}_{start} \\
\boldsymbol{\xi}(s_1) &= \boldsymbol{q}_{v,1} \\
\boldsymbol{\xi}(s_2) &= \boldsymbol{q}_{v,2} \\
\boldsymbol{\xi}(s_3) &= \boldsymbol{q}_{goal} \\
\dot{\boldsymbol{\xi}}(s_3) &= \boldsymbol{0} \\
\ddot{\boldsymbol{\xi}}(s_3) &= \boldsymbol{0}
\end{aligned}$$

These constraints can be satisfied by a piecewise polynomial trajectory consisting of a quartic

$\boldsymbol{\xi}_1(s)$ from s_0 to s_1, a cubic $\boldsymbol{\xi}_2(s)$ from s_1 to s_2 and a quartic $\boldsymbol{\xi}_3(s)$ from s_2 to s_3. The advantage of such low-degree polynomials is that now analytical expressions for the acceleration and jerk can be found, in low-degree polynomials. The constraints on the individual polynomials are:

$$\boldsymbol{\xi}_1(s_0) = \quad \boldsymbol{q}_{start} \tag{A.2}$$

$$\dot{\boldsymbol{\xi}}_1(s_0) = \quad \dot{\boldsymbol{q}}_{start} \tag{A.3}$$

$$\ddot{\boldsymbol{\xi}}_1(s_0) = \quad \ddot{\boldsymbol{q}}_{start} \tag{A.4}$$

$$\boldsymbol{\xi}_1(s_1) = \quad \boldsymbol{\xi}_2(s_1) = \boldsymbol{q}_{v,1} \tag{A.5}$$

$$\dot{\boldsymbol{\xi}}_1(s_1) = \quad \dot{\boldsymbol{\xi}}_2(s_1) \tag{A.6}$$

$$\ddot{\boldsymbol{\xi}}_1(s_1) = \quad \ddot{\boldsymbol{\xi}}_2(s_1) \tag{A.7}$$

$$\boldsymbol{\xi}_2(s_2) = \quad \boldsymbol{\xi}_3(s_2) = \boldsymbol{q}_{v,2} \tag{A.8}$$

$$\dot{\boldsymbol{\xi}}_2(s_2) = \quad \dot{\boldsymbol{\xi}}_3(s_2) \tag{A.9}$$

$$\ddot{\boldsymbol{\xi}}_2(s_2) = \quad \ddot{\boldsymbol{\xi}}_3(s_2) \tag{A.10}$$

$$\boldsymbol{\xi}_3(s_3) = \quad \boldsymbol{q}_{goal} \tag{A.11}$$

$$\dot{\boldsymbol{\xi}}_3(s_3) = \quad \mathbf{0} \tag{A.12}$$

$$\ddot{\boldsymbol{\xi}}_3(s_3) = \quad \mathbf{0} \tag{A.13}$$

The coefficients of the polynomials in terms of s_0, s_1, s_2 and s_3 can be found analytically. We express:

$$\boldsymbol{\xi}_1 = \boldsymbol{a}_0 + \boldsymbol{a}_1 s + \boldsymbol{a}_2 s^2 + \boldsymbol{a}_3 s^3 + \boldsymbol{a}_4 s^4$$

$$\boldsymbol{\xi}_2 = \boldsymbol{b}_0 + \boldsymbol{b}_1 s + \boldsymbol{b}_2 s^2 + \boldsymbol{b}_3 s^3$$

$$\boldsymbol{\xi}_3 = \boldsymbol{c}_0 + \boldsymbol{c}_1 s + \boldsymbol{c}_2 s^2 + \boldsymbol{c}_3 s^3 + \boldsymbol{c}_4 s^4$$

$$\boldsymbol{a}_0, ..., \boldsymbol{a}_4, \boldsymbol{b}_0, ..., \boldsymbol{b}_3, \boldsymbol{c}_0, ..., \boldsymbol{c}_4 \in \mathbb{R}^m$$

Letting:

$$Q = [\boldsymbol{q}_{start}, \dot{\boldsymbol{q}}_{start}, \ddot{\boldsymbol{q}}_{start}, \boldsymbol{q}_{v,1}, \mathbf{0}, \mathbf{0}, \boldsymbol{q}_{v,1}, \boldsymbol{q}_{v,2}, \mathbf{0}, \mathbf{0}, \boldsymbol{q}_{v,2}, \boldsymbol{q}_{goal}, \mathbf{0}, \mathbf{0}],$$

and constructing the matrix:

$$S = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_0 & 1 & 0 & s_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_0^2 & 2s_0 & 2 & s_1^2 & 2s_1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_0^3 & 3s_0^2 & 6s_0 & s_1^3 & 3s_1^2 & 6s_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_0^4 & 4s_0^3 & 12s_0^2 & s_1^4 & 4s_1^3 & 12s_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & s_1 & s_2 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -2s_1 & -2 & s_1^2 & s_2^2 & 2s_2 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -3s_1^2 & -6s_2 & s_1^3 & s_2^3 & 3s_2^2 & 6s_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & s_2 & s_3 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2s_2 & -2 & s_2^2 & s_3^2 & 2s_3 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3s_2^2 & -6s_2 & s_2^3 & s_3^3 & 3s_3^2 & 6s_3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4s_2^3 & -12s_2^2 & s_2^4 & s_3^4 & 4s_3^3 & 12s_3^2
\end{bmatrix}$$

Our constraints are then satisfied by the equation:

$$AS = Q, \tag{A.14}$$

Where:

$$A = [\boldsymbol{a}_0, \boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \boldsymbol{a}_4, \boldsymbol{b}_0, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3, \boldsymbol{c}_4],$$

The determinant of S is $-48(s_0 - s_1)^5(s_1 - s_2)^3(s_2 - s_3)^5(s_0 - 4s_1 + s_2 + 2s_3)$, so as long as $s_3 > s_2 > s_1 > s_0$ and $s_0 - 4s_1 + s_2 + 2s_3 \neq 0$, S is invertible. Postmultiplying (A.14) by S^{-1} gives us:

$$A = QS^{-1}$$

Having the coefficients of the trajectory polynomials $\boldsymbol{\xi}_j, j \in \{1, 2, 3\}$ in terms of s_0, s_1, s_2 and s_3, the constraints on joint acceleration and jerk can be modelled as a collection of functions in s, s_0, s_1, s_2 and s_3; s_0, s_1, s_2 and s_3 can then be chosen such that the constraints are fulfilled.

However, finding the trajectory of least duration while guaranteeing that the accelerations and jerks do not exceed the maxima involves minimising $s_3 - s_0$ subject to multivariate nonlinear inequality constraints (i.e., the jerk and acceleration constraints). Although methods for solving such nonlinear programming problems exist, [177], these require the constraint polynomials to be concave or transformed into concave functions, which is computationally expensive.

Since optimising the trajectory duration is not a high priority, we simplify this problem and stipulate $s_1 - s_0 = s_2 - s_1 = s_3 - s_2 = \Delta s$. We then choose a starting value for Δs heuristically

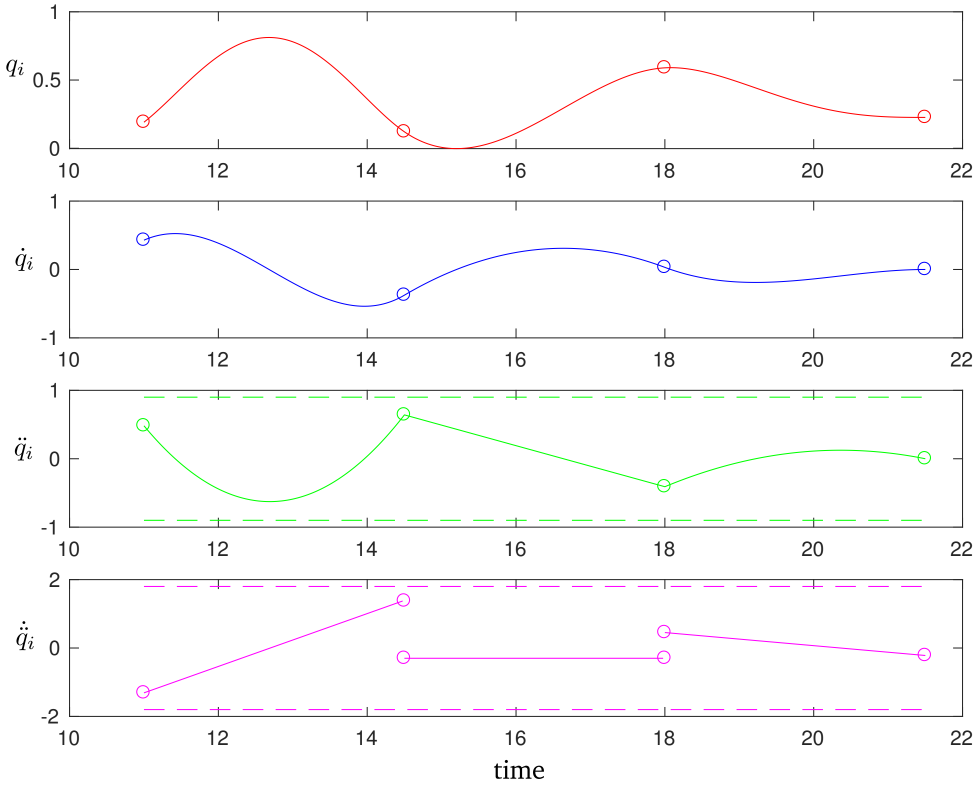


Figure A.3: The piecewise-polynomial joint space trajectory for joint i. In the acceleration and jerk, the limits are also shown as dashed lines. The circles are where the transition between polynomials occurs.

and solve iteratively for a solution which satisfies the joint acceleration and jerk constraints. We can calculate the coefficients for the $\boldsymbol{\xi}_j, j \in \{1,2,3\}$ which satisfy (A.2) - (A.13)

We derive analytical expressions for $\ddot{\boldsymbol{\xi}}_j$ and $\dddot{\boldsymbol{\xi}}_j, j \in \{1,2,3\}$ in terms of Δs and s; when Δs is fixed these are simply polynomials in s, see Fig. A.3. Since $\dddot{\boldsymbol{\xi}}_1$, $\dddot{\boldsymbol{\xi}}_3$ and $\ddot{\boldsymbol{\xi}}_2$ are linear, they have their extrema at s_0 and s_1, s_2 and s_3, and s_1 and s_2, respectively. $\dddot{\boldsymbol{\xi}}_2$ is constant. $\ddot{\boldsymbol{\xi}}_1$ and $\ddot{\boldsymbol{\xi}}_3$ are quadratic, meaning that, for $\ddot{\boldsymbol{\xi}}_1$ the extrema are either at s_0 and s_1 or, if any element of $\ddot{\boldsymbol{\xi}}_1$ has a minimum at $s^* \in [s_0, s_1]$, at this point s^* also (the same for $\ddot{\boldsymbol{\xi}}_3$, *mutatis mutandis*).

We therefore can find Δs such that the joint acceleration and jerk limits are satisfied as shown in Alg. 16, where $\xi_{i,j}$ is the i^{th} element of $\boldsymbol{\xi}_j$, where $\boldsymbol{a}_{\max}$, $\boldsymbol{a}_{\min}$, $\boldsymbol{j}_{\max}$ and $\boldsymbol{j}_{\min}$ are the vectors of maximum positive and negative joint acceleration and jerk, and $a_{i,\max}$, $a_{i,\min}$, $j_{i,\max}$ and $j_{i,\min}$ are the values of their i^{th} element.

Notice that $\ddot{\boldsymbol{\xi}}_1(s_1) = \ddot{\boldsymbol{\xi}}_2(s_1)$ and $\ddot{\boldsymbol{\xi}}_3(s_2) = \ddot{\boldsymbol{\xi}}_2(s_2)$, so we need only check acceleration of one of these. Also, $\ddot{\boldsymbol{\xi}}_3(s_3) = \mathbf{0}$

Algorithm 16 Planning a piecewise polynomial trajectory subject to constraints on acceleration and jerk

```
Input: $\boldsymbol{q}_{start}$, $\dot{\boldsymbol{q}}_{start}$, $\ddot{\boldsymbol{q}}_{start}$, $\boldsymbol{q}_{goal}$, $\boldsymbol{q}_{v,1}$, $\boldsymbol{q}_{v,2}$, $s_0$, $\boldsymbol{a}_{\max}$, $\boldsymbol{a}_{\min}$, $\boldsymbol{j}_{\max}$, $\boldsymbol{j}_{\min}$
Output: $\Delta s$
$\Delta s \leftarrow$ starting_guess
$s_1, s_2, s_3 \leftarrow$ calculate_times$(\Delta s, s_0)$
$S \leftarrow$ construct_S$(s_{0,\dots,3})$
$Q \leftarrow$ construct_Q$(\boldsymbol{q}_{start}, \dot{\boldsymbol{q}}_{start}, \ddot{\boldsymbol{q}}_{start}, \boldsymbol{q}_{goal}, \boldsymbol{q}_{v,1}, \boldsymbol{q}_{v,2})$
$[\boldsymbol{a}_{0,\dots,4}, \boldsymbol{b}_{0,\dots,3}, \boldsymbol{c}_{0,\dots,4}] \leftarrow QS^{-1}$
for $j \in \{1, ..., max\_iterations\}$ do
    for $i \in \{1, ..., m\}$ do
        $\emptyset \leftarrow accs\_to\_check$
        $\emptyset \leftarrow jerks\_to\_check$
        append_to_list$(accs\_to\_check, \ddot{\xi}_{i,1}(s_0))$
        append_to_list$(accs\_to\_check, \ddot{\xi}_{i,2}(s_1))$
        append_to_list$(accs\_to\_check, \ddot{\xi}_{i,2}(s_2))$
        append_to_list$(jerks\_to\_check, \dddot{\xi}_{i,1}(s_0))$
        append_to_list$(jerks\_to\_check, \dddot{\xi}_{i,1}(s_1))$
        append_to_list$(jerks\_to\_check, \dddot{\xi}_{i,2}(s_2))$
        append_to_list$(jerks\_to\_check, \dddot{\xi}_{i,3}(s_2))$
        append_to_list$(jerks\_to\_check, \dddot{\xi}_{i,3}(s_3))$
        if $\exists\, s^* \in [s_0, s_1]$ such that $\dddot{\xi}_1 = 0$ then
            append_to_list$(accs\_to\_check, \ddot{\xi}_{i,1}(s^*))$
        if $\exists\, s^* \in [s_2, s_3]$ such that $\dddot{\xi}_3 = 0$ then
            append_to_list$(accs\_to\_check, \ddot{\xi}_{i,3}(s^*))$
        $in\_limits(i) \leftarrow (\max(accs\_to\_check) \leq a_{i,\max}) \wedge (\min(accs\_to\_check) \geq a_{i,\min})$
        $\wedge(\max(jerks\_to\_check) \leq j_{i,\max}) \wedge (\min(jerks\_to\_check) \geq j_{i,\min})$
    if $in\_limits(1) \wedge in\_limits(2) \wedge ... \wedge in\_limits(m)$ then
        return $\Delta s$
    $\Delta s \leftarrow \Delta s + \delta s$
```

A.4 Optimal-Volume Ball and Capsule Enclosure

Lemma A.1: The operator BE encloses $\mathcal{B}_1 = \mathcal{B}(\boldsymbol{p}_1; r_1)$ and $\mathcal{B}_2 = \mathcal{B}(\boldsymbol{p}_2; r_2)$ in a ball of least volume. □

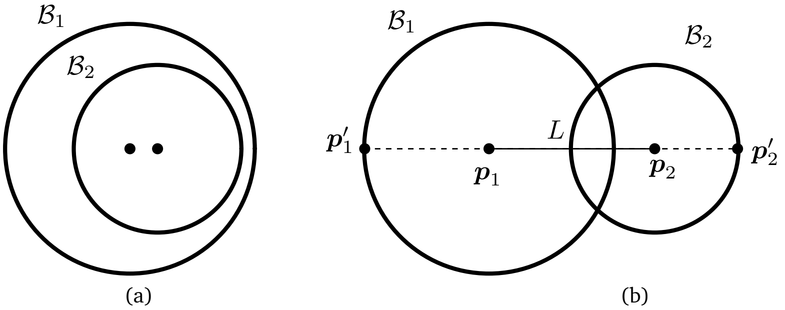


Figure A.4: The cases A (a) and B (b) in Lemma A.1. Note that in case B, the spheres need not necessarily overlap.

Proof: Without loss of generality, let $r_1 \geq r_2$. Consider the cases (A) $\mathcal{B}_2 \subseteq \mathcal{B}_1$ and (B) $\mathcal{B}_2 \nsubseteq \mathcal{B}_1$. These are shown in Fig. A.4 In case (A), $\mathtt{BE}(\mathcal{B}_1, \mathcal{B}_2) = \mathcal{B}_1$. In case (B), $\alpha = \|\boldsymbol{x}\| = \|\boldsymbol{p}_1 - \boldsymbol{p}_2\|$. Consider the line L defined by $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. The point $\boldsymbol{p}_1'$ lies at the intersection of L and the boundary of $\mathcal{B}_1$ on the opposite side of $\boldsymbol{p}_1$ to $\boldsymbol{p}_2$ and $\boldsymbol{p}_2'$ lies on the intersection of L and the boundary of $\mathcal{B}_2$ on the opposite side of $\boldsymbol{p}_2$ to $\boldsymbol{p}_1$, as shown. Then since both $\boldsymbol{p}_1'$ and $\boldsymbol{p}_2'$ must be contained in the enclosing ball, the diameter of an enclosing sphere is lower bounded by $\|\boldsymbol{p}_1' - \boldsymbol{p}_2'\| = r_1 + r_2 + \|\boldsymbol{p}_1 - \boldsymbol{p}_2\|$; since from (3.9), $\|\boldsymbol{p}_1 - \boldsymbol{p}_2\| = \alpha$, this is exactly the diameter of the ball given by $\mathtt{BE}(\mathcal{B}_1, \mathcal{B}_2)$ as defined in Sec. 3.4.1; hence BE gives the tightest possible enclosure.

Observation A.1: A capsule $\mathcal{C}$ of radius ρ_c can contain a sphere $\mathcal{S}$ of radius ρ_s iff $\rho_c \geq \rho_s$. □

Proof: Let $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ be the defining points of $\mathcal{C}$ and $\overline{\boldsymbol{p}_a, \boldsymbol{p}_b}$ be the line segment between them, and $\boldsymbol{p}_s$ be the centre of $\mathcal{S}$. Sufficiency is easily demonstrated by choosing $\boldsymbol{p}_s \in \overline{\boldsymbol{p}_a, \boldsymbol{p}_b}$, so we prove necessity. We partition the space into three subspaces using two hyperplanes $\mathcal{H}_a$ and $\mathcal{H}_b$ perpendicular to $\overline{\boldsymbol{p}_a, \boldsymbol{p}_b}$ offset at $\boldsymbol{p}_a$ and at $\boldsymbol{p}_b$ respectively, see Fig. A.5. Subspace $\mathcal{D}$ includes $\mathcal{H}_a$ and $\mathcal{H}_b$ and the space in between, while $\mathcal{A}$ and $\mathcal{B}$ are the open halfspaces on either side. As in [67], the closest point $\boldsymbol{p}_s'$ on $\overline{\boldsymbol{p}_a, \boldsymbol{p}_b}$ to $\boldsymbol{p}_s$ is either the foot of the perpendicular to $\overline{\boldsymbol{p}_a, \boldsymbol{p}_b}$ through $\boldsymbol{p}_s$, if $\boldsymbol{p}_s$ lies in $\mathcal{D}$; or $\boldsymbol{p}_a$ or $\boldsymbol{p}_b$, if $\boldsymbol{p}_s$ lies in $\mathcal{A}$ or $\mathcal{B}$ respectively.

Now consider the point $\boldsymbol{p}_s^*$ on the line defined by $\boldsymbol{p}_s$ and $\boldsymbol{p}_s'$, at a distance ρ_s from $\boldsymbol{p}_s'$ and such that $\boldsymbol{p}_s$ lies between $\boldsymbol{p}_s'$ and $\boldsymbol{p}_s^*$ (see Fig. A.5). Since $\|\boldsymbol{p}_s^* - \boldsymbol{p}_s\| \leq \rho_s$, $\boldsymbol{p}_s^* \in \mathcal{S}$. Observe that if $\boldsymbol{p}_s \in \mathcal{B}$, then $\boldsymbol{p}_s^* \in \mathcal{B}$ (since $\boldsymbol{p}_s^*$ is further away from $\mathcal{H}_b$ than $\boldsymbol{p}_s$, shown in Fig. A.5(a)), and similarly for $\mathcal{A}$, and $\mathcal{D}$ (since the line defined by $\boldsymbol{p}_s$ and $\boldsymbol{p}_s'$ is parallel to $\mathcal{H}_a$ and $\mathcal{H}_b$ and hence cannot cross them, see Fig. A.5(b)). The closest point on $\overline{\boldsymbol{p}_a, \boldsymbol{p}_b}$ to $\boldsymbol{p}_s^*$ is then also $\boldsymbol{p}_s'$, since if

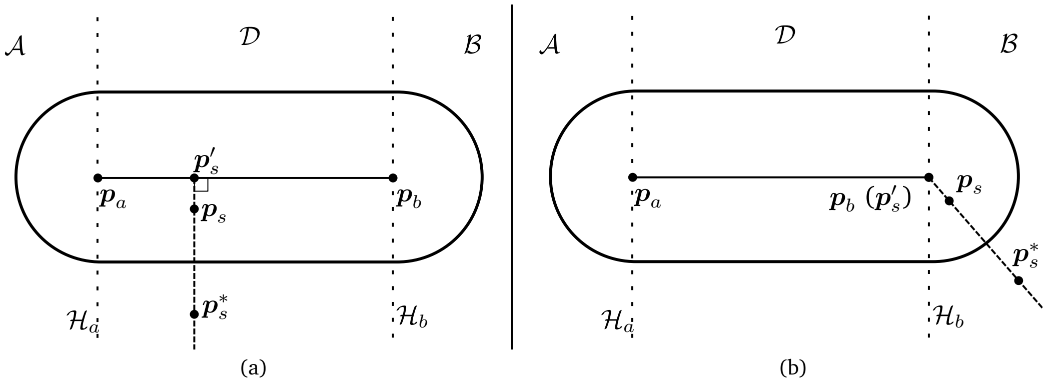


Figure A.5: The partitioning of the space in Obs. A.1. In (a), the centre $\boldsymbol{p}_s$ of the sphere $\mathcal{S}$, lies in $\mathcal{D}$, and the closest point on $\overline{\boldsymbol{p}_a, \boldsymbol{p}_b}$ is the foot of the perpendicular. In (b) $\boldsymbol{p}_s$ lies in $\mathcal{B}$, and the closest point on $\overline{\boldsymbol{p}_a, \boldsymbol{p}_b}$ is $\boldsymbol{p}_b$. Note that $\mathcal{S}$ is not depicted.

$\boldsymbol{p}_s, \boldsymbol{p}^*_s \in \mathcal{A}$, this is $\boldsymbol{p}_a$, if $\boldsymbol{p}_s, \boldsymbol{p}^*_s \in \mathcal{B}$, this is $\boldsymbol{p}_b$, else if $\boldsymbol{p}_s, \boldsymbol{p}^*_s \in \mathcal{D}$ then this is the foot of the perpendicular to $\overline{\boldsymbol{p}_a, \boldsymbol{p}_b}$. However, since $\|\boldsymbol{p}^*_s - \boldsymbol{p}'_s\| = \rho_s$, if $\rho_c < \rho_s$, then $\boldsymbol{p}^*_s \notin \mathcal{C}$. □

Observation A.2: For a capsule $\mathcal{C}$ of radius r with defining points $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$, the straight-line distance between two arbitrary points in the capsule is less than or equal to $2r + \|\boldsymbol{p}_a - \boldsymbol{p}_b\|$. □

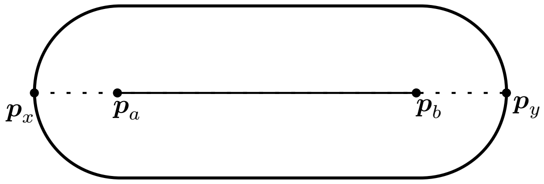


Figure A.6: The equality case of Obs. A.2. Points $\boldsymbol{p}_x$ and $\boldsymbol{p}_y$ lie on the intersection of the line defined by $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$, and the boundary of the capsule.

Proof: Consider arbitrary points $\boldsymbol{p}_x$ and $\boldsymbol{p}_y$ in $\mathcal{C}$. The points $\boldsymbol{p}'_x$ and $\boldsymbol{p}'_y$ are the points on the line segment $\overline{\boldsymbol{p}_a, \boldsymbol{p}_b}$ which are closest to $\boldsymbol{p}_x$ and $\boldsymbol{p}_y$. By the definition of a capsule, $\|\boldsymbol{p}_x - \boldsymbol{p}'_x\| \leq r$ and $\|\boldsymbol{p}_y - \boldsymbol{p}'_y\| \leq r$. Since $\boldsymbol{p}'_x$ and $\boldsymbol{p}'_y$ both lie on $\overline{\boldsymbol{p}_a, \boldsymbol{p}_b}$, $\|\boldsymbol{p}'_x - \boldsymbol{p}'_y\| \leq \|\boldsymbol{p}_a - \boldsymbol{p}_b\|$. Hence by the triangle inequality, $\|\boldsymbol{p}_x - \boldsymbol{p}_y\| \leq 2r + \|\boldsymbol{p}_a - \boldsymbol{p}_b\|$. Equality is achieved when $\boldsymbol{p}_x$ and $\boldsymbol{p}_y$ lie on the intersection of $\mathcal{C}$ and the line defined by $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$, as in Fig. A.6. □

Lemma A.2: The operator CE encloses $\mathcal{B}_1 = \mathcal{B}(\boldsymbol{p}_1; r_1)$ and $\mathcal{B}_2 = \mathcal{B}(\boldsymbol{p}_2; r_2)$ in the capsule of least volume. □

Proof: Without loss of generality, let $r_1 \geq r_2$. Consider the cases that (A) $\mathcal{B}_2 \subseteq \mathcal{B}_1$ and (B) $\mathcal{B}_2 \nsubseteq \mathcal{B}_1$. In case (A), from the definitions in (3.9) and (3.10), $\texttt{CE}(\mathcal{B}_1, \mathcal{B}_2) = \mathcal{B}_1$.

In case (B), we wish to find the enclosing capsule minimising $\frac{4}{3}\pi r^3 + a\pi r^2$, where a is the magnitude of the distance between the defining points of this enclosing capsule. From Obs. A.1,

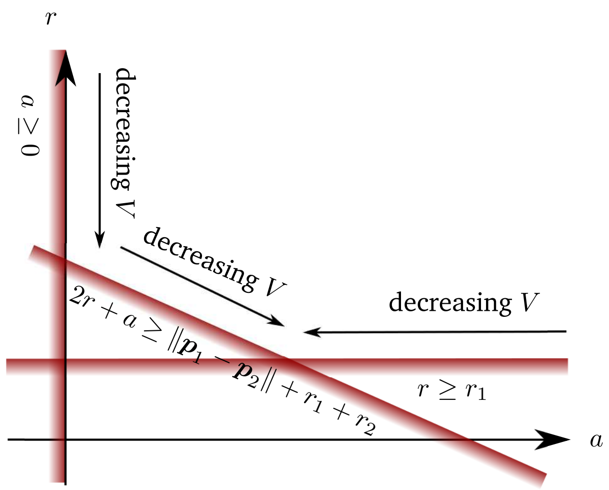


Figure A.7: The constrained optimisation problem in the proof to Lem. A.2. The bivariate function V has no minimum in the open upper-right quadrant, and the value of the function along the constraint boundaries is strictly monotonic, given $r \geq r_1$.

the radius of the smallest enclosing capsule is at least r_1. Similarly to the proof of Lemma A.1, we construct the line L defined by $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, and point $\boldsymbol{p}_1'$ at the intersection of the boundary of $\mathcal{B}_1$ and L, on the opposite side of $\boldsymbol{p}_1$ from $\boldsymbol{p}_2$, and point $\boldsymbol{p}_2'$ at the intersection of the boundary of $\mathcal{B}_2$ and L, on the opposite side of $\boldsymbol{p}_2$ from $\boldsymbol{p}_1$. This is shown in Fig. A.4(b). Since both $\boldsymbol{p}_1'$ and $\boldsymbol{p}_2'$ must be in the smallest enclosing capsule, and by Lemma A.2, $\|\boldsymbol{p}_1' - \boldsymbol{p}_2'\| = \|\boldsymbol{p}_1 - \boldsymbol{p}_2\| + r_1 + r_2 \leq 2r + a$, we have a lower bound on $2r + a$. The problem is now a bounded optimisation problem, visualised in Fig. A.7:

$$\begin{aligned} &\text{minimise:} && V = \frac{4}{3}\pi r^3 + a\pi r^2 \\ &\text{subject to:} && a \geq 0, \quad r \geq r_1, \quad 2r + a \geq \|\boldsymbol{p}_1 - \boldsymbol{p}_2\| + r_1 + r_2 \end{aligned}$$

The partial derivatives of V are:

$$\frac{\partial V}{\partial r} = 4\pi r^2 + 2a\pi r, \qquad \frac{\partial V}{\partial a} = \pi r^2$$

In the open upper-right quadrant $(a, r > 0)$, $\frac{\partial V}{\partial r}, \frac{\partial V}{\partial a} > 0$ and there are no local minima. We therefore search on the boundaries. On the boundary $a = 0$, V decreases with decreasing r. Along the boundary $2r + a = \|\boldsymbol{p}_1 - \boldsymbol{p}_2\| + r_1 + r_2$, the derivative is $-\frac{2\pi r}{\sqrt{5}}(r + a)$ in the direction of increasing a (i.e. downwards and to the right); since $a \geq 0$, this evaluates to 0 when $r = r_1 = 0$ and is negative otherwise.

On the boundary $r = r_1$, V decreases with decreasing a, except for the special case of $r_1 = 0$, where both balls $\mathcal{B}_1$ and $\mathcal{B}_2$ have zero volume. If the capsule enclosing them also has zero radius, then the volume is zero regardless of a, so any a attains the minimum volume.

Hence, a minimum is obtained at the intersection of $r = r_1$ and $2r + a = \|\boldsymbol{p}_1 - \boldsymbol{p}_2\| + r_1 + r_2$. It can be verified that these are the values of r and a of $\texttt{CE}(\mathcal{B}_1, \mathcal{B}_2)$ from the definitions in (3.9) and (3.10), Sec. 3.4.1. □

A.5 Obtaining Joint Position, Velocity and Acceleration Limits From Time Series of Marker Positions

Given time series of the elbow and wrist positions $\boldsymbol{x}_E$ and $\boldsymbol{x}_W$ in the shoulder coordinate system, and their first and second derivatives $\dot{\boldsymbol{x}}_E, \ddot{\boldsymbol{x}}_E$ and $\dot{\boldsymbol{x}}_W, \ddot{\boldsymbol{x}}_W$, we show how to obtain the time series of joint positions, velocities and accelerations. First, we find the joint angles $\boldsymbol{q}$ from the shoulder coordinate system and elbow and wrist positions $\boldsymbol{x}_E$ and $\boldsymbol{x}_W$ using inverse kinematics.

We express the concatenation $[\boldsymbol{x}_E^\top, \boldsymbol{x}_W^\top]^\top$ as $\boldsymbol{x}_{EW}$ for brevity. From $\boldsymbol{q}$, we compute the Jacobians $J_E(\boldsymbol{q})$ and $J_W(\boldsymbol{q})$ at elbow and wrist, again, concatenating them to $J_{EW}(\boldsymbol{q}) = [J_E(\boldsymbol{q})^\top, J_W(\boldsymbol{q})^\top]^\top$, hence:

$$\dot{\boldsymbol{x}}_{EW} = J_{EW}(\boldsymbol{q}) \cdot \dot{\boldsymbol{q}}. \tag{A.15}$$

The joint vector $\boldsymbol{q}$ has dimension 4×1 and the vector $\boldsymbol{x}_{EW}$ has dimension 6×1, so to obtain the joint velocity we calculate $J_{EW}^+(\boldsymbol{q})$, the pseudoinverse of $J_{EW}(\boldsymbol{q})$. We premultiply (A.15) by $J_{EW}^+(\boldsymbol{q})$, obtaining the joint velocities from $\dot{\boldsymbol{q}} = J_{EW}^+(\boldsymbol{q}) \cdot \dot{\boldsymbol{x}}_{EW}$, as long as the arm is not at a singularity.

To obtain the accelerations, we differentiate (A.15):

$$\ddot{\boldsymbol{x}}_{EW} = J_{EW}(\boldsymbol{q}) \cdot \ddot{\boldsymbol{q}} + \dot{J}_{EW}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \cdot \dot{\boldsymbol{q}},$$

we rearrange (A.5), and premultiply by $J_{EW}^+(\boldsymbol{q})$:

$$\ddot{\boldsymbol{q}} = J_{EW}^+(\boldsymbol{q}) \cdot \left(\ddot{\boldsymbol{x}}_{EW} - \dot{J}_{EW}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \cdot \dot{\boldsymbol{q}}\right).$$

A.6 Questionnaires for Study in Sec. 5.4

F0

1. Age / *Alter*

2. Gender / *Geschlecht*

3. Height / *Körpergröße*

4. Do you have any physical disabilities or sensory impairments? If yes, please explain. */Sind Ihnen körperliche oder Sinnesbeeinträchtigungen bekannt? Falls ja bitte um Erklärung.*

5. I am (a) left-handed (b) right-handed / *Ich bin (a) Linkhänder (b) Rechthänder*

6. Assess your experience with robotics, or human-robot cooperation.

   I am / have been involved with robots...

   (a) never (yet), (b) a few times, (c) once a month, (d) once a week, (e) daily

   Bewerten Sie Ihre Vorerfahrung mit Robotik, oder Mensch-Roboter-Kooperation.

   Ich beschäftige mich/habe mich beschäftigt mit Robotern...

   (a) noch nie (b) ein paar mal (c) einmal im Monat (d) einmal in der Woche (e) täglich

F1

Item		strongly disagree/ *stimme gar nicht zu*	rather disagree/ *stimme eher nicht zu*	moderately agree/ *stimme mittelmäßig zu*	rather agree/ *stimme eher zu*	strongly agree/ *stimme voll zu*	no response prefered/ *keine Angabe*
1	The system is capable of interpreting situations correctly. *Das System ist imstande Situationen richtig einzuschätzen.*	①	②	③	④	⑤	⓪
2	The system state was always clear to me. *Mir war durchgehend klar, in welchem Zustand sich das System befindet.*	①	②	③	④	⑤	⓪
3	The system works reliably. *Das System arbeitet zuverlässig.*	①	②	③	④	⑤	⓪
4	The system reacts unpredictably. *Das System reagiert unvorhersehbar.*	①	②	③	④	⑤	⓪
5	I trust the system. *Ich vertraue dem System.*	①	②	③	④	⑤	⓪
6	A system malfunction is likely. *Ein Ausfall des Systems ist wahrscheinlich.*	①	②	③	④	⑤	⓪
7	I was able to understand why things happened. *Ich konnte nachvollziehen, warum etwas passiert ist.*	①	②	③	④	⑤	⓪
8	The system is capable of taking over complicated tasks. *Das System kann wirklich komplizierte Aufgaben übernehmen.*	①	②	③	④	⑤	⓪
9	I can rely on the system. *Ich kann mich auf das System verlassen.*	①	②	③	④	⑤	⓪
10	The system might make sporadic errors. *Das System könnte stellenweise einen Fehler machen.*	①	②	③	④	⑤	⓪
11	It's difficult to identify what the system will do next. *Zu erkennen, was das System als Nächstes macht, ist schwer.*	①	②	③	④	⑤	⓪
12	I am confident about the systems capabilities. *Ich bin überzeugt von den Fähigkeiten des Systems.*	①	②	③	④	⑤	⓪

F2

Item		strongly disagree/ *stimme gar nicht zu*	rather disagree/ *stimme eher nicht zu*	moderately agree/ *stimme mittelmäßig zu*	rather agree/ *stimme eher zu*	strongly agree/ *stimme voll zu*	no response prefered/ *keine Angabe*
1	I already know similar systems. *Ich kenne bereits ähnliche Systeme.*	①	②	③	④	⑤	⓪
2	One should be careful with unfamiliar automated systems. *Bei unbekannten automatisierten Systemen sollte man eher vorsichtig sein.*	①	②	③	④	⑤	⓪
3	I trust a system more than I mistrust it. *Ich vertraue einem System eher, als dass ich ihm misstraue.*	①	②	③	④	⑤	⓪
4	I have already used similar systems. *Ich habe ähnliche Systeme bereits genutzt.*	①	②	③	④	⑤	⓪
5	Automated systems generally work well. *Automatisierte Systeme funktionieren generell gut.*	①	②	③	④	⑤	⓪
6	The developers are trustworthy. *Die Entwickler sind vertrauenswürdig.*	①	②	③	④	⑤	⓪
7	The developers take my well-being seriously. *Die Entwickler nehmen mein Wohlergehen ernst.*	①	②	③	④	⑤	⓪

A.7 Questionnaires for Study in Sec. 5.5

F0

– see F0 in Appendix A.6.

F1

Item		strongly disagree/ *stimme gar nicht zu*	rather disagree/ *stimme eher nicht zu*	neither agree nor disagree/ *teils-teils*	rather agree/ *stimme eher zu*	strongly agree/ *stimme voll zu*	no response prefered/ *keine Angabe*
1	I already know similar systems. *Ich kenne bereits ähnliche Systeme.*	①	②	③	④	⑤	⓪
2	One should be careful with unfamiliar automated systems. *Bei unbekannten automatisierten Systemen sollte man eher vorsichtig sein.*	①	②	③	④	⑤	⓪
3	I trust a system more than I mistrust it. *Ich vertraue einem System eher, als dass ich ihm misstraue.*	①	②	③	④	⑤	⓪
4	I have already used similar systems. *Ich habe ähnliche Systeme bereits genutzt.*	①	②	③	④	⑤	⓪
5	Automated systems generally work well. *Automatisierte Systeme funktionieren generell gut.*	①	②	③	④	⑤	⓪
6	The developers are trustworthy. *Die Entwickler sind vertrauenswürdig.*	①	②	③	④	⑤	⓪
7	The developers take my well-being seriously. *Die Entwickler nehmen mein Wohlergehen ernst.*	①	②	③	④	⑤	⓪

F2/F3

Item		strongly disagree/ *stimme gar nicht zu*	rather disagree/ *stimme eher nicht zu*	neither agree nor disagree/ *teils-teils*	rather agree/ *stimme eher zu*	strongly agree/ *stimme voll zu*	no response prefered/ *keine Angabe*
1	I trusted the robot to do the right thing at the right time. *Ich vertraute darauf, dass der Roboter das Richtige zum richtigen Zeitpunkt macht.*	①	②	③	④	⑤	⓪
2	The robot did not understand how I wanted to do the task. *Der Roboter verstand nicht wie ich handeln wollte.*	①	②	③	④	⑤	⓪
3	The robot kept getting in my way. *Der Roboter war oft in meinem Weg.*	①	②	③	④	⑤	⓪
4	I felt safe when working near the robot. *Ich fühlte mich sicher bei der Arbeit neben dem Roboter.*	①	②	③	④	⑤	⓪
5	The robot moved too fast. *Der Roboter bewegte sich zu schnell.*	①	②	③	④	⑤	⓪
6	The robot came too close to me. *Der Roboter kam mir zu nahe.*	①	②	③	④	⑤	⓪
7	I trusted the robot would not harm me. *Ich vertraute darauf, dass der Roboter mich nicht verletzen würde.*	①	②	③	④	⑤	⓪
8	The reaction of the robot to me was easy to comprehend. *Die Reaktion des Roboters auf mich war einfach nachzuvollziehen.*	①	②	③	④	⑤	⓪
9	The robots movement in my presence was confusing. *Die Bewegungen des Roboters in meiner Gegenwart haben mich verwirrt.*	①	②	③	④	⑤	⓪
10	I was distracted by the robots movement. *Ich war abgelenkt von der Bewegung des Roboters.*	①	②	③	④	⑤	⓪
11	The robots movement surprised me. *Ich war überrascht von der Reaktion des Roboters.*	①	②	③	④	⑤	⓪

F4

Item		strongly disagree/ *stimme gar nicht zu*	rather disagree/ *stimme eher nicht zu*	neither agree nor disagree/ *teils-teils*	rather agree/ *stimme eher zu*	strongly agree/ *stimme voll zu*	no response prefered/ *keine Angabe*
1	One should be careful with this system. *Bei diesem System sollte man eher vorsichtig sein.*	①	②	③	④	⑤	⓪
2	I trust this system more than I mistrust it. *Ich vertraue dem System eher, als ich ihm misstraue.*	①	②	③	④	⑤	⓪
3	This system generally works well. *Dieses System funktioniert generell gut.*	①	②	③	④	⑤	⓪
4	The developers of this system are trustworthy. *Die Entwickler dieser System sind vertrauenswürdig.*	①	②	③	④	⑤	⓪
5	The developers of this system take my well-being seriously. *Die Entwickler dieser System nehmen mein Wohlergehen ernst.*	①	②	③	④	⑤	⓪
6	I understand how the robot works better than at the start. *Ich verstehe besser als am Anfang wie der Roboter funktioniert.*	①	②	③	④	⑤	⓪
7	I feel safer, when the robot waits for me. *Ich fühle mich sicherer, wenn der Roboter auf mich wartet.*	①	②	③	④	⑤	⓪
8	I need more time to adapt to the behaviour of the robot. *Ich brauche mehr Zeit, mich an das Verhalten des Roboters anzupassen.*	①	②	③	④	⑤	⓪
9	It is annoying when the robot waits for me. *Mich stört es, wenn der Roboter auf mich wartet.*	①	②	③	④	⑤	⓪

References

[1] "Robots and robotic devices – collaborative robots," International Organization for Standardization, Geneva, Switzerland, TS 15066:2016, 2016.

[2] "Robots and robotic devices - safety requirements for industrial robots - part 1: Robots," International Organization for Standardization, ISO Standard 10218-1:2011, 2011.

[3] "Ridkind: Riddor reported injuries by kind of accident and broad industry group," 2018, [database], retrieved from hse.gov.uk/statistics/tables/ridkind.xlsx for years 2014-2018, Accessed: October 31, 2018.

[4] "Safety of machinery - positioning of safeguards with respect to the approach speeds of parts of the human body," International Organization for Standardization, ISO Standard 13855:2010, 2010.

[5] I. Asimov, *Liar!* Street & Smith, 1941.

[6] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Safety evaluation of physical human-robot interaction via crash-testing," in *Proc. Robotics: Science and Systems*, 2007, pp. 217–224.

[7] S. Haddadin, A. Albu-Schaffer, A. D. Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 3356–3363.

[8] R. Rossi, M. Polverini, A. Zanchettin, and P. Rocco, "A pre-collision control strategy for human-robot interaction based on dissipated energy in potential inelastic impacts," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 26–31.

[9] A. Raatz, S. Blankemeyer, G. Runge, C. Bruns, and G. Borchert, *Opportunities and Challenges for the Design of Inherently Safe Robots*. Springer, 2015, pp. 173–183.

[10] M. Grebenstein, A. Albu-Schäffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, R. Gruber, S. Haddadin, U. Hagn, R. Haslinger, H. Höppner, S. Jörg, M. Nickl, A. Nothhelfer, F. Petit, J. Reill, N. Seitz, T. Wimböck, S. Wolf, T. Wüsthoff, and G. Hirzinger, "The DLR hand arm system," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2011, pp. 3175–3182.

[11] M. Van Damme, P. Beyl, B. Vanderborght, R. Versluys, R. Van Ham, I. Vanderniepen, F. Daerden, and D. Lefeber, “The safety of a robot actuated by pneumatic muscles—a case study,” *Int. J. Social Robotics*, vol. 2, no. 3, pp. 289–303, 2010.

[12] R. V. Martinez, A. C. Glavan, C. Keplinger, A. I. Oyetibo, and G. M. Whitesides, “Soft actuators and robots that are resistant to mechanical damage,” *Advanced Functional Materials*, vol. 24, no. 20, pp. 3003–3010, 2014.

[13] O. Khatib, *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*. Springer, 1990, pp. 396–404.

[14] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, “A depth space approach to human-robot collision avoidance,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2012, pp. 338–345.

[15] D. H. Park, H. Hoffmann, P. Pastor, and S. Schaal, “Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2008, pp. 91–98.

[16] M. Polverini, A. Zanchettin, and P. Rocco, “Real-time collision avoidance in human-robot interaction based on kinetostatic safety field,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems*, 2014, pp. 4136–4141.

[17] N. M. Ceriani, A. M. Zanchettin, and P. Rocco, “Collision avoidance with task constraints and kinematic limitations for dual arm robots,” in *Proc. 13th Int. Conf Intelligent Autonomous Systems*, E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, Eds. Springer, 2016, pp. 1285–1299.

[18] M. Rauscher, M. Kimmel, and S. Hirche, “Constrained robot control using control barrier functions,” in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2016, pp. 279–285.

[19] M. Kimmel and S. Hirche, “Invariance control for safe human-robot interaction in dynamic environments,” *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1327–1342, 2017.

[20] J. Vogel, C. Castellini, and P. van der Smagt, “EMG-based teleoperation and manipulation with the DLR LWR-III,” in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2011, pp. 672–678.

[21] R. Luo and D. Berenson, “A framework for unsupervised online human reaching motion recognition and early prediction,” in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 2426–2433.

[22] H. Ding, G. Reißig, K. Wijaya, D. Bortot, K. Bengler, and O. Stursberg, "Human arm motion modeling and long-term prediction for safe and efficient human-robot-interaction," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2011, pp. 5875–5880.

[23] H. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 14–29, 2015.

[24] F. C. Van der Helm, "A finite element musculoskeletal model of the shoulder mechanism." *J. Biomech.*, vol. 27, no. 5, pp. 551–569, 1994.

[25] K. R. S. Holzbaur, W. M. Murray, and S. L. Delp, "A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control," *Ann. Biomed. Eng.*, vol. 33, no. 6, pp. 829–840, 2005.

[26] A. Platzer and E. Clarke, *The Image Computation Problem in Hybrid Systems Model Checking*. Springer, 2007, pp. 473–486.

[27] M. Althoff and B. H. Krogh, "Reachability analysis of nonlinear differential-algebraic systems," *IEEE Trans. on Automatic Control*, vol. 59, no. 2, pp. 371–383, 2014.

[28] "Tödlicher Roboter-Unfall bei VW: 29-Jähriger angeklagt," *Hessische/Niedersächsische Allgemeine*, 2017, [online]. Available: www.hna.de/kassel/kreis-kassel/baunatal-ort312516/toedlicher-roboter-unfall-bei-vw-kassel-in-baunatal-vor-gericht-8413531.html.

[29] J. Stühmer, S. Nowozin, A. Fitzgibbon, R. Szeliski, T. Perry, S. Acharya, D. Cremers, and J. Shotton, "Model-based tracking at $300hz$ using raw time-of-flight observations," in *IEEE Int. Conf. Computer Vision*, 2015, pp. 3577–3585.

[30] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3D pictorial structures for multiple human pose estimation," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 1669–1676.

[31] E. Mueggler, G. Gallego, and D. Scaramuzza, "Continuous-time trajectory estimation for event-based vision sensors," in *Proc. Robotics: Science and Systems*, 2015.

[32] G. Niemeyer, C. Preusche, and G. Hirzinger, "Telerobotics," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 741–757.

[33] P. Akella, M. Peshkin, E. Colgate, W. Wannasuphoprasit, N. Nagesh, J. Wells, S. Holland, T. Pearson, and B. Peacock, "Cobots for the automobile assembly line," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1999, pp. 728–733.

[34] C. Thomas, F. Busch, B. Kuhlenkoetter, and J. Deuse, “Process and human safety in human-robot-interaction - a hybrid assistance system for welding applications,” in *Intelligent Robotics and Applications*, S. Jeschke, H. Liu, and D. Schilberg, Eds. Springer, 2011, pp. 112–121.

[35] J. Kinugawa, Y. Kawaai, Y. Sugahara, and K. Kosuge, “PaDY: Human-friendly/cooperative working support robot for production site,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2010, pp. 5472–5479.

[36] K. Konada, J. Kinugawa, S. Arai, and K. Kosuge, “B-PaDY: Robot co-worker in a bumper assembly line: System integration of the prototype,” in *Proc. IEEE Int. Conf. Mechatronics and Automation (ICMA)*, 2017, pp. 722–727.

[37] “Safety of machinery - electrical equipment of machines,” International Organization for Standardization, IEC Standard 60204-1:2006, 2006.

[38] A. De Luca and R. Mattone, “Sensorless robot collision detection and hybrid force/motion control,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2005, pp. 999–1004.

[39] A. Hoffmann, A. Poeppel, A. Schierl, and W. Reif, “Environment-aware proximity detection with capacitive sensors for human-robot-interaction,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2016, pp. 145–150.

[40] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff, “Reachset conformance testing of hybrid automata,” in *Proc. Hybrid Systems: Computation and Control*, 2016, pp. 277–286.

[41] S. Petti and T. Fraichard, “Safe motion planning in dynamic environments,” in *Proc. IEEE-RSJ Int. Conf. Intell. Robots and Systems*, 2005, pp. 2210–2215.

[42] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, “Requirements for safe robots: Measurements, analysis and new insights,” *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, 2009.

[43] J. Hollerbach, “Dynamic scaling of manipulator trajectories,” *J. Dyn. Sys., Meas., Control*, vol. 106, no. 1, pp. 102–106, 1984.

[44] O. Dahl and L. Nielsen, “Torque-limited path following by online trajectory time scaling,” *IEEE Transactions on Robotics and Automation*, vol. 6, no. 5, pp. 554–561, 1990.

[45] C. G. L. Bianco and O. Gerelli, “Online trajectory scaling for manipulators subject to high-order kinematic and dynamic constraints,” *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1144–1152, 2011.

[46] T. Kröger and F. Wahl, "Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94–111, 2010.

[47] M. J. A. Zeestraten, S. Calinon, and D. G. Caldwell, "Variable duration movement encoding with minimal intervention control," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2016, pp. 497 – 503.

[48] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726 – 1743, 2013.

[49] F. Rohrmüller, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic mapping of dynamic obstacles using Markov chains for replanning in dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems*, 2008, pp. 2504–2510.

[50] G. Hoffman and C. Breazeal, "Effects of anticipatory action on human-robot teamwork: Efficiency, fluency, and perception of team," in *Proc. ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2007, pp. 1–8.

[51] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2013, pp. 299–306.

[52] D. Vasquez, T. Fraichard, and C. Laugier, "Incremental learning of statistical motion patterns with growing hidden Markov models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 403–416, 2009.

[53] C. Pérez-D'Arpino and J. A. Shah, "Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 6175–6182.

[54] A. M. Zanchettin and P. Rocco, "Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 6595–6600.

[55] S. Pellegrinelli, H. Admoni, S. Javdani, and S. Srinivasa, "Human-robot shared workspace collaboration via hindsight optimization," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 831–838.

[56] J. Mainprice, R. Hayne, and D. Berenson, "Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 897–908, 2016.

[57] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," *IEEE Trans. Cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[58] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2005.

[59] T. McMahon, S. Thomas, and N. Amato, "Sampling-based motion planning with reachable volumes: Theoretical foundations," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 6514–6521.

[60] H. Täubig, B. Bäuml, and U. Frese, "Real-time swept volume and distance computation for self collision detection," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2011, pp. 1585–1592.

[61] A. Giusti and M. Althoff, "On-the-fly control design of modular robot manipulators," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1484–1491, 2018.

[62] R. E. Moore, *Interval Analysis*. Wiley, 1966.

[63] T. Sunaga, "Theory of interval algebra and its application to numerical analysis," *Research Association of Applied Geometry Memoirs*, vol. 2, pp. 547–564, 1958.

[64] R. C. Young, "The algebra of multi-valued quantities," *Math. Ann.*, vol. 104, pp. 260–290, 1931.

[65] S. Rump, "INTLAB - INTerval LABoratory," in *Developments in Reliable Computing*, T. Csendes, Ed. Kluwer Academic Publishers, 1999, pp. 77–104, http://www.ti3.tuhh.de/rump/.

[66] M. Althoff, "An introduction to CORA 2015," in *Proc. Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015, pp. 120–151.

[67] C. Ericson, *Real-Time Collision Detection*. Morgan Kaufmann, 2004.

[68] K. Waldron and J. Schmiedeler, "Kinematics," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 9–33.

[69] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," PhD Thesis, Technische Universität München, 2010.

[70] K. Makino and M. Berz, "Taylor models and other validated functional inclusion methods," *International Journal of Pure and Applied Mathematics*, vol. 4, no. 4, pp. 379–456, 2003.

[71] J.-A. Ferrez, K. Fukuda, and T. Liebling, "Solving the fixed rank convex quadratic maximization in binary variables by a parallel zonotope construction algorithm," *European Journal of Operational Research*, vol. 166, no. 1, pp. 35 – 50, 2005.

[72] E. Gilbert, D. Johnson, and S. Keerthi, “A fast procedure for computing the distance between complex objects in three-dimensional space,” *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.

[73] P. Lindemann, “The Gilbert-Johnson-Keerthi distance algorithm,” in *Media Informatics Proseminar on Algorithms in Media Informatics*, 2009.

[74] S. B. Liu, H. Roehm, C. Heinzemann, I. Lütkebohle, J. Oehlerking, and M. Althoff, “Provably safe motion of mobile robots in human environments,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2017, pp. 1351–1357.

[75] C. Högfors, B. Peterson, G. Sigholm, and P. Herberts, “Biomechanical model of the human shoulder joint–ii. the shoulder rhythm,” *J. Biomech.*, vol. 24, no. 8, pp. 699–709, 1991.

[76] S. Pheasant and C. M. Haslegrave, *Bodyspace: Anthropometry, Ergonomics and the Design of Work*. Taylor & Francis CRC Press, 2006, ch. 6. Hands and Handles, pp. 143–160.

[77] E. Asarin, T. Dang, G. Frehse, A. Girard, C. Le Guernic, and O. Maler, “Recent progress in continuous and hybrid reachability analysis,” in *Proc. IEEE Conference on Computer Aided Control Systems Design*, 2006, pp. 1582–1587.

[78] N. Klopčar, M. Tomšič, and J. Lenarčič, “A kinematic model of the shoulder complex to evaluate the arm-reachable workspace,” *J. Biomech.*, vol. 40, no. 1, pp. 86–91, 2001.

[79] M. Ragaglia, A. Zanchettin, and P. Rocco, “Safety-aware trajectory scaling for human-robot collaboration with prediction of human occupancy,” in *Proc. Int. Conf. Advanced Robotics*, 2015, pp. 85–90.

[80] V. De Sapio, J. Warren, and O. Khatib, “Predicting reaching postures using a kinematically constrained shoulder model,” in *Advances in Robot Kinematics*, J. Lenarčič and B. Roth, Eds. Springer, 2006, pp. 209–218.

[81] N. Klopčar and J. Lenarčič, “Kinematic model for determination of human arm reachable workspace,” *Meccanica*, vol. 40, no. 2, pp. 203–219, 2005.

[82] A. Zanchettin, L. Bascetta, and P. Rocco, “Achieving humanlike motion: Resolving redundancy for anthropomorphic industrial manipulators,” *IEEE Robotics Automation Magazine*, vol. 20, no. 4, pp. 131–138, 2013.

[83] K. Abdel-Malek, J. Yang, R. Brand, and E. Tanbour, “Towards understanding the workspace of human limbs,” *Ergonomics*, vol. 47, no. 13, pp. 1386–1405, 2004.

[84] G. Wu, F. C. van der Helm, H. D. Veeger, M. Makhsous, P. V. Roy, C. Anglin, J. Nagels, A. R. Karduna, K. McQuade, X. Wang, F. W. Werner, and B. Buchholz, "ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion-part ii: shoulder, elbow, wrist and hand," *J. Biomech.*, vol. 38, no. 5, pp. 981 – 992, 2005.

[85] F. Zacharias, *Knowledge Representations for Planning Manipulation Tasks*. Springer, 2016, ch. Visualization and Setup Evaluation, pp. 71–92.

[86] D. Hahn, M. Olvermann, J. Richtberg, W. Seiberl, and A. Schwirtz, "Knee and ankle joint torque-angle relationships of multi-joint leg extension," *J. Biomech.*, vol. 44, no. 11, pp. 2059–2065, July 2011.

[87] W. M. Murray, T. S. Buchanan, and S. L. Delp, "Scaling of peak moment arms of elbow muscles with upper extremity bone dimensions," *Journal of Biomechanics*, vol. 35, no. 1, pp. 19–26, 2002.

[88] J. C. Otis, C. C. Jiang, T. L. Wickiewicz, M. G. Peterson, R. F. Warren, and T. J. Santner, "Changes in the moment arms of the rotator cuff and deltoid muscles with abduction and rotation." *The Journal of Bone & Joint Surgery*, vol. 76, no. 5, pp. 667–676, 1994.

[89] A. A. Amis, D. Dowson, and V. Wright, "Analysis of elbow forces due to high-speed forearm movements," *J. Biomech.*, vol. 13, no. 10, pp. 825–831, 1980.

[90] R. Featherstone and D. E. Orin, "Dynamics," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 35–65.

[91] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *Proc. 47th IEEE Conference on Decision and Control*, 2008, pp. 4042–4048.

[92] G. V. Smirnov, *Introduction to the Theory of Differential Inclusions*. American Mathematical Society, 2002.

[93] 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 *Proc. Int. Conf. on Advanced Robotics*, 2005, pp. 532–539.

[94] A. Girard, C. Le Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *Hybrid Systems: Computation & Control*, ser. LNCS 3927. Springer, 2006, pp. 257–271.

[95] T. S. Arthanari and Y. Dodge, *Mathematical Programming in Statistics*. Wiley, 1981.

[96] M. Althoff, D. Heß, and F. Gambert, "Road occupancy prediction of traffic participants," in *Proc. IEEE Int. Conf. Intell. Transportation Systems*, 2013, pp. 99–105.

[97] T. Koritnik, T. Bajd, and M. Munih, “A simple kinematic model of a human body for virtual environments,” in *Advances in Robot Kinematics: Motion in Man and Machine*, J. Lenarčič and M. M. Stanišić, Eds. Springer, 2010, pp. 401–408.

[98] M. Althoff and J. M. Dolan, “Online verification of automated road vehicles using reachability analysis,” *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 903–918, 2014.

[99] P. I. Corke, *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011.

[100] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, “Multi-Parametric Toolbox 3.0,” in *Proc. European Control Conference*, 2013, pp. 502–510, http://control.ee.ethz.ch/~mpt.

[101] G. Monheit and N. Badler, “A kinematic model of the human spine and torso,” *IEEE Computer Graphics and Applications*, vol. 11, no. 2, pp. 29–38, 1991.

[102] E. Demircan, T. Besier, S. Menon, and O. Khatib, “Human motion reconstruction and synthesis of human skills,” in *Advances in Robot Kinematics: Motion in Man and Machine*, J. Lenarčič and M. M. Stanišić, Eds. Springer, 2010, pp. 283–292.

[103] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, “Opensim: Open-source software to create and analyze dynamic simulations of movement,” *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.

[104] L. M. Schutte, “Using musculoskeletal models to explore strategies for improving performance in electrical stimulationinduced leg cycle ergometry,” PhD Thesis, Stanford University, 1993.

[105] S. M. LaValle and M. S. Branicky, *On the Relationship between Classical Grid Search and Probabilistic Roadmaps*. Springer, 2004, pp. 59–75.

[106] B. Burns and O. Brock, “Toward optimal configuration space sampling,” in *Proceedings of Robotics: Science and Systems*, 2005, pp. 105–112.

[107] A. Bhatia and E. Frazzoli, “Incremental search methods for reachability analysis of continuous and hybrid systems,” in *Proc. Hybrid Systems: Computation and Control*, R. Alur and G. J. Pappas, Eds. Springer, 2004, pp. 142–156.

[108] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[109] D. C. Boone and S. P. Azen, “Normal range of motion of joints in male subjects,” *J. Bone Joint Surg. Am.*, vol. 61, no. 5, pp. 756–759, 1979.

[110] J. D. Lee and K. A. See, “Trust in automation: Designing for appropriate reliance,” *Human Factors*, vol. 46, no. 1, pp. 50–80, 2004.

[111] R. E. Yagoda and D. J. Gillan, “You want me to trust a ROBOT? The development of a human–robot interaction trust scale,” *Int. J. Social Robotics*, vol. 4, no. 3, pp. 235–248, 2012.

[112] I. Gaudiello, E. Zibetti, S. Lefort, M. Chetouani, and S. Ivaldi, “Trust as indicator of robot functional and social acceptance. An experimental study on user conformation to iCub answers,” *Computers in Human Behavior*, vol. 61, pp. 633 – 655, 2016.

[113] D. J. McAllister, “Affect- and cognition-based trust as foundations for interpersonal cooperation in organizations,” *The Academy of Management Journal*, vol. 38, no. 1, pp. 24–59, 1995.

[114] S. M. Merritt and D. R. Ilgen, “Not all trust is created equal: Dispositional and history-based trust in human-automation interactions,” *Human Factors*, vol. 50, no. 2, pp. 194–210, 2008.

[115] P. Kaniarasu, A. Steinfeld, M. Desai, and H. Yanco, “Robot confidence and trust alignment,” in *Proc. ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2013, pp. 155–156.

[116] P. A. Hancock, D. R. Billings, K. E. Schaefer, J. Y. C. Chen, E. J. de Visser, and R. Parasuraman, “A meta-analysis of factors affecting trust in human-robot interaction,” *Human Factors*, vol. 53, no. 5, pp. 517–527, 2011.

[117] M. Coeckelbergh, *Robotics in Germany and Japan. Philosophical and Technical Perspectives*. Peter Lang Publishing, 2014, ch. Robotic Appearances and Forms of Life. A Phenomenological-Hermeneutical Approach to the Relation between Robotics and Culture.

[118] F. Kaplan, “Who is afraid of the humanoid? Investigating cultural differences in the acceptance of robots,” *International Journal of Humanoid Robotics*, vol. 1, no. 3, pp. 465–480, 2004.

[119] I. Asimov, *Foundation Trilogy: Foundation, Foundation and Empire, and Second Foundation*. Gnome Press, 1953.

[120] The Terminator Series. [website]. International Movie Database. Accessed: January 1, 2018. [Online]. Available: www.imdb.com/list/ls073555125/

[121] J. Manyika, M. Chui, M. Miremadi, J. Bughin, K. George, P. Willmott, and M. Dewhurst, “Harnessing automation for a future that works,” McKinsey Global Institute, Tech. Rep., 2017.

[122] T. Nomura, T. Kanda, T. Suzuki, and K. Kato, *Hesitancy in Interacting with Robots–Anxiety and Negative Attitudes*. CRC Press, 2004, pp. 88–99.

[123] C. Bartneck, T. Suzuki, T. Kanda, and T. Nomura, "The influence of people's culture and prior experiences with aibo on their attitude towards robots," *AI & Society: The Journal of Human-Centred Systems*, vol. 21, no. 1, 2006.

[124] H. R. Lee and S. Sabanović, "Culturally variable preferences for robot design and use in South Korea, Turkey, and the United States," in *Proc. ACM/IEEE Int. Conf. on Human-robot Interaction*, 2014, pp. 17–24.

[125] S. A. Green, M. Billinghurst, X. Chen, and J. G. Chase, "Human-robot collaboration: A literature review and augmented reality approach in design," *International Journal of Advanced Robotic Systems*, vol. 5, no. 1, p. 1, 2008.

[126] J. Perez-Osorio, H. J. Müller, E. Wiese, and A. Wykowska, "Gaze following is modulated by expectations regarding others' action goals," *PLOS ONE*, vol. 10, no. 11, pp. 1–19, 2015.

[127] S. Elprama, I. El Makrini, B. Vanderborght, and A. Jacobs, "Acceptance of collaborative robots by factory workers: a pilot study on the role of social cues of anthropomorphic robots," in *Proc. IEEE/RSJ Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*, 2016, pp. 919–924.

[128] J. K. Burgoon, L. A. Stern, and L. Dillman, *Interpersonal Adaptation: Dyadic interaction patterns*. Cambridge University Press, 1995.

[129] C. Chao and A. L. Thomaz, "Turn taking for human-robot interaction," in *Dialog with Robots: Papers from the AAAI Fall Symposium*, 2005.

[130] D. Novick, B. Hansen, and K. Ward, "Coordinating turn-taking with gaze," in *Int. Conf. Spoken Language*, vol. 3, 1996, pp. 1888–1891.

[131] D. Tannen, "Turn-taking and intercultural discourse and communication," in *The Handbook of Intercultural Discourse and Communication*. Wiley, 2012.

[132] E. Calisgan, A. Haddadi, H. Van der Loos, J. Alcazar, and E. Croft, "Identifying nonverbal cues for automated human-robot turn-taking," in *Proc. IEEE/RSJ Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*, 2012, pp. 418–423.

[133] A. Gaschler, S. Jentzsch, M. Giuliani, K. Huth, J. de Ruiter, and A. Knoll, "Social behavior recognition using body posture and head pose for human-robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2012, pp. 2128–2133.

[134] R. P. A. Petrick, M. E. Foster, and A. Isard, “Social state recognition and knowledge-level planning for human-robot interaction in a bartender domain,” in *Proc. AAAI Workshop on Grounding Language for Physical Systems*, 2012, pp. 32–38.

[135] C. Chao, “Timing multimodal turn-taking for human-robot cooperation,” in *Proc. ACM Int. Conf. on Multimodal Interaction*, 2012, pp. 309–312.

[136] T. Asfour and R. Dillmann, “Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2003, pp. 1407–1412.

[137] P. Morasso, “Three dimensional arm trajectories,” *Biological Cybernetics*, vol. 48, no. 3, pp. 187–194, 1983.

[138] D. Bortot, M. Born, and K. Bengler, “Directly or on detours? How should industrial robots approximate humans?” in *Proc. ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2013, pp. 89–90.

[139] M. Huber, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer, “Human-robot interaction in handing-over tasks,” in *Proc. IEEE Int. Symp. Robot and Human Interactive Communication (RO-MAN)*, 2008, pp. 107–112.

[140] A. Dragan and S. Srinivasa, “Generating legible motion,” in *Proc. Robotics: Science and Systems*, 2013, paper # 24.

[141] C. Lichtenthäler, T. Lorenz, and A. Kirsch, “Influence of legibility on perceived safety in a virtual human-robot path crossing task,” in *Proc. IEEE/RSJ Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*, 2012, pp. 676–681.

[142] A. Dragan, R. Holladay, and S. Srinivasa, “An analysis of deceptive robot motion,” in *Proc. Robotics: Science and Systems*, 2014, paper # 10.

[143] J. Harris and E. Sharlin, “Exploring the affect of abstract motion in social human-robot interaction,” in *Proc. IEEE Int. Symp. Robot and Human Interactive Communication (RO-MAN)*, 2011, pp. 441–448.

[144] H. Knight, R. Thielstrom, and R. Simmons, “Expressive path shape (swagger): Simple features that illustrate a robot’s attitude toward its goal in real time,” in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 1475–1482.

[145] H. Knight and R. Simmons, “Designing expressive motions for a robot requesting help: Attributions and task performance,” in *Proc. Robotics: Science and Systems (workshop)*, 2014.

[146] R. Laban and F. C. Lawrence, *Effort*. Macdonald & Evans, 1947.

[147] J. P. Dillard, D. H. Solomon, and M. T. Palmer, "Structuring the concept of relational communication," *Communication Monographs*, vol. 66, no. 1, pp. 49–65, 1999.

[148] J. Li, W. Ju, and C. Nass, "Observer perception of dominance and mirroring behavior in human-robot relationships," in *Proc. ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2015, pp. 133–140.

[149] M. Salem and K. Dautenhahn, "Evaluating trust and safety in HRI : Practical issues and ethical challenges," in *Proc. Workshop "Emerging Policy and Ethics of Human-Robot Interaction" at ACM/IEEE Int. Conf. Human Robot Interaction*, 2015.

[150] S. Bernard, T. Castelain, H. Mercier, L. Kaufmann, J.-B. Van der Henst, and F. Clément, "The boss is always right: Preschoolers endorse the testimony of a dominant over that of a subordinate," *Journal of Experimental Child Psychology*, vol. 152, no. Supplement C, pp. 307 – 317, 2016.

[151] V. V. Unhelkar, H. C. Siu, and J. A. Shah, "Comparative performance of human and mobile robotic assistants in collaborative fetch-and-deliver tasks," in *Proc. ACM/IEEE Int. Conf. on Human-robot Interaction*, 2014, pp. 82–89.

[152] 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.

[153] G. Hoffman and C. Breazeal, "Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team," in *Proc. ACM/IEEE Int. Conf. Human-robot Interaction*, 2007, pp. 1–8.

[154] J. Iqbal, Z. H. Khan, and A. Khalid, "Prospects of robotics in food industry," *Food Science and Technology*, vol. 37, pp. 159 – 165, 2017.

[155] M. Körber, E. Baseler, and K. Bengler, "Introduction matters: Manipulating trust in automation and reliance in automated driving," *Applied Ergonomics*, vol. 66, pp. 18–31, 2018.

[156] A. M. Zanchettin and P. Rocco, "Path-consistent safety in mixed human-robot collaborative manufacturing environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems*, 2013, pp. 1131–1136.

[157] A. Zanchettin, N. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 882–893, 2016.

[158] L. J. Cronbach, "Coefficient alpha and the internal structure of tests," *Psychometrika*, vol. 16, no. 3, pp. 297–334, 1951.

[159] S. S. Shapiro and M. B. Wilk, “An analysis of variance test for normality (complete samples),” *Biometrika*, vol. 52, no. 3-4, pp. 591–611, 1965.

[160] M. Friedman, “A comparison of alternative tests of significance for the problem of m rankings,” *Ann. Math. Statist.*, no. 1, pp. 86–92, 03 1940.

[161] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan, “Information gathering actions over human internal state,” in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 66–73.

[162] M. Saerbeck and C. Bartneck, “Perception of affect elicited by robot motion,” in *Proc. ACM/IEEE Int. Conf. Human-Robot Interaction*, 2010, pp. 53–60.

[163] L. Carlson and E. Covell, *Functional features in language and space*. Oxford University Press, 2004, ch. Defining functional features for spatial language, pp. 175–190.

[164] C. D. Kidd and C. Breazeal, “Robots at home: Understanding long-term human-robot interaction,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2008, pp. 3230–3235.

[165] I. Leite, C. Martinho, and A. Paiva, “Social robots for long-term interaction: A survey,” *International Journal of Social Robotics*, vol. 5, no. 2, pp. 291–308, 2013.

[166] J. F. Hoorn, E. A. Konijn, D. M. Germans, S. Burger, and A. Munneke, “The in-between machine–the unique value proposition of a robot or why we are modelling the wrong things,” in *Proc. Int. Conf on Agents and Artificial Intelligence*, 2015, pp. 464–469.

[167] A. C. Lo, P. D. Guarino, L. G. Richards, J. K. Haselkorn, G. F. Wittenberg, D. G. Federman, R. J. Ringer, T. H. Wagner, H. I. Krebs, B. T. Volpe, C. T. J. Bever, D. M. Bravata, P. W. Duncan, B. H. Corn, A. D. Maffucci, S. E. Nadeau, S. S. Conroy, J. M. Powell, G. D. Huang, and P. Peduzzi, “Robot-assisted therapy for long-term upper-limb impairment after stroke,” *New England Journal of Medicine*, vol. 362, no. 19, pp. 1772–1783, 2010.

[168] J. D. van der Laan, A. Heino, and D. D. Waard, “A simple procedure for the assessment of acceptance of advanced transport telematics,” *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 1, pp. 1–10, 1997.

[169] M. Müllhäuser, “Tactile cueing with active cyclic stick for helicopter obstacle avoidance: development and pilot acceptance,” *CEAS Aeronautical Journal*, 2017.

[170] K. L. Koay, D. S. Syrdal, M. L. Walters, and K. Dautenhahn, “Living with robots: Investigating the habituation effect in participants’ preferences during a longitudinal human-robot interaction study,” in *Proc. IEEE Int. Conf. Robot & Human Interactive Communication*, 2007.

[171] J. Lee and N. Moray, "Trust, control strategies and allocation of function in human-machine systems," *Ergonomics*, vol. 35, no. 10, pp. 1243–1270, 1992.

[172] S. Wrede, C. Emmerich, R. Grünberg, A. Nordmann, A. Swadzba, and J. Steil, "A user study on kinesthetic teaching of redundant robots in task and configuration space," *J. Human Robot Interaction*, vol. 2, no. 1, pp. 56–81, 2013.

[173] C. Bartneck, D. Kulić, E. Croft, and S. Zoghbi, "Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots," *International Journal of Social Robotics*, vol. 1, no. 1, pp. 71–81, 2009.

[174] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[175] M. L. Walters, K. Dautenhahn, R. te Boekhorst, K. L. Koay, C. Kaouri, S. Woods, C. Nehaniv, D. Lee, and I. Werry, "The influence of subjects' personality traits on personal spatial zones in a human-robot interaction experiment," in *IEEE Int. Workshop on Robot and Human Interactive Communication.*, 2005, pp. 347–352.

[176] L. Takayama and C. Pantofaru, "Influences on proxemic behaviors in human-robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2009, pp. 5495–5502.

[177] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Proc. 2nd Berkeley Symposium on Mathematical Statistics and Probability*, 1951, pp. 481–492.

Author's Publications and Student Projects

[178] A. Pereira and M. Althoff, "Safety control of robots under computed torque control using reachable sets," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 331–338.

[179] M. J. Zeestraten, A. Pereira, M. Althoff, and S. Calinon, "Online motion synthesis with minimal intervention control and formal safety guarantees," in *Proc. of IEEE Systems, Man and Cybernetics*, 2016, pp. 2116–2121.

[180] D. Beckert, A. Pereira, and M. Althoff, "Online verification of multiple safety criteria for a robot trajectory," in *Proc. IEEE Conf. Decision and Control*, 2017, pp. 6454–6461.

[181] T. Salzmann, "Dynamic replanning for safe human-robot co-existence," 2017, [seminar project].

[182] J.-H. Neudeck, "Planning heuristic based on human motion prediction," 2017, [seminar project].

[183] E. Bozkir, "Probabilistic and formal approaches to human occupancy prediction for safe human-robot collaboration," 2016, [Masters thesis].

[184] A. Pereira and M. Althoff, "Calculating human reachable occupancy for guaranteed collision-free planning," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, 2017, pp. 4473–4480.

[185] ——, "Overapproximative human arm occupancy prediction for collision avoidance," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 818–831, 2018.

[186] ——, "Overapproximative arm occupancy prediction for human-robot co-existence built from archetypal movements," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 1394–1401.

[187] C. Stark, A. Pereira, and M. Althoff, "Reachset conformance testing of human arms with a biomechanical model," in *Proc. IEEE Int. Conf. Robotic Computing*, 2018, pp. 209–216.

[188] N. Reppekus, "Representation of reachable sets in human-robot interaction with a view to online safety control," 2015, [Bachelors thesis].

[189] C. Stark, "Validation of a conservative low-order human model using a high-order biomechanical model," 2017, [Bachelors thesis].

[190] J. Reinhardt, A. Pereira, D. Beckert, and K. Bengler, "Dominance and movement cues of robot motion: A user study on trust and predictability," in *Proc. of IEEE Systems, Man and Cybernetics*, 2017, pp. 1493–1498.

[191] M. Baumann, "Conducting a medium term user study in the practical course safe human-robot co-existence," 2017, [seminar project].

[192] J. Gerstner, "Report on master practical course safe human-robot co-existence - conducting a user study," 2017, [seminar project].

[193] S. Liu, "Reachability analysis of emergency braking maneuvers of robot manipulators," 2017, [Masters thesis].