AN OPTIMIZATION-BASED FORMALISM FOR SHARED AUTONOMY IN DYNAMIC ENVIRONMENTS

Christopher Edwin Mower

Doctor of Philosophy
School of Informatics
University of Edinburgh

2021

Christopher Edwin Mower
An Optimization-based Formalism For Shared Autonomy In Dynamic Environments
Doctor of Philosophy, 2021

SUPERVISOR:
Professor Sethu Vijayakumar, FRSE

EXAMINERS:
Professor Nick Hawes, University of Oxford
Dr J. Michael Herrmann, University of Edinburgh

LAY SUMMARY

Construction tasks, such as concrete spraying, demand high levels of concentration and manual dexterity from an operator. Often, these tasks are prolonged and can lead to fatigue and low levels of situational awareness. This can have a damaging impact on safety and has been linked as a contributing factor to the construction sector having one of the highest levels of workplace fatalities - national statistics (Health and Safety Executive, Workplace fatal injuries in Great Britain, 2021) between 2020-2021 reported 39 of the 142 fatalities were from the construction sector (the highest over all sectors). As such, operators require extensive and costly training to perform these tasks at a high quality whilst ensuring minimum safety standards. In the ideal case, regulators would replace this system with an autonomous robot maximizing human safety. However, current AI methods are not able to match, let alone surpass, the contextual awareness and responsive capabilities of skilled human operators.

In order to create a safer environment, improve quality, and reduce costs, this thesis explores *shared autonomy*. Shared autonomy aims to combine the natural aptitudes of humans with the latest advancements in multi-modal sensing, the processing capacity of computers, and systematic reasoning abilities of autonomous methods to generate optimal robot motions. Shared autonomy applies not only to construction, but the likes of healthcare, remote inspection, maintenance, disaster recovery, and even space and maritime exploration. The recent developments within this branch of work has generated a number of difficult research questions, that form the focus of this thesis, such as: What kind of assistance would improve task performance without impediment of operator experience? How do we model assistance that would enable a computationally efficient system, responsive enough for online teleoperation in complex environments? And, how can we leverage intention estimation and prediction, and environment sensing, to make predictions of degenerate future events and adapt control strategies early enough to prevent these from becoming a problem?

The work presented in this thesis, develops methods to address these questions in specific cases inspired by the requirements and restrictions of the construction sector. The direction taken in this work has been to leverage numerical optimization as a framework for modeling shared autonomy that maintains the features of an operators skill and experience, whilst ensuring (changing) physical and operational constraints. We demonstrate the capabilities of the proposed work in realistic lab mock-ups, and evaluate the methods through rigorous, repeatable analysis.

ABSTRACT

Teleoperation is an integral component of various industrial processes. For example, concrete spraying, assisted welding, plastering, inspection, and maintenance. Often these systems implement direct control that maps interface signals onto robot motions. Successful completion of tasks typically requires high levels of manual dexterity and cognitive load. In addition, the operator is often present nearby dangerous machinery. Consequently, safety is of critical importance and training is expensive and prolonged – in some cases taking several months or even years.

An autonomous robot replacement would be an ideal solution since the human could be removed from danger and training costs significantly reduced. However, this is currently not possible due to the complexity and unpredictability of the environments, and the levels of situational and contextual awareness required to successfully complete these tasks.

In this thesis, the limitations of direct control are addressed by developing methods for *shared autonomy*. A shared autonomous approach combines human input with autonomy to generate optimal robot motions. The approach taken in this thesis is to formulate shared autonomy within an optimization framework that finds optimized states and controls by minimizing a cost function, modeling task objectives, given a set of (changing) physical and operational constraints.

Online shared autonomy requires the human to be continuously interacting with the system via an interface (akin to direct control). The key challenges addressed in this thesis are: 1) ensuring computational feasibility (such a method should be able to find solutions fast enough to achieve a sampling frequency bound below by 40Hz), 2) being reactive to changes in the environment and operator intention, 3) knowing how to appropriately blend operator input and autonomy, and 4) allowing the operator to supply input in an intuitive manner that is conducive to high task performance.

Various operator interfaces are investigated with regards to the control space, called a *mode of teleoperation*. Extensive evaluations were carried out to determine for which modes are most intuitive and lead to highest performance in target acquisition tasks (e.g. spraying/welding/etc). Our performance metrics quantified task difficulty based on Fitts' law, as well as a measure of how well constraints affecting the task performance were met. The experimental evaluations indicate that higher performance is achieved when humans submit commands in low-dimensional task spaces as op-

posed to joint space manipulations. In addition, our multivariate analysis indicated that those with regular exposure to computer games achieved higher performance.

Shared autonomy aims to relieve human operators of the burden of precise motor control, tracking, and localization. An optimization-based representation for shared autonomy in dynamic environments was developed. Real-time tractability is ensured by modulating the human input with information of the changing environment within the same task space, instead of adding it to the optimization cost or constraints. The method was illustrated with two real world applications: grasping objects in cluttered environments and spraying tasks requiring sprayed linings with greater homogeneity.

Maintaining motion patterns – referred to as *skills* – is often an integral part of teleoperation for various industrial processes (e.g. spraying, welding, plastering). We develop a novel model-based shared autonomous framework for incorporating the notion of skill assistance to aid operators to sustain these motion patterns whilst adhering to environment constraints. In order to achieve computational feasibility, we introduce a novel parameterization for state and control that combines skill and underlying trajectory models, leveraging a special type of curve known as Clothoids. This new parameterization allows for efficient computation of skill-based short term horizon plans, enabling the use of a model predictive control loop. Our hardware realization validates the effectiveness of our method to recognize a change of intended skill, and showing an improved quality of output motion, even under dynamically changing obstacles.

In addition, extensions of the work to supervisory control are described. An exploratory study presents an approach that improves computational feasibility for complex tasks with minimal interactive effort on the part of the human. Adaptations are theorized which might allow such a method to be applicable and beneficial to high degree of freedom systems. Finally, a system developed in our lab is described that implements sliding autonomy and shown to complete multi-objective tasks in complex environments with minimal interaction from the human.

ACKNOWLEDGEMENTS

First, I would like to thank my supervisor Professor Sethu Vijayakumar for his guidance, dedication, encouragement, and for allowing me to freely pursue and explore the field of robotics and shared autonomy.

All my love goes to my wife Nazanin. You have been a helpful critic and inspiration to me. Thank you, especially, for your unwavering support at the toughest of times. You are in every possible way the visual personification of absolute perfection.

To my parents, Alison and Gavin Mower, whose love, support, and encouragement throughout my life has helped me everyday. I am forever grateful to you both.

I am indebted to my friends and colleagues, Dr João Moura and Dr Theodoros Stouraitis. João: my thanks for the collaboration we have had over the past years; I hope we can continue to explore the fascinating work we started in Edinburgh. Theodoros: thank you for countless interesting and helpful discussions throughout the PhD. Thanks go to Dr Wu Di and Dr Stuart Turner for proof-reading this thesis.

My thanks go to the SLMC group, past and present, for helpful feedback, comments, and critiques. In particular, I would like to send thanks to Dr Vladimir Ivan and Dr Wolfgang Merkt.

I would like to thank Dr Craig Lucas for creating opportunities for me that ultimately lead and inspired me to take on the PhD.

This thesis was supported by the Costain Group PLC. My gratitude goes to Aled Davies, Andrew Irwin, and Professor Bill Hewlett for several insightful discussions, and Tim Embley for his support. In addition, I would also like to thank Dr James Edwards, Kai Barrington, Dr Henrietta Baker, and the Costain PhD community for a number of interesting conversations.

I would like to thank my Viva examiners Professor Nick Hawes and Dr Michael Herrmann for their time reviewing this thesis, helpful feedback, and an enlightening discussion.

A special thanks to all those who participated in the experiments towards this thesis. Finally, thanks to the rest of my family and friends for many warm memories.

PUBLICATIONS

Parts of the research leading to this thesis has previously appeared in the following peer-reviewed publications. Some passages have been quoted verbatim from the respective sources.

- **Christopher E. Mower**, João Moura, and Sethu Vijayakumar, *Skill-based Shared Control*, Proceedings of Robotics: Science and Systems (R:SS), Virtual Conference, 2021.
  DOI: 10.15607/RSS.2021.XVII.028
  Video 1: https://youtu.be/TwhsgA6fw6M
  Video 2: https://youtu.be/xt_6_uHtLRc
- **Christopher E. Mower**, João Moura, Aled Davies, and Sethu Vijayakumar, *Modulating Human Input for Shared Autonomy in Dynamic Environments*, 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), New Delhi, India, 2019, pp. 1-8.
  DOI: 10.1109/RO-MAN46459.2019.8956304
- **Christopher E. Mower**, Wolfgang Merkt, Aled Davies, and Sethu Vijayakumar, *Comparing Alternate Modes of Teleoperation for Constrained Tasks*, 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Vancouver, BC, Canada, 2019, pp. 1497-1504.
  DOI: 10.1109/COASE.2019.8843265
  Video: https://youtu.be/SWWtT1vHnsM
- Wolfgang Merkt, Yiming Yang, Theodoros Stouraitis, **Christopher E. Mower**, Maurice Fallon, and Sethu Vijayakumar, *Robust shared autonomy for mobile manipulation with continuous scene monitoring*, 2017 13th IEEE Conference on Automation Science and Engineering (CASE), Xi'an, China, 2017, pp. 130-137. [**First Prize for Greatest Potential for Positive Impact at Robots for Resillient Infrastructure Challenge 2017, Leeds, UK**]
  DOI: 10.1109/COASE.2017.8256092
  Video: https://youtu.be/1OyAzHcDL5M

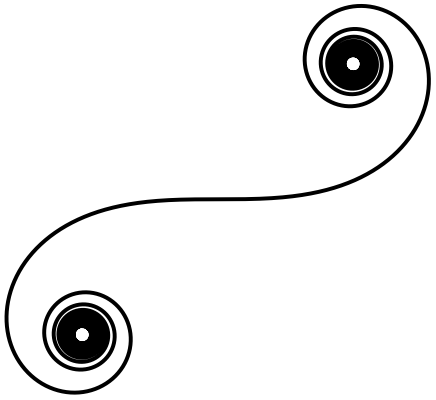
DECLARATION

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Edinburgh, 2021

Christopher Edwin Mower,
January 17, 2022

To Walter and Margaret



CONTENTS

LIST OF TABLES

LIST OF ACRONYMS

BCI	Brain Computer Interface. A device for measuring brain activity.
DoF	Degrees of Freedom. Number of degrees of freedom for a manipulator.
EXOTica	Extensible Optimization Toolkit. The EXOTica library is a generic optimization toolkit for robotics platforms.
FJ	Full Joint. Mode of teleoperation where all joints are teleoperated by an operator.
FK	Forward Kinematics. Forward kinematics is the mapping from a manipulators joint space to some task space.
FT	Full Task. Mode of teleoperation where all task space dimensions are teleoperated by an operator.

HRI	Human-Robot Interaction. Human-Robot Interaction is a multi-disciplinary field combining human-computer interaction, AI, robotics, natural language understanding, and psychology.
IK	Inverse Kinematics. Inverse kinematics is the mapping from a task space to the manipulators joint space.
MPC	Model Predictive Control. An advanced control method that continuously re-plans trajectories.
RJ	Reduced Joint. Mode of teleoperation where some joints are teleoperated by an operator and the remaining joints are autonomously driven.
ROS	Robot Operating System. An open source robotics middleware suite, it is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.
RT	Reduced Task. Mode of teleoperation where some task space dimensions are teleoperated by an operator and other dimensions are handled by autonomy.
TBM	Tunnel Boring Machine. A heavy duty machine used to excavate tunnels.

LIST OF SYMBOLS

Mathematical notation

$\mathbb{R}^n$	The space of real n-dimensional vectors.
$\mathbb{R}^{m \times n}$	The space of real m-by-n matrices.
I_n	The n-by-n identity matrix, i. e. the n-by-n matrix with ones along the diagonal and zeros elsewhere.
0_n	The n-dimensional vector where each element is zero.
$0_{m \times n}$	The m-by-n matrix where each element is zero.

M^{-1}	Superscript "-1" denotes the inverse for a square non-singular matrix M.
$M^{\dagger}$	Superscript "$\dagger$" denotes the Moore-Penrose pseudo-inverse of a matrix M.
M^{T}	Superscript "T" denotes transpose of a matrix M.
$a{:}b$	A range of integers, i.e. $1{:}3 = \{1, 2, 3\}$.
$\frac{dy}{dx}$	The derivative of a variable y with respect to a variable x.
$[a, b]$	Range of values between a, b, if a, b are vectors then the range is element-wise. Note, a, b will always be chosen such that $a \leq b$.
$a \leq b$	Standard inequality notation for scalars $a, b \in \mathbb{R}$.
$x \preccurlyeq y$	$x_i \leq y_i$ for all $i = 1{:}n$ where $x, y \in \mathbb{R}^n$.

System notation

$t \in \mathbb{R}$	Time $[\mathrm{s}]$.
$\mathbb{T}$	Window in time, $\mathbb{T} = [t_0, t_1]$ where t_0, t_1 are given time stamps.
f_s	Sampling frequency.
$x \in \mathbb{R}^{n_x}$	The n_x-dimensional system state.
$u \in \mathbb{R}^{n_u}$	The n_u-dimensional control input to a system.
$h \in \mathbb{R}^{n_h}$	The n_h-dimensional operator interface signal.
$\widetilde{e}$	Environment model, either defined (1) statically (e.g. walls), (2) dynamically (e.g. moving object in robot workspace), or (3) using a combination of (1) and (2).
$\mathbb{X} \subseteq \mathbb{R}^{n_x}$	Space of feasible system states.
$\mathscr{T} \subseteq \mathbb{R}^{n_\phi}$	The n_ϕ-dimensional manipulator task space. The forward kinematics $\phi(\cdot)$ (see below) describes the relation between the configuration space $\mathbb{C}$ and $\mathscr{T}$.
$\mathbb{U} \subseteq \mathbb{R}^{n_u}$	Space of feasible control inputs.
$\mathbb{H} \subseteq \mathbb{R}^{n_h}$	Space of possible interface signals.

Common operators

$\dot{a}$	Derivative of a with respect to time t, i.e. $\dot{a} \equiv \frac{da}{dt}$.
a'	Derivative of a with respect to some variable. The variable of differentiation will be specified as necessary.
$a \cdot b$	Dot-product for two vectors a, b such that $a \cdot b = \sum_i \ a_i b_i$.
$\|\cdot\|$	Vector norm such that $\|a\|^2 = a \cdot a$.
$\|\cdot\|_W$	Weighted vector norm such that $\|a\|_W^2 = a^T W a$ where W is a positive semi-definite weighting matrix.
$\mathcal{R}(\cdot)$	Rotation matrix of order 2 or 3, notation used interchangeably but will be specified when necessary. Also, $\mathcal{R}_a$ is used to represent a 2-by-2 rotation matrix for an angle a $[\mathrm{rad}]$.
$\phi : \mathbb{C} \to \mathscr{T}$	The forward kinematics function that maps configurations $x \in \mathbb{C}$ to the task space $\mathscr{T}$.
$f(a; b)$	A function defined on variables a and parameterized by b.
$\mathrm{diag}(\cdot)$	Diagonal matrix.
$\arg\min_a c(a)$	$\arg\min$ returns a, local to some initial guess, that minimizes the scalar valued function $c(\cdot)$.
$\mathrm{cost}(\cdot)$	Non-negative cost function, high values indicate less optimal, whereas low values indicate more optimal.
$f(\cdot)$	Equation of motion, also known as the system dynamics.
$\mathcal{P}(\cdot)$	Probability of some event.

1

INTRODUCTION

(a) (b)

(c) (d)

Figure 1.1: Industrial teleoperation tasks such as (a) concrete spraying, (b) welding, (c) remote drone piloting for inspection, and (d) nuclear decommissioning each require highly skilled operators.

Teleoperation is a method that enables a human to control a robotic system via an interface (e.g., joystick). Such a method is motivated by issues such as human safety in hazardous environments (e.g., construction site, nuclear power plant, disaster site, war zone), environments where it can be expensive to send humans (e.g., space, underwater, off-shore facilities such as oil-rigs), task performance requires precise micro-manipulation (e.g., minimally invasive surgery, welding), human anatomy is

unable to generate enough power (e.g., excavation), and many others. This thesis is concerned primarily with tasks inspired by the civil engineering sector. Figure 1.1 shows a sample of key tasks needed for the delivery of construction projects.

In construction, there is a high risk of injury or even fatality, potential damage to expensive machinery, a substantial likelihood that raw materials (e.g., concrete) are wasted if the operator is not appropriately skilled at controlling the manipulator, and low task performance can lead to disaster (e.g., tunnel collapse when concrete is not appropriately laid). Ideally, regulators would replace the human workers with an autonomous system in order to maximize safety and improve quality. However, the workspace of the robot is quite often cluttered, dirty, and prone to change. In addition, unpredictable events can further complicate the task; for example, in concrete spraying (Figure 1.1a) some of the sprayed concrete may fall away during operation requiring the operator to readjust their strategy to ensure the tunnel wall meets the required design profile. Without an enormous wealth of computational power and data curation, making classical model-based and modern model-free machine learning approaches scale to such complex tasks would be very difficult. Furthermore, to satisfy regulators, such an autonomous system will be required to, demonstrably, surpass human-level situational and contextual awareness raising difficult questions about how to appropriately represent causal inference at a large scale that is explainable to humans.

The control method most often implemented on devices used in industry is *direct control* - also known as *manual control*. This process takes the signals from the interface and maps them directly onto control signals generating robot motion. This means that the motion of the robot is the sole responsibility of the human operator. This motivates the need for highly skilled operators requiring expensive training taking several months or even years for the worker to become certified. In addition, the tasks can be prolonged and require high levels of manual dexterity. The heavy burden on operators often leads to fatigue that can have a damaging effect on safety.

An alternate approach to direct control is *shared autonomy*. The goal of this approach is to share the load of a task between an autonomous system and the human (Sheridan, 1992). This can mean that the autonomous system relieves the human of certain sub-tasks or extends the capabilities of the human. Another form shared autonomy places one agent (either human or autonomy) in the role of the main decision maker and the other agent acts to refine the chosen actions. Historically, this was known as *cooperative control*. However, more recent literature tends to term this as *shared control* (Niemeyer et al., 2016). Another form of shared autonomy is known as *supervisory control*. In this case, the human tends to give high-level commands to

the system that are translated into low-level motion plans.

The goal of this thesis is to develop novel shared autonomous approaches that assist human operators to perform teleoperation tasks, inspired by the requirements and restrictions of the civil engineering sector, with greater performance and efficiency in order to enable safer working environments for workers, cost effective training, and improved job quality in the future.

CHALLENGES The challenges addressed in this thesis are as follows.

(C1) *Human interface*: the interface that maps a human's physical actions onto the control space needs to be designed so that it is intuitive and promotes high task performance, despite the skill level of the user.

(C2) *Human intention estimation*: to assist the human and improve task performance in the face of changing intentions, a shared autonomous system must be able to obtain the human's intention online. Furthermore, the human should not be burdened by additional sensors (e.g. heart-rate monitors) in order to capture their intention. Thus, the intention should be estimated utilizing already known quantities, i.e. the system, environment, and interface state.

(C3) *Influence of autonomy*: the influence by the autonomy must be sufficiently rich to assist the human when performing a task but not overbearing that might lead to confusion or failure.

(C4) *Dynamic environment*: the environment that the robot is situated in is dynamic. The shared autonomous system should track these changes and update motion plans on-the-fly according to the current goal and state of the environment.

(C5) *Online teleoperation*: the tasks addressed in this thesis require the human to continuously interact with the system and thus adaptable motion plans must be generated online.

RESEARCH QUESTIONS With regards to the challenges stated previously, the research questions addressed by this thesis are stated as follows.

(Q1) How to design an appropriate interface for an operator in target acquisition[1] teleoperation tasks; specifically, what are appropriate mappings and their respective dimensionality that elicit highest performance?

1 *Target acquisition*: a task where the robot must align the central axis of it's end-effector with some point on a surface in order to *acquire* the *target* (e.g. concrete spraying, welding).

(Q2) How to effectively combine human input and autonomy informed by environment sensing to produce optimal robot motions?

(Q3) Can we utilize knowledge of the changing environment and operator intention, to provide effective assistance that adapt plans online?

(Q4) What are the salient features of the operator's expertise? Furthermore, can we optimize the robot motion online which simultaneously respects these features and satisfies physical constraints?

(Q5) Can we find approaches, addressing all the above, that are computationally feasible?

The above list summarizes the main questions addressed by this thesis. Each correspond to one of more of the challenges listed previously.

The first question (Q1) is directly related with the first challenge (C1). When considering the mapping from a humans physical actions to the system control space, there are several important practical considerations. For example, the specific device used by the human and its design, the fidelity of the link between the device and the system (e.g. communication delays), and the specific model used to map the interface signals to the control space. Thus, (Q1) extracts the focus for this these out of these challenges.

The second question (Q2) is related to challenges (C1)-(C3). The question attempts to encapsulate the problem of determining the correct level of autonomy (see discussion in Section 2.1.2), and combining the autonomy with human input to produce meaningful robot actions.

The third question (Q3) relates to challenges (C2) and (C4). Both the human intention and environment are assumed to change over time. This question aims to highlight the problem that these plans should be adaptable and online re-planning is necessary.

The fourth question (Q4) is corresponds to challenges (C1)-(C3). An expert will likely exhibit certain patterns/features when performing tasks. This question aims to identify what are those features and how they can be utilized when assisting less skilled operators.

The fifth question (Q5) directly corresponds to challenge (C5). The tasks considered by this thesis require continuous human interaction and so the computation time must be sufficiently small to facilitate online teleoperation.

CONTRIBUTIONS This thesis makes the following contributions.

Evaluation for several modes of teleoperation[2] (Chapter 3): Several user studies are presented that further understanding about optimal interfaces for humans interacting with teleoperation systems. A number of hypotheses are made regarding how the user performance is affected by the *mode of teleoperation* - categorized by the mapping between interface signals and controls and its dimensionality. To test the presented hypotheses, a number of experiments were developed inspired by industrial tasks that include spraying in a simulator and target acquisition tasks performed on a lab mock-up implemented on a KUKA LWR arm. A number of modes of teleoperation are compared, exploring both the number of dimensions of the control input as well as the most intuitive control spaces (i.e. joint or task space). Subsequent analysis indicates that higher performance is achieved when commands are submitted in low-dimensional task spaces as opposed to joint space manipulations. Through an empirically driven investigation, an analysis of the sub-task allocation problem is provided. Extensive data sets of results have been collected containing both objective and subjective metrics. A comparison between two subsets of the participants based on the participants personal habits identified to effect performance. In addition, two generalized performance metrics for Fitts' law are proposed relating to teleoperated target acquisition tasks.

Improving performance by modulating human input utilizing task and environment tracking evolution and modulating[3] (Chapter 4): Building on the understanding gained in the aforementioned user studies, an optimization-based representation for shared autonomy in dynamic environments is proposed. Real-time tractability is ensured by modulating the human input with the information of the changing environment in the same low-dimensional task space, instead of adding it to the optimization cost or constraints. The method is applied to realistic applications: grasping objects in a cluttered environment, and a spraying task requiring sprayed linings with greater homogeneity. Furthermore, evidence is provided that indicates the method, when applied to spraying leads, to higher coverage than the current direct control approach. A description of how to apply the proposed method to two case studies by defining various objective terms: a grasping task where the human guides the robot and is assisted in collision avoidance and, by a prediction scheme, choosing appropriate grasp orientations; and a spraying task that demonstrates how to modulate the human input with changes in the environment introduced by the sprayed mate-

2 Christopher E. Mower, Wolfgang Merkt, Aled Davies, and Sethu Vijayakumar, *Comparing Alternate Modes of Teleoperation for Constrained Tasks*, IEEE International Conference on Automation Science and Engineering (CASE), 2019. DOI: 10.1109/COASE.2019.8843265

3 Christopher E. Mower, João Moura, Aled Davies, and Sethu Vijayakumar, *Modulating Human Input for Shared Autonomy in Dynamic Environments*, IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 2019. DOI: 10.1109/RO-MAN46459.2019.8956304

rial. Several evaluations were implemented on a simulated 7 *Degrees of Freedom* (DoF) KUKA LWR arm.

Skill-assistance for patterned motion[4] (Chapter 5): Performing a number of motion patterns – referred to as *skills* – (e.g., wave, spiral, sweeping motions) during teleoperation is an integral part of many industrial processes such as spraying, welding, and wiping (cleaning, polishing). Maintaining these motions whilst simultaneously avoiding obstacles and traversing complex terrain requires expert operators. In this work, a novel skill-based shared control framework is proposed for incorporating the notion of skill assistance to aid novice operators to sustain these motion patterns whilst adhering to environmental constraints. The shared control method uses streaming joystick data to estimate the model parameters that provide a description of the operator's intention. We introduce a novel parametrization for state and control that combines skill and underlying trajectory models, leveraging a special type of curve known as Clothoids. This new parameterization allows for efficient computation of skill-based short term horizon plans, enabling the use of a *Model Predictive Control* (MPC) loop. Several experiments on a realistic hardware mock-up is presented, validating the effectiveness of the method to recognize a switch of intended skill, and indicates an improved quality of output motion, even under dynamically changing obstacles.

THESIS OUTLINE The remainder of this thesis is organized as follows:

- Chapter 2 starts by introducing basic concepts and notation for teleoperation. Next, an overview of the application is presented that provides the main source of inspiration for this work (i.e. concrete spraying), then a summary of related literature from that domain. Finally, the chapter is completed with a review of related literature on teleoperation interfaces, intention recognition, and shared autonomy.
- Chapter 3 formalizes the concept of modes of teleoperation and introduces several used in our experiments. A preliminary simulation-based study is reported that gives early indications of modes that lead to higher performance. Fitts' law, and two generalized metrics for performance evaluation are introduced. Finally, results are reported from an extensive user study that aid certain design choices in future chapters - and indeed future work.

4 Christopher E. Mower, João Moura, and Sethu Vijayakumar, *Skill-based Shared Control*, Proceedings of Robotics: Science and Systems (R:SS), 2021. DOI: 10.15607/RSS.2021.XVII.028

– Chapter 4 introduces a proposed input modulation method for combining human input and autonomous assistance in the context of continuously changing environments. Several experiments are reported that highlight the benefits of this method.
– Chapter 5 identifies key features of an operator's expertise for industrial tasks such as concrete spraying, welding, plastering, and cleaning. Our skill-based shared control method is proposed for planning motions that respect these features whilst ensuring system constraints. Finally, a description of a potential extension of the method is presented.
– Chapter 6 summarizes the thesis, discusses limitations, and outlines future work.
– Appendix A describes the implementation of a system for robust *sliding autonomy* - a form of supervisory control that allows the system to regulate itself between different levels of autonomy depending on the state of the human and environment.

2

BACKGROUND AND RELATED WORK

In this chapter, background and relevant work for shared autonomy is reviewed. This thesis has been supported by the Costain Group PLC and so the source of inspiration draws heavily on the civil engineering sector, specifically the tunnel construction method known as *concrete spraying*, also called *shotcreting* (Figure 1.1a). First, Section 2.1 starts by giving a formal treatment of the teleoperation setup, i.e. key models and notation used in this thesis. Then in Section 2.2, an overview of industry-relevant literature is given. Next, Section 2.4 gives a review of various teleoperation interfaces and highlights some of the difficulties for an operator to provide inputs to a teleoperation system. Sections 2.2 and 2.4 will help inform design decisions in subsequent chapters. Next, intention estimation and prediction methods are discussed in Section 2.6 where an operator's intention is inferred from data. A brief summary of relevant works in motion planning are given in Section 2.7. Finally, Section 2.8 reviews relevant background material for shared autonomy.

2.1 TELEOPERATION PRELIMINARIES

2.1.1 *Overview*

Teleoperation, also known as *telerobotics*, is illustrated in Figure 2.1. Applications of teleoperation are vast, examples include construction (Honegger, Schweitzer, et al., 1997; Honegger and Codourey, 1998; Girmscheid and Moser, 2001), manufacturing processes (e.g., welding) (Erden and Billard, 2015), disaster response (McGill et al., 2015), space (Hirzinger et al., 1993; Bejczy, 1994; Leidner et al., 2019), undersea and science (geology, biology) applications (Yoerger and Slotine, 1987; T. Wang et al., 2020; Brantner and Khatib, 2021), toxic waste clean up (nuclear decommissioning) (Goertz, 1952; Goertz, 1954; Marturi et al., 2016), agriculture (Murakami et al., 2008; Peña et al., 2018), mining (Kwitowski, Lewis, et al., 1989; Kwitowski, Mayercheck, et al., 1992; Hainsworth, 2001), military (Yamauchi, 2004), policing (Gong et al., 2017),

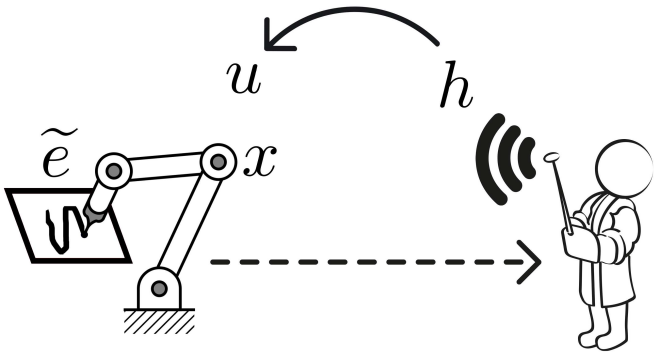


Figure 2.1: Teleoperation overview. The operator signal h is transformed to a control input command u that updates the system state x. The mapping $h \rightarrow u$ in shared autonomy is informed by a model of the environment $\widetilde{e}$ that can be some pre-specified model or data collected online leveraging multi-modal sensing.

firefighting (Tamura et al., 2020), assistive devices for the disabled (Campeau-Lecours, Lamontagne, et al., 2017), robotic surgery (Guthart and J. Salisbury, 2000; Isaac-Lowry et al., 2017), and entertainment (Pereira et al., 2017). In-fact, teleoperation is particularly pervasive in robotics, the majority of robot systems have a manual control mode allowing a human to specify the motion (Sheridan, 1992). In addition to the aforementioned applications, driving can also be interpreted as a teleoperation task. Many groups and companies across the world are working towards autonomous driving, relevant to this work is the literature surrounding semi-autonomous driving (Seshia et al., 2015) - some of the relevant works on this as apart of Section 2.8 is discussed. The following formalizes the process of teleoperation.

The teleoperation process involves a human, the operator (sometimes called the teleoperator), who interacts with an interface (e.g., a joystick) and a robotic system with state $x(t) \in \mathbb{X}$ such that $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ is the state space and t is time. The interface produces signals $h(t) \in \mathbb{H}$ that are transformed to control signals $u(t) \in \mathbb{U}$ such that $\mathbb{H} \subseteq \mathbb{U}$ and $\mathbb{U} \subseteq \mathbb{R}^{n_u}$ are the spaces of interface signals and controls respectively. The controls induce movement in the system that effects the state of the world. The environment is modeled and/or tracked (utilizing multi-modal sensing), and denoted by $\widetilde{e}$ - this can represent any arbitrary measure. The space of interface signals is assumed to a subset of the controls, i.e. $\mathbb{H} \subseteq \mathbb{U}$, however in general that is not always the case. Industrial manipulators implement velocity-based controls (Honegger and Codourey, 1998). As such, let us assume in this work that the controls $\mathbb{U}$ represent velocities. In addition, the operator requires some feedback regarding the system state (this is represented by the dashed line in Figure 2.1). Industrial tasks like concrete spraying, welding, and nuclear decommissioning will typically involve the human being nearby the robot (Figures 1.1a, 1.1b, and 1.1d). However, in cases where the operator is remote they must rely on visualization and/or camera feeds (Figure 1.1c).

Finding appropriate methods for choosing controls $u(t)$ is ubiquitous in robotics. In this thesis, an optimization-based framework is leveraged to achieve this goal. Let t_c denote the current time, and t_f be a future time, and $\mathbb{T} = [t_c, t_f]$ represents the time horizon. Optimal controls are chosen by solving the receding horizon optimal control problem

$$x^*, u^* = \underset{x,u}{\arg\min}\ \text{cost}(x, u) \quad \text{subject to} \quad \begin{cases} x(t) \in \mathbb{X}(\widetilde{e}), u(t) \in \mathbb{U} \\ \dot{x} = f(x, u), t \in \mathbb{T} \end{cases} \tag{2.1}$$

where $\mathbb{X}(\widetilde{e}) \subseteq \mathbb{X}$ represents the space of feasible states parameterized by the model of the environment $\widetilde{e}$, $\dot{x} \equiv \frac{dx}{dt}$ is the derivative of state with respect to time $f(\cdot)$ is a state-dependent deterministic transition function (equations of motion). Actuation of industrial manipulators tend to be high-precision, hence the choice (an assumption) that the transition function f is deterministic. Equation 2.1 can be transcribed to a numerical optimization problem (Kelly, 2017), and then solved by off-the-shelf software such as SNOPT (Gill et al., 2002), qpOASES (Ferreau et al., 2014), and IPOPT (Wächter, 2009).

2.1.2 *Levels of automation and terminology*

Teleoperation and shared autonomy are broad concepts with many variants. Often, the term "shared autonomy" refers to the idea that a system is executing actions informed by two streams of information or agents: the human and autonomy. As described in the previous section, the human interacts with the system via an interface (e.g. joystick) and often has some feedback from the task (e.g. visual and/or force). The autonomy is informed by sensing data or prior information. Typically, one agent acts to assist the other. Often, it is the case that the human is assisted by the autonomy (A. Dragan and Srinivasa, 2013; Reddy et al., 2018; Javdani et al., 2018). Alternatively, the human could be utilized instead as corrective input to an otherwise autonomous system (Bajcsy et al., 2017; Hagenow et al., 2021a).

A fitting analogy for shared autonomy was presented by Paolo Robuffo Giordano at the International Conference on Robotics and Automation (ICRA) 2020 Workshop entitled "Shared Autonomy: Learning and Control". Direct control is like riding a bicycle: the person must be fully focused on riding, otherwise they will crash. Shared autonomy is like riding a horse: the horse will handle the balance, rough terrain, and avoid obstacles whereas the human gives high level commands[1].

1 The talk by Paolo Robuffo Giordano can be found here: https://youtu.be/P5J8khEa1WI.

There are, in addition, works that propose systems that have the ability to switch between different levels of autonomy. In the literature this has been referred to as *sliding autonomy* (Dias et al., 2008; Sellner et al., 2006), see also Appendix A, and *variable autonomy* (Chiou, 2017).

The wealth of work in teleoperation covers a spectrum of automation. To categorize these works and facilitate regulatory considerations for applying this technology in the real world, several taxonomies have been developed. Levels of automation are most well-defined in the field of autonomous vehicles (SAE International, 2016). However, taxonomies for several other scientific and industrial domains have been proposed, such as human-robot interaction (Beer et al., 2014), undersea teleoperation (Sheridan and Verplank, 1978), manufacturing (Frohm et al., 2008), and medical robotics (G.-Z. Yang et al., 2017). A comprehensive review is presented by Vagia et al., 2016. Taxonomies, such as those mentioned above, can be useful to clearly define evaluation criteria, support ethical and legal considerations, as well as define the context of use (i.e. the tasks where certain levels can be used and otherwise).

The terminology in thesis has been aligned with the categorization by Niemeyer et al., 2016, Sec 43.3: direct teleoperation (simply executes user input), shared control (user continuously interacts with system and robot motion is a result of some combination of the user and autonomy), and supervisory control (human supplies discrete high-level commands that are mapped to sequences of autonomous motions).

2.2 INDUSTRY RELATED LITERATURE

In this section, literature from industry is discussed, specifically work in the domain of underground tunnel construction - concrete spraying. Inspiration is drawn throughout this thesis from concrete spraying - the process of spraying wet concrete onto an excavated surface for tunnel support. Section 2.2.1 gives an overview of the concrete spraying process, highlights performance indicators, and then summarizes recommendations from the technical literature on how an operator, or nozzleman, should operate a concrete spraying unit. In Section 2.2.2 an overview of related concrete spraying literature is given that have worked towards moving the concrete spraying process towards automated and semi-automated approaches.

2.2.1 *Overview of the concrete spraying process*

Concrete spraying is a key stage in many tunnel construction projects where wet concrete is sprayed onto a wall to provide temporary or final structural support. The method, originally known as gunite, was developed early in the 20th century by the

Figure 2.2: Concrete application in a freshly excavated tunnel using a 5-DoF concrete spraying unit. Image provided by Costain Laing O'Rourke Joint Venture.

American inventor Carl Ethan Akleley. The process entailed a nozzleman wielding a spraying nozzle by hand and consequently the application process was considerably tiring. By the 1980's specialized heavy machinery such as the MEYCO Oruga in Figure 2.2 was developed where the nozzle was attached to the end of a robotic arm able to handle greater flow rates and allow the manipulator arm to reach into more difficult places.

The concrete spraying process is an iterative procedure. Each iteration constructs approximately one meter of the tunnel consisting of three stages:

1. ***Excavation***: a small amount of dirt/rock is taken away either by digging or a controlled explosion.

2. ***Temporary support layer***: a lining of water-proofing material and then concrete is sprayed onto the excavated tunnel cavity.

3. ***Long lasting support layer***: either pre-cast reinforced concrete blocks are put in place or another thickener layer of concrete is sprayed to support the tunnel.

The quality of the sprayed concrete is measured in terms of the layer thickness, compaction, homogeneity, and proportion of rebound[2]. The layer thickness is tracked over successive iterations by taking a LIDAR scan of the tunnel cavity before spraying begins, then another after the nozzleman has finished spraying - the thickness at a point on the cavity can found by the difference between these measurements. A surveyor will sample a number of key points from the tunnel cavity in order to estimate the overall structural integrity of the sprayed lining.

The manipulation of the device by the operator during spraying is a critical factor contributing to job quality. To minimize rebound and improve quality, Ballou, 2003

2 *Rebound* is the amount of concrete that falls away from the wall and consequently wasted. Rebound tends to generate a lot of dust.

recommends that the nozzleman should keep the spraying nozzle perpendicular to the surface of the tunnel and at a specific stand-off distance (roughly 1m-2m), use small circular motions for building up layers quickly, and large sweeping motions for building up layers slowly, start at the bottom and move upwards, and do not reduce the pressure or flow rate.

2.2.2 *Developments towards (semi-)automation*

A number of articles in the robotics and civil engineering literature have made developments towards automated and semi-automated concrete spraying - in this section an overview of those relevant works are given.

Automating the concrete spraying process is an extremely difficult task due to the varying complexity of jobs and different tunnel conditions. Not only the environment, but the process is prone to unpredictable events (e.g., rebound, as mentioned above). Nonetheless, attempts have been made in certain cases. A. J. Rodriguez and Río, 2007 studies what such a system would require. This includes, data that could be transmitted back and monitored by a human supervisor, systems that could monitor complex sensing and predict potential failures and/or disturbances, and a number of requirements regarding the concrete mixture to reduce rebound.

A concrete spraying system was developed by Girmscheid and Moser, 2001 capable of manual control, semi-automated spraying, and fully automated spraying. The system was implemented on an 8-DoF MEYCO Robojet Concrete Spraying Unit, and featured vision support that allowed the operator to visualize the sprayed concrete coverage (Honegger, Schweitzer, et al., 1997). In the manual, and semi-automated modes the operator controlled the device using a 6-DoF space mouse, and the system mapped these commands into the robot task space as end-effector velocities. This was converted to joint velocities by an inverse kinematic mapping (Honegger and Codourey, 1998). The semi-automated mode assisted the operator by keeping the nozzle at a constant stand-off distance and perpendicular to the spraying surface; this mode however required a pre-collected (static) model of the tunnel cavity. For the case of the fully automated mode a trajectory was generated using a squared-off raster scan; the human (now acting as a supervisor), could adjust some parameters prior to spraying in order to modify the trajectory for different site conditions. Each mode was used in specific circumstances; the manual control mode was used when the conditions were extremely irregular; the fully automated mode was only applicable when the excavated surface was sufficiently smooth; and the semi-automated mode could be used in other cases.

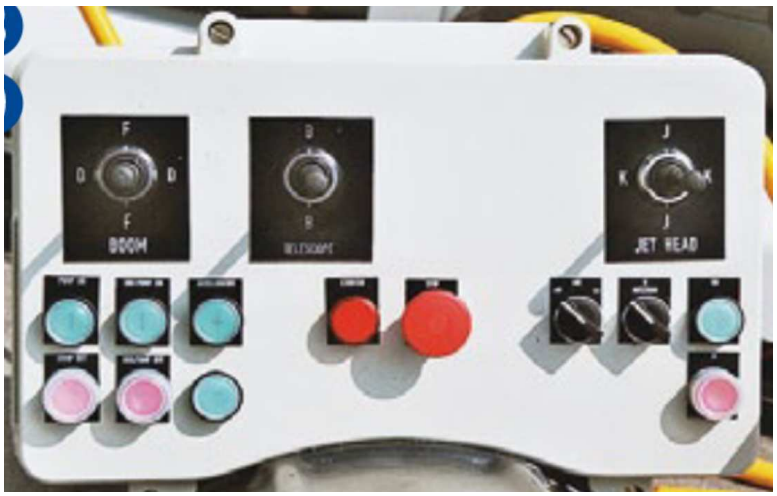


Figure 2.3: The control pad an operator uses to manipulate a MEYCO Oruga Concrete Spraying Unit. Several buttons at the bottom modify operational parameters, and the three joysticks above control the individual joint velocities of the manipulator.

An alternative, fully automated system, was developed on a 5-DoF Sika-Putzmeister PM-407 by Nabulsi et al., 2010 that implemented a similar system to the system described previously. However, in addition, this system adjusted the trajectory online with respect to an estimated concrete layer thickness (A. Rodriguez et al., 2009). More recently, G. Liu et al., 2021 developed an automated concrete spraying system for an HPS3016C concrete spraying unit that plans a trajectory by sub-dividing a tunnel section into a square grid. A starting point is chosen as the central position of one of the bottom corner segments, then a path is created using a raster scan. The end-effector velocity is adjusted at each grid point with respect to the estimated depth (i.e. if the depth to be filled is large then the velocity will be slow, and fast otherwise). Again, both these automated systems are only applicable to conditions where the surface is sufficiently smooth.

Whilst there has been a number of approaches to concrete spraying with varying levels of autonomy, companies such as the Costain Group PLC typically utilize the 5-DoF MEYCO Oruga concrete spraying unit. This uses a manual mode where each joint velocity is determined by a joystick axes. The operator interfaces with the system using a 3-joystick control pad, shown in Figure 2.3.

2.2.3 *Observations from the industry related literature*

In the previous two sections, an overview of the concrete spraying process was given, and a summary of some key works that have moved towards autonomous and shared autonomous systems in concrete spraying. In this section, some observations that will help us determine key assumptions regarding the setting of tunnel construction are

stated. These assumptions will be used in later chapters like "pegs of knowledge" to hang future ideas.

The MEYCO Oruga concrete spraying unit (Figure 1.1a and Figure 2.2) is used in many construction projects in the UK, these are typically used due to their small size[3]. These manipulators use direct control where each joystick axis indicates a joint velocity. To the best of the authors knowledge, task-based control has not been implemented on these manipulators as has on the larger manipulators, i.e. the 8-DoF MEYCO Robojet (Girmscheid and Moser, 2001). In addition, early promising reports indicate that task-based control simplifies the manipulator operation and thus improves overall job quality (Honegger, Schweitzer, et al., 1997). However, again, to the best of the authors knowledge there exists no published works that explore the best operator interface with regards to joint-based control as opposed to task-based control and the number of human-controlled dimensions for these types of tasks.

An accurate model of the environment is collected using LIDAR sensors pre-spraying and post-spraying in order to measure the layer thickness. There are works that leverage this data to update a visualization of the sprayed concrete (Honegger, Schweitzer, et al., 1997). Thus, access to a pre-collected static model of the environment is assumed. Online estimation of the environment is not possible to collect due to the high levels of dust and clutter. It can be assumed however that it is possible to monitor progress, e.g., virtual positions along the collected surface that represent the path taken by the spraying unit.

The concrete spraying task and environment is unpredictable; the humans experience is required for dealing with problems such as obstacles, sharp corners, holes, water leaks, and other potential faults. Given the unstructured and complex nature of the tunneling construction method, there is an inherent difficulty in the automation of the concrete spraying process.

Typically the prismatic joints used in industrial manipulators require higher oil consumption. Thus, it is better to minimize the motion of these joints by fixing them during spraying (Honegger and Codourey, 1998; Girmscheid and Moser, 2001). Thus the robot can be assumed to be comprised of only rotational joints.

It is possible that there exists obstacles in the scene to avoid spraying (Honegger, Schweitzer, et al., 1997; Girmscheid and Moser, 2001). In these cases, the operator must avoid spraying in these areas. Also, these areas can be created during operation, i.e. once an area has been sprayed that meets the design profile it should be avoided in the future. Whilst it is assumed that is is not possible to accurately measure layer thickness using sensing, mentioned above, it is assumed that this can be estimated

3 Evidenced by conversations and correspondences with Costain personnel.

using empirical models (A. Rodriguez et al., 2009). These models, can be tuned by an expert for different concrete mixes and job-sites.

Current concrete spraying systems cannot vary the spraying velocity (i.e. the speed at which a volume of concrete is ejected from the nozzle) during operation - this is a parameter that is adjusted before the task is started and fixed during the task. Thus, the rate of concrete deposition must be regulated by the operator.

Fatigue affects a person's ability to think clearly and respond appropriately to stimuli; it has been linked with workplace injury (Swaen et al., 2003). The effects of fatigue have been shown to reduce goal-directed attention and promote stimulus-driven actions (Boksem et al., 2005), i.e. people become easily distracted and act in response to events/issues as they happen without considering the long term impact of their actions. With regard to construction, Zhang et al., 2015 have shown that workers who reported frequently feeling tired were more likely to report difficulty with cognitive and physical functions. Prolonged working hours, as in concrete spraying, undoubtedly result in fatigue and as a result a goal for the development of shared autonomous systems by the robotics community and indeed in this work is to reduce the cognitive load/burden on the operator.

2.3 LIMITATIONS OF TELEOPERATION

Satisfactory completion of complex robotic tasks often requires human intervention, via teleoperation, due to high-risk and unpredictability of the task and environment. However, there are a number of factors that negatively impact direct teleoperation, such as (note "L" stands for *Limitation*):

(L1) inadequate or unintuitive interfaces,

(L2) coarse and highly variant input commands from the operator,

(L3) poor observability of the task by the operator,

(L4) a limited supply of skilled workers,

(L5) deficient fidelity of the link between operator and the robot (for instance caused by network latency), and

(L6) operator fatigue due to high levels of concentration for prolonged periods of time.

These limitations often lead to excessive cognitive loads and consequently dangerous working environments.

With regards to the limitations listed above, improving operator observability (L3) and accounting for computer communication channels (L4) is out-of-scope of this thesis.

2.4 TELEOPERATION INTERFACES

There are a vast number of interfaces that have been developed for teleoperation tasks. In this section, an overview of these interfaces and associated applications is given, and the limitations for providing teleoperation control signals is highlighted.

A wide variety of input modalities for a large number of tasks have been developed in the literature over the years. Mechanical links used to manipulate radioactive materials from a safe distance (Goertz, 1952; Goertz, 1954). Exoskeletons map operator arm motions onto robot joint motion (Bergamasco et al., 1994; Y. S. Kim et al., 2005; Frisoli et al., 2005; Schiele and Hirzinger, 2011) and utilized in underwater manipulation tasks (Gancet et al., 2016). A 6-DoF space mouse was used to control the end-effector MEYCO Robojet concrete spraying unit (Girmscheid and Moser, 2001). A computer mouse has been used as an interface to define goal poses for robot reaching and grasping tasks (You and Hauser, 2011; Leeper et al., 2012; Hauser, 2013). Several works have developed systems for controlling a robot arm mounted on a wheelchair for assistive tasks, Tsui et al., 2008 uses a touch screen interface, Campeau-Lecours, Côté-Allard, et al., 2019 use a gamepad controller, and Driessen et al., 2001 use a combination of a joystick for timed events and a keyboard for discrete events. Sian et al., 2002 achieves whole body control of a humanoid using two joysticks. A fingertip controller was developed for touching virtual objects (Massie, J. K. Salisbury, et al., 1994). Hand controllers have been used for remote teleoperation in space (Hirzinger et al., 1993). Similarly hand-motion capture systems for controlling a robot arm was explored by Rakita et al., 2017. Haptic interfaces (A. M. Okamura, 2004; A. M. Okamura, 2009) are often used in surgical robotics (Simorov et al., 2012).

Since their inception, there has been growing interest in the use of *Brain Computer Interface* (BCI) (Schwartz et al., 2006; Luth et al., 2007; Galán et al., 2008; Collinger et al., 2013; Schröer et al., 2015). These devices measure brain activity and transform the signals into control commands. Recent work highlights the potential benefits of a BCI in construction tasks (Y. Liu et al., 2021). However, these devices suffer greatly from inter-subject variability (Saha and Baumert, 2020) meaning that the generalization between different workers and tasks requires great calibration efforts. Until these issues are resolved a BCI will unlikely to be a viable solution for construction tasks involving a complex manipulator arm.

Prior work has explored techniques that guide robot motions via synthesized constraints, i.e., potential fields and virtual fixtures. Potential fields, originally developed by Khatib (Khatib, 1985), guide a user towards or away from a goal or obstacle (Crandall and M. A. Goodrich, 2002). Virtual fixtures specify guidance and/or forbidden regions in task space (Rosenberg, 1993; Kosari, Rydén, et al., 2014a).

The work of A. Dragan and Srinivasa, 2013 formalize assistive teleoperation under the framework of policy blending. The system, grounded inverse reinforcement learning, estimates and predicts a human policy then arbitrates this with the observed human policy. Other learning-based methods have also been developed by Abi-Farraj et al. (Abi-Farraj, Osa, et al., 2017) and attempt to refine unskilled operator input based on skilled operator input learned by exploiting *learning from demonstration* techniques.

A number of strategies with varying levels of assistance were compared by You and Hauser, 2011 for a reach-to-target task. Their user study indicated participants preferred those strategies with higher levels of autonomy. D.-J. Kim et al., 2012 compared manual and autonomous control modes for a pick-and-place task. The results of their experiments concluded that in-fact participants preferred less autonomy. A study by Leeper et al., 2012 compares a number of strategies for assisted and non-assisted remote robot grasping. These studies compare various methods for varying levels of autonomy and interface designs. The conflicting results for pick-and-place tasks provides motivation for this work on related tasks (target acquisition tasks, e.g. spraying/welding/wiping). An investigation and analysis to provide an explanation for these contradictory results is warranted, however, this is out-of-scope of this work.

To the best of the author's knowledge, there has not been work in comparing control spaces that are intuitive and lead to high task performance for unskilled operator's.

2.6 INTENTION RECOGNITION

Various works in the field of *Human-Robot Interaction* (HRI) suggest the use of implicit information, such as human intention, rather than solely explicit information, e.g., interface signals, in teleoperation systems (M. Goodrich and Olsen, 2003; Green et al., 2008; Burridge and Hambuchen, 2009). The are several reasons why it is a good idea to incorporate human intentions in a shared autonomous system. For example: in the presence of time-delays, knowing the human's intention allows the system to anticipate the humans actions; human-input can be noisy, the intent estimation can be

used like a filter to circumvent dangerous actions; and knowing the human intention can allow a system to apply appropriate assistance.

There are numerous works that utilize human intentions in a shared autonomous system. M. Li and A. Okamura, 2003 use Hidden Markov Models to estimate the human's intention in order to regulate the level of assistance. Hauser, 2013 developed a freeform intent inference technique that enabled a 6-DoF robot arm to complete reaching tasks using 2D cursor input from a computer mouse. A. Dragan and Srinivasa, 2013 formalize policy blending where the humans estimation intention is used to generate an autonomous policy that is then merged with the human input via arbitration. Muelling et al., 2017 demonstrate how their system could be integrated with a BCI to assist a human in grasping tasks. Javdani et al., 2018 formulates shared autonomy as a partially observable Markov Decision Process, and applies the system to a multi-object grasping task. Many of these works are made possible by the use of maximum entropy inverse optimal control (Ziebart et al., 2008).

2.7 MOTION PLANNING

Various methods allow robots to operate in dynamic environments. For example, sampling based planners use sensory data in the current time frame (Karaman et al., 2011). Whilst typically robust, this method is generally unable to ensure accuracy and efficiency.

Another approach for handling dynamic environments is to track the changes using some representation, e.g., an occupancy grid, and to continuously check for the intersection between the predicted robot motion or pre-defined plan and the collision map (Hermann et al., 2015). The proposed method is similar in approach to these methods, however, here tasks such as spraying where the full history of the task affects the future motion of the robot are considered, as opposed to a snapshot of the current environment state.

When demonstrations are available, Dynamic Movement Primitives (DMP) (Schaal, 2006) can encode the demonstrated trajectories as a set of differential equations, being able to adapt the robot motion and to handle perturbations during execution. Motivated by the dissimilarities between job sites in the concrete spraying application, no access to demonstrated trajectories is assumed.

A dynamical system approach (Khansari-Zadeh and Billard, 2012) also allows the adaptation of the robot motion on-the-fly with respect to some original plan while ensuring collision-free motions with multiple convex shaped objects. The approach only considers collisions with the end-effector.

The potential field approach (Khatib, 1985) generates collision-free motion within some static environment by summing attractive and repulsive virtual forces defined by some known goal position and obstacle positions, respectively. In order to produce more robust plans, Park et al., 2008 extends it to handle dynamic obstacles. However, the main drawback of the potential field approach is its propensity to local minima, especially in high dimensions.

A number of methods in the literature handle dynamic environments by projecting task goals into alternative representations. For example, Y. Yang, Ivan, and Vijayakumar, 2015 uses a relative distance space representation for real-time null-space motion adaptation to avoid self-collisions and collisions with known objects. Topological-based representations using a combination of writhe and interaction mesh space, as in (Zarubin et al., 2012), can handle complex tasks such as wrapping. Another method by Ivan and Vijayakumar, 2015 leverages a computation scheme typically used to estimate electric flux in the field of electro-dynamics for coverage tasks involving wrapping.

The methods discussed above propose various approaches to deal with dynamic environments without incorporating human-level intervention, thus, being unable to handle complex tasks that require situational and/or contextual awareness.

2.8 SHARED AUTONOMY

There exists a body of research based on shared autonomy that addresses the limitations of direct control. A subset of these works develop arbitration frameworks utilizing operator intent prediction that assumes a probability distribution over the possible goals of the operator. These methods typically attempt to address problems (L1, L2). Hauser, 2013 and Javdani et al., 2018 develop intent recognition systems that blend assistance based on a prediction of the operator's goal. These works enable assistance even when the confidence in the prediction is low and are important when it is difficult to predict a single goal from multiple possibilities. A. Dragan and Srinivasa, 2013 propose a policy blending formalism that relies on the concept of arbitration for blending operator input with a prediction of the operator's intent. From their user study, they verify the importance of the confidence in the prediction for moderating the level of arbitration. In this work, a given intention/goal is assumed and it is shown how to integrate it within this shared autonomy framework.

To address (L1, L2, L5) and to prevent unsafe robot motions, forbidden regions can be defined by specifying virtual fixtures in the task space. These can be pre-specified by an expert (Rosenberg, 1993), or computed on-the-fly with respect to sensory data

(Rydén and Chizeck, 2012). However, in both cases the misplacement of such virtual fixtures due to human error or sensor noise can still lead to unsafe motions.

Rakita et al., 2018 address problem (L3) by continuously providing an effective viewpoint to a remote user using an additional camera-in-hand robot arm. Abi-Farraj, Osa, et al., 2017 addresses problem (L4) by encoding tasks as trajectory distributions demonstrated by expert operators using direct control. The proposed system assists an unskilled operator in manipulation tasks by providing force cues to the user via a haptic interface based on the demonstrated trajectory distributions.

Several works, demonstrated at the DARPA Robotics Challenge Finals, address problems (L5) and (L6). For example, Marion et al., 2017 proposes a piloting system, called Director for the Atlas robot, where the pilot specifies task sequences using high-level motion primitives, and the shared autonomous system incorporates perception and optimization-based trajectory motion planning.

There exists a body of research based on shared autonomy that addresses the limitations of direct control. A subset of these works develop arbitration frameworks utilizing operator intent prediction that assumes a probability distribution over the possible goals of the operator. Hauser, 2013 and Javdani et al., 2018 develop intent recognition systems that blend assistance based on a prediction of the operator's goal. These works enable assistance even when the confidence in the prediction is low and are important when it is difficult to predict a single goal from multiple possibilities. A. Dragan and Srinivasa, 2013 propose a policy blending formalism that relies on the concept of arbitration for blending operator input with a prediction of the operator's intent. From their user study, they verify the importance of the confidence in the prediction for moderating the level of arbitration.

To prevent unsafe robot motions, forbidden regions can be defined by specifying virtual fixtures in the task space. These can be pre-specified by an expert (Rosenberg, 1993), or computed on-the-fly with respect to sensory data (Rydén and Chizeck, 2012). However, in both cases the misplacement of such virtual fixtures due to human error or sensor noise can still lead to unsafe motions.

An alternate branch of work exists that develops *corrective shared autonomy* (Hagenow et al., 2021a; Hagenow et al., 2021b). In this case the roles of the human and autonomy are reversed: the human input is treated as corrective actions for an otherwise autonomous robot. These methods leverage learning from demonstration (Argall et al., 2009) to find a generative policy and projects the operator signals as corrections in the motion synthesis. In our case, high levels of contextual awareness are required to complete these tasks and such methods would struggle to capture the required adaptability.

Cognitive load reduction is generally accepted as an important goal for shared autonomous systems. However, as a system becomes more autonomous under supervisory control techniques, issues inevitably arise with the human operator such as boredom (Cummings, Mastracchio, et al., 2013), distraction (Cummings and Ryan, 2014), and trust (J. D. Lee and See, 2004). To address these issues and maintain the operator's situational awareness of the state of the system and environment the system should maintain a certain level of transparency (Alonso and Puente, 2018). That is, an operator should "feel" like they are in control of the system so that their input is perturbed enough such that the stays within system limitations. A key idea that comes from the shared autonomous driving literature is the principle of minimal intervention (Broad et al., 2019; Schwarting et al., 2017). This idea has a connection with the legibility of robot motion. A. D. Dragan et al., 2013 provide a representation for legibility that can be incorporated inside a cost function.

Many approaches in shared autonomy leverage policy blending (A. Dragan and Srinivasa, 2013; Hauser, 2013). As mentioned above, these take a demonstrated policy for the human, make an estimate of the operator's intent, and predict a policy, then blend the two. This often assumes the operator has one goal/intention to be estimated. However, in general operator's/humans change their mind on-the-fly motivating a method which continuously re-estimates the intention and accounts for any potential change. Part of the focus of this thesis aims to address this issue for online shared control.

3

COMPARING ALTERNATE MODES OF TELEOPERATION

3.1 INTRODUCTION

Teleoperation of industrial manipulators generally requires high levels of concentration and manual dexterity. Classical automation methods do not apply due to the vast differences between job sites that, in addition to being cluttered, constantly evolve over time. Excessive cognitive loads invariably lead to fatigue that can become dangerous. This danger is most prevalent in the construction sector seen by having one of the highest levels of incidents involving fatalities per annum in the Great Britian, see Figure 3.1.

An example of such a task from construction is concrete spraying as shown in Figure 2.2. Here, a skilled human operator manipulates the device via some interface to spray a lining of wet concrete onto a excavated tunnel surface. The tunnel surface is often unstructured, due to excavation, and the operator's visibility is restricted by high amounts of dust. Despite these restrictions, operators are required to manipulate these devices so that they (1) ensure job-site safety, (2) achieve high task performance, and (3) minimize costs (Ballou, 2003). Simultaneously accounting for safety and task performance in teleoperation tasks is required in a variety of applications: nuclear waste disposal (Abi-Farraj, Pedemonte, et al., 2016), space robotics (Lii et al., 2010), and subsea (Murphy et al., 2011).

Devices, such as the concrete sprayer systems, are generally controlled by a human operator on a per-actuator level and consequently these control architectures do not lend themselves to easy operation since they force the operator to submit commands directly in the joint space. Since humans typically model tasks in the three-dimensional Cartesian space they must naturally learn an *Inverse Kinematics* (IK) mapping imposing high monetary costs and time for specialized training. In the literature, methods such as the IK method (Khatib, 1987) and optimization techniques (Zucker et al., 2013) have been developed that allow control commands to be submitted in alternative control spaces. A number of works leveraging these ad-

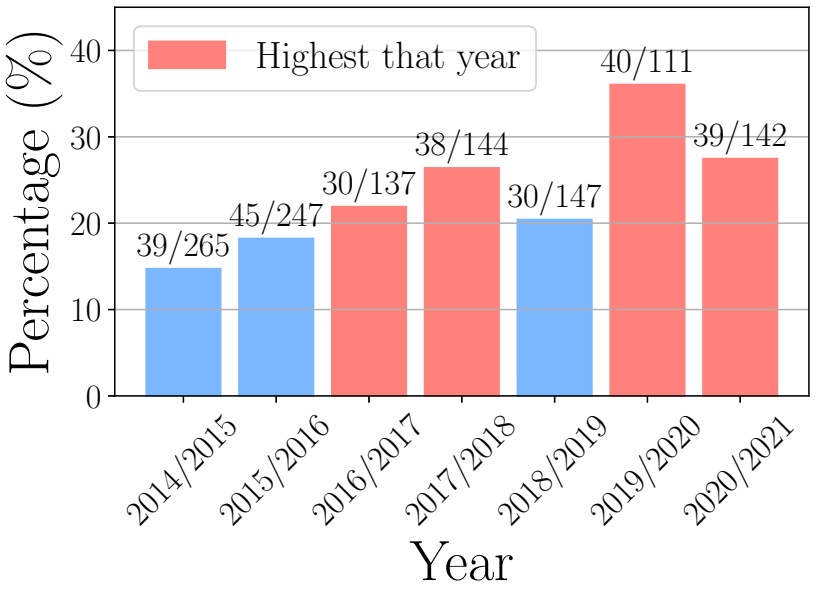


Figure 3.1: Percentage of fatalities from the construction sector in Great Britian. Note, each bar reports the total number of fatailities in construction out of the total number of workplace fatalities recorded that year. By red, the years are indicated when the construction sector had the highest percentage of all sectors. Source: Health and Safety Executive (HSE), *Fatal injuries in the workplace in Great Britain*, 2015-2021, National Statistics.

vancements have developed assistive techniques for teleoperation, for example, virtual fixtures (Rosenberg, 1993), shared control frameworks that merge human input and autonomy (Abi-Farraj, Pedemonte, et al., 2016), and human supervisory capabilities implementing sliding autonomy (Dias et al., 2008; Merkt, Y. Yang, et al., 2017). However, for assistive techniques, in general, the issue of sub-task allocation, i.e., assignment of control dimensions to human or autonomous control, remains one of the main challenges (Inagaki, 2003).

3.2 CONTROL METHODOLOGY FRAMEWORK

In this section, the method by which interface signals h are transformed to control signals u is described, and then the concept of a *mode of teleoperation* is formalized.

3.2.1 *Extracting operator signals from an interface*

Various examples of the interfaces developed for teleoperation were discussed in Section 2.4. These interfaces produce signals h that can represent the interaction by the human in some manner (e.g., joystick data, hand poses, speech, etc.). The space of

possible signals is denoted $\mathbb{H}$ and can be discrete (e.g., buttons), continuous (e.g., joysticks), or a combination of both (e.g., gamepads).

Teleoperation tasks in construction often implement control pads containing buttons/switches and joysticks, see Figure 2.3. For interfaces such as joysticks, gamepad axes, space mouse, the signals h collected from a driver running on a computer are typically drawn from a space given by

$$\mathbb{H} := \{h \in \mathbb{R}^{n_h} : h_i^- \leq h_i \leq h_i^+ \text{ for all } i = 1{:}n_h\} \tag{3.1}$$

where $h_i^-, h_i^+ \in \mathbb{R}$ are lower and upper bounds respectively. Common choices for the bounds are $h_i^- = -1, h_i^+ = 1$ or $h_i^- = 0, h_i^+ = 1$. Note, as mentioned above, the space $\mathbb{H}$ can contain discrete elements – buttons will often be used to customize the manipulator operational parameters (e.g., concrete sprayer flow-rate), these discrete elements are neglected here since the authors interest is the operator manipulation of the robot device. Once h is collected from the interface, it then needs to be projected, by some mapping, onto the control space $\mathbb{U}$.

3.2.2 *Mapping operator signals to control commands*

Operator signals $h \in \mathbb{H}$ are mapped onto the control input space $\mathbb{U}$ using a function

$$u_h = \xi(h) \tag{3.2}$$

where $\xi : \mathbb{H} \to \mathbb{U}$. The control signal u_h can be either a position, velocity, acceleration, or some combination. Prior work suggests that velocity-based controls lead to higher tracking precision (Massimino et al., 1989) and industrial manipulators tend to implement velocity-based control - in this thesis, it is assumed u_h represents velocity inputs.

The choice of ξ is dependent on the interface itself, the task to be completed, and whether the input is a position/velocity/acceleration. The mapping can be complex when using devices such as brain-computer interfaces (Philips et al., 2007; Carlson et al., 2012), and depth cameras (Schultz et al., 2017). These devices are inherently noisy and thus require treatment such as filtering and classification models. It is assumed in this work that the inherent noise from the interface is negligible. In the following sub-sections two potential choices utilized in this thesis are described.

3.2.2.1 *Linear mapping*

A straightforward choice for ξ in Equation 3.2 is a linear mapping. This choice can be formalized by

$$\xi(h) := Mh + b \tag{3.3}$$

where $M = \text{diag}(m_1, \ldots, m_{n_h}) \in \mathbb{R}^{n_h \times n_h}$ is a diagonal matrix that scales each element of h, and $b = [b_1, \ldots, b_{n_h}]^T \in \mathbb{R}^{n_h}$ is a vector that shifts the scaled elements. This choice has been utilized in several examples; Klamt et al., 2020 scale joystick values to velocities that drive the omnidirectional base of the Centauro robot, for a driving setup Johns et al., 2016 map the steering wheel and pedals angles onto the turning angle and the longitudinal acceleration; and Equation 3.3 is used for mapping joystick control pad inputs on the MEYCO Oruga Concrete Spraying Unit and its joint actuators[1]. By choosing appropriate values for m_i, b_i, the elements h_i can be scaled and shifted to suit each respective control dimension.

Joint limits are easily handled by predicting the next joint position and clipping the goal to the joint limit.

3.2.2.2 *Isometric mapping*

The space of operator signals $\mathbb{H}$ is very often of the form defined in Equation 3.1, e.g., joystick, gamepad, spacemouse. In many teleoperation setups control dimensions are orthogonal. For example, the end-effector of the JACO robot arm is controlled by regulating the its Cartesian position and orientation (Campeau-Lecours, Maheu, et al., 2016). When applying joystick commands using the space defined by Equation 3.1 and implementing the map given by Equation 3.3, this can lead to inconsistent maximum control magnitude. Take for example the simple case where an operator teleoperates under direct control a point on a plane whose control u_h is the two dimensional velocity in the base frame. Consider the case when the linear mapping given by Equation 3.3 is implemented with $M = I_2$ and $b = 0_2$. Let us assume the user pushes the joystick to a maximum $h = [0, 1]^T$, then the magnitude of the velocity is $\|u_h\| = 1$. Let us now conceive that instead the user pushes the joystick to another maximum $h = [1, 1]^T$. In this case the magnitude of the velocity is $\|u_h\| = \sqrt{2}$. This is not ideal, since for different maximal joystick inputs the point-mass will have different target velocities.

1 Verified during discussions with expert personnel at Costain.

To preserve an isometric maximal control magnitude an alternative mapping is defined by

$$\xi(h) := \begin{cases} \nu^+ \min(\|h\|, 1) \frac{h}{\|h\|} & \text{if } \|h\| > 0 \\ 0_{n_h} & \text{if } \|h\| = 0 \end{cases} \tag{3.4}$$

where $0 < \nu^+ \in \mathbb{R}$ is the scalar representing the magnitude of the maximum control, and $\min(\cdot)$ returns the value whose value is minimal.

Some tasks, such as concrete spraying, wiping, and welding, can be thought of two dimensional task along a surface. In these cases h has dimension $n_h = 2$. For this special case, Equation 3.4 can be re-defined without using an *if*-clause. Thus, define the mapping

$$\xi(h) := \nu^+ \min(1, \|h\|) \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \tag{3.5}$$

where $h = [h_0, h_1]^T \in \mathbb{H}$, and $\theta = \text{arctan2}(h_1, h_0)$.

3.2.3 *Generating robot motion from command signals*

For this chapter, let the state x represent the joint positions of the robot, and let us model the process of teleoperation by

$$x^* = \arg\min_x \text{cost}(x; u_h) \quad \text{subject to} \quad x \in \mathbb{X} \tag{3.6}$$

where $x \in \mathbb{X}$ is the robot state, $u_h = \xi(h_c)$ such that $h_c = h(t_c)$ is the current operator input and ξ is Equation 3.2, $\text{cost}()$ is a scalar valued cost function modeling optimal robot motion (i.e. low values indicate better states x), and $\mathbb{X}$ is the feasible state space represented by a set of equality and inequality constraints. Motion is generated by sequentially solving Equation 3.6 at a given sampling frequency f_s [Hz] and setting x^* as the target joint state. For this chapter, the space $\mathbb{X}$ is given by

$$\mathbb{X} := \left\{ x \in \mathbb{R}^{n_x} \;\middle|\; \begin{array}{ll} x^- \preccurlyeq x \preccurlyeq x^+ & \text{(joint position limits)} \\ \dot{x}^- \preccurlyeq \dot{x} \preccurlyeq \dot{x}^+ & \text{(joint velocity limits)} \end{array} \right\} \tag{3.7}$$

where x^-, x^+ are lower and upper joint position limits, $\dot{x} = \frac{1}{\delta t}(x - x_c)$ is the joint velocity, x_c is the current joint state, $\delta t = 1/f_s$ is the control loop duration, and $\dot{x}^+$ are lower and upper joint velocity limits. Let us assume that the previous joint position is

used to seed every conservative optimization. Given also that the constraints (Equation 3.7) are linear, let us further assume Equation 3.6 can be solved fast enough for real-time actuation.

3.2.4 *Task allocation*

Task allocation, in the context of teleoperation, is the assignment of control dimensions to human or autonomous regulation – a design choice. Determining the best assignment between interface and control axes is a non-trivial problem, with severe consequences to user experience (Inagaki, 2003) - .

In teleoperation, n_u-dimensional inputs $u \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ are mapped to robot motions by solving Equation 3.6. Individual axes of u can be aligned with joint axes (i.e. individual manipulator *Degrees of Freedom* (DoF)'s) or task space axes (e.g., end-effector positional and rotational axes) – part of the design decision.

The dimensionality of the operator input space n_h is bounded by $n_h \leq n_u$. Equality is clearly attained for direct control systems. For shared control systems, equality can be attained however it is often the case that the operator regulates a subset of the controls and the autonomy handles other dimensions (Herlant et al., 2014; You and Hauser, 2011). Additionally, it can be the case that the human and autonomy can occupy the same space. This will form the focus of Chapter 4 and Chapter 5, in this chapter it is assumed there is a split between the control dimensions assigned to human and autonomy.

3.2.5 *Mode of teleoperation*

Let us formalize the concept of a *mode of teleoperation*.

Definition 3.1 (mode of teleoperation)**.** The chosen value for n_h coupled with the space (i.e. joint/task space) in which the operator commands h are submitted categorizes what is defined as the *mode of teleoperation*.

In order to design effective shared autonomy systems several design choices must be made. Under the frame work of modes of teleoperation, this chapter explores which is the most appropriate mode for target acquisition tasks inspired by construction – hypotheses are derived from observations from industry and stated in later sections.

3.2.6 *Defining several modes of teleoperation*

In this section, a number of modes of teleoperation are defined that will be used in later comparisons. Let us assume for each mode the operator input space $\mathbb{H}$ is given by Equation 3.1 such that $h_i^- = -1, h_i^+ = 1$ for all $i = 1{:}n_h$. For each mode let us define the values of n_h and n_a, as the dimensions of the space of human inputs and the dimension of the autonomous inputs respectively.

FULL JOINT MODE (FJ) Industrial manipulators are operated on a per-actuator level. That is, there exists a one-to-one mapping between each joystick axis and manipulator actuator. Therefore, for an n-DoF manipulator $n_h = n$ and $n_a = 0$. The *Full Joint* (FJ) mode can be modeled using

$$\text{cost}(x; u_h) := \|x_g - x\|_W^2 \tag{3.8}$$

where $x_g = x_c + \delta t u_h$ is a goal system state parameterized by the current operator input signal $u_h = \xi(h_c)$ such that ξ is given by the mapping Equation 3.3 with $M = \nu^+ I_n$ and $b = 0_n$, $0 < \nu^+ \in \mathbb{R}$ is the maximum velocity for a single joint, and δt is the control loop time step.

REDUCED JOINT MODE (RJ) An alternative could be where the control of a number of joints is relieved from the human and assign to autonomous regulation. Let $x = [x_h^T, x_a^T]^T \in \mathbb{X}^n$ where $x_h \in \mathbb{X}^{n_h}$ denote the joints under human control, and $x_a \in \mathbb{X}^{n_a}$ denote the autonomous joints such that $n = n_h + n_a$. As an example and inspired by the concrete spraying task, joints x_a could be assigned the task of end-effector alignment with the surface normal with respect to some pre-collected model of the environment $\widetilde{e}$. The *Reduced Joint* (RJ) mode is modeled using

$$\begin{aligned} \text{cost}(x; u_h) &:= \text{cost}_h(x_h; u_h) + \text{cost}_a(x_a; \widetilde{e}) && \text{(3.9a)} \\ &= \|x_{g,h} - x_h\|_{W_h}^2 + \|\widehat{\phi}_a(\widetilde{e}) - \phi_a(x_a)\|_{W_a}^2 && \text{(3.9b)} \end{aligned}$$

where $x_{g,h} = x_{c,h} + \delta t u_h$ is the goal state for the human regulated joints, $u_h = \xi(h_c)$ is human the control input such that ξ is the mapping defined by Equation 3.3 with $M = \nu^+ I_{n_h}$ and $b = 0_{n_h}$, $\widehat{\phi}_a(\widetilde{e})$ is a task space goal for the autonomous joints such that $x_c = [x_{x,h}^T, x_{c,a}^T]^T$ is the current joint configuration.

FULL TASK MODE (FT) There is evidence to suggest that humans model manipulation tasks in the task space (Mistry and Schaal, 2015). An alternative class of modes can be defined when the human instead submits commands to the manipulator task

space rather than the joint space (as in the previous two modes described above). For spraying, the manipulator task space comprises three translational and two rotational dimensions of the end-effector. The *Full Task* (FT) mode is thus expressed by

$$\text{cost}(x; u_h) := \|\widehat{\phi}_g - \phi(x)\|^2 \tag{3.10}$$

where $\phi : \mathbb{X} \to \mathscr{T}$ is the *Forward Kinematics* (FK) function that maps the configuration space to an n_ϕ-dimensional task space $\mathscr{T} \subseteq \mathbb{R}^{n_\phi}$, $\widehat{\phi}_g = \widehat{\phi}_c + \delta t u_h$ is a user-defined task space goal, and $\widehat{\phi}_c$ is the current task space goal that is tracked over time (i.e. in the next iteration $\widehat{\phi}_c$ will be assigned $\widehat{\phi}_g$ from the current iteration). The task space $\mathscr{T}$ can be a combination of spaces, for example the positional and rotational components of the end-effector pose. As such, multiple mappings may be required for the sub-dimensions of the operator control input h_c to the control input u_h. For example, if $\mathscr{T}$ is comprised of the end-effector position and rotation then the positional components will use a isometric mapping Equation 3.4 and the rotational components will use a linear mapping Equation 3.3.

REDUCED TASK MODE (RT) In this mode, a number of task space dimensions are relieved from the human and allow these to be moderated by the autonomy. In order for the autonomy to perform this moderation, however, again, a pre-collected model of the environment $\widetilde{e}$ is assumed to be available. For example, a spraying task requires five task space dimensions, however, to a human this can be considered a two-dimensional task, i.e., the position on the wall to spray. In this example, the human controls a target position defined on a two-dimensional surface (i.e. the spraying surface is modeled by $\widetilde{e}$) and pass control of the three remaining dimensions to the robot. The *Reduced Task* (RT) mode is modeled by

$$\text{cost}(x; u_h) := \left\| \begin{bmatrix} \widehat{\phi}_h(u_h) \\ \widehat{\phi}_a(\widetilde{e}) \end{bmatrix} - \phi(x) \right\|^2 \tag{3.11}$$

where $\widehat{\phi}_h(u_h) = \widehat{\phi}_{h,c} + \delta t u_h$ such that $u_h = \xi(h_c)$, $\widehat{\phi}_a(e)$ is some k-dimensional task space goal computed using the m-dimensional input from the human such that $m < k$. As in the spraying example, the computed value of $\widehat{\phi}_a(\widetilde{e})$ can be used to encode ideals such as perpendicularity to the spraying plane and a given stand-off distance from the surface.

Operators in industry must maintain a number of motion constraints to achieve high task performance. To determine the performance in target acquisition tasks, e.g., spraying, several measures of quality are derived: completion time T, angle to the plane L_a, standoff distance maintenance L_δ. These measures were inspired by the concrete spraying use-case. Over-spraying is undesirable since material can be wasted and can effect the tunnels structural integrity. The rate of concrete deposition is constant during operation, thus completing a section in a timely fashion is ideal. This motivates the use of completion time T as a metric of performance. Ensuring the nozzle is kept perpendicular to the spraying plane and a certain standoff distance is maintained is also a requirement of operators (Ballou, 2003).

As a pre-cursor, let us define the FK transform between a base frame B and end-effector frame E by

$$T_B^E = \begin{bmatrix} \widehat{e}_x & \widehat{e}_y & \widehat{e}_z & p_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

where $\widehat{e}_x, \widehat{e}_y, \widehat{e}_z \in \mathbb{R}^3$ are orthogonal unit-direction vectors, and $p_e \in \mathbb{R}^3$ is the position vector for the end-effector. Typically $\widehat{e}_z$ is defined as the "pointing" vector from the end-effector.

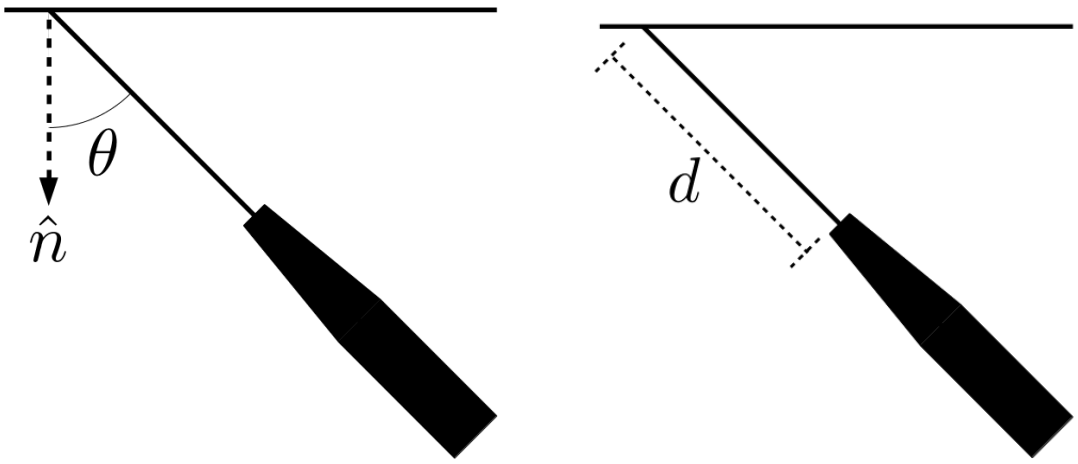
Define the *angular length* L_a as the total change in θ_t for a complete move and expressed by

$$L_a := \frac{1}{T} \int_0^T \theta_t \, dt \tag{3.13}$$

where θ_t as in Figure 3.2a. The angle θ is computed by applying the dot-product rule

$$\theta = \arccos(\widehat{n} \cdot \widehat{e}_z) \tag{3.14}$$

where $\widehat{n}$ is the normal to the surface, and $\widehat{e}$ is the unit-direction vector for the end-effector central axis (easily accessible using the FK method). The quantity L_a in Equation 3.13 describes how well a user is able to maintain perpendicularity to the wall during transitioning between one target and the next.



(a) (b)

Figure 3.2: (a) Standoff distance from the wall. (b) Angles θ and ϕ to the wall.

Define the *delta length* denoted L_δ as the least absolute deviation in the stand-off distance with respect to a given ideal stand-off distance δ^* over the duration of a complete move and expressed by

$$L_\delta := \frac{1}{T}\int_0^T |\delta_t - \delta^*|\, dt \tag{3.15}$$

where δ_t is the stand-off distance as in Figure 3.5b. The stand-off distance δ is computed using

$$\delta = \|p_e - p_w\| \tag{3.16}$$

where $p_w = p_e + \alpha \hat{e}_z$ is the position on the surface that the end-effector is pointing at such that $\alpha = \frac{\hat{n}\cdot(p_0 - p_e)}{\hat{n}\cdot\hat{e}_z}$ and p_0 is any position on the wall.

3.4 Effect of input dimensionality for a mock-up spraying task

In any human motor control task, increasing the number of dimensions increases the redundancy in possible solutions. The human nervous system expertly manages redundancy, however the particular method the body employs is still unclear (Mistry and Schaal, 2015). On the other hand, trading off redundancy by reducing the number of controllable dimensions simplifies a task but in-turn removes control authority. In teleoperation, one must decide the appropriate level of control authority that benefits a range of operators.

There are mixed results in the literature with regards to the appropriate level of control authority; some support more autonomy (You and Hauser, 2011; A. Dragan

and Srinivasa, 2013), others less autonomy (D.-J. Kim et al., 2012), and some report mixed results (Javdani et al., 2018). In the concrete spraying application, devices are most often controlled at a per-actuator level using a FJ mode of teleoperation – see the common MEYCO Oruga concrete spraying Unit in Figure 2.2. Girmscheid and Moser, 2001 report the development of manual (FJ mode of teleoperation), shared control (RT mode of teleoperation), and a fully autonomous mode[2] for an 8-DoF MEYCO RoboJet concrete spraying manipulator. To the best of the author's knowledge, user performance and experience was not addressed as part of this work or any later work.

For the purpose of the thesis, an initial, preliminary, investigation was conducted using a spraying mock-up developed to investigate whether operators perform better or worse given higher or lower levels of control authority.

3.4.1 *Hypothesis*

A study was conducted to ascertain, quantitatively, the effect of adjusting the control authority on task performance. Despite conflicting results in the literature, as mentioned above, for the case reported by D.-J. Kim et al., 2012, patients reported that whilst they felt autonomous modes of control could be beneficial, they favored being in charge of the robot during the manual modes of control. This particular study centers on assitive robotics for patients with spinal cord injuries and as such the users reported favor is with regards to quality of life. In contrast, You and Hauser, 2011 does not necessarily apply to rehabilitation and assitive robotics for patients, and report that users preferred more autonomy. Since industrial tasks are not conducted by an operator in the same vein as a patient, it seems more likely humans will prefer more assistance in target acquisition tasks. Thus, the hypothesis in this section reflects this reasoning and is stated as follows.

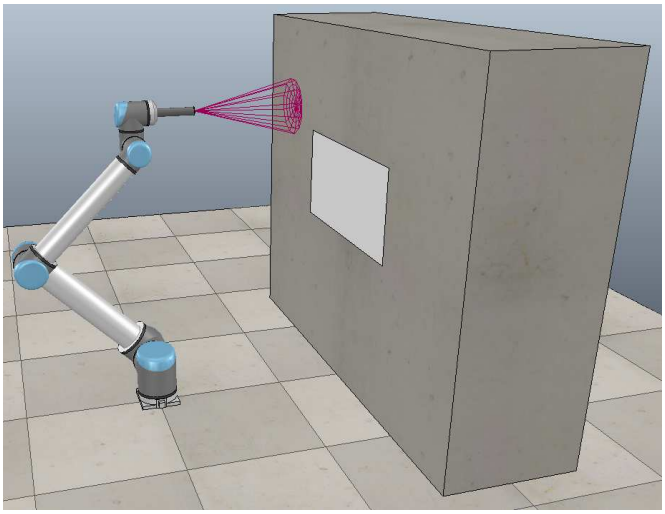
Hypothesis 3.1. Reducing the number of dimensions the operator controls, i.e. n_h, will lead to an improved performance (i.e., small values for T, L_α, and L_δ).

3.4.2 *System description*

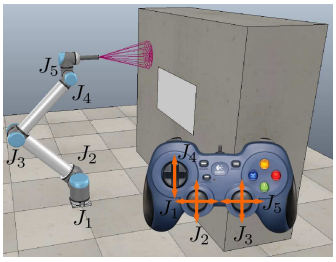
A 6-DoF UR10 manipulator arm was simulated in the CoppeliaSim[3] (Rohmer et al., 2013) simulation environment with a painting nozzle attached to the end-effector. A grey block represents the spraying surface, and the a rectangular patch highlights a

2 Recall, in certain cases when the excavated surface is very smooth, i.e., when a *Tunnel Boring Machine* (TBM) is used, spraying can be autonomous by following a pre-defined task space trajectory. The author's interest is in cases where the earth is removed by hand or controlled explosion. In these cases, automated methods cannot be employed.

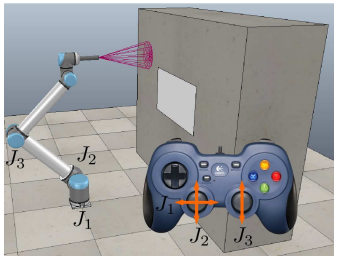
3 CoppeliaSim is formally known as V-REP.



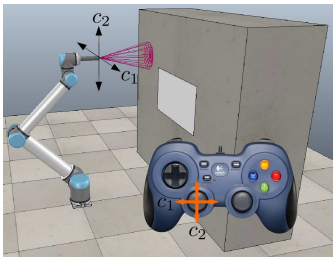
(a)



(b)



(c)



(d)

Figure 3.3: Experimental setup for the mock-up simulated spraying experiment showing (a) the operator's perspective, and the controller mappings for the (b) FJ mode, (c) RJ mode, and (d) RT mode.

section that is required to be sprayed, see Figure 3.3a. Participants interfaced with the simulation via a Logitech F710 gamepad. Streaming data was collected from the gamepad using the PyGame library (Shinners, 2011). In this experiment, the modes FJ, RJ, and RT were considered. Each mode of teleoperation was implemented in the *Extensible Optimization Toolkit* (EXOTica) framework (Ivan, Y. Yang, et al., 2019) using the Python programming language. Figures 3.3b, 3.3c, and 3.3d illustrate these modes in the simulation environment with the corresponding alignment between interface and control axes. Target joints states were transmitted to CoppeliaSim using a remote API for Python.

Data retrieval scripts were implemented in Lua within CoplliaSim that recorded timestamps, joint states, and end-effector poses. Data from the simulator was retrieved at 50Hz. In addition, CoplliaSim provides spraying simulation out-of-the-box. The sprayed points generated using this simulation were also tracked.

Note that in this section the UR10 is treated as a 5-DoF manipulator rather than a 6-DoF manipulator. The reason is because the roll of the end-effector, and as such the joint angle of the sixth joint, does not affect the spraying quality.

3.4.3 *Experimental design*

In this preliminary study, three modes of teleoperation were compared; the FJ, the RJ, and the RT. A within-subjects study was conducted with five participants (4 male, 1 female). All participants were familiar with robotics but none had worked specifically in the field of teleoperation.

The role of the participant is to operate the UR10 via a game-pad controller and spray paint the light gray plaque on the dark gray wall under each mode of teleoperation. The order in which modes were presented was randomized. At the start of a trial the robot was moved into an initial state, common for all participants and every mode of teleoperation. Participants were instructed to paint the plaque whilst maintaining the distance between the nozzle and wall, and ensuring the nozzle kept perpendicular to the wall as best they could for the duration of the spraying.

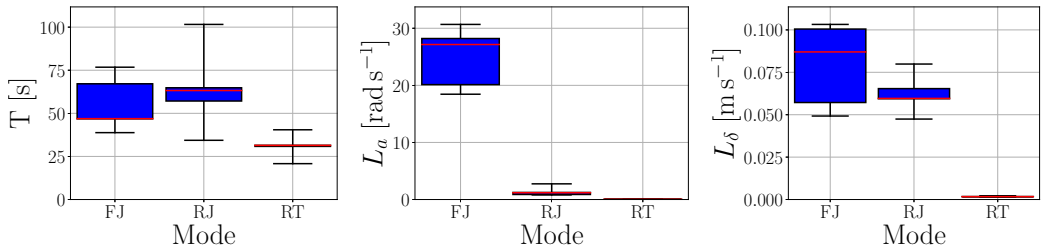
In order to estimate the sprayed coverage, the 2-dimensional sprayed points from the simulator in a local coordinate frame of the plaque were collected. The plane of the plaque was discretized in a pixel-like fashion. If a point landed within a pixel then it was considered "covered". Once every pixel was "covered", the experiment was terminated, and the recorded data was saved to disk.

An ideal stand-off distance of $\delta^* = 0.4$ was chosen, justified as follows. In industry, the MEYCO Oruga, shown in Figure 2.2, has an approximate maximum reach of $r_{Oruga} = 6\text{m}$ and the recommended stand-off distance is $\delta^*_{Oruga} = 2\text{m}$. The UR10 arm has a reach radius of $r_{UR10} = 1.3\text{m}$. Thus, δ^*_{UR10}, denoted δ^* in the experiments, is found by balancing ratios and rounding to the nearest decimal point. The robot initial position is always such that the nozzle is at the required standoff distance and perpendicular to the plane.

3.4.4 *Results*

Three different control strategies were compared for a spraying task using a UR10 manipulator with a paint spraying nozzle attached to the end-effector. Three modes of teleoperation were implemented in this experiment; the FJ, RJ, and RT modes defined in Section 3.2.6. Joint angles and timestamps were collected for every trial and using this information the completion time T, angular length L_a, and delta length L_δ are computed that measure a participants performance. Results are illustrated Figure 3.4.

Completion time results are shown in Figure 3.4a. Notice that operators performed best using the RT mode seen by having the smallest average completion time. Comparing RT to the other two modes, the results are in agreement with Hypothesis 3.1. However, intriguingly the average time for completion using the RJ mode is highest –



(a) (b) (c)

Figure 3.4: Results from experiments on input dimensionality. (a) completion time T, (b) angular length L_α, (c) delta length L_δ.

not in agreement with Hypothesis 3.1. The variances for each mode reduce with the dimensionality of the modes; the variance for the FJ is highest, whilst the variance for the RT mode is very small in comparison. The average and standard deviations for the FJ mode was 55.3 ± 15.9s, the RJ mode was 64.2 ± 24.2s, and the RT mode is 30 ± 6.96s.

The angular length results are shown in Figure 3.4b. The average angular lengths reduce with mode dimensionality – in agreement with Hypothesis 3.1. The variances for each mode is also reduced with dimensionality. The average and standard deviations for the FJ mode was $24.9 \pm 5.33 \mathrm{rad\,s^{-1}}$, the RJ mode was $1.37 \pm 0.790 \mathrm{rad\,s^{-1}}$, and the RT mode is $0.0373 \pm 0.00732 \mathrm{rad\,s^{-1}}$.

The delta length results are shown in Figure 3.4c. The results, similar to the angular length, i.e. they agree with Hypothesis 3.1. The average and standard deviations for the FJ mode was $0.0794 \pm 0.0249 \mathrm{m\,s^{-1}}$, the RJ mode was $0.0623 \pm 0.0118 \mathrm{m\,s^{-1}}$, and the RT mode is $0.00175 \pm 0.000228 \mathrm{m\,s^{-1}}$.

Recalling Hypothesis 3.1, and given the above results, a trend has been observed; reducing the dimensionality of the mode of teleoperation leads to higher performance. During the experiments, participants noted that the RT mode was "much easier than the other two modes". This could be a possible indication that the participant cognitive load was reduced, however this is not in any way conclusive and requires further study.

3.5 Alternate Modes of Teleoperation for Target Acquisition Tasks

Concrete spraying operators are often positioned in close proximity to the spraying unit. The previous experiment (Section 3.4) was performed in simulation. This could have introduced bias in the way the participants interacted with and perceived the system.

In this section, a more extensive investigation is described for all the modes of teleoperation described in Section 3.2.6 that aims to deduce for which modes lead to higher performance for novice operators. This investigation was performed using on a real-world target acquisition mock-up. The experiment leverages Fitts' law for quantifying task difficulty.

3.5.1 *Hypotheses*

In this section, the hypotheses are stated. As mentioned in Section 3.1, humans typically model tasks in the task space as opposed to the manipulator joint configuration space and thus, intuitively, it is expected that modes of teleoperation that allow control commands to be submitted in task spaces will achieve higher performance. Therefore, the hypotheses are based upon this intuition.

Hypothesis 3.2. Modes of teleoperation with lower dimensions will see higher performance.

Hypothesis 3.3. Participants will rate higher those modes of teleoperation with fewer dimensions.

Hypothesis 3.4. Participants will rate higher those modes of teleoperation with fewer dimensions.

Hypothesis 3.5. Participants will rate higher those modes of teleoperation that submit commands in the manipulator task space.

3.5.2 *System description*

Each mode of teleoperation, described in Section 3.2.6, is implemented using an open-loop position control framework on a 7-DoF KUKA LWR robot arm. The operator interfaced with the system using an F710 Logitech gamepad. Figure 3.6 illustrates the mappings between the interface and robot for each mode. Each experiment was initiated by the user clicking a button on the gamepad. As a safety feature, in order to stream control commands to the system the participant held down the top-left shoulder button akin to, but less straining, than a dead-man switch on a teleoperated machine. Inter-process communication is handled by *Robot Operating System* (ROS) (Quigley et al., 2009) and modes requiring numerical optimization were computed using EXOTica (Ivan, Y. Yang, et al., 2019). Goal joint positions were streamed at $100\,\mathrm{Hz}$. Targets and the focus point were displayed on a $1.65\,\mathrm{m}$ cross-diagonal display placed specifically such that the transform between the robot base and the television

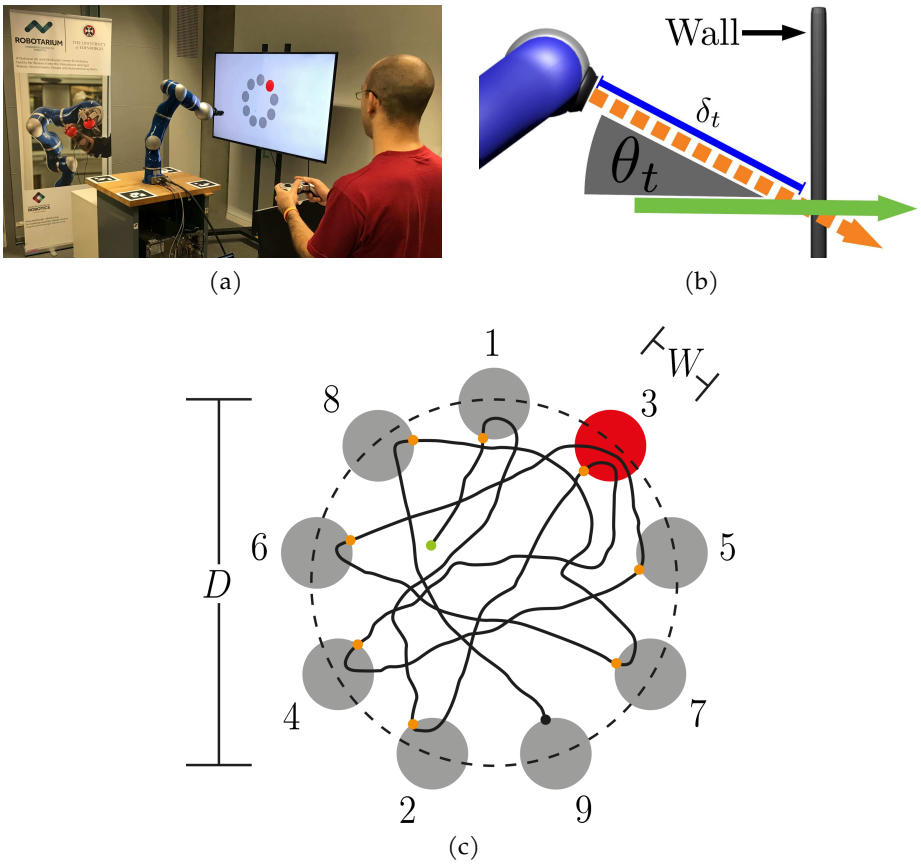


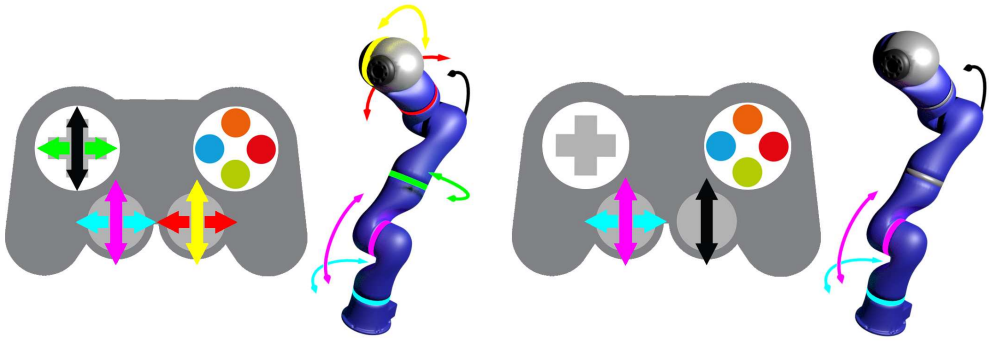
Figure 3.5: Experimental setup in (a) where an operator controls the KUKA LWR using a gamepad. (b) shows the angle to the plane θ_t and stand-off distance δ_t used to compute performance metrics. Whilst condition order of appearance is randomized, target order was always kept the same as in (c). The black line in (c) indicates an example path of the focus point during manipulation; the path was not shown to participants, only the focus point indicated by the green dot.

screen was known. Targets were scaled to the real world and the position of the focus point was found using the robot FK.

An ideal stand-off distance of $\delta^* = 0.55\,\mathrm{m}$ was chosen, justified, similar to that in Section 3.4.3, as follows. In industry, the MEYCO Oruga (Figure 2.2) has an approximate maximum reach of $r_O = 6\,\mathrm{m}$ and the ideal stand-off distance is $\delta^*_O = 2\,\mathrm{m}$. The KUKA LWR arm has an approximate maximum reach of $r_K = 1.5\,\mathrm{m}$. Thus δ^*_K is found by balancing the ratios and adding a ten percent safety distance.

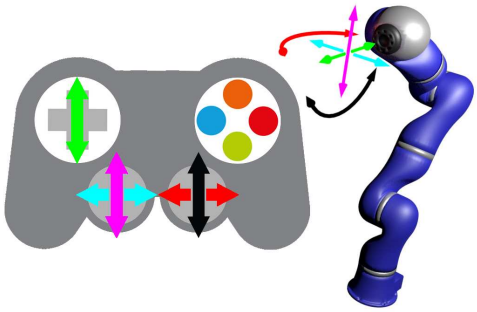
3.5.3 *Experimental design*

Results were obtained in this study by evaluating the performance of 21 participants (16 male, 5 female). The age distribution of the participants were 7 (21-25), 11 (26-30), 2 (31-35), and 1 (36+).

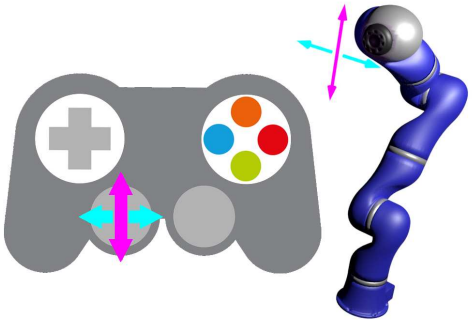


(a) Full joint space control.

(b) Reduced joint space control.



(c) Full task space control.



(d) Reduced task space control.

Figure 3.6: Gamepad mappings for the modes of teleoperation implemented in experiments. Markings indicate where the human interacts with on the gamepad and where that interaction is perceived to be on the robot model.

During preliminary investigations a difference in performance was noticed for participants who regularly played video games. During the final experiments, participants were asked to provide a rating on how often they play video games[4]. They were asked "How often do you play video games (e.g., Xbox, PS4, PC)? [1 Never], [2 Bi-monthly], [3 Monthly], [4 Bi-weekly], [5 Weekly], [6 Regularly (but not daily)], [7 Daily]". A participant was considered a *gamer* if they gave a rating greater than or equal to 4, and a *non-gamer* otherwise. The responses are shown in Table 3.1 as follows.

Table 3.1: Habitual traits for participants.

Gaming regularity	1	2	3	4	5	6	7
Count	7	3	2	2	3	3	1

Participants were tasked with teleoperating a KUKA LWR robot arm to reach-and-point to a number of circular targets, indicated in red, arranged in a circle as in Figure 3.5c. Targets were displayed on a screen in a known position and orientation with

4 Note, those who participated in the preliminary investigation were barred from the final experiments to avoid skill transfer.

respect to the robot base, see Figure 3.5a. For each condition, targets were presented in the order as in Figure 3.5c. Conditions were randomized for every participant to minimize skill transfer. The participants were instructed to use as many controllable dimensions available to them in order to manipulate the robot in such a way that completes the task as fast as possible while maintaining the angular and delta length constraints; i.e., simultaneously minimize T, L_α, and L_δ to the best of their ability.

Targets and conditions were presented to the participant in succession. As one target was deemed acquired the next immediately followed and as one condition was completed (i.e., one full cycle of targets in the order shown in Figure 3.5c) the next immediately followed. Target acquisition is when the focus point (i.e., the point on the display screen the robot is pointing) comes into contact with the target. At the start of every mode the robot was reset to the same starting configuration. Following each mode participants were asked to fill out a questionnaire (Section 3.5.5).

Ultimately, during each run of the experiments the values for W and D in Figure 3.5c were modified to yield a different index of difficulty. This was performed for each mode of teleoperation. Controlled parameters included the observation pose and the ideal standoff distance.

3.5.4 *Quantifying task difficulty using Fitts' law*

Fitts' law is widely considered to be a robust measure of performance for target acquisition (pointing) tasks making the analogy between movement time and transmission of information (Fitts, 1954). A common usage of Fitts' law from the human-computer interaction literature is to compare the usability of computer input devices (e.g., a mouse, trackball, and a stylus with a tablet) (MacKenzie et al., 1991). The law makes two underlying assumptions: (1) task difficulty is linearly correlated with performance, and (2) a complete move is performed through a number of iterations of feedback-guided corrective sub-movements, i.e., the *deterministic iterative-corrections model* (Crossman and Goodeve, 1983). In this work, Fitts' law is used as a method to specify task difficulty.

Fitts' established the information capacity of the human motor system by deriving a model for the *index of performance* I_p (in $\mathrm{bit\,s^{-1}}$) expressed by

$$I_p := I_d / T \tag{3.17}$$

where I_d is the *index of difficulty* (in bit) and T (in sec) is the average movement time. The index of performance is a metric that quantifies task performance; higher values for I_p indicate better performance. The index of difficulty is a metric that defines

the task difficulty; higher values for I_d imply the task is more difficult. Under the deterministic iterative-corrections model and by analogy with Shannon's Theorem 17 (Shannon, 1948) a formula for I_d is derived, see (Crossman and Goodeve, 1983) for details, given by

$$I_d := \log_2(2D/W) \tag{3.18}$$

where D is the distance to a target and W is the width of the target, see Figure 3.5c. Define a *condition* by the tuple (W, D). A difficult task (i.e. high I_d) can be seen when, for example, W is small and D is large, and an easy task (i.e. small I_d) when W is large and D is small.

For each mode of teleoperation, participants completed 17 conditions: 16 have been generated by scaling the condition values used in Fitts' original work (Fitts, 1954) while one has been hand-tuned to add spread to the I_d values. The conditions used in this study are shown in Table 3.2 representing the range $I_d \in [1.3219, 5.9069]$.

As described above, Fitts' law is typically specified for tasks where completion time is a measure such that lower duration's are better. The law was been generalized to two additional metrics for performance that were in Section 3.3, i.e. the angular length L_α, and delta length L_δ.

In addition, we utilize *throughput* that is another common indicator of overall performance (Fitts, 1954). For the time metric, the throughput Th is estimated by the reciprocal of a scalar b where a, b are regression parameters such that $a + bI_d = T$; so $Th = 1/b$. For the angular length and delta length, throughput for each has been estimated in the same fashion. Throughput is computed by computing the line of best fit, and so in turn we can calculate the Pearson correlation coefficient that measures the linear correlation between the two variables - i.e. the measure of performance (e.g. time) and the index of difficulty.

3.5.5 *Questionnaire*

Participants were asked to provide a rating for each mode on their speed perception, accuracy perception, fatigue, and the mental capacity required. The questionnaire, shown in Table 3.3, was devised from the ISO 9241 standard and work by Douglas et al. (Douglas et al., 1999). Questions 1-5 use a 7-point scale and 6-7 are open-ended.

Table 3.2: Condition values used in experiments.

W [m]	D [m]
0.0125	0.0469, 0.0938, 0.1875, 0.3750
0.0500	0.0938, 0.3750, 0.7500
0.0250	0.0469, 0.0938, 0.1875, 0.7500
0.1000	0.1875, 0.3750, 0.7500
0.1500	0.1875, 0.3750, 0.7500

Table 3.3: Questionnaire used in investigation.

	Questions:	Rating: 1	7
1)	The mental effort required for operation was	high	low
2)	Accurate pointing was	difficult	easy
3)	Operation speed was	fast	slow
4)	Finger fatigue	high	none
5)	Overall, the mode of teleoperation was	difficult	easy
6)	Did you have any trouble with this mode of teleoperation?		
7)	Do you have any comments in general about using this mode of teleoperation?		

3.5.6 *Results*

3.5.6.1 *Low-dimensional task space elicits high performance*

Data for 21 participants was collected for 17 conditions with 9 targets for each condition. In order to compute the performance metrics, target acquisition timestamps and joint states were collected.

The data was filtered based on one criteria. Data for the first target of every condition was ignored since when transitioning between one condition and the next the data is not representative of skill in either condition. An extensive database of results was collected constituting, in total, 8882 data points. Using each metric (time, angular length, and delta length) and the index of difficulty allows us to compute an index of performance for every metric and for each point in the data set. The distributions of $\log(I_p)$ is shown in Figure 3.7. Using this data, a one-sided paired-sampled t-test to test the hypotheses was used.

Precisely the same conclusions for time and delta length metrics have been observed. Hypothesis 3.2 is accepted as expected for reducing control space-dimensionality, except the FJ-RJ comparison. For the angular length Hypothesis 3.2 is accepted; this is

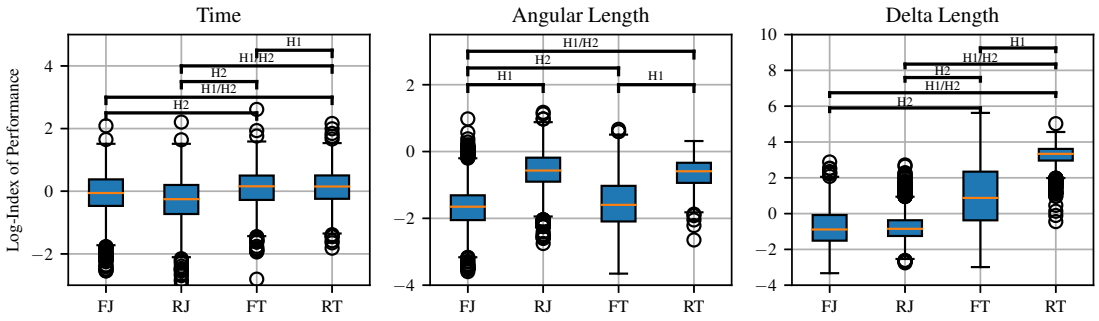


Figure 3.7: Log-Index of performance distribution results. By means of a one-sided paired-sample t-test, pairs accepted under Hypothesis 3.2 and Hypothesis 3.3 ($\alpha = 0.01$) are indicated above.

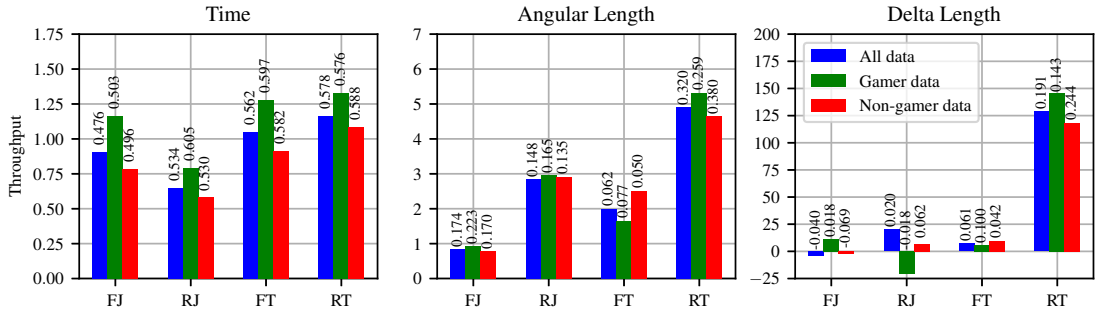


Figure 3.8: Throughput results that estimate the index of performance. The results for the full data set and the two subsets: gamer and non-gamer. Also, the Pearson correlation coefficient is shown above each bar.

unsurprising since RJ supports end-effector alignment. For cross-modes, e.g., FJ-RT, it is observed that Hypothesis 3.2 is accepted.

When comparing task space modes and joint space modes it is seen that Hypothesis 3.3 has been accepted. An interesting point to note for the time metric and delta length metric is that Hypothesis 3.3 is accepted when comparing the RJ and FT modes. This suggests that low-dimensional joint spaces in some cases elicit worse performance than higher-dimensional task spaces. It is postulated that the RJ mode may fight intuition resulting in the machine performing motions that do not match the operator's innate model of the system dynamics.

3.5.6.2 *Habitual traits effect performance*

During preliminary investigations, it was noted that participants who were known to play video games regularly seemed to give higher performance than those that did not. In the final experimental design it was decided to include this comparison as part of a multivariate analysis. As described in Section 3.5.3, participants were asked to indicate their gaming regularity indicating a distinction between gamers and non-gamers. Using this allows us to split the data set into two subsets.

By the assumption that difficulty and performance are linearly correlated it is possible to compare across the various modes of teleoperation using linear regression. To compare the general performance for each trait, an estimate of throughput is used. Throughput estimates for the full data set and the gamer/non-gamer subsets are shown in Figure 3.8.

Regarding time performance, in general, excluding the result for the RJ mode, a similar pattern is seen when compared to the results in Figure 3.7; the RT mode has highest throughput in general and task space modes have higher throughput than joint space. For each mode of teleoperation a higher throughput for gamers was observed as opposed to the non-gamers. It was noted that gamers generally seemed more familiar using two analog joysticks at once as opposed to the non-gamers. The author posits, due to this ability, gamers were able to achieve faster times.

Regarding angular length performance, a similar general trend in performance as discussed in the previous section is seen. Gamers have higher throughput than non-gamers for each mode except the FT mode. It should be noted that the correlation coefficients for these are generally low which renders these results as potentially spurious. Drawing conclusions from these results may be unreliable. The throughput results for the delta length suffer from similar issues and so reliable conclusions unfortunately cannot be made for these either.

3.5.6.3 *Participants approve low-dimensional task spaces*

The results of the questionnaire are shown in Table 3.3. The mean and standard deviation of the responses on questions 1-6 are shown above a paired-sampled t-test to determine the responses' statistical significance.

The results for question 1 show that participants felt the FJ mode required the highest amount of mental effort. There seems to be no statistical difference between the RJ and FT modes. Participants indicated the RT required the least amount of mental effort.

Overall, participants indicated the task space modes were the easiest to point accurately. There is no statistical difference between FJ and RJ modes both having ratings indicating that participants felt these were the most difficult to accurately point. The results of the t-test indicate Hypothesis 3.4/Hypothesis 3.5 be rejected when comparing FJ, RJ, and FT modes. However, whilst Hypothesis 3.4/Hypothesis 3.5 are accepted when comparing between RT and the joint space modes, Hypothesis 3.4 is rejected when comparing the task space modes. The p-value for the RJ-FT comparison is reasonably high and the FJ-FT comparison is higher than the p-value for the FT-RT comparison. Comparing the results of this question to the responses given for questions 6 and 7 (shown below), the author suggests that, despite the computed

Table 3.4: Results of the questionnaire and paired-sampled t-test ($\alpha = 0.05$). Bold indicates the one-sided Hypothesis 3.4/Hypothesis 3.5 is accepted. *Note, since Q3 does not evaluate the users preference (highlighted in bold) indicates the result of a two-sided significance test.

Question	FJ	RJ	FT	RT
1	1.81 ± 1.33	3.29 ± 1.59	3.57 ± 1.57	5.05 ± 1.36
2	2.57 ± 1.66	2.90 ± 1.73	3.14 ± 1.39	2.95 ± 1.69
3*	4.62 ± 0.92	4.71 ± 1.06	3.81 ± 1.08	4.29 ± 1.15
4	3.29 ± 2.00	3.62 ± 1.88	4.67 ± 1.43	4.90 ± 1.37
5	3.10 ± 1.55	4.24 ± 1.26	4.86 ± 1.28	5.95 ± 1.07
Question		RJ	FT	RT
1	FJ	**0.0048 (3)**	**0.0009 (3)**	**5.2e-7 (3/4)**
	RJ	-	0.5867	**0.0003 (3/4)**
	FT	-	-	**0.0011 (3)**
2	FJ	0.5538	0.1435	**0.0281 (3/4)**
	RJ	-	0.6339	**0.0104 (3/4)**
	FT	-	-	0.0727
3*	FJ	0.7245	**0.0202**	0.2596
	RJ	-	**0.0068**	0.1428
	FT	-	-	0.1158
4	FJ	0.4907	**0.0120 (3)**	**0.0044 (3/4)**
	RJ	-	**0.0100 (4)**	**0.0005 (3/4)**
	FT	-	-	0.3086
5	FJ	**0.0244 (3)**	**0.0009 (3)**	**3.3e-6 (3/4)**
	RJ	-	0.1198	**1.0e-5 (3/4)**
	FT	-	-	**0.0047 (3)**

p-value, accepting Hypothesis 3.4 for the FT-RT comparison has potential grounding as a conclusion.

During the experiments, maximum joint velocities were reduced to a conservative range for safety. Question 3 attempted to ascertain whether the participants felt the robot motion was too slow or indeed too fast. Mean values indicate participants felt the operation speed was neither too fast nor too slow. There are not significant differences between the results apart from the FT mode compared to the joint space modes.

The results for question 4 indicate the participants experienced the least finger fatigue for task space control modes with no statistical difference between the two. The joint space modes caused the most fatigue.

For question 5, participants rated the RT mode as the easiest to use and the FJ the most difficult. There is no significant difference between the FT and RJ modes.

Common responses to questions 6 and 7 are sumarized. Selected are common statements made by participants and shown in Table 3.5.

Upon observation, the responses generally correlate with the answers to questions (1)-(5). The mappings for both joint modes were identified as inverted by many of the participants. The direction the joint moved under these modes was matched with the joystick direction. For example, for the FJ mode, pushing forward on the left joystick moved joint 2 in the direction such that the end-effector and focus point moved downwards. Participants felt the robot end-effector and focus point should instead move upwards. An issue for some participants for the task space modes is that the robot would make unexpected motions in certain configurations. The source of this issue is that at times a target configuration was computed requiring a joint velocity surpassing the maximum; an unfortunate consequence of unconstrained optimization. Whilst these issues were, at times, observed they did not render the task impossible to complete.

3.6 DISCUSSION

In this chapter, two studies were developed and implemented to determine for which modes of teleoperation lead to high task performance for unskilled human operators on tasks inspired by concrete spraying in industry. The first experiment compared modes for a spraying mock-up task in a simulated environment. In the second study, a Fitts' law paradigm was used to quantify difficulty and performance of each mode. Two metrics were generalized for Fitts' law as additional performance metrics and an extensive data set was collected from the experiments. The results and analysis performed support several conclusions regarding control, sub-task allocation between human and autonomy, and how habitual traits can effect performance.

The preliminary study described in Section 3.4 showed early evidence that the mode of teleoperation is a significant factor of task performance. One limitation of the experiment was that only one viewpoint was presented to the operator. Lathan and Tracey, 2002 observed a significant correlation between an operator's spatial perception ability[5] and teleoperation performance – their experiments were for a navigation task. In the construction sector there is a drive to extract the operator, and perform the spraying task remotely. Future work should explore performance as a function of spatial perception ability in target acquisition tasks.

In the second study, detailed in Section 3.5, a reduced task space control mode has been shown to outperform all three other modes of teleoperation with regards

5 The spatial perception ability is the ability to navigate or manipulate objects in a three dimensional environment.

Table 3.5: Responses to questions 6 and 7.

Full joint mode	
6)	*"Wrist joints felt slower than base joint."*
	"Joint mappings felt inverted and was easy to get into strange configurations."
7)	*"Hard to maintain constraints."*
	"Only used two joints at a time."
Reduced joint mode	
6)	*"Attention was directed to focus point, not robot, making operation very difficult."*
	"Often requiring to re-adjust."
7)	*"Slightly better than full joint mode."*
	"Easy for small D values."
Full task mode	
6)	*"Unexpected motions at times."*
	"Maintaining delta length was effectively impossible. Perhaps with more practice this mode would be more efficient."
7)	*"Felt it was possible to go faster."*
	"Favored over both joint space control modes; intuitive, mostly easy to use."
Reduced task mode	
6)	*"Some unexpected motions."*
	"Sometimes felt too slow."
7)	*"Very intuitive and easy to use."*

to three performance metrics; time, angular length, and delta length. The results of the questionnaire support this conclusion as the RT mode was generally favored the highest. The RT mode is the closest model that directly regulates the performance parameters. Given that the hypotheses were accepted this opens new avenues for shared control design. Based on these results, the author posits that there exists a positive correlation between cognitive load and number of human controllable dimensions for task space control modes. Both joint space modes were not favored by participants in this study with some noting that the reduced joint space control felt like having the *"worst of both worlds"* with regards to joint space and task space control.

Generally, the analysis of the data collected during the experiments in this chapter indicate operators elicit higher performance in task space modes rather than joint space. This is in partial agreement with other studies. S. Wang et al., 2020 studied joint and task space modes for a humanoid teleoperation setup and found little difference in the performance between these modes, however task space modes were found to be more intuitive for human operators.

This work assumed unskilled participants and so the long-term learning effects on performance are not considered here. Future work should study the effect on the par-

ticipants learning curve in order to predict average learning time. The ability of the participants classed as gamers to achieve higher performance than non-gamers may be a consequence of their familiarity with the game pad controller used as the interface in the experiments. Future investigations should also take into account the effect of the interface used by comparing other interfaces such as a joystick, 6-DoF space mouse, and a combination of a computer mouse and keyboard. Metrics quantifying cognitive load, e.g. NASA TLX (Hart and Staveland, 1988), should, in addition, be explored and compared against the number of controllable dimensions in-order to investigate our proposition of a positive correlation between the two quantities.

The experiments were designed using a flat plane to represent a spraying surface. This design choice was necessary in our experimental setup which used a television screen and would be challenging to develop in simulation. Otherwise, a realistic setup would be very challenging (e.g., displaying visuals on a curved surface).

Determining the appropriate level of autonomy in teleoperation with regards to user preference is difficult. The literature is conflicted: some studies suggest less automation is desirable (D.-J. Kim et al., 2012), whereas others indicate the opposite (You and Hauser, 2011). The experimental analysis in this Chapter suggest that for target acquisition tasks more automation is preferred by participants. A. Dragan and Srinivasa, 2013 suggest the conflict can be explained in terms of context (i.e. task, user) and the choice in arbitration (i.e. how the human input and autonomy is combined). The analysis of this work aligns with the former. However, the studies in this chapter did not examine this question in detail, and so it can not be claimed as a conclusion - further investigation is required to address this in the context of target acquisition tasks. Furthermore, these tasks are multi-objective (i.e. fast and accurate). This raises the question whether participants will attempt to optimize all objectives or chose one. Future work should explore this question and investigate if weighted assistance positively affects task performance.

This study indicates the mode specification highly impacts the design and performance of shared control systems. Future studes should use the knowledge acquired here to inform the development of new formulations for shared autonomous and collaborative methods that adapt dynamic motion constraints on-the-fly using multimodal sensory data. Future chapters will leverage the work here for design decisions in their formulations.

3.7 LESSONS LEARNED

The experiments described in this chapter constitute some of the earliest investigations performed toward this thesis. The FT mode was not included in the experi-

ments described in Section 3.4 due to a matter of oversight. In order to investigate the relation between modes of different spaces, i.e. task and joint space, the mode was included in later investigations. In addition, only objective measurements were collected. Optimal modes depend, also, on the subjective opinions/preferences of the participants.

In addition, proper planning and infrastructure design for code and archiving data should also be a key consideration in the early development of experiments. Without makes future analysis very difficult, if not impossible.

4

MODULATING HUMAN INPUT FOR SHARED CONTROL

4.1 INTRODUCTION

There are numerous shared autonomy methods that address some of the above limitations of teleoperation, such as: predict-then-act (A. Dragan and Srinivasa, 2013), next-best viewpoint for an external camera-in-hand robot (Rakita et al., 2018), learning from demonstration approaches (Abi-Farraj, Osa, et al., 2017), and sliding autonomy (Merkt, Y. Yang, et al., 2017). An aim of shared autonomy is to reduce the cognitive load on the human operator by leveraging autonomous capabilities informed by available sensory information and task heuristics addressing, therefore, the limitations of direct teleoperation. Determining an appropriate balance between human input and autonomous assistance that is intuitive for the operator and reliable for the completion of complex tasks is a key challenge for the robotics community.

A common approach when blending human input and autonomous reasoning is to allocate specific sub task-space dimensions as either human or autonomously regulated. For example, Abi-Farraj, Pedemonte, et al., 2016 proposes a shared control architecture for a pick-and-place task, that allocates one translation and three rotational dimensions to the human and the additional two translation dimensions to a fully autonomous system. Sub-task allocation is often favorable since it avoids the difficulty of dealing with conflicting interests, but fails to address limitation (L2).

Additionally, despite literature, e.g., (Simon, 1979) suggesting a "*bounded rationality*" on the cognitive capabilities of the human, most shared autonomy methods either consider a static environment or implicitly assume that the operator keeps track of all the changes in environment (Hauser, 2013; Abi-Farraj, Pedemonte, et al., 2016; Rosenberg, 1993; Marion et al., 2017). Therefore, some of these methods might still lead to heightened fatigue, highlighted in limitation (L6). In addition, the level of trust between human and the autonomous reasoning is also evidently a key factor when addressing limitations (L1) and (L6) (A. Dragan and Srinivasa, 2013; Nikolaidis et al., 2017).

In Chapter 3, various control spaces (e.g., joint space or task space) were investigated, in which unskilled operators submit commands, and their dimensionality affect overall task performance for target acquisition tasks, addressing the limitations (L1) and (L4). A key conclusion was that unskilled human operators achieve higher performance by commanding the robot in a lower dimensional task space. This knowledge is leveraged here in order to design a more effective shared autonomous system.

4.2 HUMAN INPUT MODULATION PROBLEM FORMULATION

Let us model motion generation as a discrete time state-dependent policy function

$$x_{t+1} = \pi(x_t u_t; \widetilde{e}_{0:t}), \tag{4.1}$$

where x_t represents the robotic system state at the current time t, u_t the control input, and $\widetilde{e}_t$ the state of the environment. The nth order Markovian property is assumed for the policy π, i.e., the current state depends on the previous n inputs, and that the initial state x_0 is known.

There are, in the literature, multiple approaches for describing a policy π. From simple analytical expressions for trivial examples to various heuristic algorithms purposely suited for a given application. Recent years have popularized descriptions based on numerical optimization, since they are able to account for multiple motion constraints whilst minimizing a given objective function. Moreover, an optimization-based problem description allows for the use of readily available robust algorithms and libraries (such as SNOPT and IPOPT), instead of ad-hoc implementations.

In this chapter the policy (Equation 4.1) is formulated as the optimization problem of finding the next-best state with respect to an objective function describing task goals that consider the full history of the environment, and motion constraints as

$$x^*_{t+1} = \underset{x_{t+1}}{\arg\min}\ \text{cost}(x_t; u_t) \tag{4.2a}$$

subject to

$$x_{t+1} = f(x_t, u_t) \tag{4.2b}$$

$$x_{t+1} \in \mathbb{X}(\widetilde{e}_{0:t}) \tag{4.2c}$$

where $\text{cost}(\cdot)$ is a real-valued cost function, the space of feasible is

$$\mathbb{X} = \{x \in \mathbb{R}^{n_x} : c_i(x) \leq 0 \; \forall \; i \in \mathcal{I}\} \tag{4.3}$$

described by a set of inequality constraints c_i, and $\mathbb{I}$ is a set of indices. Note, in general this formulation could also contain equality constraints, however in this chapter, only cases where inequalities are required are described.

For complex tasks in dynamic environments it is often very difficult to find robust methods that specify appropriate control u_t to autonomously complete the task. Complexities in tasks often derive from poor situational and contextual awareness of the part of the autonomous system, often translating as a poor representation or handling of $\widetilde{e}_t$.

As an alternative, teleoperation methods express the control input u_t as a function of the human input h_t (often $u_t \propto h_t$ and projected in the joint or task space). These systems typically consider the human as an observer of the environment and assume that $\widetilde{e}_t$ is a parameter of the human's inherent model of the task, rather than a parameter of Equation 4.2. Therefore, those control schemes can be thought of as a one-to-one mapping between the human input and robot motion. Shared autonomy attempts to relax this assumption and redefine the control input as a blend of human and autonomous reasoning.

4.3 AN APPROACH FOR HUMAN INPUT MODULATION

In this section details are described about the proposed shared autonomy optimization-based approach for obtaining the robot policy, using the formulation Equation 4.2. In order to achieve real-time tractability when solving this optimization, special care must be taken in setting up the shared autonomy problem. For instance, providing an analytical Jacobian of the cost function cost to the optimizer, as opposed to some estimation method, considerably speeds up solving times. In the next section, the specification of cost is extended by various objective terms.

However, for complex and dynamic environments, the analytical Jacobians for the collision map are typically unavailable or slow to compute, easily rendering the problem intractable for real-time purposes. In this work, instead of neglecting the dynamic environment from Equation 4.2 altogether, its representation is offloaded from the user's inherent model to the autonomous system by incorporating it in the control input function as

$$u_t(h_t^{(p)}, \widetilde{e}_{0:t}) = g_H(h_t^{(p)}) + g_A(\widetilde{e}_{0:t}), \tag{4.4}$$

where g_H and g_A respectively map the human input h_t and the full task history $\widetilde{e}_{0:t}$ to a control action. Note, in this chapter the position $h_t^{(p)} = \phi_t + h_t$ is defined where h_t is the interface signal, and $\phi_t = \phi(x_t)$ maps the system state space to the task

space of interest, e.g., ϕ typically maps the robot joint positions to the translation and rotational components of the end-effector. Therefore, within this formulation, the environment information is a parameter of the control, rather than incorporated in the optimization.

It is assumed that $\widetilde{e}_t$ appropriately describes the task and the environment, and that at each time-step all $\widetilde{e}_{0:t}$ can be tracked. For example, in a shared autonomous spraying system, $\widetilde{e}_t$ can be the current surface location being sprayed, easily computed by the forward kinematics and a model of the surface.

Blending the human input and task/environment information in the same space by means of a summation as in Equation 4.4 raises questions about conflicting input. That conflict is resolved by treating the human input $h_t^{(p)}$ as a goal that attracts the robot into some particular task position, and the task/environment information $\widetilde{e}_{0:t}$ either as an attractor or a repeller conditioning the robot task motion. Because both human input and task goals/environment vary with time, this can be thought of as blending, essentially building a different task potential field for each time iteration, which determines the corresponding control action.

The control from the human g_H is expressed as

$$g_H(h_t^{(p)}) = \alpha_H \big(h_t^{(p)} - \phi_t\big) \tag{4.5}$$

where $0 < \alpha_H \in \mathbb{R}$ is some scalar parameter representing the strength of the human part. In a spraying task, ϕ could represent the two dimensional position on the spraying surface that the robot end-effector is pointing.

The control map for the autonomy g_A is expressed as a linear control law given by

$$g_A(\widetilde{e}_{0:t}) = \alpha_A \sum_{\tau=0}^{t} \beta_\tau (\widetilde{e}_\tau - \phi_t) \tag{4.6}$$

where $0 < \alpha_A \in \mathbb{R}$ is some scalar parameter representing the strength of the autonomous part, and $\beta_\tau \in \mathbb{R}$ is a scalar parameter tuned with respect to every point $\widetilde{e}_\tau$. It is assumed that the choice of each β_τ can be either constant or updated by some strategy during the task. If $\beta_\tau > 0$ then this implies $\widetilde{e}_\tau$ is a goal, otherwise if $\beta_\tau < 0$ then $\widetilde{e}_\tau$ is an obstacle.

4.4 CASE STUDIES

The proposed method can be applied to various tasks such as grasping, welding, wiping, and spraying. In this section, how to use the proposed optimization-based ap-

proach is illustrated for a shared autonomous grasping task and a shared autonomous spraying task. The goals of each task are modeled as a weighted objective function

$$\text{cost}(x; x_t, u_t, \widetilde{e}_{0:t}) = \sum_i \rho_i \, \text{cost}_i(x; x_t, u_t, \widetilde{e}_{0:t}) \tag{4.7}$$

where $0 \neq \rho_i \in \mathbb{R}$ is a weighting term that reflects the importance of each sub-task, and each cost_i models a specific sub-task. In the following sub-sections the possible objective terms cost_i are specified for the grasping and spraying tasks.

Please note, the index i for each term described in the following sections relates the term with a task map in the software package EXOTica (Ivan, Y. Yang, et al., 2019).

4.4.1 *Shared control for grasping*

Scenarios such as assistive robots in every day tasks (A. Dragan and Srinivasa, 2013) and nuclear waste disposal (Abi-Farraj, Pedemonte, et al., 2016) motivate the use of teleoperated grasping. One typical example is clearing, where the robot has to grasp a number of static or dynamic objects in a particular order and in a changing and cluttered environment, whilst avoiding collisions. Such tasks require contextual and situational awareness, as shown in Figure 4.5.

Under direct control, in order to complete this task, the operator must decide the ordering in which to grasp the objects, track the changes in environment, and maneuver the grasping tool to pick and place the items at a goal position (e.g., in the bin) whilst avoiding collisions. The combination of planning motion for multiple goals, tracking and localization, and ensuring precise collision-free robot motion undoubtedly results in high cognitive loads for the operator.

Let us assume the human provides commands in the three-dimensional end-effector velocity space. Collision-free end-effector motion is represented using

$$\text{cost}_{EffPos} := \|\phi_{pos}(x) - (\phi_{pos}(x_t) + u_t(h_t^{(p)}, \widetilde{e}_{0:t}))\|^2 \tag{4.8}$$

where ϕ_{pos} is the forward kinematics mapping that computes the position of the end-effector, and u_t is the modulated input Equation 4.4. In this case the user is assisted to produce collision-free motions by merging the operator's goal and the environment model.

Moreover, a goal orientation g_t can be incorporated, i.e. a function of the predicted item the operator intends to grasp, by using

$$\text{cost}_{EffOri} := \|\phi_{ori}(x) - g_t(h_t^{(p)})\|^2 \tag{4.9}$$

where ϕ_{ori} is the forward kinematics mapping that computes the orientation of the end-effector. For g_t a simple prediction rule is used in this work that chooses the appropriate orientation according to the closest object to the end-effector. More elaborate prediction rules are described by A. Dragan and Srinivasa, 2013.

Another goal for the shared autonomous grasping task is to have collision free null-space motion, that can be achieved by introducing an additional cost term (e.g., see (Y. Yang, Ivan, and Vijayakumar, 2015)) or constraint equation, e.g., see (Kanehiro et al., 2008).

4.4.2 *Shared control for spraying*

In this example, inspiration is taken from the main application source, concrete spraying. A criteria for high job quality concrete spraying is to ensure the sprayed lining is as homogeneous as possible (Ballou, 2003). This requires the operator to mentally track all the positions along the spraying surface previously visited in order to ensure minimal overlap. The aim of the shared autonomous system to aid the operator performing more accurate spraying.

For spraying tasks, there are five main task objectives of interest: (i) given a position on the wall to spray, orient the end-effector to "look at" this position, (ii) given the normal to the wall at this point, align end-effector axis with this normal, (iii) maintain a given standoff distance, (iv) ensure smooth motion synthesis, (v) help the operator to avoid over-spraying. Note, all points of interest discussed in the rest of this section are shown in Figure 4.1.

The $LookAt$ task relies on the cost term

$$\text{cost}_{LookAt} := \|p_t - a_t\|^2 \tag{4.10}$$

where $p_t := \phi_t + u_t$ is the target point, a function of the current sprayed position on the surface ϕ_t and control u_t, and a_t is the orthogonal projection of p_t onto the end-effector approach axis.

The $EffAxisAlign$ task consists in aligning the end-effector with the spraying surface using the cost term

$$\text{cost}_{EffAxisAlign} := \|\cos(\theta_t) - 1\|^2 \tag{4.11}$$

where θ_t denotes the angle between the surface normal $\widehat{n}$ and the end-effector approach axis.

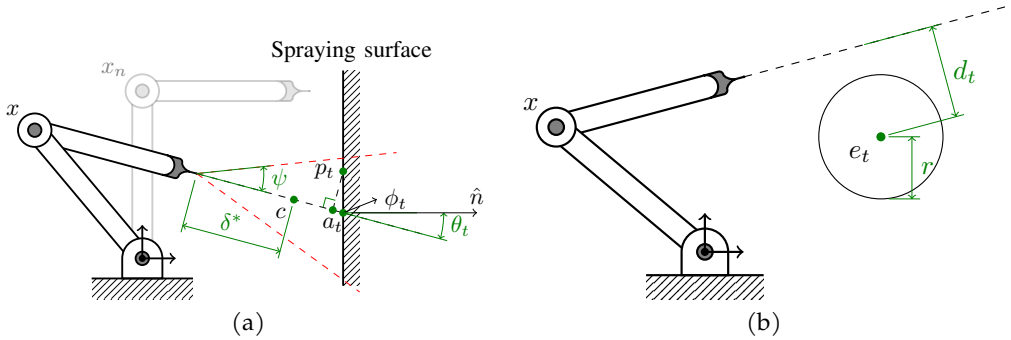


Figure 4.1: Schematic diagrams showing main points of interest for the shared autonomous spraying task. (a) Representation of a robot in some state x, some nominal state x_n, and points of interest with respect to the spraying surface. The dotted red line indicates an estimation of the spraying cone. (b) Depiction of the $AvoidLookAtSphere$ task map.

A given optimal standoff distance $0 < \delta^* \in \mathbb{R}$ specified by an operator is assumed (Honegger and Codourey, 1998). In this case the $StandoffDist$ task is modeled by

$$\text{cost}_{StandoffDist} := \|c - p_t\|^2 \tag{4.12}$$

where c is a constant point in the end-effector frame such that its distance away is δ^*.

In order to draw the robot to a more desirable configuration, for example a configuration which keeps an elbow joint up, an additional cost term is specified by

$$\text{cost}_{Nominal} := \|x - x_n\|^2 \tag{4.13}$$

where x_n is some given manipulator specific nominal configuration.

To ensure smooth joint motion, joint velocity $\dot{x}$, acceleration $\ddot{x}$, and jerk $\dddot{x}$ is minimized using

$$\text{cost}_{MotionSmooth} := \omega_1\|\dot{x}_t\|^2 + \omega_2\|\ddot{x}_t\|^2 + \omega_3\|\dddot{x}_t\|^2 \tag{4.14}$$

where $0 < \omega_1, \omega_2, \omega_3 \in \mathbb{R}$ are weighting terms, and $\dot{x}, \ddot{x}$, and $\dddot{x}$ are estimated using backward differencing using a window of the previous three solutions. The values for ω_i are tuned to reflect the relative importance of each term - for a given term i, a higher value for ω_i implies higher smoothing for term i. In the experiments described in later sections, these values were set to 1 for all i. Smoothing joint space motion up to the third derivative is generally beneficial in terms of wear and tear on the robot.

Let $\widetilde{e}_{0:t}$ represent all points previously sprayed. Two potential approaches for avoiding over-spray are described. The first includes the goal in the optimization problem

itself using an inequality constraint, and the second modulates the input using the sprayed locations information.

INEQUALITY CONSTRAINT APPROACH The constraint for every $\widetilde{e}_t$ is described as

$$c_{AvoidLookAtSphere} := r^2 - d_t^2, \tag{4.15}$$

which ensures that the distance d_t between the end-effector approach axis and the point $\widetilde{e}_t$ is always greater than a given radius r. Of course, over the duration of the task the number of points $\widetilde{e}_t$ grows.

INPUT MODULATION APPROACH In this approach the previous inequality constraint is removed and instead a control input u_t is used that directly modulates the human input with the environment model $\widetilde{e}_{0:t}$ using Equation 4.4.

ENSURING A TARGET IS WITHIN THE SPRAYING CONE Finally, inequality constraints enforce practical motion constraints such as joint limits. In the concrete spraying task, when the shotcrete is ejected from the nozzle the distribution can be estimated by a cone A. Rodriguez et al., 2009. Thus, the target position should lie within this cone. The following *GazeAt* constraint ensures this property.

Let $p_t = (p_t^x, p_t^y, p_t^z)^T$ define the target point defined in the end-effector frame. Keeping p_t within the spraying cone can be represented as an inequality constraint given by

$$c_{GazeAt} := \begin{pmatrix} p_t^{x2} + p_t^{y2} - \tan(\psi)^2 p_t^{z2} \\ -p_t^z \end{pmatrix} \tag{4.16}$$

where ψ is the angle of the spraying cone. Note, the bottom term in Equation 4.16 ensures p_t is in-front of the end-effector – not behind. Regarding the top part of Equation 4.16 it derived from the equation of a double cone as follows.

The double cone in three dimensions (x, y, z) is given by

$$r^2 = z^2 \tan(\psi)^2$$

where $r^2 = x^2 + y^2$. Note, ψ represents the angle between the z-axis and the cone surface. The interior of the cone can thus be represented by the inequality $r^2 \leq z^2 \tan(\psi)^2$, and by some rearrangement yields the top part of Equation 4.16.

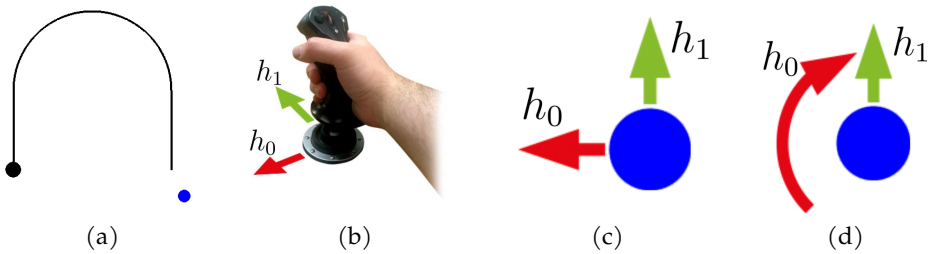


Figure 4.2: Joystick mappings experiment: (a) visualizer for the experiment, the robot (blue) is in its starting position, the line to follow is in black and the black dot indicates the final goal position for the robot, (b) the axes of the joystick, (c) Mode 1 is a linear mapping, and (d) Mode 2 is a car-like mapping.

4.5 INVESTIGATING APPROPRIATE JOYSTICK MAPPINGS

MOTIVATION During development of the experiments described in later sections, the following question arose: what should be the mapping between the joystick and control axes? Two options were considered: (1) joystick axes align with the local transform frame (Figure 4.2c), and (2) one joystick axis moved the robot along a certain heading and the other axis rotated the heading (Figure 4.2d).

EXPERIMENTAL DESIGN AND SETUP A within-subjects experiment was designed in order to test for which mapping would lead to higher performance in a direct teleoperation setting. In this case, higher performance is when the operator can maneuver the robot to precisely track a path from a start state to a finish state whilst trying to complete the task in the shortest time as possible. The experiment required the participant to move the robot so that it tracked a line from a start position (the same for each participant) to an end goal (also the same for each participant). Figure 4.2a shows the visualizer that the users were presented. They were told to track the line as precisely as possible and to complete the task as quickly as they could. The task was conducted in both mappings and in each case the participant was allowed to attempt the task in a dry-run, then the recorded trial was conducted. For every participant, to reduce the effect of training bias the order in which the mappings were presented was randomized. In applications such as spraying and wiping, during the task the operator can see where the robot has been. To simulate this scenario the trail of the robot was visualized. Results for 8 participants (6 male, 2 female) were collected.

The visualizer, experiment, and data collection was implemented in Python leveraging the PyGame library (Shinners, 2011). Data was collected at 40Hz.

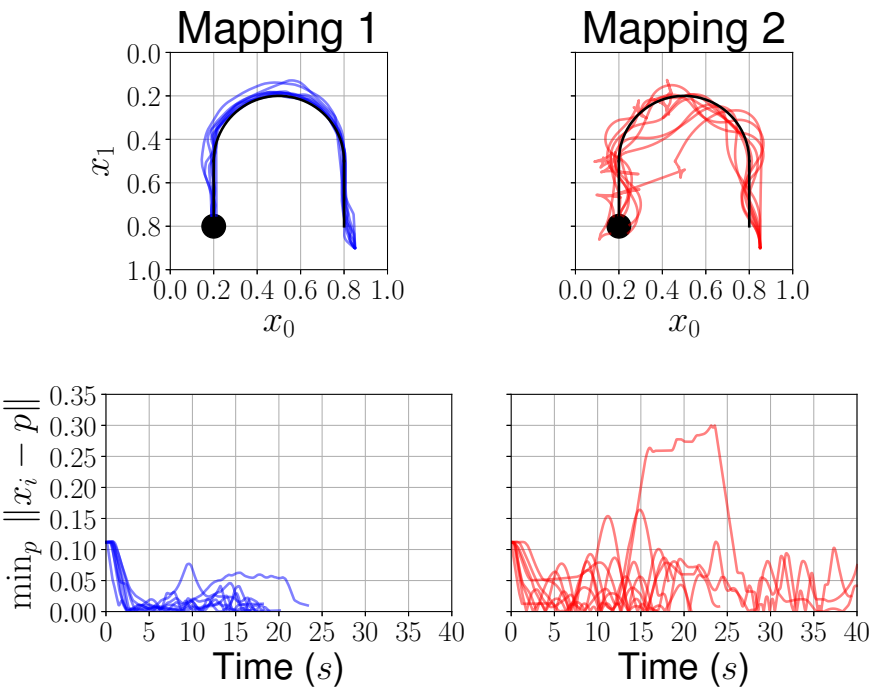


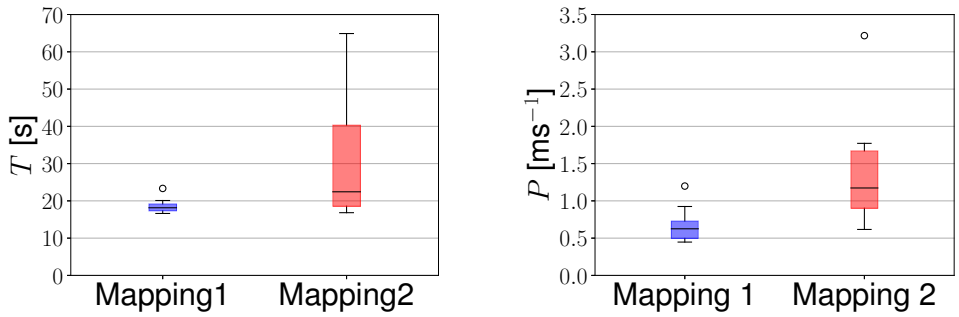
Figure 4.3: Trajectories for the joypad mapping experiment. Left show the results for Mode 1, and the right shows results for Mode 2. The two upper plots show the trajectories by participants. The lower figures plots the minimum distance between the state x and the line $\mathcal{L}$ over time.

MEASURES OF PERFORMANCE Performance is measured using two metrics: (i) completion time T, and (ii) tracking precision denoted P. Clearly, lower values in each metric indicate higher performance. The tracking precision is given by

$$P = \frac{1}{T} \sum_{i=1}^{D} \min_{p \in \mathcal{L}} \|x_i - p\| \tag{4.17}$$

where $x_i \in \mathbb{R}^2$ is the position of the robot represented by the blue dot in Figure 4.2a, D is the total number of data points collected during the trail, $\mathcal{L} \subset \mathbb{R}^2$ is the space containing all the points on the line shown in Figure 4.2a in black, and p is a point on that line.

RESULTS Examples of the participants trajectories and the profile for the distance to the line are shown in Figure 4.3. The results for the completion times, and tracking precision metrics are visualized in Figure 4.4. On average, participants completed the task in 18.719 ± 2.018s, and 30.775 ± 15.974s for mappings 1 and 2 respectively. The average tracking precision for mapping 1 and 2 are $0.684 \pm 0.240 \mathrm{m\,s^{-1}}$ and $1.422 \pm 0.769 \mathrm{m\,s^{-1}}$ respectively.



(a) (b)

Figure 4.4: Results for the joystick mapping experiment: (a) completion times, and (b) precision metric.

DISCUSSION The results for the completion time indicate participants were able to complete the task faster using mapping 1. In addition, the smaller variance for mapping 1 indicates that participants were able to achieve faster times more consistently. The precision tracking results indicate that participants were able to track the line $\mathcal{L}$ with higher accuracy under mapping 1. Overall, the results indicate mapping 1 leads to higher performance; future experiments will use this type of input method.

This experiment has helped us to design effective interfaces. However, the experiment and its analysis is limited. The results are not taken to be conclusive facts but useful indicators. Future work should extend this investigation to more complex tasks with obstacles, the effects of viewpoint, and extensions to a physical system.

4.6 CLEARING EXPERIMENT: DIRECT CONTROL VS SHARED CONTROL

The potential benefits of this formulation is illustrated by a grasping scenario, as described in Section 4.4.1. The goal of the task consists of clearing a number of items from the table in a particular order by grasping each item and placing it into the bin without collisions as timely as possible, as depicted in Figure 4.5.

Two different modes were tested: in the first, *direct control* mode, the human operator regulates the three translation dimensions of the end-effector; the second, *input modulation* mode, is the proposed method. The system has access to a collision map of the scene, appropriate grasp orientations for each item, and it is able to keep track of objects. The operator has to place the objects in a given order according to their color, and this contextual knowledge is only known by the human in both modes. The environment is static, however this need not be the case if tracking is available via perception.

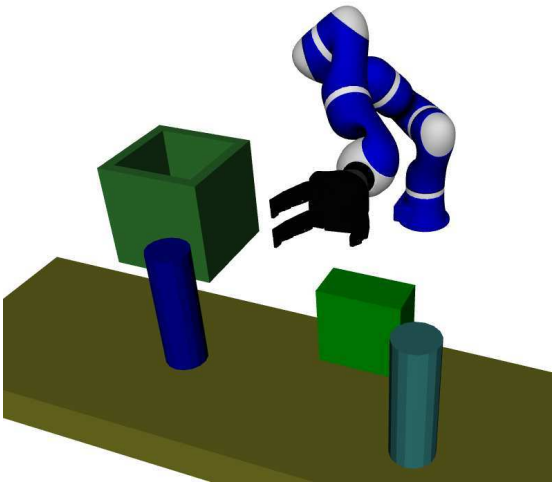


Figure 4.5: Kuka grasping experimental setup.

To compare each mode the completion time for five trials under each mode were collected. The direct control mode resulted in an average time of 31.6 ± 3.0 seconds, whilst the input modulation mode resulted in an average time of 17.6 ± 1.2 seconds, as shown in Figure 4.6.

4.7 Constraint-based optimization versus input modulation for a spraying task

Two strategies were compared for shared autonomy applied to the spraying task. The sub-task under consideration is to assist the user to avoid re-spraying areas and reduce overlap. The setup for this experiment is the same as in the previous section. The first method encodes $\widetilde{e}_{0:t}$ as avoidance regions using the $AvoidLookAtSphere$ constraint Equation 4.15 in the optimization formulation Equation 4.2c. The second, based on the proposed method, modulates the human input using Equation 4.4. The goal of this experiment is to demonstrate the computational savings the proposed method can have, rendering it applicable for a real-time system.

Each control cycle is split into two phases: a setup phase and a solver phase. The setup phase initializes a problem by specifying the target p_t. The solver phase parses the problem to an optimization solver; here, SNOPT (Gill et al., 2002) is used. For the first formulation, the setup phase consists of specifying target position p_t. For the second formulation, the setup phase uses the proposed method to modulate the human input. Each experiment is repeated 20 times for each mode and present the average computation times in Figure 4.7.

For the setup phase (Figure 4.7a), the completion time for the inequality constraint mode is negligible whereas a positive correlation between the CPU time and the num-

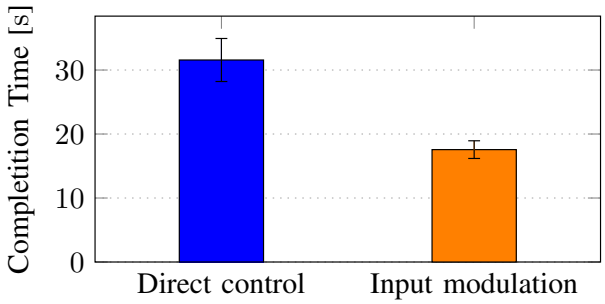


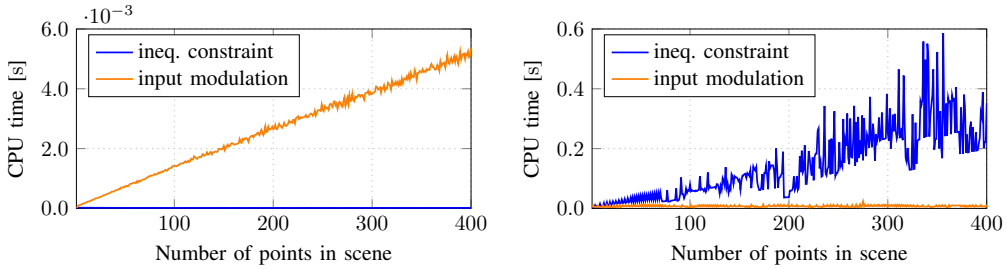
Figure 4.6: Completion time for the direct control mode versus the input modulation mode for a clearing task.

ber of points in the scene is clearly observed. For the solver phase, Figure 4.7b shows that the input modulation mode scales better for high numbers of points in the scene as opposed to the inequality constraint mode. Additionally, for each mode the average number of iterations for the solver phase and the mean-variance for the setup and solver phases is reported. The average number of iterations for the inequality constraint mode and input modulation mode is 12.73 ± 5.90 and 12.82 ± 5.72 iterations respectively. For the inequality constraint mode, the mean-variance for the setup and solver time was 5.58×10^{-12} and 3.57×10^{-6} respectively. For the input modulation mode, the mean-variance for the setup and solver time was 6.11×10^{-8} and 5.84×10^{-7} respectively.

4.8 Direct control versus input modulation shared control for a spraying task

In this section, the task comprises spraying a surface with minimal sprayed overlap. The goal here is two-fold: (1) illustrate a *qualitative* comparison between a direct control method and the input modulation method in the outcome of a spraying outcome, and (2) validate that the quality of the motion generated by the modulation method is not deficient when compared to direct control. Similarly to the grasping experiment, two modes are compared: a direct control mode, and an input modulation mode based on the proposed method. In both cases the human defines a two-dimensional trajectory for the robot to follow along the spraying surface. The direct control method tracks this trajectory as best as possible. The input modulation method uses the proposed method to modulate this input and avoid sprayed areas.

Figure 4.8 shows the experimental setup. For each mode, the same kinematic model, human-defined trajectory, and environment model are assumed. In total, 469 data points are synthesized corresponding to the input from the human operator in the task space. At each iteration in time, the environment variable $\widetilde{e}_t$ stores the currently



(a) Setup phase.

(b) Solver phase.

Figure 4.7: Completion time results plotted against number of points in the scene for (a) the setup phase, and (b) the solver phase for the shared autonomous spraying task.

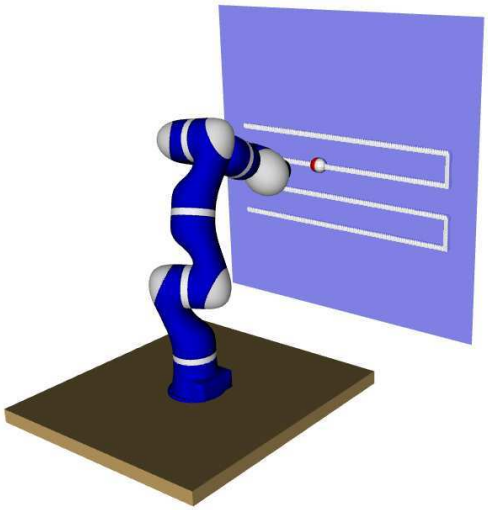
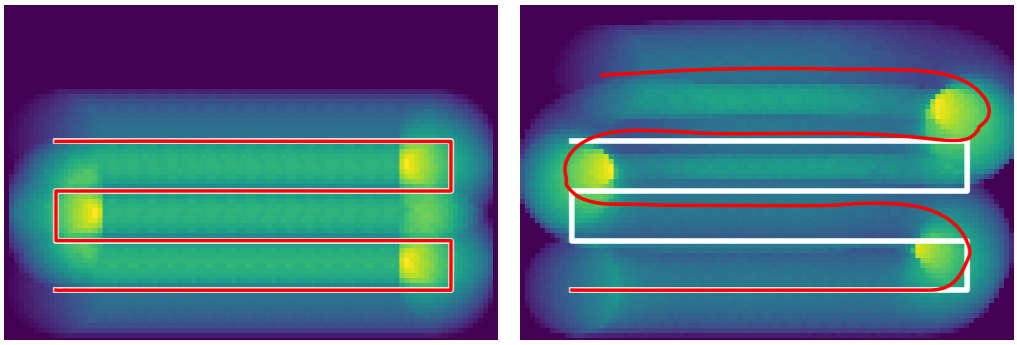


Figure 4.8: Experimental setup for a mock spraying task. The blue surface represents the spraying surface and the white line indicates synthesized human input.

sprayed position ϕ_t. However, a delay was introduced when logging $\widetilde{e}_t$ to the environment model $\widetilde{e}_{0:t}$ in order to avoid large and unrealistic repulsive potentials.

Figure 4.9 shows a color-map representing the sprayed regions, where brighter colors correspond to over-spraying. It is observed, qualitatively, that the precision tracking of the direct control mode results in a high overlap and minimal coverage, whereas the input modulation mode results in an increased coverage and less overlap. However, overlap is still high in the corners.

In both modes of control, the cost function – used by the solver to determine optimized robot motions – is the same, therefore the final cost for each time step is used to compare the quality of motion. Figure 4.10 plots the final costs for the objective function at each control cycle. The left-hand plot shows the final costs verses time, whilst the right-hand plot shows a box plot for all the data. For the performed motion the average final cost over the time steps was 5.054 ± 2.334 for direct control and 5.079 ± 2.109 for the input modulation method.



(a) Direct control. (b) Input modulation.

Figure 4.9: Comparison between direct control mode and the input modulation mode for a mock spraying task. The white line represents the synthesized human input. The red line represents the spraying trajectory.

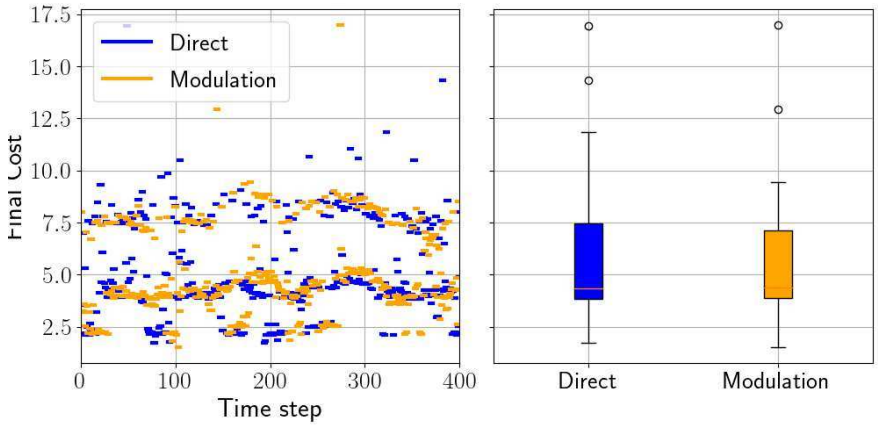


Figure 4.10: Comparing the final cost of optimization for direct control and shared control.

4.9 USER STUDY: DIRECT CONTROL VERSUS INPUT MODULATION FOR SPRAYING

MOTIVATION A small study was conducted to support the formulations and proposed method in this chapter within the context of spraying tasks inspired by the concrete spraying example. The goal is to provide evidence that these assitive methods can be beneficial in spraying scenarios. Application of material via spraying can be costly since material is being ejected continuously over the duration of the task. Thus, it is typically required that operators complete the job in a timely manner to not risk exceeding the design profile and wasting material (Ballou, 2003). Based on this information, the hypothesis is stated as follows.

Hypothesis 4.1. Autonomous modulation of teleoperated trajectories leads to faster completion times.

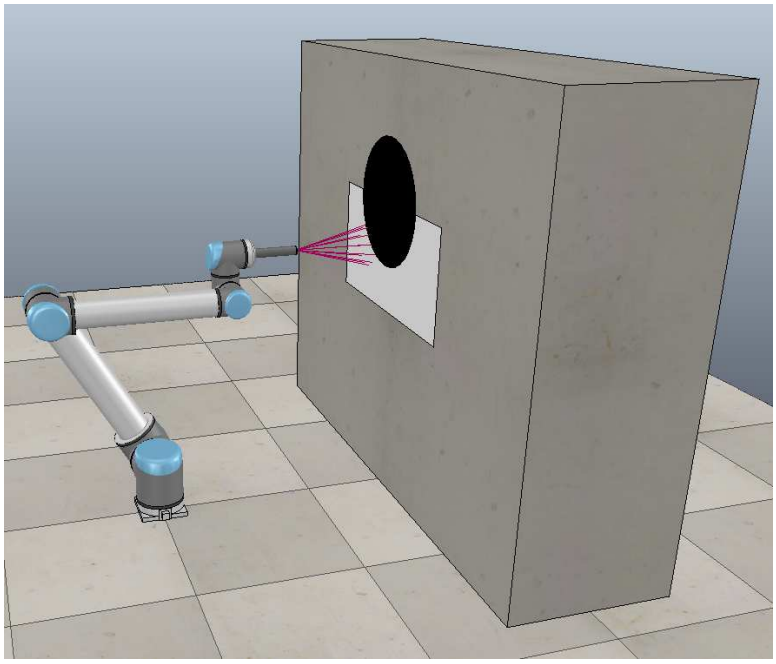


Figure 4.11: Spraying task using a UR10 manipulator in CopelliaSim (V-REP). The black circle on the wall represents a section that should be avoided being sprayed.

EXPERIMENTAL DESIGN AND SETUP To test this hypothesis, a spraying mock-up simulator was developed using CopelliaSim (Rohmer et al., 2013). Experiments were conducted using a paint spraying task simulation shown in Figure 4.11. The implementation for this experiment extended the experiments reported in Section 3.4. The participant protocol and stopping criteria is the same, however a black circle was placed into the scene that represents a section of the wall that the operator should *not* spray resulting in a non-convex spraying surface. A within-subjects experiment was performed that compared direct control verses the input modulation method proposed in this chapter, both under the *Reduced Task* (RT) mode of teleoperation.

Results were collected for five participants (3 male, 2 female). All participants were familiar with robotics but none had worked specifically in the field of teleoperation. The measure of performance compared in this task is completion time.

RESULTS AND DISCUSSION The results for this experiment are shown in Figure 4.12. The average time for completion under direct control was 24.805 ± 2.886s and under the input modulation method 16.581 ± 2.113s. This result indicates that users were able to complete the task in a more timely manner using the input modulation method rather than under the RT mode of teleoperation. The input modulation method roughly halves the average completion time indicating that Hypothesis 4.1 is true.

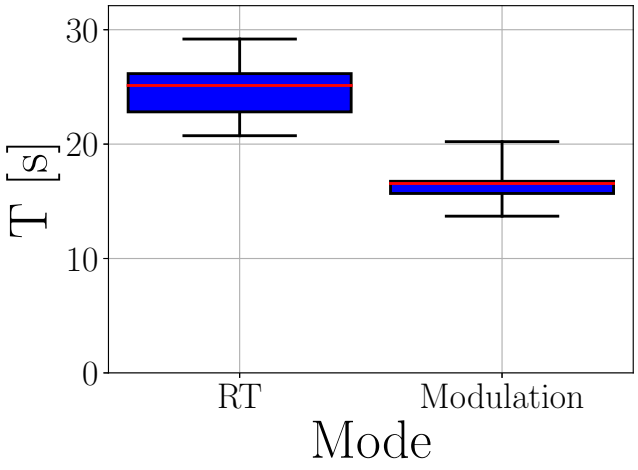


Figure 4.12: Completion times results for the RT mode and operator input modulation method.

4.10 ASSISTING OPERATORS TO IMPROVE COVERAGE

MOTIVATION Ensuring coverage in tasks such as spraying and wiping has two important priorities: (1) maximize the area visited, and (2) reduce overlap. These tasks often require the operator to follow a pattern; e.g., raster scan (Moura et al., 2018). Those patterns can lead to overlap if the operator does ont precisely control the robot. In spraying tasks, such as concrete spraying, patterned motions are required for the successful completion of the task and overlaps lead to wasted material (Ballou, 2003). In this experiment, the aim is to provide evidence that the proposed method can lead to higher coverage than direct control in terms of area coverage and minimal overlap.

EXPERIMENT SETUP AND DESIGN An experiment based on a wiping task was developed using a simulated KUKA LWR arm, see Figure 4.13. The operator is tasked with controlling the robot to perform a raster scan as accurately as possible across a whiteboard. During the task the operator could observe the current state of the robot and the trial which has been wiped. However, they are not given any additional information that informs them of the amount of overlap. For each trial, the robot was moved into an initial position (the same for every trial) and the trial was started when the user pulled the trigger on the joystick. Each trial lasting 30 seconds.

MEASURES OF PERFORMANCE After each trial, the center visualizer (red and white) is saved to disk as an image. Let us assume p_{ij} defines the ijth pixel, the visualizer is transformed to an array with $p_{ij} = 0$ in the place of red pixels, and $p_{ij} = 1$ otherwise.

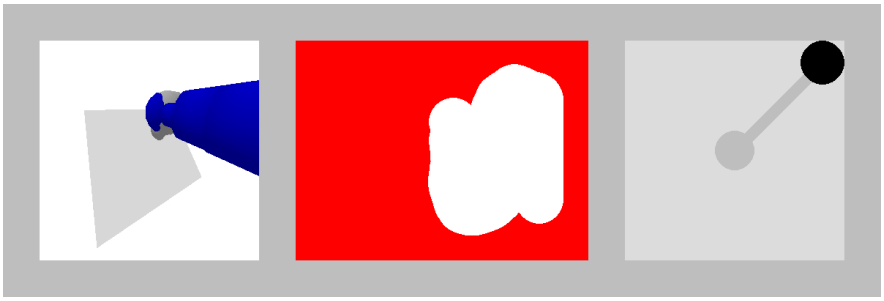


Figure 4.13: Example of the visualizer used in the experiments. The left shows the robot, in simulation, the middle shows a two dimensional representation of the whiteboard as if covered in red that the robot is wiping out, the right is a visualization of the joystick state.

This gives us a straightforward method to estimate area coverage using the metric defined by

$$C = \frac{100}{N_{total}} \sum_{i,j} p_{ij} \tag{4.18}$$

where N_{total} is the total number of pixels in the visualizer. Note C is scaled such that its value represents a percentage. Clearly, higher values for this metric are desirable.

Given the end-effector positions in the whiteboard x_t and the robots radius r, the overlap metric is defined by

$$O = \frac{1}{N_{total}} \sum_{i \neq j} \text{overlap}(x_i, x_j; r) \tag{4.19}$$

where overlap$(\cdot)$ returns the area overlap of two circles with equal radius r (note, when there is no overlap then its value is zero). The area computed, represented by overlap$(\cdot)$, is also known as the *symmetric lens* formed by the intersection of two equal disks (Weisstein, 2021a; Weisstein, 2021b). The function is defined by

$$\text{overlap}(x_i, x_j; r) = r^2\pi - 2r^2 \arctan\left(\frac{d}{\sqrt{4r^2 - d^2}}\right) - \frac{1}{2} d\sqrt{4r^2 - d^2} \tag{4.20}$$

where $d = \|x_i - x_j\|$ is the offset between the two disks. Clearly, Equation 4.20 is only defined for $2r > d$. When $2r \leq d$ the disks do not intersect, thus $\text{overlap}(x_i, x_j; r) = 0$ is defined. As mentioned above, reducing overall overlap is a priority therefore smaller values for O indicate higher performance.

RESULTS AND DISCUSSION In total, 10 trials were performed. During each trial end-effector positions x_t in the whiteboard frame were collected, and at the end the final visualizer image was saved to disk. Post-processing computed the values Equa-

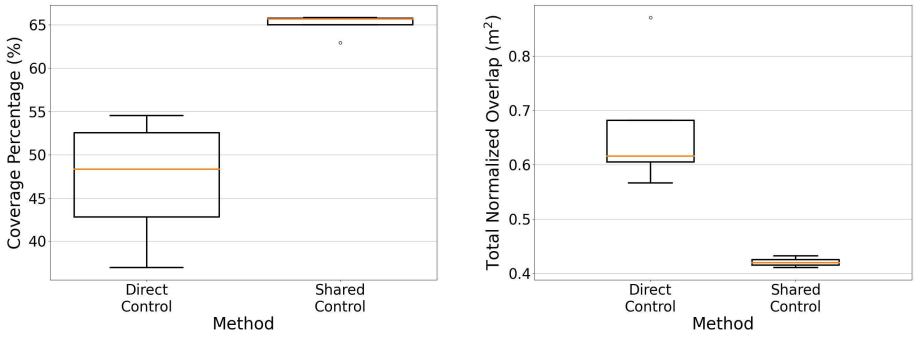
tion 4.18 and Equation 4.19 for each trial. The results are shown in Figure 4.14. This experiment has indicates that the input modulation method leads to beneficial improvements in increasing area coverage and reducing overlap.

4.11 DISCUSSION

In Section 4.5, an experiment was performed under controlled conditions to compared two different mappings between the joystick axes and controls. It was found that mapping 1 lead to higher performance than mapping 2. Recall for mapping 1, the vector describing the robot motion is aligned with the direction the operator positions the joystick (Figure 4.2c). The design choices for future experiments will take this into account. One of the experimental controls was that each participant visualized the experiment in the same viewer. Future work should investigate how much influence the viewpoint has on the performance.

A direct control mode was compared against the proposed input modulation mode for a grasping task in Section 4.6. The experiments demonstrate that the input modulation mode leads to significantly faster completion times. The completion time for the direct control mode is nearly double that of the proposed method. This is unsurprising, since the proposed method modulates the operator input in order to produce collision-free motion whereas for the direct control this is a responsibility of the human. This modulation removes the need for precise input relegating it to the autonomous system relaxing, therefore, the burden on the operator. However, this is just an illustrative experiment. More conclusive results require further user-study experiments.

In addition, a direct control mode was compared against the proposed method for a mock spraying task in Section 4.8. Figure 4.9a shows the spraying profile when using the direct control mode. There is clearly a large observable amount of overlap between the sprayed layers. There is also a significant portion left to spray in the top part of the plane thus requiring additional, and consequently, wasted material. Figure 4.9b shows a more homogeneous color distribution. Figure 4.9 demonstrates how the method is able to regulate the robot motion in order to produce greater homogeneity in the sprayed lining. In Figure 4.9b, the resulting curved edges of the spraying trajectory can be seen which is a desirable motion for obtaining smoothness of the laid material. Additionally, for both modes a high amount of overlap in the corners is observed. In the experiments every β_τ is tuned, as in Equation 4.6, to a constant value. The method could potentially use different β_τ at different points in the trajectory to minimize this overlap effect. In addition, the resulting cost profiles were compared,



(a) Percentage coverage.

(b) Total normalized overlap.

Figure 4.14: Comparing direct control and proposed modulation in terms of area coverage and overlap.

visualized in Figure 4.10, and observed that the proposed method does not lead to any decrease in joint motion optimality.

The setup and solver phase completion times were compared for two different shared autonomy formulations, an inequality constraint mode and input modulation mode. Figure 4.7b shows that the inequality constraint mode requires larger solver times, whilst Figure 4.7a shows the opposite relation for the setup time. However, given the difference in their order of magnitude, it is reasonable to essentially neglect the effect of the setup time in the total computation time. Therefore, the total computation time scales better for the input modulation mode This happens because the input modulation mode only requires an additional small computation in the task space per additional point in the scene as in Equation 4.6, whereas the inequality constraint mode requires the computation of an additional Jacobian per additional point in the scene, which is more costly. It was also observed that the number of solver iterations remains roughly constant when increasing the number of points in the scene, which suggests that the time per iteration increases for every additional point in the scene.

In Section 4.10 the method's ability to improve coverage and reduce overlap in spraying and wiping tasks was assessed. A direct control method implementing a RT mode of teleoperation was compared with the input modulation method. In both cases the modulation method exhibits significantly improved average results; modulation leads to increased coverage and reduced overlap. In addition, in both cases the variance in the modulation is smaller indicating that the modulation method performs at an improved level with greater reliability.

In this chapter, it was identified from past literature that humans are inherently bounded in their rationality (Simon, 1979), i.e. humans cannot be expected to make perfectly rational decisions/actions based on the full history of the task. A method

was introduced that makes local adjustments by modulating the operator's input based on appropriate streaming multi-modal sensory data $\widetilde{e}_t$. Through a number of experiments the applicability and benefits of this method for a number of industry inspired tasks was demonstrated.

5

SKILL-BASED SHARED CONTROL

5.1 INTRODUCTION

Many industrial applications require a human to teleoperate a robotic device, typically under direct control (Niemeyer et al., 2016), to perform a manipulation task as part of a construction or manufacturing process (e.g., Figure 1.1). During task execution, the operator will often employ various motion patterns to achieve different goals: in concrete spraying, an operator will switch between circular and sweeping motions to regulate the rate of concrete deposition and create a smooth lining (Ballou, 2003); in robot assisted welding, the operator's expertise on the desired weld determine the choice among different weave patterns (Silva Pereira et al., 2019); in plastering, different patterns yield different moulds (Bard et al., 2015); and in the cleaning process of the train's front panels, expert cleaners repeatedly employ spiral brushing motion patterns (Moura et al., 2018). These motion patterns, referred to as *skills*, are crucial to the success of the tasks.

Sustaining and smoothly executing these skills over the duration of a task can be difficult, requiring extensive and expensive training regimes, yet critical to performance. For example, in concrete spraying during tunnel construction, appropriately laying concrete is critical for preventing wasted material and collapse. Such critical teleoperation tasks also put considerable cognitive load on the operator. Further, there are significant challenges to ensure the safety of the operators when they operate close to the site (e.g., concrete falls away from the wall after being sprayed, known as rebound (Ballou, 2003)). These circumstances make the automation of such operations desirable. However, the crucial role of the operator's domain knowledge for the successful completion of such tasks motivates a human-in-the-loop *shared control* approach.

Many shared control paradigms developed in the literature can be thought of as direct control within a restricted workspace. This property benefits a novice by only allowing them to operate within safe limits – a concept that lets the operator largely dic-

Figure 5.1: Lab mock-up on a KUKA LWR in shared control mode; with online MPC based skill estimation and switching.

tate motion, under the principle of minimal intervention (Broad et al., 2019; Schwarting et al., 2017). While the principle achieves the crucial goal of minimally disrupting an operator, this author advocates that shared control systems should, in addition, assist novice operators (Enayati et al., 2018; Erden and Billard, 2015). Thus, the focus in this chapter is to realize a shared control framework that captures key skills while ensuring the environment physical constraints are continuously satisfied.

Other critical issues that the shared control literature addresses are, for instance, assisted guidance (Rosenberg, 1993), human intention prediction (Hauser, 2013; Javdani et al., 2018), appropriate blending between human and autonomous policies (A. Dragan and Srinivasa, 2013), large time-delays (Ya-kun et al., 2017), appropriate teleoperation spaces (Chapter 3), obstacle avoidance (Broad et al., 2019; Rubagotti et al., 2019) (Chapter 4), and representation and utilization of expert demonstrations (Abi-Farraj, Osa, et al., 2017). However, these approaches tend to focus on cases where the intention of the operator consists of some distinct goal, e.g., an object to grasp. To the best of the authors knowledge, very few approaches to shared control target the problem of *how* the robot moves with regards to some skill. In this work, the goal is to be able to reproduce skills, rather than simply getting from one place to another.

Some shared autonomous methods provide assistance to the operators by blending the current operator command with an autonomous policy responsible for the adaptation to the dynamic environment (Kosari, Rydén, et al., 2014b) (Chapter 4). However, these strategies lack any anticipatory reasoning, i.e. early adaptation to predicted future events. Consider the example of an unskilled operator inadvertently

driving a robot towards an obstacle. In such an example, predicting the collision event and taking it in consideration by continuously re-planning a sequence of control commands over a receding horizon, in a *Model Predictive Control* (MPC) fashion, could significantly improve the level of assistance from the shared controller to the less skilled operator. Moreover, such capabilities are key to being able to produce skill-based controls, estimated from the operator's actions, in the face of changing physical constraints.

5.2 PROBLEM FORMULATION

A key nuance, specific to shared control, is the need for incorporating a prediction of operator commands $\widehat{h}(t; h^r)$ over a future time period $t \in \mathbb{T}_f = [t_c, t_f]$, where t_c and t_f are the current and future time-stamp respectively, and raw human signals from an interface (e.g., joystick) are given by $h^r(t)$ for a window of previous commands $t \in \mathbb{T}_p = [t_p, t_c]$ such that t_p is a previous time-stamp. It is assumed the operator input commands are drawn from a space equivalent to the control action space.

It is possible mathematically formalize a receding horizon shared controller as a standard optimal control problem. A sequence of the optimal control commands $u(t) \in \mathbb{R}^{n_u}$ and the corresponding sequence of system states $x(t) \in \mathbb{R}^{n_x}$ are the result from solving

$$x^*, u^* = \arg\min_{x,u} \ \text{cost}(x, u; \widehat{h}) \tag{5.1a}$$

$$\text{subject to } \dot{x} = f(x, u), \ x \in \mathbb{X}(\widetilde{e}), \ u \in \mathbb{U} \tag{5.1b}$$

where $f(\cdot)$ represents the equations of motion, $\mathbb{X}(\cdot)$ and $\mathbb{U}$ are the sets of feasible states and controls respectively represented by a combination of equality and inequality constraints, and $\widetilde{e}(t)$ is an environment model from sensing data.

Recent literature introduces two different approaches to solve Equation 5.1 for obstacle avoidance (Broad et al., 2019; Rubagotti et al., 2019). Both of these approaches minimize an objective function defined by $\text{cost}(x, u; \widehat{h}) = \int_{\mathbb{T}_f} \|x - \widehat{x}\|^2_{W_x} + \|u - \widehat{h}\|^2_{W_u} \, dt$ where $\widehat{x}$ is the solution of $\dot{x} = f(x, \widehat{h})$ such that $x(t_c) = x_c$ is known. Broad et al., 2019 sample $N \gg 1$ controls $\{u\}_N \sim \mathbb{U}$, exclude samples such that $x \notin \mathbb{X}(\widetilde{e})$, and choose x, u with minimal cost. Rubagotti et al., 2019 use an off-the-shelf solver to compute a local minimizer. Both works adhere to the principle of minimal intervention - evident by the choice of cost function - i.e. minimally adjust the operator input commands such that the system constraints are satisfied. Additionally, despite both approaches handling arbitrary prediction models $\widehat{h}$, both their experiments assume a simplistic model, i.e. $\widehat{h}(t) = h^r_c$ for all $t \in \mathbb{T}_f$ such that h^r_c is the current raw operator command.

The main focus of this chapter is to address the problem of planning motions that respect a skill whilst satisfying system constraints. The cost functions and prediction models in the related literature fail to capture the key goal of maintaining a prescribed motion pattern; therefore, the need for an alternative approach that follows the principle of minimal intervention while being the least disruptive to the operator's intentions or skill choice.

The goal is to assist novice users to produce accurate skill patterns. Thus, the role of the human is to provide the necessary input so that the skill can be identified reliably from the interface data. The global plan should be dictated by the human, and the role of the autonomy is to correct the generated motion locally.

A generalized treatment of human-policy identification or human intention detection is out-of-scope of this work – it is a challenging problem with many factors often leading to models that are computationally infeasible for online prediction (Oguz et al., 2018). However, several domains are identified as the target are conducive to some simplifying assumptions: it is assumed (i) a given finite set of skill models denoted $\mathcal{S}$, by an expert operator, that contain all motion patterns required for the completion of the task at hand, and (ii) the novice operator is skilled enough to enact teleoperation motions that, albeit sub-optimal, are identifiable by an off-the-shelf policy prediction method given $\mathcal{S}$.

Based on the above formulation, the questions addressed in this chapter are summarized as follows.

Question 5.1. What is an appropriate state and control representation that both captures the intended skills and allows the trajectories to bend and adapt in order to avoid obstacles?

Question 5.2. How to capture intentions of a human operator in the form of motion patterns?

Question 5.3. How to plan trajectories that respect both the estimated intended skill and the principle of minimal intervention, while avoiding obstacles and remaining computationally feasible for online operation?

5.3 OBSERVING SKILLS IN INDUSTRY

First, an observation on motion patterns is made that is intended to be reproduced. Observe three examples of welds in Figure Figure 5.2 (upper); these show instances of patterned motions around some central axis. Figure 5.2 (lower) shows two instances of patterns, a wave and cycloid, in a two-dimensional plane exhibiting the same feature (pattern around a central axis). This central axis (red) is termed as the

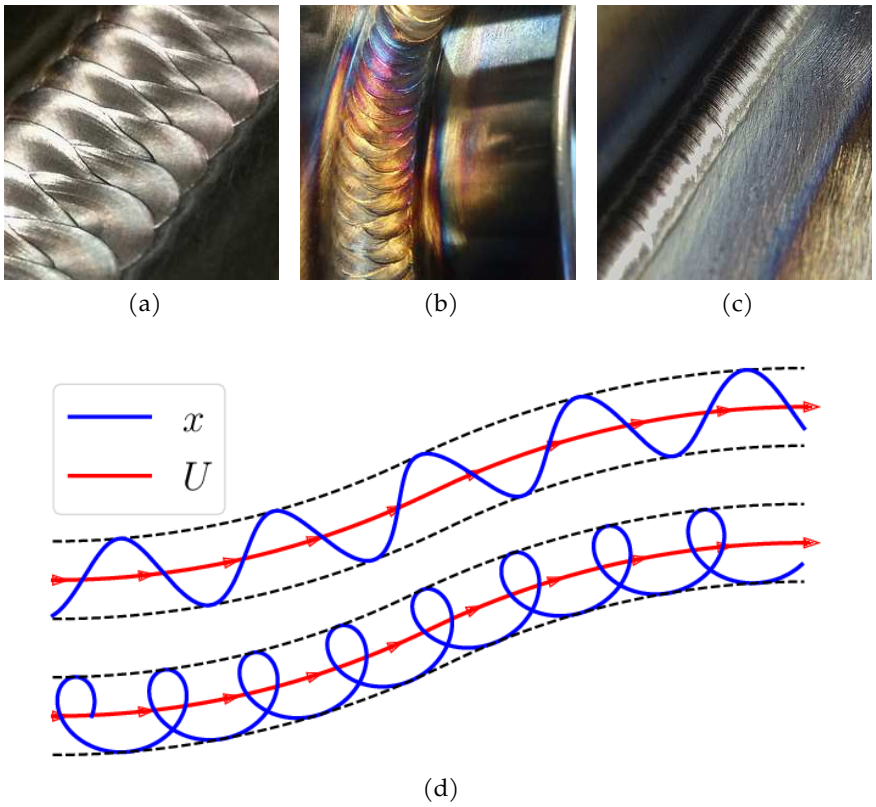


Figure 5.2: Motion pattern examples. Upper: weave patterns used in welding, credit Josh Welton Welton, 2014. These highlight patterns implemented in industry to achieve different goals. Lower: model-based trajectories for a wave and cycloid skill.

underlying trajectory and denote by U. Let us denote the state trajectory - the trajectory a robot will actually follow - by x (blue). This work relies on a distinction between the underlying trajectory U, a skill representation S_i that produces a pattern such that subscript i denotes the ith skill from the skill set $\mathcal{S}$, and the method by which these are combined to form states x and controls u.

5.4 NOVEL PARAMETRIC MODELS FOR STATE AND CONTROL TRAJECTORIES

A key feature of the underlying trajectory is that it must sustain the intended motion pattern while having the ability to bend to avoid obstacles and satisfy changing environmental constraints. Two approaches could be considered: (1) a model-free approach where the optimization handles these requirements via constraints, or (2) a model-based approach where these requirements are inherent to the model. It has been found that a model-based approach provides a suitable solution.

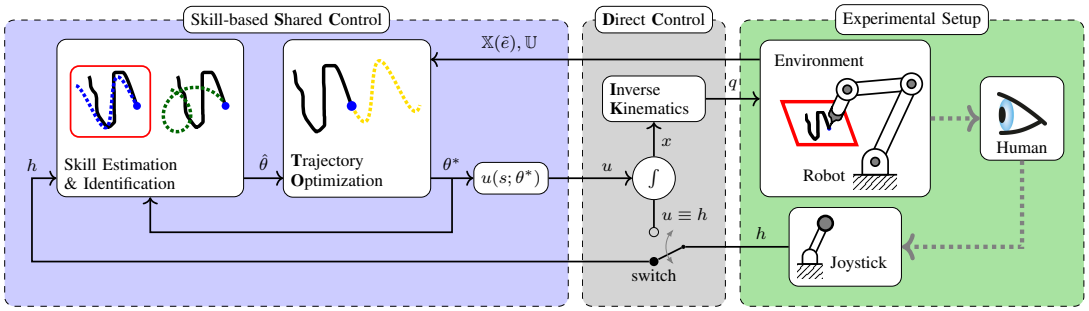


Figure 5.3: Method outline. From right to left: the *Experimental Setup* illustrates an operator controlling the robot with a joystick only using visual feedback of the task; the ***D**irect **C**ontrol* maps control signals to joint state targets; and the *Skill-based **S**hared **C**ontrol* estimates skill parameters and optimizes the robot motions given continuous environment sensing.

5.4.1 *Mapping time to a spatial domain*

In order to describe the method, a change of variable is introduced – let there exist some spatial parameter s related to time by

$$\frac{ds}{dt} = v(t) \tag{5.2}$$

where $v(t)$ is some velocity profile. Thus, for a time period $\mathbb{T} = [t_a, t_b]$, integrating Equation 5.2 such that a point, e.g., $s(t_a) = s_a$, is known leads to $s(t)$ defining $\mathbb{S} = [s_a, s_b]$. In this work, $v(t) = v_0$ is set where $v_0 \in \mathbb{R}$ is a constant parameter.

5.4.2 *State and control trajectory representations*

Let the underlying trajectory and skill representations be described by $U(s;\psi)$ and $S_i(s;\rho_i)$ respectively where ψ, ρ_i are model parameters. For brevity, all model parameters are collected and denote by $\theta_i = (\psi^T, \rho_i^T)^T$. These models are combined to describe the state trajectory by

$$x(s;\theta_i) = U + S_i U' \tag{5.3}$$

where a dash $'$ represents the derivative with respect to s. Let us highlight here the need for the change of variables Equation 5.2: the model U' in Equation 5.3 must have unit-norm so not to interfere with the skill pattern S_i. Due to a parameterized model Equation 5.3, the control actions $u(\cdot)$ are given by $x'(\cdot)$, and thus defined by

$$u(s;\theta_i) = U' + S_i' U' + S_i U'' \tag{5.4}$$

where the product rule for differentiation has been applied.

5.5 SKILL REPRESENTATION

Following assumption (i) in Section 5.2, the skill set $\mathcal{S}$ is sub-divided by skill models S_i, each parameterized by ρ_i, i.e. $\mathcal{S} = \cup_i \{S_i(s;\rho_i) : \rho_i \in \mathbb{R}^{m_i}\}$.

The specific skill representation is assumed given and dependent on the specific application domain. In this section, two general formulations for skill representation are described; one based on polar coordinates and another based on Cartesian coordinates. These general formulations allow an domain expert to easily define their own functions. Several motivating examples for skills, summarized in Figure 5.4, are provided.

It is acknowledged that some skills may not be easily defined by handwritten equations. This motivates a learning-based approach where a representation $\widehat{S}_i(\cdot)$ is estimated from expert demonstrations. The focus in this work is to provide a shared control framework that incorporates the notion of skills, and thus this is left for future work.

5.5.1 *Polar coordinate representation*

Let the skill model $S_i(s;\rho_i)$ describe some motion pattern – that is specific to a given application domain, e.g., (Silva Pereira et al., 2019) exemplifies various weave patterns in welding. Here, a particular general form is used that can represent several skills (Section 5.5.3). Let S_i be a composition of the two functions $\sigma_i(\cdot)$ and $\omega_i(\cdot)$:

$$S_i(s;\rho_i) = \sigma_i \mathcal{R}(\omega_i) \tag{5.5}$$

where $\mathcal{R}(\cdot)$ defines a two-dimensional rotation matrix about some angle, $\sigma_i(\cdot)$ can be thought of as a scaling and $\omega_i(\cdot)$ as a rotation angle. A skill type can be therefore given by simply defining the scalar valued functions σ_i and ω_i. The derivative of Equation 5.5 with respect to s is

$$S_i'(s;\rho_i) = \sigma_i' \mathcal{R}(\omega_i) + \sigma_i \omega_i' \mathcal{R}'(\omega_i) \tag{5.6}$$

found by an application of the chain rule for differentiation.

5.5.2 *Cartesian coordinate representation*

The model for S given in Section 5.5.1 can be extended further. Let us imagine that at the current position of the underlying trajectory U_c a coordinate system is defined by the base unit-vectors U' and $\perp U'$. By scaling these unit vectors by functions $\sigma_0 = \sigma_0(s;\rho)$ and $\sigma_1 = \sigma_1(s;\rho)$, the generalized skill model is given by

$$S(s;\rho) = \sigma_0 I_2 + \sigma_1 \mathcal{R}_{\pi/2} \tag{5.7}$$

where $\mathcal{R}_{\pi/2}$ is the rotation matrix through 90 degrees. The derivative of Equation 5.7 with respect to s is thus given by

$$S'(s;\rho) = \sigma_0' I_2 + \sigma_1' \mathcal{R}_{\pi/2}\,. \tag{5.8}$$

5.5.3 *Skill representation definitions*

In this section, several motivating examples of skills based on the general representations presented in Section 5.5.1 and Section 5.5.2 are defined. Each skill is shown in Figure 5.4.

5.5.3.1 *Line skill*

First, the simplest skill is presented; the *line* skill. Ultimately, this skill results in a similar approach as in (Rubagotti et al., 2019) but under the proposed parameterization for state and control. The benefit of potentially including a line skill is that an operator may not want to produce a skill at certain points in the task and want the flexibility to maneuver the robot without there being a pattern - this can be useful to prevent the user being required to manually switch between direct and shared control modes by pressing a button and enable seamless interaction.

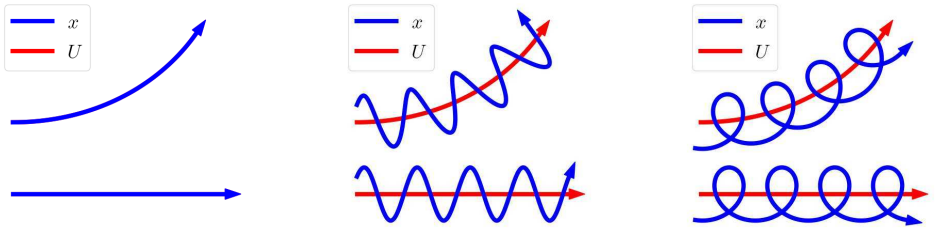
The line skill can be represented by

$$\sigma = 0, \tag{5.9a}$$
$$\omega = 0. \tag{5.9b}$$

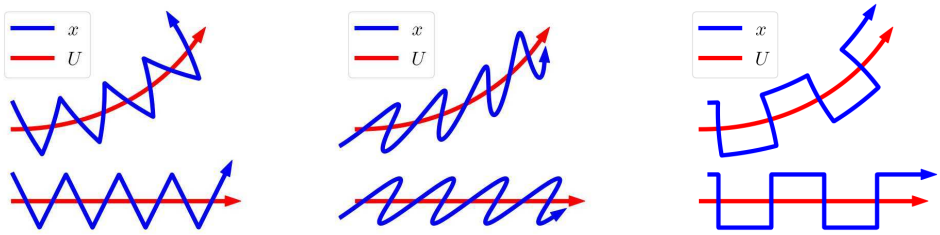
Notice that this skill does not have parameters, i.e. $m_i = 0$ for the line skill.

5.5.3.2 *Wave skill*

The second skill presented is a *wave* skill. Waves are used in applications such as concrete spraying (Ballou, 2003) and welding (Silva Pereira et al., 2019).



(a) Line. (b) Wave. (c) Cycloid.



(d) Zig-zag. (e) Skewed wave. (f) Box-wave.

Figure 5.4: Skills. Each of these trajectories have the same underlying trajectory. The skill in each has been defined in this chapter.

This skill is based on a sine wave and has the representation

$$\sigma = \rho_0 \sin(2\pi\rho_1 s + \rho_2), \tag{5.10a}$$

$$\omega = \frac{\pi}{2}. \tag{5.10b}$$

where three parameters represent the wave amplitude ρ_0, the wavelength ρ_1, and the phase ρ_2.

5.5.3.3 *Cycloid skill*

The third skill produces circular motions that are termed a *cycloid* skill. The cycloid skill, or concentric circles skill, are used, similar to the wave skill, in applications such as concrete spraying (Ballou, 2003) and welding (Silva Pereira et al., 2019).

The cycloid skill has the representation

$$\sigma = \rho_0 \tag{5.11a}$$

$$\omega = 2\pi\rho_1 s + \rho_2 \tag{5.11b}$$

where three parameters represent the cycloids radius ρ_0, the wave length ρ_1, and phase ρ_2.

5.5.3.4 *Zig-zag skill*

The fourth is defined as a *zig-zag* skill. The zig-zag motion pattern was referred to in (Silva Pereira et al., 2019) as a potential weave pattern in welding.

The zig-zag skill can be modeled by the representation

$$\sigma = \left(\rho_1 \left(s - \rho_2\right) - \text{floor}\left(\rho_1 \left(s - \rho_2\right)\right)\right) a(s) + b(s) + \frac{\rho_0}{2} \tag{5.12a}$$

$$\omega = \frac{\pi}{2} \tag{5.12b}$$

where floor$(\cdot)$ is the floor-operator that returns the greatest integer less than or equal to the input, the functions $a(\cdot)$ and $b(\cdot)$ are given by

$$a(s) = \rho_0 \left(-1 + 2 \,\text{mod}\left(\text{floor}\left(\rho_1 \left(s - \rho_2\right)\right), 2\right)\right) \text{ and} \tag{5.13a}$$

$$b(s) = -\rho_0 \,\text{mod}\left(\text{floor}\left(\rho_1 \left(s - \rho_2\right)\right), 2\right), \tag{5.13b}$$

and mod$(\cdot)$ is the modulo-operator (i.e. it returns the remainder or signed remainder of a division). The zig-zag skill is defined by three parameters, similarly to the wave skill ρ_0 is an amplitude, ρ_1 a frequency, and ρ_2 a phase.

5.5.3.5 *Skewed-wave skill*

The fifth skill is a generalization of the wave skill in Equation Equation 5.10. This skews the wave and is thus termed the *skewed-wave* skill and has the representation

$$\sigma = \rho_0 \sin(2\pi\rho_1(s + \rho_2)) \tag{5.14a}$$

$$\omega = \rho_3 \tag{5.14b}$$

where, as in Equation 5.10, ρ_0, ρ_1, and ρ_2 represent the amplitude, wavelength, and phase respectively, and here ρ_3 is an angle that skews the wave.

5.5.3.6 *Box-wave skill*

The final motivating example makes use of the Cartesian representation in Equation Equation 5.7. This skill produces a box-like wave and is thus called the *box-wave* skill. This is another example of a weave pattern used in welding (Silva Pereira et al., 2019).

The box-wave is comprised of four parts of the motion. Each with a given model determined by the value of the s. Recall, σ_0, σ_1 are the scaling functions as in Equation 5.7. For convenience, let us define the values

$$m = \text{mod}(s + \rho_3, \rho_2), \quad \xi = \frac{4\text{mod}(s + \rho_3, \frac{1}{4}\rho_2)}{\rho_2} \tag{5.15}$$

where mod$(\cdot)$ is the modulo-operator as used in the zig-zag skill in Section 5.5.3.4.

Let $\sigma(\cdot)$ denote the function which stacks the scaling functions σ_0, σ_1. The box-wave skill is thus defined by

$$\sigma(s;\rho) = \begin{bmatrix} \sigma_0(s;\rho) \\ \sigma_1(s;\rho) \end{bmatrix} = \begin{cases} [\rho_0(2\xi - 1), -\rho_1]^T & \text{if } 0 \leq m \leq \frac{1}{4}\rho_2 \\ [\rho_0(1 - 2\xi), \rho_1(2\xi - 1)]^T & \text{if } \frac{1}{4}\rho_2 \leq m \leq \frac{1}{2}\rho_2 \\ [\rho_0(2\xi - 1), \rho_1]^T & \text{if } \frac{1}{2}\rho_2 \leq m \leq \frac{3}{4}\rho_2 \\ [\rho_0(1 - 2\xi), \rho_1(1 - 2\xi)]^T & \text{if } \frac{3}{4}\rho_2 \leq m \leq \rho_2 \end{cases} \tag{5.16}$$

where m and ξ are given by Equation 5.15.

5.6 UNDERLYING TRAJECTORY REPRESENTATION

In this section, two approaches are described to model an underlying trajectory representation that were considered during the development of this work. The underlying trajectory is the central axis that the state trajectory (i.e. pattern) repeats about. The representation should be versatile enough to bend and curve so that obstacles can be avoided and adapted to the operators intention. An important consideration for the choice of underlying trajectory model is that the features of the pattern should be maintained.

5.6.1 *Polynomial model approach*

A first attempt to model the underlying trajectory used a polynomial representation. Such a function was defined by

$$U' = \begin{bmatrix} \cos(\phi_0) \\ \sin(\phi_0) \end{bmatrix} + \phi_1(s - s_c)\, \mathcal{R}_{\pi/2} \begin{bmatrix} \cos(\phi_0) \\ \sin(\phi_0) \end{bmatrix} \tag{5.17}$$

where ϕ_0 defines an orientation parameter, and ϕ_1 is a parameter which bends the trajectory.

This model allows us to bend the trajectory based on a single parameter ϕ_1. Unfortunately, it was observed that near to constraints this model lead to high velocities in the state trajectory and did not maintain the features of the skill (e.g., the wave's frequency). In an attempt to counteract this issue, additional constraints on the state velocity were imposed. However, this lead to numerical instabilities in the trajectory optimization.

5.6.2 *Clothoid-based approach*

A model-based approach using Clothoids as a representation for the underlying trajectory was found to provide a suitable solution. The Clothoid representation – also known as Euler curves, has some inherent properties such as (1) linearly varying curvature, and (2) a compact representation (small number of parameters), that will allow us to realize online adaptation within the optimization framework. Clothoids have been utilized in road design (Marzbani et al., 2015), path (Brezak and Petrović, 2014) and attitude (Girbés et al., 2019) planning, autonomous driving (Lima et al., 2015; Silva and Grassi, 2018), and continuum robotics (Gonthina et al., 2019).

The Clothoid model of the underlying trajectory is given by

$$U' = [\cos(\alpha), \sin(\alpha)]^T, \text{ such that} \tag{5.18a}$$

$$\alpha' = \phi_0 + \phi_1(s - s_c) \tag{5.18b}$$

for all $s \in \mathbb{S}_f = [s_c, s_f]$ where $\alpha(s)$ describes the orientation and $\phi = [\phi_0, \phi_1]^T \in \mathbb{R}^2$ describe the curvature of the Clothoid trajectory. Note, $\alpha(s)$ is found by integrating Equation 5.18b such that $\alpha(s_c) = \alpha_c$ is known. The underlying trajectory parameters are $\psi = [\phi_0, \phi_1, \alpha_c]^T$. Notice, for all α, U' has unit-norm.

A representation for U is found by integrating Equation 5.18a given that $U(s_c; \psi) = U_c$ is known. If the current robot state x_c is known, then $U_c = x_c - S_i U'$ is obtained by re-arranging Equation 5.3. Note, the analytic integral of Equation 5.18a can only be found in terms of a special class of functions known as the Fresnel integrals (Heald, 1985). To avoid this difficulty, U is instead computed by approximation using the multi-variable version of the Runge–Kutta 4 method. Additionally, by simply defining more curvature parameters ϕ_0, ϕ_1, multiple Clothoid segments can be concatenated to form more elaborate underlying trajectories.

Specifically, the Clothoid model was considered suitable since it did not exhibit the same issues as the parabola model, i.e. it did not lead to unrealistic velocities in the state trajectory and did not require additional constraints on the problem. Clothoids have linearly varying curvature that is proportional to the arc length. A question can be raised whether proportionality to time or distance may be useful also. In the case of an autonomous robot this may indeed be useful. However, there is a subtle issue that needs to be addressed. The human's intention must be synchronized with the phase of the pattern being produced. For example, if the human intends to generate motion that is currently in a peak of a wave, and the robot moves as if it is in the trough, this could cause confusion.

In this section, the skill estimation and identification steps are described. The intention recognition problem is succinctly described by A. Dragan and Srinivasa, 2013 as

$$g^* = \arg\max_g \ \mathcal{P}(g|\Omega) \quad \text{subject to} \quad g \in \mathbb{G} \tag{5.19}$$

where $g \in \mathbb{G}$ is a goal in the space of potential goals $\mathbb{G}$ (this space can be continuous or discrete), g^* is the recognized operator goal which maximizes the posterior probability, and Ω are a collection of observations and parameters typically containing the operator input. For the purposes here, g represents an intended skill model and its respective parameterization.

5.7.1 *Skill estimation*

As in related literature (Hauser, 2013; A. Dragan and Srinivasa, 2013; Javdani et al., 2018), the assumption is made that the operator intent can be inferred conditioned on a window of their previous input – note, intent may change. In this case, the intent estimation consists of a selection of parameters $\widehat{\theta}_i$, providing a description of the operator's intent within the context of skills.

The following describes how to estimate $\widehat{\theta}_i$ from a window of raw interface signals $h^r(s)$ for $s \in \mathbb{S}_p = [s_p, s_c]$ that correspond to the time window $\mathbb{T}_p = [t_p, t_c]$ where t_c is the current time, and t_p is some previous time stamp such that $t_w = t_c - t_p$ is a specified window duration. Note, since the method relies on this window duration for t_w seconds at the start of a task the robot must be operated using direct control - see the switch in Figure 5.3. The encoding $\theta_i \in \mathbb{R}^{m_i}$ is treated as a decision variable. Under assumption (ii) in Section 5.2, $\widehat{\theta}_i$ is inferred by solving

$$\widehat{\theta}_i = \arg\min_{\theta_i} \varepsilon_i(\theta_i) \quad \text{subject to} \quad \theta_i \in \Theta_i^{SE} \tag{5.20}$$

such that $\varepsilon_i(\cdot)$ is an error function defined by

$$\varepsilon_i(\theta_i) = \int_{\mathbb{S}_p} \|u(s;\theta_i) - h^r(s)\|^2 \, ds + R(\theta_i) \tag{5.21}$$

where $u(\cdot)$ is defined in Equation 5.4, $R(\cdot)$ is a regularization term, and Θ_i^{SE} is the set of possible values for θ_i represented by a combination of inequality and equality constraints.

Regarding the regularization term $R(\cdot)$, it can be used to bias the solution towards pre-specified goals. As an example, to reduce the rate of change in the skill parameters ρ_i, sustaining characteristics of the skill. For instance, in the case of highly variable inputs, it is possible to define the regularization term as $R(\theta_i) = \|\rho_i - \widehat{\rho}_i^{prev}\|^2_{W_R}$ where $\widehat{\rho}_i^{prev}$ is the previous skill parameter solution and W_R is some appropriate weighting matrix. Alternatively, setting $R(\theta_i) = \|\phi\|^2_{W_R}$ penalizes ϕ, favoring low curvature trajectories. Additionally, a combination of weighted terms can address multiple goals.

5.7.2 *Skill identification*

Recall from Section 5.5 the skill set $\mathcal{S}$ is sub-divided by a number of skill models S_i. The model S_i is identified and its corresponding values for $\widehat{\theta}_i$ found by solving Equation 5.20 for each model and chose the skill that results in the smallest error ε_i.

5.8 TRAJECTORY OPTIMIZATION FOR SKILL-BASED SHARED CONTROL

Now, it is established how to identify a skill model S_i and a corresponding parameter description $\widehat{\theta}_i$ for the intended state and control signals from a window of the operator's raw interface signals. Ideally, in the next control loop step $s_n = s(t_n)$ (i.e. $t_n = t_c + \delta t$ where δt is the control loop cycle duration), the trajectory that most closely resembles the operator's intent should be executed, i.e. $x(s_n; \widehat{\theta}_i), u(s_n; \widehat{\theta}_i)$. However, the proposed shared control system must also be able to assist a novice operator in avoiding collision with obstacles while simultaneously maintaining the intended skill. Since the state and control trajectories are functions of the parameters θ_i, finding optimal parameters θ_i^* by solving Equation 5.1 directly leads to a highly nonlinear scheme. During the development of the method, it was found that constrained optimization solvers often fail to converge in this case. To resolve this issue, Equation 5.1 was adapted by using a modified version of the cost function, leading to the following optimization problem

$$\theta_i^* = \arg\min_{\theta_i} \text{cost}(\theta_i; \widehat{\theta}_i) \tag{5.22a}$$

subject to

$$x(s;\theta_i) \in \mathbb{X}(\widetilde{e}),\ u(s;\theta_i) \in \mathbb{U}, \theta_i \in \Theta_i^{TO} \tag{5.22b}$$

for all $s \in \mathbb{S}_f$

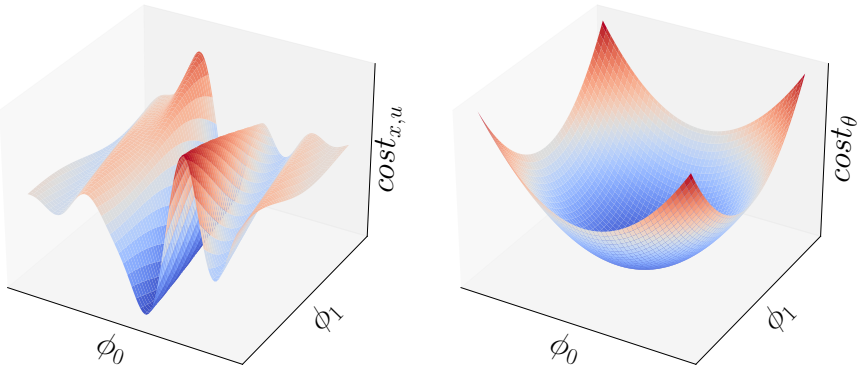


Figure 5.5: Cost landscape comparison highlighting the convexity in the proposed objective function (right) as opposed to the typical principle of minimal intervention cost function by applying the proposed state and trajectory models (left). Note, $\text{cost}_{x,u} = \text{cost}\big(x(\theta_i), u(\theta_i)\big)$ where cost is Equation 5.1 substituting the proposed models for x, u given by Equation 5.3 and Equation 5.4 respectively, and $\text{cost}_\theta = \text{cost}(\theta_i; \widehat{\theta}_i)$ is the proposed cost function Equation 5.23. Note, in both cases all variables in θ_i are fixed apart from ϕ_0, ϕ_1 for illustration purposes.

such that

$$\text{cost}(\theta_i; \widehat{\theta}_i) = \|\theta_i - \widehat{\theta}_i\|^2_{W_{TO}} \tag{5.23}$$

where $x(\cdot)$ and $u(\cdot)$ are defined by Equation 5.3 and Equation 5.4 respectively, Θ_i^{TO} is the set of possible values for θ_i, and $\mathbb{S}_f = [s_c, s_f]$ corresponds the time horizon $\mathbb{T}_f$ as in Equation 5.1. Note that a distinction is made between Θ_i^{SE} and Θ_i^{TO}. Depending on the application goals you can restrict the range of possible values that θ_i can take in both Equation 5.20 and Equation 5.22: for example, setting the underlying trajectory initial orientation α_c to the value estimated in Equation 5.20 – this is enforced in Equation 5.22 by setting the constraint $\alpha_c = \widehat{\alpha}_c$ where $\widehat{\alpha}_c$ is the value found by solving Equation 5.20. Notice also that since explicit equations are derived for x, u in terms of parameters θ_i, it is possible to remove the equations of motion[1] as an equality constraint.

The cost function proposed here follows the criteria of the principle of minimal intervention; however, the cost function attempts to maintain the operator's intentions as described by the estimated parameters. Additionally, the proposed cost function design has the numerical benefit of being strictly convex with respect to parameters θ_i, as schematically illustrated in Figure Figure 5.5. However, note the constraints are non-convex, meaning Equation 5.22 is still a nonlinear optimization problem - only convex in its objective function Equation 5.23.

1 Recall the equations of motion are represented by $f(\cdot)$ in Equation 5.1b

Now, an experimental setup is described for skill-based shared control, specifically for a wiping task involving two different skills defined explicitly. Note the conceptual similarity between wiping, welding, and spraying, where the two-dimensional task can directly map to a three-dimensional workspace.

The goal of the experimental mock-up task is for a robot to sweep a given area of a surface, e.g., see Figure 5.6. Let us assume the operator enacts one of two skills: (i) a wave skill where the robot should travel in a wave back-and-forth across the underlying trajectory, and (ii) a cycloid skill where the robot should travel in circles around the underlying trajectory. Let the states $x \in \mathbb{R}^2$ define the position of the end-effector on the wiping surface and the control input $u \in \mathbb{R}^2$ as the 2D input planar velocity. The operator gives two-dimensional inputs $h^r \in \mathbb{R}^2$ via a joystick.

Equation Equation 5.5 defines a particular version of the skill representation in terms of the scaling and rotational functions $\sigma_i(\cdot)$ and $\omega_i(\cdot)$ respectively. In these experiments the wave and cycloid skills described in Section 5.5.3.2 and Section 5.5.3.3 respectively are utilized.

For the skill estimation stage, the regularization function is defined by $R(\theta_i) = \|\theta_i - \widehat{\theta}_i^{prev}\|^2_{W_R}$ where $\widehat{\theta}_i^{prev}$ is the previous solution, and the constraints that define Θ_i^{SE} are as follows: $\theta_i^{min} \leq \theta_i \leq \theta_i^{max}$ bound the values of θ_i within lower θ_i^{min} and upper θ_i^{max} limits, and $\alpha_c = \alpha^*_{prev}$ where α^*_{prev} is the previous solution from the trajectory optimization retrieved via a feedback loop as in Figure 5.3. Note that on the very first iteration of the method that $\widehat{\theta}_i^{prev}$ and α^*_{prev} are undefined, and so the regularization term R and this particular constraint are relaxed for a single iteration.

In both the skill estimation and trajectory optimization, the solvers are seeded with the previous solution. In the very first iteration, a guess is given to the skill-estimation whereas the trajectory optimization uses the solution from the skill estimation $\widehat{\theta}_i$ as the initial seed.

5.9.1 *System description*

The method was implemented within the CasADi framework (Andersson et al., 2019), and utilizing the plugin for SNOPT (Gill et al., 2002) to solve Equation 5.20 and Equation 5.22. The method runs in an MPC loop: given a window of previous operator inputs, $\widehat{\theta}_i$ is estimated for all skills i in separate processing threads, identify the intended skill by comparing the cost of fittings Equation 5.21, compute an optimal θ_i^* by solving Equation 5.22 – resulting in a trajectory of states $x(\theta_i^*)$ and controls $u(\theta_i^*)$, from which the first step is executed.

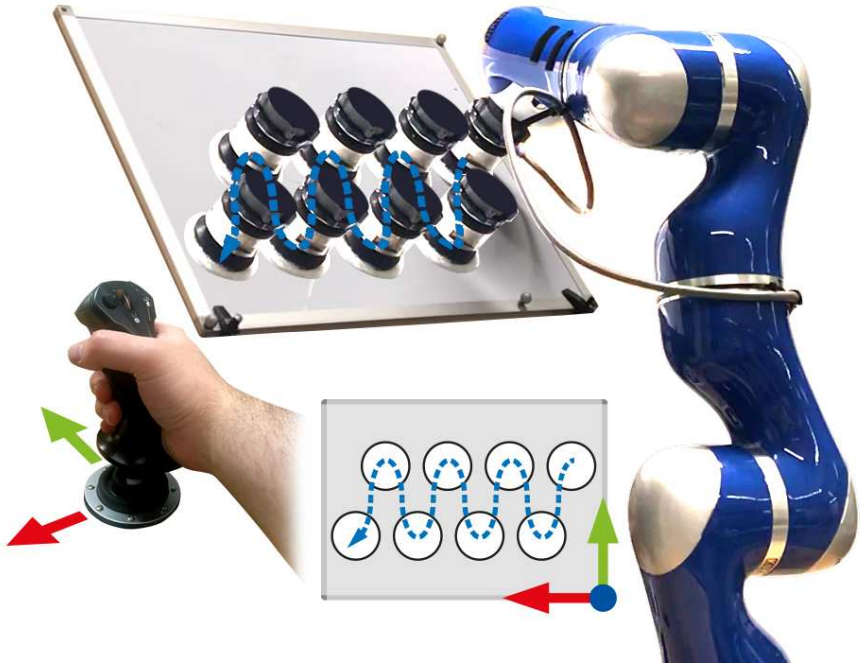
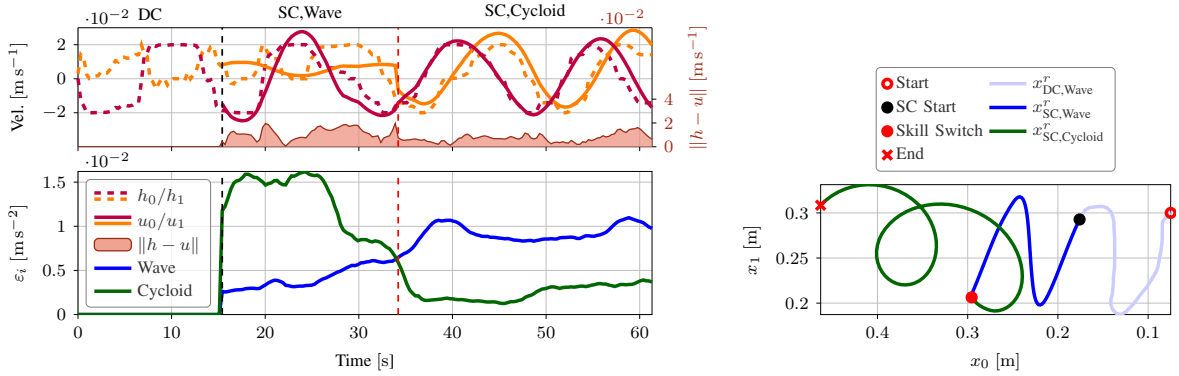


Figure 5.6: Wiping mock-up task where a user teleoperates a robot to perform a wave motion pattern from *right to left*.

The pose of the whiteboard, used as the wiping surface for the robot, is tracked using a Vicon motion capture system. Participants interface with the system using a Thrustmaster T-Flight HOTAS X joystick. The two main axes of the joystick are mapped corresponding to the two dimensions of the whiteboard, see Figure 5.6. The experiments were executed on a PC running 64-bit Ubuntu 20.04 with a 16-core Intel Core i9-9900KF CPU at 3.60GHz. Data was collected using a 7-DoF KUKA LWR Arm.

5.10 SWITCHES IN AN OPERATOR'S INTENDED SKILL

First, the method's capability to identify a skill and a change in the operator's intention solely from streaming joystick data is demonstrated. An operator starts by performing a wave skill, and then switches to a cycloid roughly 20 seconds after the method starts. Figure 5.7 (right) shows the trajectory resulting from applying control commands using the method. Figure 5.7 (upper-left) shows the velocity profile of the commanded versus applied control as well as the instantaneous deviation between the two. Figure 5.7 (lower-left) demonstrates the evolution of the skill fitting costs and instance of the skill switch identification.



(a) (b)

Figure 5.7: Skill switch experiment. Upper-left: input commands in the velocity space, the resulting velocity controls, and the difference between the two signals. Lower-left: corresponding fitting error $\varepsilon_i(\cdot)$ for both skills indicating (by the vertical dashed red line) the moment the skill switches. Right: robot trajectory (note that the robot moves from *right to left*).

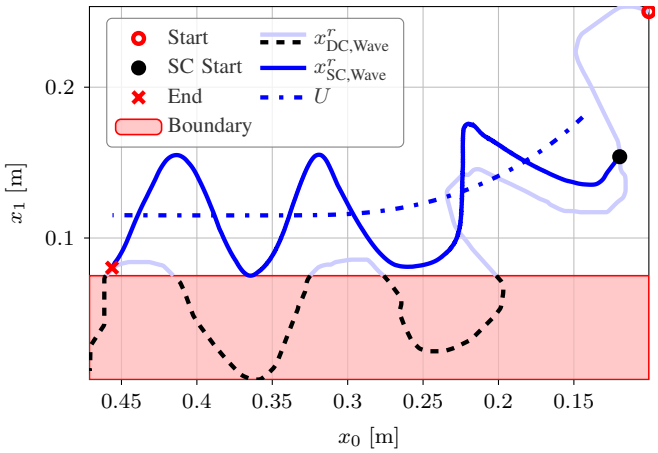


Figure 5.8: Static obstacle avoidance experiment. The trajectory produced by direct control where the operator inadvertently leads the robot into constraint violation and also the trajectory produced by shared control - the motion is from *right to left*.

5.11 STATIC OBSTACLE AVOIDANCE

Next the method's ability to continuously adhere to environmental constraints while maintaining the features of the estimated skill is demonstrated. Now, a boundary is introduced, represented in Equation 5.22b as an inequality constraint given by $x_{min} \leq x(s;\theta_i) \leq x_{max}$ for all $s \in \mathbb{S}_f$. Figure 5.8 (upper) shows the trajectory taken by the robot and its corresponding underlying trajectory. It also shows the trajectory that results from applying the same operator commands in a direct control mode, leading to a collision – highlighted by the black dotted lines.

An important consideration of any teleoperation system is that the overall computational time should remain below a certain threshold to allow for reasonable sam-

pling frequencies. Figure Figure 5.8 (lower) shows the skill estimation and trajectory optimization solver duration for each MPC loop cycle for this obstacle avoidance experiment. The average number of iterations was 16.9 ± 5.1 and 2.4 ± 3.3 and the average CPU time was $4.9 \pm 1.1\text{ms}$ and $1.4 \pm 0.2\text{ms}$ for the skill estimation and trajectory optimization, respectively. The average total CPU time was $6.3 \pm 1.2\text{ms}$, well under the 20ms threshold for a 50Hz operation.

5.12 Obstacle avoidance in a dynamic environment

A key requirement for the adoption of shared control systems in industry, such as the construction sector, is the ability to modify control inputs in the face of dynamic (changing) environmental constraints. Due to the MPC-nature of the method it is able to adapt online to changes in the environment whilst maintaining the skill features.

A straight edge was tracked using a Vicon motion capture system (see Figure 5.10) and incorporated this in the trajectory optimization step by including $d(x, \nu) \geq 0$ for all $s \in \mathbb{S}_f$ as a dynamic constraint in Equation 5.22b. Here, $d(\cdot)$ is a signed distance function, which computes the distance to the line for feasible states (negative otherwise) while ν is the pose of a Vicon tracking marker – updated at each control loop cycle.

An experimenter moved the straight edge within the workspace of the robot using a motion unknown to the operator. Figure 5.10 demonstrates that the proposed method was able to avoid the obstacle whilst maintaining the required skill.

5.13 User study: direct control versus shared control

In this section, a user study is described. The goal of this experiment is to analyze the relation between the quality of the inputs, provided by the participants, and the quality of the resulting motion patterns for the method in comparison to direct control.

5.13.1 *Participants*

A within-subjects experiment was conducted, evaluating the performance of 11 participants (9 male, 2 female). In Chapter 3 it was observed that certain habits (i.e. playing computer games) can effect task performance. Using the same criteria as in (Section 3.5.3), four participants were classified as highly familiar with computer games.

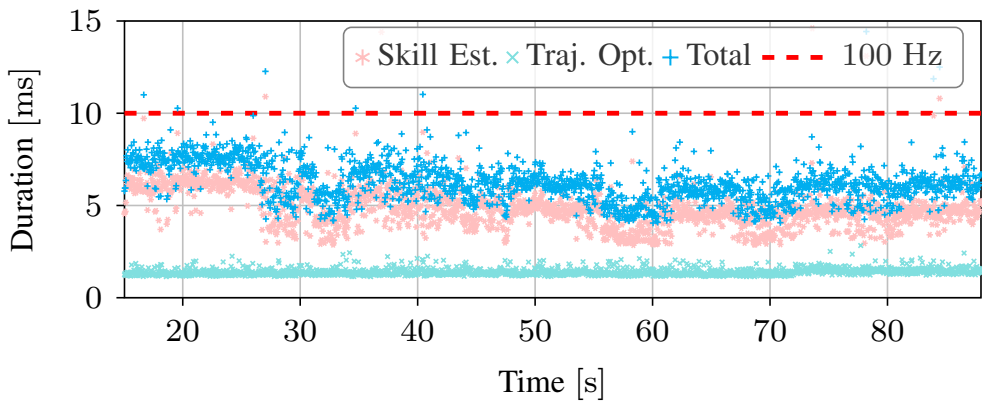


Figure 5.9: Static obstacle avoidance experiment. Computation times for the skill estimation, the trajectory optimization, and the total time.

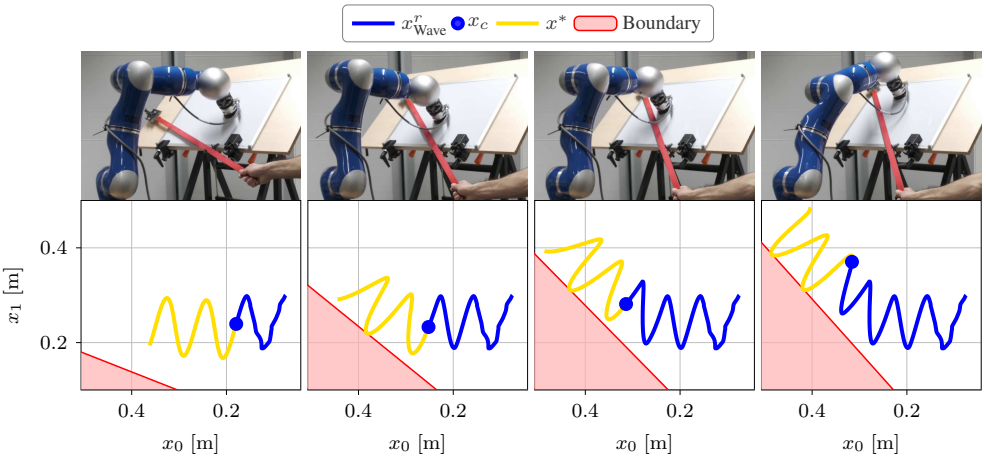


Figure 5.10: Dynamic obstacle avoidance experiments at four snapshots in time. The obstacle (red boundary) is sensed using the Vicon Motion Capture system, and represented as parameterized constraints in Equation 5.22. Note, motion in the plots are *right to left*.

5.13.2 *Participant protocol*

Participants were tasked with controlling the robot to perform a wiping task straight along the whiteboard, right to left, for two different skills (wave and cycloid) and in two different modes: direct control (DC) and shared control (SC). After consent was taken, the participant was given a practice run to familiarize themselves with the joystick.

The skills were presented to the user in a fixed order: wave then cycloid. The order in which the modes DC and SC were presented was randomized. The participants were informed they would teleoperate under an assistive (i.e. shared control) and non-assistive (i.e. direct control) modes of control without specifying the sequence. At the start of each skill block, a demonstration of the skill was played on the robot

so they could easily envisage the task they were required to perform. For each of the four trials, the user was asked to perform one unrecorded practice run, then another recorded trial. Participants were able to visually observe the current state of the robot without being given any visual feedback about the robots trail across the board, and thus were unable to monitor the resulting trajectory.

5.13.3 *Measures and analysis*

During each trial, joystick signals h^r, and target positions x^r (in the whiteboard coordinate frame) at a sampling frequency of 50Hz was collected. Note that superscript r refers to the fact that these values are raw signals. Recall from Section 5.7.1, The method relies on a direct control initialization that lasts for a certain window duration t_w before shared control is activated. The data corresponding to this initialization was removed for both the direct control and shared control results so that the comparison between the two modes was fair.

The quality of the joystick and applied control signals is measured by fitting an ideal signal using the model Equation 5.4, i.e. for each trial

$$\begin{aligned} F_h = \frac{1}{T} \min_{\theta_i} \sum_{k=0}^{N} \left\| u\big(s(t_k);\theta_i\big) - h_k^r \right\|^2 \\ \text{subject to } \theta_i \in \Theta_i \end{aligned} \tag{5.24}$$

was computed where T is the duration of the trial used here to normalize the fitting, t_k are time stamps, N is the number of data points collected. Similarly, the quality of the target position signals x^r is measured by computing

$$\begin{aligned} F_x = \frac{1}{T} \min_{\theta_i} \sum_{k=0}^{N} \left\| x\big(s(t_k);\theta_i\big) - x_k^r \right\|^2 \\ \text{subject to } \theta_i \in \Theta_i \end{aligned} \tag{5.25}$$

where $x(\cdot)$ is the model defined by Equation 5.3. In both Equation 5.24 and Equation 5.25 the parameter $s_c = 0$ is set and constrain the variables for the Clothoid curvature ϕ to zero, since the task was to perform a skill around an underlying trajectory with no curvature. All other parameters in θ_i are left unconstrained.

For both F_h and F_x, smaller values indicate more accurate fitting and thus higher performance. Note that even though the shared control method produces trajectories based on the models fitted here, a zero fitting error is *not* expected for the method since the method allows us to adapt to the varying input from the human (i.e. $\widehat{\theta}_i$ is modified online).

An example from two participants for each skill are shown in Figure 5.12. Visually, it is observed that the quality of the direct control signals is far poorer than the shared control.

The results of the computed fittings are shown in Figure 5.12. For the wave skill, it is observed that shared control leads to similar average performance for F_x, whereas the results for the cycloid skill clearly lead to a reduction in the fitting error. For the shared control mode, both the mean and variance of F_h are higher than that of the direct control signals, i.e., the operator's joystick signals become more varied in the shared control mode as opposed to the direct control. This *might* be an indication that the proposed shared control method leads to a reduction in the participants cognitive load - however, it is acknowledged that further study would be required to prove or disprove this hypothesis. Despite more varied joystick signals, the proposed method is able to produce more accurate signals for the cycloid skill. However, there is little change when comparing the average fitting F_x for the wave. This could be an indication that the method has more corrective interventions in scenarios with more complex skills.

5.14 CLOTHOID-BASED TRAJECTORY OPTIMIZATION

Teleoperation systems generally require a control loop to run at frequencies roughly between $50-100$Hz. Two key factors affecting an optimization solver performance for trajectory optimization problems is the number of decision variables, and the quality of the initial guess (i.e. how close the initial guess is to a local optima). In this work, Clothoids provided a compact representation enabling computation times amenable for teleoperation, see in Figure 5.9.

Whilst the method can handle three dimensional surfaces by projecting the two dimensional commands into the space of the surface (e.g., see Figure Figure 5.1), it cannot handle three dimensional tasks. Clothoids have been extended to three dimensions (Girbés et al., 2019) for attitude planning for drones however to the best of the authors knowledge their applicability to motion planning and manipulation tasks has not been explored. Trajectory optimization is a popular technique (Toussaint, 2009; Kalakrishnan et al., 2011; Zucker et al., 2013) that suffers from high computation times, especially in high-*Degrees of Freedom* (DoF) systems.

In this section, the general trajectory optimization problem is re-formulated in three dimensions. The goal here is to provide a starting point for future research on the scalability of clothoids.

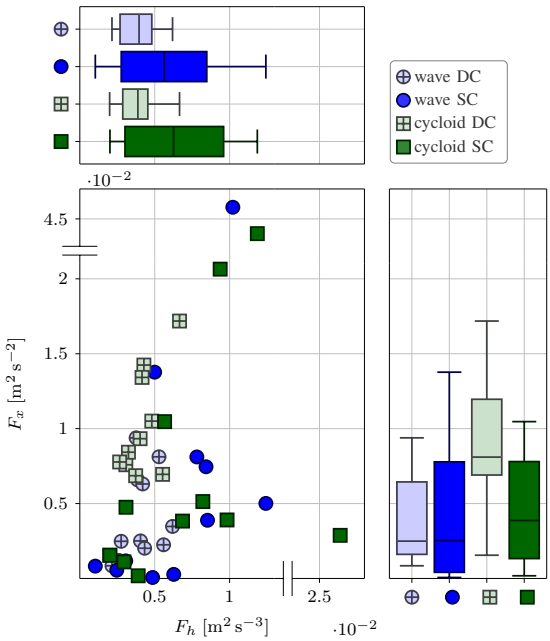


Figure 5.11: Scatter plot comparing fitting error in the input commands and resulting trajectories with the corresponding box plots. Note, the box plots share the same axes as the scatter plot.

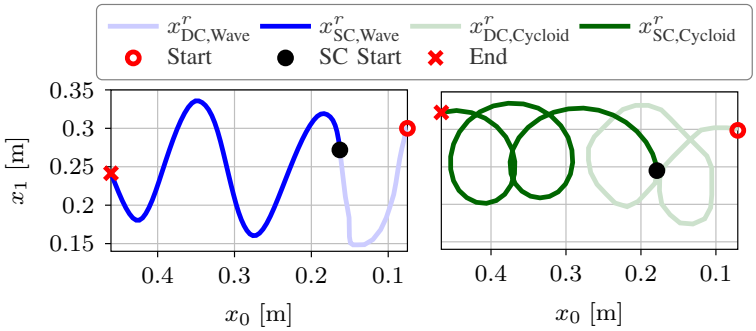


Figure 5.12: Participant shared control examples where the motion is from *right to left*.

Let the system state $x \in \mathbb{X}$ be defined in a three dimensional Cartesian coordinate frame. The clothoid is defined in terms a derivative with respect to a spatial parameter s given by

$$x' = \begin{pmatrix} \sin(\alpha)\cos(\beta) \\ \cos(\alpha)\cos(\beta) \\ \sin(\beta) \end{pmatrix} \tag{5.26}$$

where α and β are orientations that define the unit direction vector. These orientations are both given by linear models defining their gradient

$$\alpha' = \phi_0 + \phi_1 s \tag{5.27a}$$

$$\beta' = \psi_0 + \psi_1 s \tag{5.27b}$$

where $\phi_0, \phi_1, \psi_0, \psi_1$ are parameters. Given an initial position $x(s_0) = x_c$ the trajectory is found for x by integrating Equation 5.26, and, as before, the controls are given by taking the derivative of $x(s)$ with respect to s.

Let us define a trajectory over some horizon $s \in \mathbb{S} = [s_c, s_f]$, then the optimal trajectories are found by solving

$$\phi^*, \psi^* = \arg\min_{\phi,\psi} \text{cost}(\phi, \psi) \quad \text{subject to} \quad \begin{cases} x(s; \phi, \psi) \in \mathbb{X} \\ u(s; \phi, \psi) \in \mathbb{U} \\ x(s_c; \phi, \psi) = x_c \\ s \in \mathbb{S} \end{cases} \tag{5.28}$$

where the cost function is defined in terms of the parameters ϕ, ψ. Let us assume the goal is for the robot to reach a target x_g then this could be enforced as a constraint by setting $x(s_f; \phi, \psi) = x_g$ or as a cost $\text{cost}(\phi, \psi) = \|x(s_f; \phi, \psi) - x_g\|^2$ depending on the application.

As a small demonstration, Equation 5.28 is implemented within the CasADi framework. It is shown in here that it is indeed possible solve such a problem in the face of a cluttered environment. Figure 5.13 shows the trajectory produced by the solver.

5.15 DISCUSSION

Next, a few interesting observations are discussed in depth and future directions are sketched. In the user study (Section 5.13), it was reported verbally by some participants that the speed of the robot changed at certain points of the pattern. Setting the speed of the underlying trajectory in Equation 5.2 as a constant, i.e. $v(t) = v_0$, enables a spatial-temporal coupling between the underlying trajectory and state trajectory that leads to increases and decreases in speed at certain stages of the pattern (e.g., for the wave, at peaks/troughs the speed through the pattern is slow, whereas the speed at the mid points is faster). Whilst this parameter adapts to the users commands, the change in speed through the state trajectory is still perceivable and it is

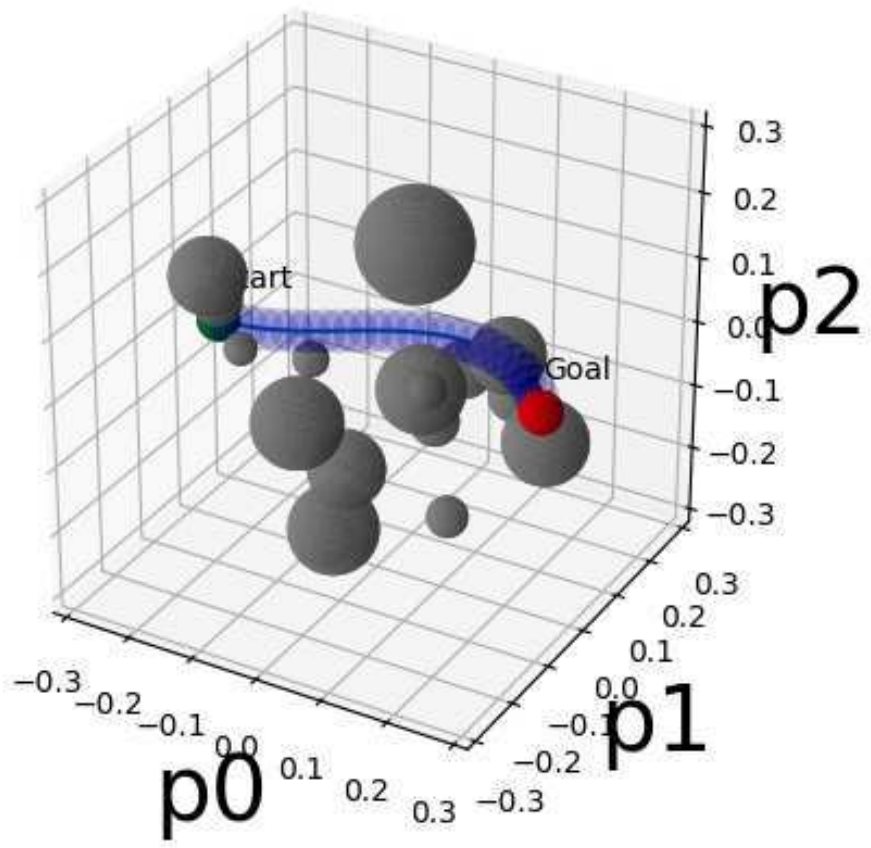


Figure 5.13: Trajectory optimization using a clothoid-based model in a cluttered environment.

hypothesized this may adversely affect user experience. To counter this effect, future work will explore richer models for $v(t)$ that removes this coupling.

The skill switching experiment (Section 5.10) demonstrates the ability of the shared controller to react to an operator's change of intention. Figure 5.7 (right) highlights this change with a red dot. While conducting the experiment, it was observed that there is a slight delay from when the user starts to produce a cycloid to when the controller identifies this switch. The question of whether this leads to user dissatisfaction remains open. Potentially, this misalignment could be minimized by accelerating the skill-identification process. However, the focus in this work was to introduce a novel conceptual framework for skill-based shared control, and thus this analysis and extension is relegated to future work.

The method, in its current formulation, is restricted to a certain window duration t_w that must be enforced for all skill models. This could potentially lead to difficulties when tuning parameters, especially in the face of, for example, a line skill (defined by $\sigma_i = 0, \omega_i = 0$) - a bias may be seen in the skill identification step towards a particular skill as opposed to any other without an adaptive window size. Future work

should investigate the effect of additional skills on the skill estimation and identification steps.

Figure 5.8 (upper) shows how the proposed skill-based shared control method adapts a trajectory to avoid a boundary. The trajectory curves smoothly whilst ensuring the wave skill. Visually, the wavelength of the resulting trajectory seems slightly smaller. This might be due to the regularization term of the skill identification, added in order to avoid over-fitting the operator's commands which have a fairly high variance (Figure 5.7).

Figure 5.8 (lower) shows the solver duration's for the skill identification and trajectory optimization steps. For the majority of cases ($\sim$99%), the total duration stays under 10ms, compatible for online teleoperation. This particular experiment considers only a single skill. Considering more skills will increase the total duration of the skill identification, if performed sequentially. However, the skill identification step for multiple skills are mutually exclusive, and so this issue is avoided by exploiting multi-threading or GPU hardware to perform these computations in parallel.

A relationship between programming by demonstration (Billard et al., 2008) and the work presented in this chapter is identified. In short, programming by demonstration establishes some parameterized representation and fits it to a database of demonstrations, then uses the learned model to imitate or reproduce some desired behavior. The method presented in this chapter, ultimately, performs this inside a control loop. A parameterized representation is established (Equation 5.3), a model is learned (Section 5.7), and a behavior is reproduced (Section 5.8). Each skill can be thought of as a behavioral primitive (Jenkins and Mataric, 2002) representing some periodic motion. The connection identified here suggests the possibility for learning new skills from expert demonstrations. However, direct application of these techniques, e.g. dynamic movement primitives (Schaal, 2006), may be difficult. This is because the learning method would be required to reason about two sets of parameters: those that can change during execution (e.g. amplitude of a wave) and those that do not (e.g. the components of the model that classifies a wave skill from a cycloid skill).

In this chapter, a method for skill-based shared control is presented. The proposed framework estimates parameters that describe the operator intentions and optimizes a predicted trajectory within the parameter space maintaining the principle of minimal intervention. Of interest to industry, if the skill parameters could be fixed for practical purposes, the operator's input could be modeled in the underlying trajectory. In other words, the operator could define the underlying trajectory and simply switch between skill modes using a button on the interface. The proposed optimization framework would still apply commanding the same computational savings how-

ever the need for the skill estimation and identification steps would be avoided. This could be a particular consideration during implementation of this work in a real industry setting.

6

CONCLUSIONS

This chapter concludes the work in this thesis, Section 6.1 gives an overview chapter-by-chapter, Section 6.2 discusses some of the limitations of the presented approaches, Section 6.3 gives some ideas for future work, and Section 6.4 completes the thesis with some final thoughts.

6.1 OVERVIEW

This thesis has studied and developed several novel approaches to shared autonomy for teleoperation in the context of tasks inspired by the requirements and restrictions of the construction sector. This thesis presented a number of user studies that attempt to understand the modes of teleoperation that lead to higher performance for novice operators, and proposes optimization-based approaches to shared autonomy that enable assisted teleoperation in tasks such as spraying, welding, and plastering. Furthermore, several evaluations are presented that highlight the benefits of the proposed methods.

The key ideas introduced by this thesis are summarized as follows.

- Low-dimensional task spaces lead to optimal performance for novice operators.
- Operator assistance can be provided using an attractor-repeller method in target acquisition tasks (e.g., spraying, welding, plastering). In addition, the notion of continuous operator intention recognition can be incorporated into such a system.
- The notion of skill assistance can be incorporated into a shared control system in a computationally efficient manner by decoupling the state trajectory into an underlying trajectory and skill model. Clothoids provide a suitable model that facilitates this efficiency due their compact representation.
- Despite the complexity and nuances in human behavior, for certain tasks (i.e. skills) it is useful to implement model-based methods when continuously re-estimating operator intentions and adapting to dynamic environments.

– There is an indication that an operator can assist trajectory optimization convergence via modest interactive effort.

The ideas explored in this thesis have lead to novel approaches for addressing several problems and limitations in teleoperation. The work presented and its analysis has progressed the state-of-the-art in the following ways.

With regards to intuitiveness of control interfaces, this work has produced new insights for target acquisition tasks previously unexplored in the literature - to the best of the authors knowledge. Many works in the literature explore similar questions regarding the human interface: wheelchairs (Rebsamen et al., 2010), drones (D. Lee et al., 2013), autonomous vehicles (Habibovic et al., 2017), mobile robots (Luo et al., 2020), robot surgery (Yamamoto et al., 2012), and pick-and-place (Leeper et al., 2012). Some of these works make use of additional or alternate sensors to infer the state of the human, e.g. brain-computer interface (Rebsamen et al., 2010), and EMG sensors (Luo et al., 2020). This allows a wider range of internals states for the human to be inferred. Equipping the operator with additional sensors could be useful for improving the effectiveness of inferring their intentions and measuring cognitive load, however it could be equally destructive towards their situational awareness. In a similar way that objective surgical assessment techniques must resemble realistic surgical settings (Darzi et al., 1999), the operator in construction settings should not be burdened by additional hardware that may adversely impact the way they interact with the teleoperation task. In addition, as teleoperation in industry become more autonomous, studies such as (Habibovic et al., 2017) indicate that users may indeed want to have the ability to influence motion plans during execution.

A thread of work in shared control utilizes policy blending (A. Dragan and Srinivasa, 2013) and can be thought of as a community standard in shared autonomy research. The work presented in Chapter 4 aligns with this approach in that two control laws (human and autonomy) are combined through a direct summation in the task space (i.e. blending). The approach proposed here, however, goes beyond this by introducing alternate models to handle dynamic changes in the environment efficiently for online teleoperation. Furthermore, the ability to incorporate adaptation to changing goals through intention prediction is preserved.

Handling changing environments explicitly in the control framework in shared autonomy is relatively unexplored. Two works address this problem directly, i.e. an optimization approach (Rubagotti et al., 2019) and a sampling-based approach (Broad et al., 2019). Both these enable *Model Predictive Control* (MPC), by simplifying the model for the future human inputs, and environment. These methods use straight line models for the future human inputs and are thus unable to capture complex rhythmic

motions as presented in this thesis. The Clothoid model is compact enough to enable adaption to changes in operator intention and the environment.

The rest of this section makes concluding remarks on each chapter individually.

ALTERNATE MODES OF TELEOPERATION Chapter 3 describes several studies to determine for which modes of teleoperation elicit high task performance for unskilled human operators on a task inspired by concrete spraying in industry. A Fitts' law paradigm was used to quantify difficulty and performance of each mode. Fitts' law was generalized for two additional performance metrics and an extensive data set was collected from the conducted experiments. The results and analysis performed support several conclusions regarding teleoperation, sub-task allocation between human and autonomy, and how habitual traits can effect performance.

A *Reduced Task* (RT) mode has been shown to outperform all other modes of teleoperation with regards to the performance metrics; time, angular length, and delta length. The results of the questionnaire support this conclusion as the RT mode was generally favored the highest. The RT mode is the closest model that directly regulates the performance parameters. Given that the stated hypotheses were accepted this opens new avenues for shared control design. Based on these results, the author posits there exists a positive correlation between cognitive load and number of human controllable dimensions for task space control modes. Both joint space modes were not favored by participants in this study with some noting that the *Reduced Joint* (RJ) mode felt like having the "*worst of both worlds*" with regards to joint space and task space control. This study indicates the mode specification highly impacts the design and performance of shared control systems.

The ability of the participants classed as gamers to achieve higher performance than non-gamers may be a consequence of their familiarity with the game pad controller used as the interface in our experiments. Future investigations will take into account the effect the interface used by comparing other interfaces such as a joystick, 6-DoF space mouse, and a combination of a computer mouse and keyboard. Metrics quantifying cognitive load (e.g. NASA TLX) will be explored and compared against the number of controllable dimensions in-order to investigate the proposition of a positive correlation between the two quantities.

HUMAN INPUT MODULATION FOR SHARED CONTROL Chapter 4 presented an optimization based method that blends human and autonomous reasoning within the same task space. Computational tractability is achieved by modulating the human input with respect to changes in the environment rather than including these changes in the optimization itself by means of constraints. Two realizations of the method are

described for two different case studies and the respective experiments. The experiments illustrate the effectiveness of the method and demonstrate its relevance to real world applications. The analysis presented show that modulating the human input, as opposed to representing task objectives in the constraints of the optimization formulation, results in large computational savings.

SKILL-BASED SHARED CONTROL Chapter 5 introduced a novel framework for skill-based shared control. Evidence was presented that demonstrates that the system is able to identify different skills, react to changes in operator intentions, and re-plan whilst accounting for changing environmental constraints. The framework follows from a novel representation for state and control that exploits a well-established geometrical primitive, i.e. Clothoids, and a parametric model of skills. It was shown that the method is computationally efficient for online shared control within an MPC framework, that enables its adaptability to dynamically changing constraints. Furthermore, the method was validated in a lab mock-up on a KUKA LWR and showed its capability to identify skill switches. The user study provides some early evidence that this approach to shared control leads to improved quality of output motions, while potentially reducing the operators cognitive load.

6.2 LIMITATIONS

In this thesis, a number of novel contributions have been presented and analyzed. However, the work is limited in a several ways.

The user studies presented in Chapter 3 explore task performance under the assumption that the participant is a novice with minimal training. We do not take into account the effect of training. In the context of the thesis we make the assumption that the user is unskilled, meaning our investigation is relevant to our subsequent design choices. However, further investigations should explore the effect of training. The formalism as presented is restricted to shared control approaches where the human and autonomy occupy the same control dimensions; alternative methods exist, however our setup was inspired by the current state-of-the-art in concrete spraying technologies (Girmscheid and Moser, 2001). Our data was recorded with a limited number of participants in a lab setting that may have introduced bias in the way they interact with the system. The results of Chapter 3 should be interpreted as useful trends to inform future design choices for teleoperation systems, rather than indisputable facts.

In Chapter 4, an input modulation method was presented that adjusted local inputs in response to dynamic environments. However, our method does not take into

account of potential future events. The global path planning is the responsibility of the operator.

Chapter 5 presented a skill-based shared control method. The method is limited in a number of ways. The method, as presented, relies on a given analytical skill model - ideally we should be able to learn skills from expert demonstrations. In addition, whilst the system can certainly include as many skills as necessary, similar skills (e.g. wave and zig-zag) could lead to difficulties in the skill identification step.

6.3 FUTURE WORK

In this section, potential future work following his thesis are described. Some of the work comes directly as extensions, and additional analysis from the chapters in this document. Others ideas are capabilities and potential future methods inspired by the work in this thesis.

HUMAN FACTORS IN TARGET ACQUISITION TASKS Chapter 3 investigates how the choice of dimensionality and control space, i.e. the mode of teleoperation, has on overall performance. Additionally, it was observed that habitual traits like playing computer games influenced performance. Participants in our study were novices and were only given enough training to ensure that they could complete the task.

From a human factors point of view, it would be curious if further studies examining the effect of training suggested an alternative design choices. However, from a more practical perspective - such as a regulator - knowing the amount of training required for users to consistently achieve high performance is a key factor in order to determine the monetary cost to certify an operator as an expert.

Undoubtedly, no matter how verifiable the safety of a shared control system, there will always be an element of training required by industrial regulators. Further to the experiments conducted in this thesis an analysis of the learning curve, i.e. the change in performance over successive training sessions, is a key next step in the extension of this work. Other factors that could be considered are the interface design, and kinematic structure of the robot.

In particular, the interface design is often overlooked in industry, see Figure Figure 2.3. In talks with industry experts from the tunnel construction community, it was noted that workers often complain about the un-intuitiveness of these controller pads. Among other factors, interface design has been listed among the most important factors in teleoperation systems (Chen et al., 2007; Murphy, 2015). Yanco et al., 2015 noted "*minimal research has been dedicated to the development of scientifically validated control interfaces*."

LEARNING SKILL REPRESENTATIONS FROM EXPERT DEMONSTRATIONS A key contribution of the skill-based shared control framework presented in Chapter 5 is the notion of skill assistance within a teleoperation system. The proposed model for state and controls rely on a given analytical skill representation - Equation 5.3. Based on industry examples (Ballou, 2003; Welton, 2014; Silva Pereira et al., 2019), several skill models were derived in Section 5.5 and illustrated in Figure 5.4. Future work should explore how could such a skill model $S(s;\rho)$ be *learned* from expert demonstrations. Learning rhythmic skills from demonstration has been addressed in the literature (Ernesti et al., 2012; Armesto et al., 2018; Kulak et al., 2020). However, to the best of the authors knowledge, this idea has not been applied to shared control.

SCALABILITY OF SKILL-BASED SHARED CONTROL The skill-based shared control method proposed in Chapter 5 presented experiments for wiping across a flat surface. By leveraging a force-torque sensor mounted at the wrist the methods extends to three dimensional surfaces – see Figure 5.1. However, this still raises the following questions: i) how can the system scale to additional dimensions, for example to include 3D motion and end-effector rotations? and ii) can Clothoids handle such additional dimensions or are they restricted to representing motions on a plane?

As presented, the method leverages clothoids in 2D inspired by the aforementioned use-cases (welding, spraying, cleaning), that allows testing and validation of the proposed novel concept of a task-based shared control framework. However, it would be straightforward to extend the method to 3D case-scenarios, for instance by using a 3D clothoid, as used in (Girbés et al., 2019). For scenarios including end-effector rotations, there is a possible limitation of using Clothoids. For such cases, further research is likely needed to find another parameterization for the underlying trajectory that appropriately handles orientation.

In addition, there are also questions regarding the scalability of the method to more than two skills. That is, would the method would succeed if there were more options to choose from and some skills were similar to each other?

The framework has no specific limit on the number of skills, and since the estimation step can be implemented in parallel, solve duration is does not increase with more skills. However, adding skills can pose a problem on the reliability of the skill identification. Therefore, when dealing with larger number of skills and/or similar skills we may require more sophisticated estimation methods. Nevertheless, our framework allows us to plug-in different estimation/identification methods, so long as the computation time remains within acceptable limits for online teleoperation.

MINIMIZING OPERATOR INTERACTION Supervisory control is a branch of teleoperation that allows a supervisor (operator) to give high-level commands and receive summary information from the system (Ferrell and Sheridan, 1967; Sheridan, 1992; Niemeyer et al., 2016). Recent examples of supervisory control system have proved effective in completing complex tasks, e.g., (Marion et al., 2017; Quere et al., 2020) and the system described in Appendix A. However, these tend to rely on the system having access to a library of possible actions/templates and fast access methods (Merkt, Ivan, et al., 2018).

A human operator tends to be able to quickly assess uncertainty in the task and environment and internally represents that knowledge in a manner, currently, inaccessible to the system. However, the author submits that some simple gestures via some interface (e.g., joystick, haptic interface, virtual reality system) could transmit this information to the planner. For this idea, inspiration was taken from the graphics literature. Rother et al., 2004 proposes the GrabCut method used to extract subjects from the foreground of an image. In most cases, the user need only provide a bounding box around the subject of interest. Of interest in this case, is that it does not require any learned feature extraction in order to perform well; their segmentation algorithm is based on iterative energy minimization. The author envisages a similar case where the operator provides some simple gestures and the system translates this to a high dimensional motion plan and perhaps involving discrete and continuous planning problems (Toussaint, 2015).

RISK-AWARE SHARED AUTONOMY Uncertainty in the environment and sometimes the task is an inherent problem pervasive in all robotics applications. In particular, estimating risk over several possible future events and predicting likely outcomes given the current system state is an important problem to mitigate failures and potentially disasters. Partially observable Markov decision processes have had very successful results when applied to planning (Hanna Kurniawati, 2008), and attempts have been made to apply the formulation to shared autonomy (Javdani et al., 2018). Continuously re-evaluating the state of risk and adapting control strategies that account for deficient future predictions is still, to the best of the authors knowledge, unexplored.

ADDITIONAL PROMISING DIRECTIONS Finally, there are several additional directions worth noting.

It is very typical, and indeed essential, for shared autonomy literature to report results for user studies to validate and give confidence in their proposed approaches – several user studies were performed in support of this thesis. However, this can be very time/resource consuming and in the wake of the recent COVID-19 global

pandemic, restrictions can mean these experiments are not allowed to happen at all. This motivates the development of automatic assessment techniques of shared autonomous approaches (Fontaine and Nikolaidis, 2021).

Typically cost terms in an optimization representation for planning motion are chosen to maximize task efficiency in some respect (e.g., minimize joint torques), and the motion of the robot may not be intuitive to the user and could even lead to confusion. Finding controls that account for the legibility of robot motion is therefore an important quality in for shared autonomous systems (A. D. Dragan et al., 2013).

6.4 EPILOGUE

This thesis has proposed frameworks for shared autonomy that have been shown to improve performance in several realistic lab and simulated mock-ups for industrial tasks. Due to the COVID-19 pandemic, the way in which people perform tasks at work will undoubtedly change, requiring remotely operated robotic solutions. The author envisages teleoperation and shared autonomy in-particular to be key technologies in the coming years. It is the author's hope that the presented frameworks, analysis, and demonstrations herein may bring us, even a small step, towards these goals.

APPENDIX

A

ROBUST SLIDING AUTONOMY

This appendix describes an integration project for a shared autonomous system implementing sliding autonomy. Sliding autonomy can be thought of a type of shared autonomy such that the system has the ability to switch between different levels of autonomy from direct control to fully autonomous sequences. The project was lead by Wolfgang Merkt and presented in (Merkt, Y. Yang, et al., 2017) and was achieved in collaboration with several contributors, including the author of this thesis.

A.1 INTRODUCTION

Utilizing robots in extreme applications (e.g. disaster recovery, space and maritime) requires sequencing several actions (e.g. move to a goal, pick object from table, pull lever, etc). Autonomous robots in such environments do not yet have the capability to be effective, provably safe, and re-plan sequences effectively to unpredictable changes. As such, a human-in-the-loop is required.

In this work we present a shared autonomous system that builds on top of the DIRECTOR interface (Marion et al., 2017) originally used to pilot the ATLAS robot by Team MIT at DARPA Robotics Challenge. This system is fully integrated with environment mapping, autonomous pose selection and navigation, model-free object segmentation, automatic grasp affordance selection, collision-free motion planning and execution, continuous scene monitoring, and failure recovery.

A.2 SYSTEM OVERVIEW

The core components of the shared autonomous system are described: perception, scene monitoring, user interface, and motion planning.

An OctoMap (Hornung et al., 2013) representation is used to enable efficient collision avoidance, change detection, tracking, and mapping. In order to segment objects of interest without a prior model, a combination of geometric insights and task informed assumptions are used in an algorithm similar to (Rusu et al., 2009). First, it is assumed objects of interest are placed on a horizontal surface and spaced a sufficient distance apart given some tolerances. These assumptions allow us to segment continuous planes (e.g., a table top) by surface normals that then informs a Euclidean

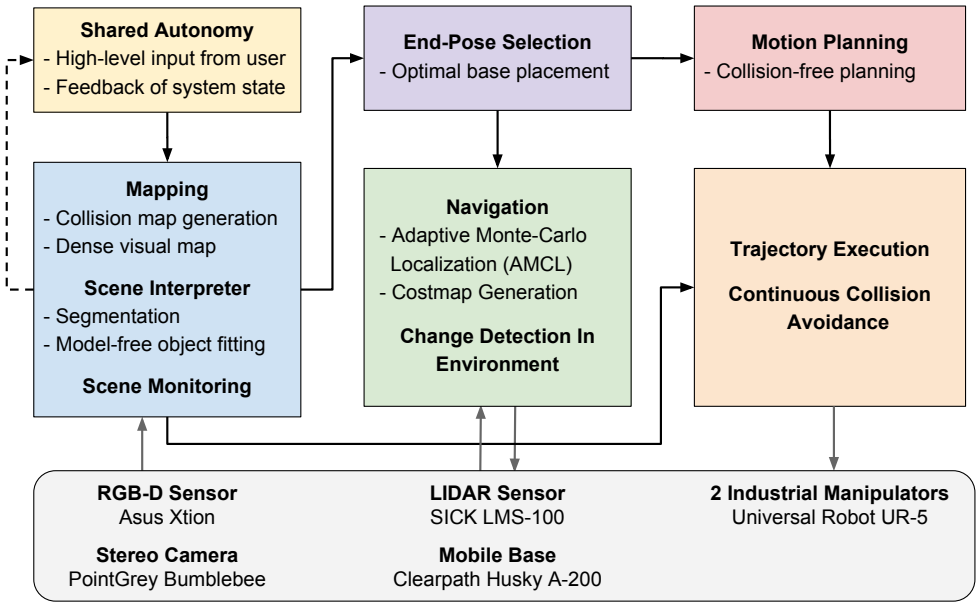


Figure A.1: Overview of the system architecture.

clustering process to extract distinct objects that can be used to generate coordinate frames for grasping. The initial seed for the algorithm is given by the human through a GUI interface.

For an autonomous device working in a shared workspace with humans, continuous scene monitoring is essential. Such a system must have the ability to flag changes in the environment that would impede or otherwise cause damage to the robot itself or a human. The continually updating discretized occupancy map is at the core of the system to track changes. If a collision of a future key pose with the map update is detected, execution is halted and the system falls back to the human operator - a feature of sliding autonomy. The scene is continually monitored so that motion execution may be resumed once the trajectory would be collision free again.

The core of the sliding autonomy system comes via a user interface that serves as an abstraction layer above task complexity allowing the human to provide high-level task objectives and provides feedback. The interface, which can be seen in Figure A.2, builds on top of the task execution system described by Marion et al., 2017 allowing the user to pass high-level objectives to the system as task primitives that can be adapted by online operator-assisted perception. In this system, this is extended by automatic review and approval of planned motions through continuous scene monitoring and reasoning to reduce the amount of human intervention required, and only fall back to the operator when absolutely necessary.

The system implemented employs a combination of sampling- and optimization-based planning algorithms utilizing *Extensible Optimization Toolkit* (EXOTica) to find

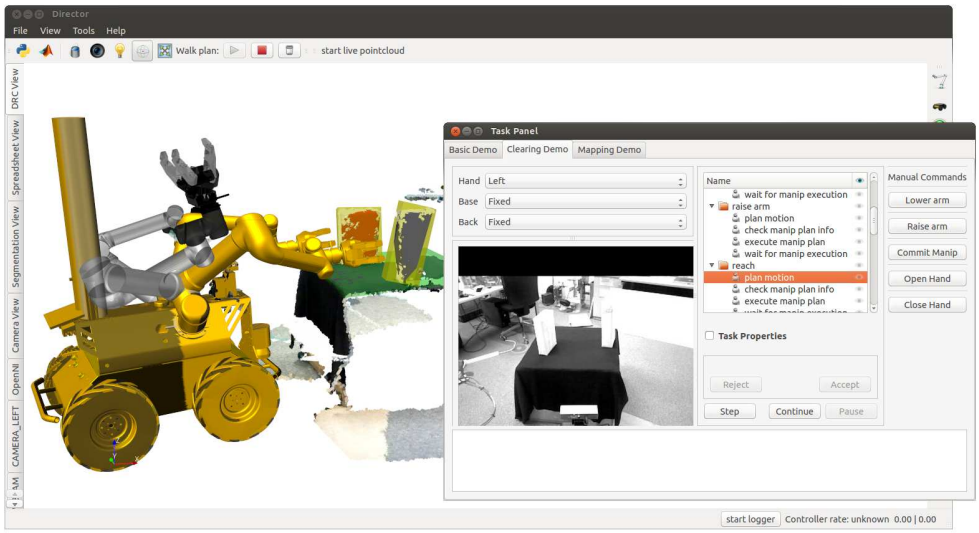


Figure A.2: The user interface for the bi-manual mobile manipulator.

collision-free motion trajectories (Ivan, Y. Yang, et al., 2019). Given an environment, reachability map, and manipulation constraints of an affordance, the goal pose is found via optimization-based inverse kinematics. Given the start and end goal configurations we employ RRT-Connect (Kuffner and LaValle, 2000) to compute a trajectory.

A.3 DEPLOYMENT

The system as described in the previous section is generic and applied to two platforms, a fixed-based manipulator arm and a mobile-base dual arm manipulator, each equipped with the required sensory devices.

To improve manipulability of the task, the position of the mobile base was included in the end-pose selection. Maximizing manipulability is important to cope with dynamic changes and perception inaccuracies to an extent as it maximizes the likelihood that nearby task space can be reached with the same base placement. The chosen end-pose selection method is iDRM (Y. Yang, Ivan, Z. Li, et al., 2016). The iDRM method consists of two different stages: an offline processing stage and an online updating stage. During the offline processing stage, a large number of self-collision-free configurations are generated and transformed such that the end-effector is at the origin of a voxelgrid. A filtering stage is completed to remove unsatisfactory samples, based on given constraints Y. Yang, Ivan, Z. Li, et al., 2016, Sec. IV (e.g., remove samples where the end-effector is below the floor). During the online stage, queries return an end pose that satisfies collision-free and maximal manipulability constraints.

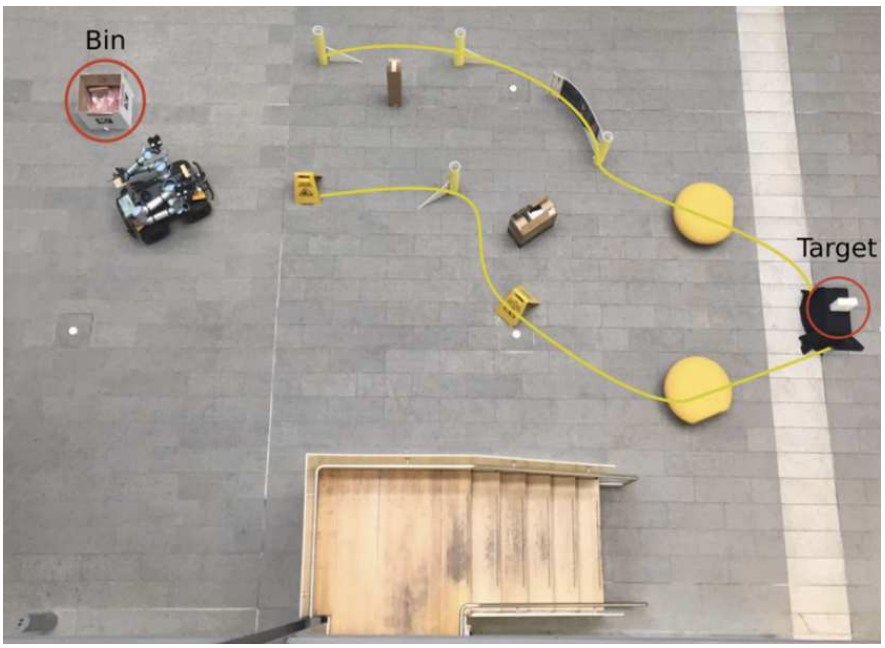


Figure A.3: Obstacle course completed by robot utilizing our system.

We used a LIDAR sensor for navigation and localization against a pre-collected static map; the method of choice was *Adaptive Monte-Carlo Localization* (Fox et al., 1999). Goal base positions are computed using iDRM and passed to the ROS navigation stack which provides cost-map generation and path planning out-of-the-box.

DEMONSTRATIONS OUTSIDE THE LAB In June 2017 the University of Leeds hosted a two day robotics challenge. Our team demonstrated the system described above. The challenge was to perform the following steps through an obstacle course shown in Figure A.3.

1. Navigate through a passage way.
2. Remove objects that impede progress using the arms.
3. Find a table with an object to pick up.
4. Segment the table in a three dimensional point cloud.
5. Plan to a pre-grasp frame, then to a grasp frame, then grasp the object.
6. Move arm back to nominal position.
7. Navigate to and drop item into bin.

Our team won first prize for "Greatest Potential For Positive Impact" for the submission.

BIBLIOGRAPHY

Abi-Farraj, F., T. Osa, et al. (May 2017). "A learning-based shared control architecture for interactive task execution." In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 329–335. DOI: 10.1109/ICRA.2017.7989042 (Cited on pages 19, 22, 53, 76).

Abi-Farraj, F., N. Pedemonte, and P. R. Giordano (Oct. 2016). "A visual-based shared control architecture for remote telemanipulation." In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4266–4273. DOI: 10.1109/IROS.2016.7759628 (Cited on pages 25, 26, 53, 57).

Alonso, V. and P. de la Puente (Nov. 2018). "System Transparency in Shared Autonomy: A Mini Review." In: *Frontiers in Neurorobotics* 12, p. 83. ISSN: 1662-5218. DOI: 10.3389/fnbot.2018.00083 (Cited on page 23).

Andersson, J. A. et al. (July 2019). "CasADi: a software framework for nonlinear optimization and optimal control." In: *Mathematical Programming Computation* 11.1, pp. 1–36. DOI: 10.1007/s12532-018-0139-4 (Cited on page 90).

Argall, B. D. et al. (2009). "A survey of robot learning from demonstration." In: *Robotics and Autonomous Systems* 57.5, pp. 469–483. ISSN: 0921-8890. DOI: 10.1016/j.robot.2008.10.024 (Cited on page 22).

Armesto, L. et al. (July 2018). "Constraint-aware learning of policies by demonstration." In: *The International Journal of Robotics Research* 37.13-14, pp. 1673–1689. DOI: 10.1177/0278364918784354 (Cited on page 108).

Bajcsy, A. et al. (Nov. 2017). "Learning Robot Objectives from Physical Human Interaction." In: *Proceedings of the 1st Annual Conference on Robot Learning*. Ed. by S. Levine, V. Vanhoucke, and K. Goldberg. Vol. 78. Proceedings of Machine Learning Research. PMLR, pp. 217–226. URL: http://proceedings.mlr.press/v78/bajcsy17a.html (Cited on page 11).

Ballou, M. (2003). "Shotcrete Rebound – How Much is Enough?" In: *Shotcrete magazine, American Shotcrete Association* (Cited on pages 13, 25, 33, 58, 67, 69, 75, 82, 83, 108).

Bard, J. D. et al. (2015). "Decorative Robotic Plastering: A Case Study of Real-Time Human Machine-Collaboration in High-Skill Domains." In: *Proceedings of the 33rd eCAADe Conference - Volume 2, Vienna University of Technology, Vienna, Austria*. URL: http://papers.cumincad.org/cgi-bin/works/BrowseTree=series:acadia+year:2011/Show?ecaade2015_74 (Cited on page 75).

Beer, J. M., A. D. Fisk, and W. A. Rogers (July 2014). "Toward a Framework for Levels of Robot Autonomy in Human-Robot Interaction." In: *J. Hum.-Robot Interact.* 3.2, pp. 74–99. DOI: 10.5898/JHRI.3.2.Beer (Cited on page 12).

Bejczy, A. K. (Jan. 1994). "Toward advanced teleoperation in space." In: *Teleoperation and Robotics in Space*. Vol. 161. AMERICAN INST OF AERONAUTICS and ASTRONAUTICS, pp. 107–107. DOI: 10.2514/5.9781600866333.0107.0138 (Cited on page 9).

Bergamasco, M. et al. (1994). "An arm exoskeleton system for teleoperation and virtual environments applications." In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE Comput. Soc. Press, 1449–1454 vol.2. DOI: 10.1109/ROBOT.1994.351286 (Cited on page 18).

Billard, A. et al. (2008). "Robot Programming by Demonstration." In: *Springer Handbook of Robotics*. Ed. by B. Siciliano and O. Khatib. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1371–1394. ISBN: 978-3-540-30301-5. DOI: 10.1007/978-3-540-30301-5_60 (Cited on page 100).

Boksem, M. A., T. F. Meijman, and M. M. Lorist (Sept. 2005). "Effects of mental fatigue on attention: An ERP study." In: *Cognitive Brain Research* 25.1, pp. 107–116. DOI: 10.1016/j.cogbrainres.2005.04.011 (Cited on page 17).

Brantner, G. and O. Khatib (June 2021). "Controlling Ocean One: Human–robot collaboration for deep-sea manipulation." In: *Journal of Field Robotics* 38.1, pp. 28–51. DOI: 10.1002/rob.21960 (Cited on page 9).

Brezak, M. and I. Petrović (Apr. 2014). "Real-time Approximation of Clothoids With Bounded Error for Path Planning Applications." In: *IEEE Transactions on Robotics* 30.2, pp. 507–515. DOI: 10.1109/TRO.2013.2283928 (Cited on page 86).

Broad, A., T. Murphey, and B. Argall (June 2019). "Highly Parallelized Data-Driven MPC for Minimal Intervention Shared Control." In: *Proceedings of Robotics: Science and Systems (RSS)*. FreiburgimBreisgau, Germany: Robotics: Science and Systems Foundation. DOI: 10.15607/RSS.2019.XV.008 (Cited on pages 23, 76, 77, 104).

Burridge, R. R. and K. A. Hambuchen (Oct. 2009). "Using prediction to enhance remote robot supervision across time delay." In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 5628–5634. DOI: 10.1109/IROS.2009.5354233 (Cited on page 19).

Campeau-Lecours, A., U. Côté-Allard, et al. (Apr. 2019). "Intuitive Adaptive Orientation Control for Enhanced Human–Robot Interaction." In: *IEEE Transactions on Robotics* 35.2, pp. 509–520. DOI: 10.1109/TRO.2018.2885464 (Cited on page 18).

Campeau-Lecours, A., H. Lamontagne, et al. (July 2017). "Kinova Modular Robot Arms for Service Robotics Applications." In: *Int. J. Robot. Appl. Technol.* 5.2, pp. 49–71. ISSN: 2166-7195. DOI: 10.4018/IJRAT.2017070104 (Cited on page 10).

Campeau-Lecours, A., V. Maheu, et al. (2016). "Jaco assistive robotic device: Empowering people with disabilities through innovative algorithms." In: (Cited on page 28).

Carlson, T. et al. (Oct. 2012). "The birth of the brain-controlled wheelchair." In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 5444–5445. DOI: 10.1109/IROS.2012.6386299 (Cited on page 27).

Chen, J. Y. C., E. C. Haas, and M. J. Barnes (Nov. 2007). "Human Performance Issues and User Interface Design for Teleoperated Robots." In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6, pp. 1231–1245. DOI: 10.1109/TSMCC.2007.905819 (Cited on page 107).

Chiou, E. (2017). "Flexible robotic control via co-operation between an operator and an ai-based control system." PhD thesis. University of Birmingham (Cited on page 12).

Collinger, J. L. et al. (Feb. 2013). "High-performance neuroprosthetic control by an individual with tetraplegia." In: *The Lancet* 381.9866, pp. 557–564. ISSN: 0140-6736. DOI: 10.1016/S0140-6736(12)61816-9 (Cited on page 18).

Crandall, J. W. and M. A. Goodrich (2002). "Characterizing efficiency of human robot interaction: a case study of shared-control teleoperation." In: *IEEE/RSJ IROS*. IEEE, pp. 1290–1295. DOI: 10.1109/IRDS.2002.1043932 (Cited on page 19).

Crossman, E. R. F. W. and P. J. Goodeve (May 1983). "Feedback Control of Hand-Movement and Fitts' Law." In: *The Quarterly Journal of Experimental Psychology Section A* 35.2, pp. 251–278. DOI: 10.1080/14640748308402133 (Cited on pages 42, 43).

Cummings, M., C. Mastracchio, et al. (Jan. 2013). "Boredom and Distraction in Multiple Unmanned Vehicle Supervisory Control." In: *Interacting with Computers* 25.1, pp. 34–47. DOI: 10.1093/iwc/iws011 (Cited on page 23).

Cummings, M. and J. Ryan (2014). *Shared authority concerns in automated driving applications*. Tech. rep. MIT Humans and Autonmation Labrartory (HAL) Reports (Cited on page 23).

Darzi, A., S. Smith, and N. Taffinder (Apr. 1999). "Assessing operative skill. Needs to become more objective." In: *BMJ* 318.7188, pp. 887–888 (Cited on page 104).

Dias, M. B. et al. (2008). "Sliding autonomy for peer-to-peer human-robot teams." In: *Proceedings of the international conference on intelligent autonomous systems*, pp. 332–341 (Cited on pages 12, 26).

Douglas, S. A., A. E. Kirkpatrick, and I. S. MacKenzie (1999). "Testing Pointing Device Performance and User Assessment with the ISO 9241, Part 9 Standard." In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. CHI '99. Pittsburgh, Pennsylvania, USA: ACM, pp. 215–222. DOI: 10.1145/302979.303042 (Cited on page 43).

Dragan, A. and S. Srinivasa (June 2013). "A policy-blending formalism for shared control." In: *The International Journal of Robotics Research* 32.7, pp. 790–805. DOI: 10.1177/0278364913490324 (Cited on pages 11, 19–23, 34, 50, 53, 57, 58, 76, 87, 104).

Dragan, A. D., K. C. Lee, and S. S. Srinivasa (Mar. 2013). "Legibility and predictability of robot motion." In: *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, pp. 301–308. DOI: 10.1109/HRI.2013.6483603 (Cited on pages 23, 110).

Driessen, B., H. Evers, and J. Woerden (Mar. 2001). "MANUS—a wheelchair-mounted rehabilitation robot." In: *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 215.3, pp. 285–290. DOI: 10.1243/0954411011535876 (Cited on page 18).

Enayati, N., G. Ferrigno, and E. De Momi (Nov. 2018). "Skill-based human–robot cooperation in tele-operated path tracking." In: *Autonomous Robots* 42.5, pp. 997–1009. DOI: 10.1007/s10514-017-9675-4 (Cited on page 76).

Erden, M. S. and A. Billard (Feb. 2015). "Hand Impedance Measurements During Interactive Manual Welding With a Robot." In: *IEEE Transactions on Robotics* (*T-RO*) 31.1, pp. 168–179. DOI: 10.1109/TRO.2014.2385212 (Cited on pages 9, 76).

Ernesti, J. et al. (2012). "Encoding of periodic and their transient motions by a single dynamic movement primitive." In: *2012 12th IEEE-RAS International Conference on Humanoid Robots* (*Humanoids 2012*), pp. 57–64. DOI: 10.1109/HUMANOIDS.2012.6651499 (Cited on page 108).

Ferreau, H. J. et al. (Dec. 2014). "qpOASES: a parametric active-set algorithm for quadratic programming." In: *Mathematical Programming Computation* 6.4, pp. 327–363. ISSN: 1867-2957. DOI: 10.1007/s12532-014-0071-1 (Cited on page 11).

Ferrell, W. R. and T. B. Sheridan (Oct. 1967). "Supervisory control of remote manipulation." In: *IEEE Spectrum* 4.10, pp. 81–88. DOI: 10.1109/MSPEC.1967.5217126 (Cited on page 109).

Fitts, P. M. (1954). "The information capacity of the human motor system in controlling the amplitude of movement." In: *Journal of Experimental Psychology* 47.6, pp. 381–391. DOI: 10.1037/h0055392 (Cited on pages 42, 43).

Fontaine, M. and S. Nikolaidis (July 2021). "A Quality Diversity Approach to Automatically Generating Human-Robot Interaction Scenarios in Shared Autonomy." In: *Proceedings of Robotics: Science and Systems*. Virtual. DOI: 10.15607/RSS.2021.XVII.036 (Cited on page 110).

Fox, D. et al. (1999). "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots." In: *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*. AAAI '99/IAAI '99. Orlando, Florida, USA: American Association for Artificial Intelligence, pp. 343–349. ISBN: 0262511061 (Cited on page 116).

Frisoli, A. et al. (2005). "A new force-feedback arm exoskeleton for haptic interaction in virtual environments." In: *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. IEEE, pp. 195–201. DOI: 10.1109/WHC.2005.15 (Cited on page 18).

Frohm, J. et al. (Jan. 2008). "Levels of Automation in Manufacturing." In: *Ergonomia - International Journal of Ergonomics and Human Factors* 30, pp. 181–207 (Cited on page 12).

Galán, F. et al. (Sept. 2008). "A brain-actuated wheelchair: Asynchronous and non-invasive Brain–computer interfaces for continuous control of robots." In: *Clinical*

Neurophysiology 119.9, pp. 2159–2169. ISSN: 1388-2457. DOI: 10.1016/j.clinph.2008.06.001 (Cited on page 18).

Gancet, J. et al. (2016). "Dexterous Undersea Interventions with Far Distance Onshore Supervision: the DexROV Project." In: *IFAC-PapersOnLine* 49.23. 10th IFAC Conference on Control Applications in Marine SystemsCAMS 2016, pp. 414–419. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2016.10.439 (Cited on page 18).

Gill, P. E., W. Murray, and M. A. Saunders (Apr. 2002). "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization." In: *SIAM J. on Optimization* 12.4, pp. 979–1006. ISSN: 1052-6234. DOI: 10.1137/S1052623499350013 (Cited on pages 11, 64, 90).

Girbés, V., G. Vanegas, and. L. Armesto (Aug. 2019). "Clothoid-Based Three-Dimensional Curve for Attitude Planning." In: *Journal of Guidance, Control, and Dynamics* 42.8, pp. 1886–1898. DOI: 10.2514/1.G003551 (Cited on pages 86, 96, 108).

Girmscheid, G. and S. Moser (May 2001). "Fully Automated Shotcrete Robot for Rock Support." In: *Comput.-Aided Civ. Infrastruct. Eng.* 16.3, pp. 200–215. DOI: 10.1111/0885-9507.00226 (Cited on pages 9, 14, 16, 18, 35, 106).

Goertz, R. C. (1954). "Mechanical master-slave manipulator." In: *Nucleonics (US) Ceased publication* 12 (Cited on pages 9, 18).

Goertz, R. C. (1952). "Fundamentals of general-purpose remote manipulators." In: *Nucleonics* 10.11, pp. 36–42 (Cited on pages 9, 18).

Gong, L. et al. (Aug. 2017). "Real-time human-in-the-loop remote control for a life-size traffic police robot with multiple augmented reality aided display terminals." In: *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, pp. 420–425. DOI: 10.1109/ICARM.2017.8273199 (Cited on page 9).

Gonthina, P. S. et al. (May 2019). "Modeling Variable Curvature Parallel Continuum Robots Using Euler Curves." In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEE. DOI: 10.1109/ICRA.2019.8794238 (Cited on page 86).

Goodrich, M. and D. Olsen (2003). "Seven principles of efficient human robot interaction." In: *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*. Vol. 4. IEEE, 3942–3948 vol.4. DOI: 10.1109/ICSMC.2003.1244504 (Cited on page 19).

Green, S. A. et al. (Jan. 2008). "Human-Robot Collaboration: A Literature Review and Augmented Reality Approach in Design." In: *International Journal of Advanced Robotic Systems* 5.1, p. 1. DOI: 10.5772/5664 (Cited on page 19).

Guthart, G. and J. Salisbury (2000). "The Intuitive/sup TM/ telesurgery system: overview and application." In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*. Vol. 1. IEEE, 618–621 vol.1. DOI: 10.1109/ROBOT.2000.844121 (Cited on page 10).

Habibovic, A. et al. (2017). "Command-Based Driving for Tactical Control of Highly Automated Vehicles." In: *Advances in Human Aspects of Transportation*. Ed. by N. A. Stanton et al. Cham: Springer International Publishing, pp. 499–510. ISBN: 978-3-319-41682-3 (Cited on page 104).

Hagenow, M. et al. (2021a). "Corrective Shared Autonomy for Addressing Task Variability." In: *IEEE Robotics and Automation Letters* 6.2, pp. 3720–3727. DOI: 10.1109/LRA.2021.3064500 (Cited on pages 11, 22).

Hagenow, M. et al. (2021b). "Informing Real-Time Corrections in Corrective Shared Autonomy Through Expert Demonstrations." In: *IEEE Robotics and Automation Letters* 6.4, pp. 6442–6449. DOI: 10.1109/LRA.2021.3094480 (Cited on page 22).

Hainsworth, D. W. (July 2001). "Teleoperation User Interfaces for Mining Robotics." In: *Autonomous Robots* 11.1, pp. 19–28. DOI: 10.1023/A:1011299910904 (Cited on page 9).

Hanna Kurniawati David Hsu, W. S. L. (June 2008). "SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces." In: *Proceedings of Robotics: Science and Systems IV*. Zurich, Switzerland. DOI: 10.15607/RSS.2008.IV.009 (Cited on page 109).

Hart, S. G. and L. E. Staveland (1988). "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research." In: *Advances in psychology*. Vol. 52. Elsevier, pp. 139–183 (Cited on page 50).

Hauser, K. (July 2013). "Recognition, prediction, and planning for assisted teleoperation of freeform tasks." In: *Autonomous Robots* 35.4, pp. 241–254. ISSN: 1573-7527. DOI: 10.1007/s10514-013-9350-3 (Cited on pages 18, 20–23, 53, 76, 87).

Heald, M. A. (May 1985). "Rational approximations for the Fresnel integrals." In: *Mathematics of Computation* 44.170, pp. 459–461. DOI: 10.1090/S0025-5718-1985-0777277-6 (Cited on page 86).

Herlant, L., B. Weinstein-Raun, and S. Srinivasa (2014). "Shared Control in Modal Teleoperation." In: *IROS 2014 Workshop on Rehabilitation and Assistive Robotics: Bridging the Gap Between Clinicians and Roboticists* (Cited on page 30).

Hermann, A. et al. (Sept. 2015). "Anticipate your surroundings: Predictive collision detection between dynamic obstacles and planned robot trajectories on the GPU." In: *2015 European Conference on Mobile Robots (ECMR)*. IEEE, pp. 1–8. DOI: 10.1109/ECMR.2015.7324047 (Cited on page 20).

Hirzinger, G. et al. (1993). "A sensor-based telerobotic system for the space robot experiment ROTEX." In: *Experimental Robotics II*. Ed. by R. Chatila and G. Hirzinger. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 222–238. ISBN: 978-3-540-39323-8. DOI: 10.1007/bfb0036142 (Cited on pages 9, 18).

Honegger, M. and A. Codourey (1998). "Redundancy resolution of a Cartesian space operated heavy industrial manipulator." In: *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*. Vol. 3. IEEE, 2094–2098 vol.3. DOI: 10.1109/ROBOT.1998.680628 (Cited on pages 9, 10, 14, 16, 59).

Honegger, M., G. Schweitzer, et al. (1997). "Vision supported operation of a concrete spraying robot for tunneling work." In: *Proceedings Fourth Annual Conference on Mechatronics and Machine Vision in Practice*. IEEE Comput. Soc, pp. 230–233. DOI: 10.1109/MMVIP.1997.625333 (Cited on pages 9, 14, 16).

Hornung, A. et al. (2013). "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees." In: *Autonomous Robots*. Software available at http://octomap.github.com. DOI: 10.1007/s10514-012-9321-0. URL: http://octomap.github.com (Cited on page 113).

Inagaki, T. (2003). "Adaptive automation: Sharing and trading of control." In: *Handbook of cognitive task design*. CRC Press, pp. 171–194. DOI: 10.1299/jsmetld.2001.10.79 (Cited on pages 26, 30).

Isaac-Lowry, O. J. et al. (Mar. 2017). "Compact teleoperated laparoendoscopic single-site robotic surgical system: Kinematics, control, and operation." In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 13.4. e1811 RCS-16-0121.R2, e1811. DOI: 10.1002/rcs.1811 (Cited on page 10).

Ivan, V. and S. Vijayakumar (July 2015). "Space-time area coverage control for robot motion synthesis." In: *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, pp. 207–212. DOI: 10.1109/ICAR.2015.7251457 (Cited on page 21).

Ivan, V., Y. Yang, et al. (July 2019). "EXOTica: An Extensible Optimization Toolset for Prototyping and Benchmarking Motion Planning and Control." In: *Robot Operating System (ROS): The Complete Reference (Volume 3)*. Ed. by A. Koubaa. Cham: Springer International Publishing, pp. 211–240. DOI: 10.1007/978-3-319-91590-6_7 (Cited on pages 36, 39, 57, 115).

Javdani, S. et al. (June 2018). "Shared autonomy via hindsight optimization for teleoperation and teaming." In: *The International Journal of Robotics Research* 37.7, pp. 717–742. DOI: 10.1177/0278364918776060 (Cited on pages 11, 20–22, 35, 76, 87, 109).

Jenkins, O. and M. Mataric (2002). "Deriving action and behavior primitives from human motion data." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 3, 2551–2556 vol.3. DOI: 10.1109/IRDS.2002.1041654 (Cited on page 100).

Johns, M. et al. (Mar. 2016). "Exploring shared control in automated driving." In: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, pp. 91–98. DOI: 10.1109/HRI.2016.7451738 (Cited on page 28).

Kalakrishnan, M. et al. (May 2011). "STOMP: Stochastic trajectory optimization for motion planning." In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, pp. 4569–4574. DOI: 10.1109/ICRA.2011.5980280 (Cited on page 96).

Kanehiro, F. et al. (June 2008). "A local collision avoidance method for non-strictly convex polyhedra." In: *Robotics: Science and Systems IV*. Robotics Science and Systems (RSS) Foundation. DOI: 10.15607/RSS.2008.IV.020 (Cited on page 58).

Karaman, S. et al. (May 2011). "Anytime Motion Planning using the RRT*." In: *IEEE International Conference on Robotics and Automation*. IEEE, pp. 1478–1483. DOI: 10.1109/ICRA.2011.5980479 (Cited on page 20).

Kelly, M. (2017). "An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation." In: *SIAM Review* 59.4, pp. 849–904. DOI: 10.1137/16M1062569 (Cited on page 11).

Khansari-Zadeh, S. M. and A. Billard (Mar. 2012). "A dynamical system approach to realtime obstacle avoidance." In: *Autonomous Robots* 32.4, pp. 433–454. ISSN: 1573-7527. DOI: 10.1007/s10514-012-9287-y (Cited on page 20).

Khatib, O. (1985). "Real-time obstacle avoidance for manipulators and mobile robots." In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. Vol. 2. Institute of Electrical and Electronics Engineers, pp. 500–505. DOI: 10.1109/ROBOT.1985.1087247 (Cited on pages 19, 21).

Khatib, O. (Feb. 1987). "A unified approach for motion and force control of robot manipulators: The operational space formulation." In: *IEEE Journal on Robotics and Automation* 3.1, pp. 43–53. ISSN: 0882-4967. DOI: 10.1109/JRA.1987.1087068 (Cited on page 25).

Kim, D.-J. et al. (Jan. 2012). "How Autonomy Impacts Performance and Satisfaction: Results From a Study With Spinal Cord Injured Subjects Using an Assistive Robot." In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 42.1, pp. 2–14. DOI: 10.1109/TSMCA.2011.2159589 (Cited on pages 19, 35, 50).

Kim, Y. S. et al. (Mar. 2005). "A force reflected exoskeleton-type masterarm for human-robot interaction." In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 35.2, pp. 198–212. DOI: 10.1109/TSMCA.2004.832836 (Cited on page 18).

Klamt, T. et al. (July 2020). "Remote mobile manipulation with the centauro robot: Full-body telepresence and autonomous operator assistance." In: *Journal of Field Robotics* 37.5, pp. 889–919. DOI: 10.1002/rob.21895 (Cited on page 28).

Kosari, S. N., F. Rydén, et al. (Nov. 2014a). "Forbidden region virtual fixtures from streaming point clouds." In: *Advanced Robotics* 28.22, pp. 1507–1518. DOI: 10.1080/01691864.2014.962613 (Cited on page 19).

Kosari, S. N., F. Rydén, et al. (Nov. 2014b). "Forbidden region virtual fixtures from streaming point clouds." In: *Advanced Robotics* 28.22, pp. 1507–1518. DOI: 10.1080/01691864.2014.962613 (Cited on page 76).

Kuffner, J. J. and S. M. LaValle (2000). "RRT-connect: An efficient approach to single-query path planning." In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*. Vol. 2. IEEE, 995–1001 vol.2. DOI: 10.1109/ROBOT.2000.844730 (Cited on page 115).

Kulak, T., J. Silverio, and S. Calinon (July 2020). "Fourier movement primitives: an approach for learning rhythmic robot skills from demonstrations." In: *Proceedings of Robotics: Science and Systems*. Corvalis, Oregon, USA: Robotics: Science and Systems Foundation. DOI: 10.15607/RSS.2020.XVI.056 (Cited on page 108).

Ya-kun, Z. et al. (Aug. 2017). "Shared control on lunar spacecraft teleoperation rendezvous operations with large time delay." In: *Acta Astronautica* 137, pp. 312–319. DOI: 10.1016/j.actaastro.2017.04.014 (Cited on page 76).

Kwitowski, A., W. Lewis, and W. Mayercheck (May 1989). "Computer-based, teleoperation of a new highwall mining system." In: *1989 IEEE International Conference on Robotics and Automation*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 1478, 1479, 1480, 1481, 1482, 1483. DOI: 10.1109/ROBOT.1989.100188 (Cited on page 9).

Kwitowski, A., W. Mayercheck, and A. Brautigam (1992). "Teleoperation for continuous miners and haulage equipment." In: *IEEE Transactions on Industry Applications* 28.5, pp. 1118–1125. DOI: 10.1109/28.158837 (Cited on page 9).

Lathan, C. E. and M. Tracey (Aug. 2002). "The Effects of Operator Spatial Perception and Sensory Feedback on Human-Robot Teleoperation Performance." In: *Presence: Teleoperators and Virtual Environments* 11.4, pp. 368–377. DOI: 10.1162/105474602760204282 (Cited on page 48).

Lee, D. et al. (2013). "Semiautonomous Haptic Teleoperation Control Architecture of Multiple Unmanned Aerial Vehicles." In: *IEEE/ASME Transactions on Mechatronics* 18.4, pp. 1334–1345. DOI: 10.1109/TMECH.2013.2263963 (Cited on page 104).

Lee, J. D. and K. A. See (Jan. 2004). "Trust in Automation: Designing for Appropriate Reliance." In: *Human Factors* 46.1, pp. 50–80. DOI: 10.1518/hfes.46.1.50_30392 (Cited on page 23).

Leeper, A. et al. (2012). "Strategies for human-in-the-loop robotic grasping." In: *2012 7th ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*. ACM Press, pp. 1–8. DOI: 10.1145/2157689.2157691 (Cited on pages 18, 19, 104).

Leidner, D. et al. (July 2019). "Global Remote Operation of Intelligent Space Robot Assistants." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.1, pp. 9863–9864. DOI: 10.1609/aaai.v33i01.33019863 (Cited on page 9).

Li, M. and A. Okamura (2003). "Recognition of operator motions for real-time assistance using virtual fixtures." In: *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings.* IEEE Comput. Soc, pp. 125–131. DOI: 10.1109/HAPTIC.2003.1191253 (Cited on page 20).

Lii, N. Y. et al. (Oct. 2010). "Toward understanding the effects of visual- and force-feedback on robotic hand grasping performance for space teleoperation." In: *IEEE/RSJ IROS*. IEEE, pp. 3745–3752. DOI: 10.1109/IROS.2010.5650186 (Cited on page 25).

Lima, P. F. et al. (July 2015). "Clothoid-based model predictive control for autonomous driving." In: *European Control Conference (ECC)*. IEEE. DOI: 10.1109/ecc.2015.7330991 (Cited on page 86).

Liu, G. et al. (Feb. 2021). "Automatic spraying motion planning of a shotcrete manipulator." In: *Intelligent Service Robotics*. DOI: 10.1007/s11370-021-00348-9 (Cited on page 15).

Liu, Y., M. Habibnezhad, and H. Jebelli (Mar. 2021). "Brain-computer interface for hands-free teleoperation of construction robots." In: *Automation in Construction* 123, p. 103523. ISSN: 0926-5805. DOI: 10.1016/j.autcon.2020.103523 (Cited on page 18).

Luo, J. et al. (2020). "A Teleoperation Framework for Mobile Robots Based on Shared Control." In: *IEEE Robotics and Automation Letters* 5.2, pp. 377–384. DOI: 10.1109/LRA.2019.2959442 (Cited on page 104).

Luth, T. et al. (June 2007). "Low level control in a semi-autonomous rehabilitation robotic system via a Brain-Computer Interface." In: *2007 IEEE 10th International Conference on Rehabilitation Robotics*. IEEE, pp. 721–728. DOI: 10.1109/ICORR.2007.4428505 (Cited on page 18).

MacKenzie, I. S., A. Sellen, and W. A. S. Buxton (1991). "A Comparison of Input Devices in Element Pointing and Dragging Tasks." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '91. New Orleans, Louisiana, USA: Association for Computing Machinery, pp. 161–166. DOI: 10.1145/108844.108868 (Cited on page 42).

Marion, P. et al. (Dec. 2017). "Director: A user interface designed for robot operation with shared autonomy." In: *Journal of Field Robotics* 34.2, pp. 262–280. DOI: 10.1002/rob.21681 (Cited on pages 22, 53, 109, 113, 114).

Marturi, N. et al. (Dec. 2016). "Towards advanced robotic manipulation for nuclear decommissioning: A pilot study on tele-operation and autonomy." In: *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)*. IEEE, pp. 1–8. DOI: 10.1109/RAHA.2016.7931866 (Cited on page 9).

Marzbani, H., R. N. Jazar, and M.Fard (2015). "Better Road Design Using Clothoids." In: *Sustainable Automotive Technologies 2014*. Ed. by I. Denbratt, A. Subic, and J. Wellnitz. Springer International Publishing, pp. 25–40. DOI: 10.1007/978-3-319-17999-5_3 (Cited on page 86).

Massie, T. H., J. K. Salisbury, et al. (1994). "The phantom haptic interface: A device for probing virtual objects." In: *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*. Vol. 55. 1. Chicago, IL, pp. 295–300 (Cited on page 18).

Massimino, M. J., T. B. Sheridan, and J. B. Roseborough (1989). "One handed tracking in six degrees of freedom." In: *Conference Proceedings., IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 498–503 vol.2. DOI: 10.1109/ICSMC.1989.71346 (Cited on page 27).

McGill, S., S.-J. Yi, and D. D. Lee (Nov. 2015). "Team THOR's adaptive autonomy for disaster response humanoids." In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 453–460. DOI: 10.1109/HUMANOIDS.2015.7363589 (Cited on page 9).

Merkt, W., V. Ivan, and S. Vijayakumar (Oct. 2018). "Leveraging Precomputation with Problem Encoding for Warm-Starting Trajectory Optimization in Complex Environments." In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 5877–5884. DOI: 10.1109/IROS.2018.8593977 (Cited on page 109).

Merkt, W., Y. Yang, et al. (Aug. 2017). "Robust shared autonomy for mobile manipulation with continuous scene monitoring." In: *2017 13th IEEE Conference on Au-*

tomation Science and Engineering (*CASE*). IEEE, pp. 130–137. DOI: 10.1109/COASE.2017.8256092 (Cited on pages 26, 53, 113).

Mistry, M. and S. Schaal (Dec. 2015). "Representation and control of the task space in humans and humanoid robots." In: *Frontiers in Neuroengineering Series*. CRC Press/Taylor & Francis, pp. 121–151. DOI: 10.1201/b17949-9 (Cited on pages 31, 34).

Moura, J. et al. (Oct. 2018). "Automation of Train Cab Front Cleaning With a Robot Manipulator." In: *IEEE Robotics and Automation Letters* (*RA-L*) 3.4, pp. 3058–3065. DOI: 10.1109/LRA.2018.2849591 (Cited on pages 69, 75).

Muelling, K. et al. (Aug. 2017). "Autonomy infused teleoperation with application to brain computer interface controlled manipulation." In: *Autonomous Robots* 41.6, pp. 1401–1422. ISSN: 1573-7527. DOI: 10.1007/s10514-017-9622-4 (Cited on page 20).

Murakami, N. et al. (Aug. 2008). "Development of a teleoperation system for agricultural vehicles." In: *Computers and Electronics in Agriculture* 63.1. Special issue on bio-robotics, pp. 81–88. ISSN: 0168-1699. DOI: 10.1016/j.compag.2008.01.015 (Cited on page 9).

Murphy, R. R. (Mar. 2015). "Meta-analysis of Autonomy at the DARPA Robotics Challenge Trials." In: *J. Field Robot.* 32.2, pp. 189–191. ISSN: 1556-4959. DOI: 10.1002/rob.21578 (Cited on page 107).

Murphy, R. R. et al. (Nov. 2011). "Use of remotely operated marine vehicles at Minamisanriku and Rikuzentakata Japan for disaster recovery." In: *2011 IEEE Int. Symposium on Safety, Security, and Rescue Robotics*. IEEE, pp. 19–25. DOI: 10.1109/SSRR.2011.6106798 (Cited on page 25).

Nabulsi, S., A. Rodriguez, and O. Rio (2010). "Robotic Machine for High-Quality Shotcreting Process." In: *ISR/ROBOTIK 2010, Proceedings for the joint conference of ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics), 7-9 June 2010, Munich, Germany - Parallel to AUTOMATICA*, pp. 1–8. URL: https://ieeexplore.ieee.org/document/5756929 (Cited on page 15).

Niemeyer, G. et al. (2016). "Telerobotics." In: *Springer Handbook of Robotics*. Ed. by B. Siciliano and O. Khatib. Cham: Springer International Publishing, pp. 1085–1108. DOI: 10.1007/978-3-319-32552-1_43 (Cited on pages 2, 12, 75, 109).

Nikolaidis, S. et al. (2017). "Human-Robot Mutual Adaptation in Shared Autonomy." In: *2017 12th ACM/IEEE International Conference on Human-Robot Interaction* (*HRI*), pp. 294–302 (Cited on page 53).

Oguz, O. S., Z. Zhou, and D. Wollherr (Mar. 2018). "A Hybrid Framework for Understanding and Predicting Human Reaching Motions." In: *Frontiers in Robotics and AI* 5, p. 27. DOI: 10.3389/frobt.2018.00027 (Cited on page 78).

Okamura, A. M. (Jan. 2004). "Methods for haptic feedback in teleoperated robot-assisted surgery." In: *Industrial Robot: An International Journal* 31.6, pp. 499–508. ISSN: 0143-991X. DOI: 10.1108/01439910410566362 (Cited on page 18).

Okamura, A. M. (Jan. 2009). "Haptic feedback in robot-assisted minimally invasive surgery." In: *Current opinion in urology* 19.1, pp. 102–107. ISSN: 1473-6586. DOI: 10.1097/MOU.0b013e32831a478c (Cited on page 18).

Park, D.-H. et al. (Dec. 2008). "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields." In: *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 91–98. DOI: 10.1109/ICHR.2008.4755937 (Cited on page 21).

Peña, C., C. Riaño, and G. Moreno (Nov. 2018). "RobotGreen. A Teleoperated Agricultural Robot for Structured Environments." In: *Journal of Engineering Science & Technology Review* 11.6, pp. 144–155. ISSN: 17912377. DOI: 10.25103/jestr.116.18 (Cited on page 9).

Pereira, A. et al. (Aug. 2017). "Augmented reality dialog interface for multimodal teleoperation." In: *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, pp. 764–771. DOI: 10.1109/ROMAN.2017.8172389 (Cited on page 10).

Philips, J. et al. (June 2007). "Adaptive Shared Control of a Brain-Actuated Simulated Wheelchair." In: *2007 IEEE 10th International Conference on Rehabilitation Robotics*. IEEE, pp. 408–414. DOI: 10.1109/ICORR.2007.4428457 (Cited on page 27).

Quere, G. et al. (May 2020). "Shared Control Templates for Assistive Robotics." In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1956–1962. DOI: 10.1109/ICRA40945.2020.9197041 (Cited on page 109).

Quigley, M. et al. (2009). "ROS: an open-source Robot Operating System." In: *ICRA Workshop on Open Source Software*. Kobe, Japan (Cited on page 39).

Rakita, D., B. Mutlu, and M. Gleicher (Mar. 2017). "A Motion Retargeting Method for Effective Mimicry-Based Teleoperation of Robot Arms." In: *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '17. Vienna, Austria: Association for Computing Machinery, pp. 361–370. DOI: 10.1145/2909824.3020254 (Cited on page 18).

Rakita, D., B. Mutlu, and M. Gleicher (Feb. 2018). "An Autonomous Dynamic Camera Method for Effective Remote Teleoperation." In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '18. Chicago, IL, USA: ACM, pp. 325–333. DOI: 10.1145/3171221.3171279 (Cited on pages 22, 53).

Rebsamen, B. et al. (2010). "A Brain Controlled Wheelchair to Navigate in Familiar Environments." In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 18.6, pp. 590–598. DOI: 10.1109/TNSRE.2010.2049862 (Cited on page 104).

Reddy, S., A. Dragan, and S. Levine (June 2018). "Shared Autonomy via Deep Reinforcement Learning." In: *Proceedings of Robotics: Science and Systems*. Pittsburgh, Pennsylvania: Robotics: Science and Systems Foundation. DOI: 10.15607/RSS.2018.XIV.005 (Cited on page 11).

Rodriguez, A. J. and O. Río (2007). "Analysis of Real Time Technical Data Obtained While Shotcreting: An Approach Towards Automation." In: *International Conference On Computational Methods in Tunneling EURO:TUN* (Cited on page 14).

Rodriguez, A., S. Nabulsi, and O. Rio (2009). "A method for estimating thickness of sprayed concrete layers from pumped volume." In: *EURO-TUN: Computational Methods in Tunnelling* (Cited on pages 15, 17, 60).

Rohmer, E., S. P. N. Singh, and M. Freese (Nov. 2013). "CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework." In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1321–1326. DOI: 10.1109/IROS.2013.6696520 (Cited on pages 35, 68).

Rosenberg, L. B. (1993). "Virtual Fixtures: Perceptual Tools for Telerobotic Manipulation." In: *Proceedings of the 1993 IEEE Virtual Reality Annual International Symposium*. VRAIS '93. Washington, DC, USA: IEEE Computer Society, pp. 76–82. DOI: 10.1109/VRAIS.1993.380795 (Cited on pages 19, 21, 22, 26, 53, 76).

Rother, C., V. Kolmogorov, and A. Blake (Aug. 2004). ""GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts." In: *ACM Trans. Graph.* 23.3, pp. 309–314. ISSN: 0730-0301. DOI: 10.1145/1015706.1015720 (Cited on page 109).

Rubagotti, M. et al. (July 2019). "Semi-Autonomous Robot Teleoperation With Obstacle Avoidance via Model Predictive Control." In: *IEEE Robotics and Automation Letters (RA-L)* 4.3, pp. 2746–2753. DOI: 10.1109/LRA.2019.2917707 (Cited on pages 76, 77, 82, 104).

Rusu, R. B. et al. (Oct. 2009). "Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1–6. DOI: 10.1109/IROS.2009.5354683 (Cited on page 113).

Rydén, F. and H. J. Chizeck (Oct. 2012). "Forbidden-region virtual fixtures from streaming point clouds: Remotely touching and protecting a beating heart." In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3308–3313. DOI: 10.1109/IROS.2012.6386012 (Cited on page 22).

SAE International (2016). "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems." In: *http://standards.sae.org/j3016_201609* (Cited on page 12).

Saha, S. and M. Baumert (Jan. 2020). "Intra- and Inter-subject Variability in EEG-Based Sensorimotor Brain Computer Interface: A Review." In: *Frontiers in Computational Neuroscience* 13, p. 87. ISSN: 1662-5188. DOI: 10.3389/fncom.2019.00087 (Cited on page 18).

Schaal, S. (2006). "Dynamic Movement Primitives - A Framework for Motor Control in Humans and Humanoid Robotics." In: *Adaptive Motion of Animals and Machines*. Ed. by H. Kimura et al. Tokyo: Springer Tokyo, pp. 261–280. DOI: 10.1007/4-431-31381-8_23 (Cited on pages 20, 100).

Schiele, A. and G. Hirzinger (Sept. 2011). "A new generation of ergonomic exoskeletons - The high-performance X-Arm-2 for space robotics telepresence." In: *2011*

IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 2158–2165. DOI: 10.1109/IROS.2011.6094868 (Cited on page 18).

Schröer, S. et al. (May 2015). "An autonomous robotic assistant for drinking." In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 6482–6487. DOI: 10.1109/ICRA.2015.7140110 (Cited on page 18).

Schultz, C. et al. (May 2017). "Goal-predictive robotic teleoperation from noisy sensors." In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5377–5383. DOI: 10.1109/ICRA.2017.7989633 (Cited on page 27).

Schwarting, W. et al. (May 2017). "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention." In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. DOI: 10.1109/icra.2017.7989224 (Cited on pages 23, 76).

Schwartz, A. B. et al. (Oct. 2006). "Brain-Controlled Interfaces: Movement Restoration with Neural Prosthetics." In: *Neuron* 52.1, pp. 205–220. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2006.09.019 (Cited on page 18).

Sellner, B. et al. (2006). "Coordinated Multiagent Teams and Sliding Autonomy for Large-Scale Assembly." In: *Proceedings of the IEEE* 94.7, pp. 1425–1444. DOI: 10.1109/JPROC.2006.876966 (Cited on page 12).

Seshia, S. A., D. Sadigh, and S. S. Sastry (June 2015). "Formal methods for semi-autonomous driving." In: *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. ACM, pp. 1–5. DOI: 10.1145/2744769.2747927 (Cited on page 10).

Shannon, C. E. (July 1948). "A Mathematical Theory of Communication." In: *Bell System Technical Journal* 27.3, pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x (Cited on page 43).

Sheridan, T. B. and W. L. Verplank (1978). *Human and computer control of undersea teleoperators*. Tech. rep. Massachusetts Inst of Tech Cambridge Man-Machine Systems Lab (Cited on page 12).

Sheridan, T. B. (1992). *Telerobotics, Automation, and Human Supervisory Control*. MIT Press (Cited on pages 2, 10, 109).

Shinners, P. (2011). *PyGame*. http://pygame.org/ (Cited on pages 36, 61).

Sian, N. et al. (2002). "Whole body teleoperation of a humanoid robot - development of a simple master device using joysticks." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 3. IEEE, 2569–2574 vol.3. DOI: 10.1109/IRDS.2002.1041657 (Cited on page 18).

Silva, J. A. R. and V. Grassi (May 2018). "Clothoid-Based Global Path Planning for Autonomous Vehicles in Urban Scenarios." In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEE. DOI: 10.1109/ICRA.2018.8461201 (Cited on page 86).

Silva Pereira, H. A. da, M. C. Rodrigues, and J. V. L. de Carvalho Firmino (June 2019). "Implementation of weave patterns by path parameterization in the simulation of welding processes by the finite element method." In: *The International Journal*

of Advanced Manufacturing Technology 104.1, pp. 477–487. DOI: 10.1007/s00170-019-03861-5 (Cited on pages 75, 81–84, 108).

Simon, H. A. (1979). "Rational Decision Making in Business Organizations." In: *The American Economic Review* 69.4, pp. 493–513. ISSN: 00028282 (Cited on pages 53, 72).

Simorov, A. et al. (Aug. 2012). "Review of surgical robotics user interface: what is the best way to control robotic surgery?" In: *Surgical Endoscopy* 26.8, pp. 2117–2125. ISSN: 1432-2218. DOI: 10.1007/s00464-012-2182-y (Cited on page 18).

Swaen, G. M. H. et al. (June 2003). "Fatigue as a risk factor for being injured in an occupational accident: results from the Maastricht Cohort Study." In: *Occupational and environmental medicine* 60 Suppl 1.Suppl 1. PMC1765730[pmcid], pp. i88–i92. ISSN: 1351-0711. DOI: 10.1136/oem.60.suppl_1.i88 (Cited on page 17).

Tamura, Y., H. Amano, and J. Ota (June 2020). "Analysis of firefighting skill with a teleoperated robot." In: *ROBOMECH Journal* 7.1, p. 26. ISSN: 2197-4225. DOI: 10.1186/s40648-020-00177-y (Cited on page 10).

Toussaint, M. (2009). "Robot Trajectory Optimization Using Approximate Inference." In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: Association for Computing Machinery, pp. 1049–1056. ISBN: 9781605585161. DOI: 10.1145/1553374.1553508 (Cited on page 96).

Toussaint, M. (2015). "Logic-Geometric Programming: An Optimization-Based Approach to Combined Task and Motion Planning." In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, pp. 1930–1936. ISBN: 9781577357384. URL: https://dl.acm.org/doi/10.5555/2832415.2832517 (Cited on page 109).

Tsui, K. et al. (2008). "Development and Evaluation of a Flexible Interface for a Wheelchair Mounted Robotic Arm." In: *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction*. HRI '08. Amsterdam, The Netherlands: Association for Computing Machinery, pp. 105–112. ISBN: 9781605580173. DOI: 10.1145/1349822.1349837 (Cited on page 18).

Vagia, M., A. A. Transeth, and S. A. Fjerdingen (2016). "A literature review on the levels of automation during the years. What are the different taxonomies that have been proposed?" In: *Applied Ergonomics* 53, pp. 190–202. ISSN: 0003-6870. DOI: 10.1016/j.apergo.2015.09.013 (Cited on page 12).

Wächter, A. (2009). "Short Tutorial: Getting Started With Ipopt in 90 Minutes." In: *Combinatorial Scientific Computing*. Ed. by U. Naumann et al. Dagstuhl Seminar Proceedings 9061. Dagstuhl, Germany: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany. URL: https://projects.coin-or.org/CoinBinary/export/837/CoinAll/trunk/Installer/files/doc/Short%20tutorial%20Ipopt.pdf (Cited on page 11).

Wang, S. et al. (2020). *A Comparison Between Joint Space and Task Space Mappings for Dynamic Teleoperation of an Anthropomorphic Robotic Arm in Reaction Tests*. arXiv: 2011.02508 [cs.RO] (Cited on page 49).

Wang, T. et al. (June 2020). "A novel bilateral impedance controls for underwater tele-operation systems." In: *Applied Soft Computing* 91, p. 106194. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2020.106194 (Cited on page 9).

Weisstein, E. W. (2021a). *"Circle-Circle Intersection". From MathWorld—A Wolfram Web Resource*. URL: https://mathworld.wolfram.com/Circle-CircleIntersection.html (Cited on page 70).

Weisstein, E. W. (2021b). ""Lens"." In: *From MathWorld—A Wolfram Web Resource*. URL: https://mathworld.wolfram.com/Lens.html (Cited on page 70).

Welton, J. (2014). *How do you get those weaved welds?* thefabricator.com/thewelder (Cited on pages 79, 108).

Yamamoto, T. et al. (2012). "Augmented reality and haptic interfaces for robot-assisted surgery." In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 8.1, pp. 45–56. DOI: https://doi.org/10.1002/rcs.421 (Cited on page 104).

Yamauchi, B. M. (Sept. 2004). "PackBot: a versatile platform for military robotics." In: *Unmanned Ground Vehicle Technology VI*. Ed. by G. R. Gerhart, C. M. Shoemaker, and D. W. Gage. Vol. 5422. International Society for Optics and Photonics. SPIE, pp. 228–237. DOI: 10.1117/12.538328 (Cited on page 9).

Yanco, H. A. et al. (Mar. 2015). "Analysis of Human-robot Interaction at the DARPA Robotics Challenge Trials." In: *Journal of Field Robotics* 32.3, pp. 420–444. DOI: 10.1002/rob.21568 (Cited on page 107).

Yang, G.-Z. et al. (2017). "Medical robotics—Regulatory, ethical, and legal considerations for increasing levels of autonomy." In: *Science Robotics* 2.4, eaam8638. DOI: 10.1126/scirobotics.aam8638. eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.aam8638. URL: https://www.science.org/doi/abs/10.1126/scirobotics.aam8638 (Cited on page 12).

Yang, Y., V. Ivan, Z. Li, et al. (Nov. 2016). "iDRM: Humanoid motion planning with realtime end-pose selection in complex environments." In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 271–278. DOI: 10.1109/HUMANOIDS.2016.7803288 (Cited on page 115).

Yang, Y., V. Ivan, and S. Vijayakumar (July 2015). "Real-time motion adaptation using relative distance space representation." In: *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, pp. 21–27. DOI: 10.1109/ICAR.2015.7251428 (Cited on pages 21, 58).

Yoerger, D. and J.-J. Slotine (1987). "Supervisory control architecture for underwater teleoperation." In: *Proceedings. 1987 IEEE International Conference on Robotics and Automation*. Vol. 4. Institute of Electrical and Electronics Engineers, pp. 2068–2073. DOI: 10.1109/ROBOT.1987.1087890 (Cited on page 9).

You, E. and K. Hauser (June 2011). "Assisted Teleoperation Strategies for Aggressively Controlling a Robot Arm with 2D Input." In: *Proc. of Robotics: Science and Systems*. Los Angeles, CA, USA: Robotics: Science and Systems Foundation. DOI: 10.15607/RSS.2011.VII.045 (Cited on pages 18, 19, 30, 34, 35, 50).

Zarubin, D. et al. (July 2012). "Hierarchical Motion Planning in Topological Representations." In: *Proceedings of Robotics: Science and Systems*. Sydney, Australia: Robotics: Science and Systems Foundation. DOI: 10.15607/RSS.2012.VIII.059 (Cited on page 21).

Zhang, M. et al. (Apr. 2015). "Influence of fatigue on construction workers' physical and cognitive function." In: *Occupational medicine (Oxford, England)* 65.3, pp. 245–250. ISSN: 1471-8405. DOI: 10.1093/occmed/kqu215 (Cited on page 17).

Ziebart, B. D. et al. (2008). "Maximum entropy inverse reinforcement learning." In: *Aaai*. Vol. 8. Chicago, IL, USA, pp. 1433–1438 (Cited on page 20).

Zucker, M. et al. (Aug. 2013). "CHOMP: Covariant Hamiltonian optimization for motion planning." In: *The International Journal of Robotics Research* 32.9-10, pp. 1164–1193. DOI: 10.1177/0278364913488805 (Cited on pages 25, 96).