Imperial College London
Department of Electrical and Electronic Engineering

Thesis submitted for the degree of Doctor of Philosophy

Learning to assist in triadic human-robot interaction

Vinícius Barbosa Schettino
Author

Prof. Yiannis Demiris
Supervisor

December 2020

Statement of Originality

I hereby declare that this thesis and the work herein detailed was composed and originated by myself, except where appropriately referenced and credited.

London, December 2020

Vinícius Barbosa Schettino

Copyright Declaration

Abstract

For robots that aim at providing physical assistance, a significant challenge is adapting to the different needs of people. A way to overcome this is to ask human experts to provide demonstrations of how to help a specific person; a setting known as Learning Assistance by Demonstration (LAD). In this thesis, we consider the application of robotic wheelchairs and investigate: *how can demonstrations of assistance be used to improve the navigation performance of powered wheelchair users with hand-control disabilities?* For this, we first contribute a custom teleoperation platform that enables demonstrations of assistance and uses haptic and virtual reality interfaces to facilitate the interpretation of raw sensor data. This platform is used to test claimed features of LAD, and we show that the technique can adapt to different hand-control impairments and also generates personalised assistive models. Furthermore, we explore the issue of model generalisation to physically different environments, which had not been investigated before. We show that this is a challenging problem and, to overcome it, propose solutions in terms of data collection and preprocessing, training and evaluation procedures, and learning algorithms. With these adaptations, we demonstrate that our model can provide useful assistance, even in previously unseen environments. To accelerate research in this field, we developed software that can simulate the full triadic interaction (assistant/robot/driver) concerning the application of LAD for robotic wheelchairs. This software permits the simulation of multiple disabilities and environments, and also allows people to take up the roles of the simulated driver and/or assistant. Finally, to keep our assumptions and developments in check, we conducted two experimental evaluations with humans, assessing how multimodal interfaces affect one's capability to infer the intention of a driver, and how different learning algorithms impact the generalisation and assistive performance of LAD.

Acknowledgements

A PhD journey can be an inspiring and exciting experience, but at times it is also very demanding. For me, it would not have been possible to complete this journey without the help of many, many people. Here, I would like to thank some of those that were particularly supportive over the past years.

First, I thank my supervisor, Yiannis Demiris. Your leadership and receptiveness inspired me, and I hope to have learned enough from it to one day pass it on to my future students. Our discussions were paramount in pushing me to do my best and keeping me on the right track. A sincere thank you for guiding me through my PhD and for allowing me to be part of the Personal Robotics Lab, which I have grown to like so much.

I also thank Dr Nicola Bellotto and Dr Jeremy Pitt, for taking the time to study this thesis and being the examiners of my viva. Your thorough review and insightful comments greatly improved the quality of this document.

I will always remember the Personal Robotics Lab as a place where I have learned immensely, but also as a place where I made many good friends. My colleagues were remarkable in finding the right balance between sharing technical knowledge and sharing good laughs. For this, in no particular order, I thank Mark, Rodrigo, Fan, Tobi, Urbano, Ahmed, Josh, Cat, Pierluigi, Ruohan and Vicent, all people that I will carry with my me as the fondest memories.

Living abroad for three years, the feeling of missing home was always lurking. Thankfully, I met at the Department a small but amazing community of Brazilian friends, who were always there to help, chat or drink coffee. For the many times you made me laugh, and the many tips for living in London and studying at Imperial, thank you Thiago, David, Alexandre and Henrique. I hope we will meet again soon.

My family is my biggest treasure. They raised me with more love than one can hope for and continue to inspire me every day through their hard work and displays of character and kindness. I have no words to thank you for the support you gave me throughout my life and throughout my PhD. Obrigado Vinicius, Juçara, Isabella, Júlia e Marcos. Amo vocês!

I mentioned how I could not have gone through the PhD without the help of many people. However, there is one person that, without her support and encouragement, the journey would never even have started. Rachel, thank you so, so much for being by my side every single day of this adventure. I know it was not easy at times, and I will forever appreciate your sacrifices. Te amo!

Lastly, I want to thank CAPES and CEFET-MG for funding my studies. Deeply studying robotics and machine learning was a dream of mine, and would not have been possible without the financial support of these institutions.

Contents

List of Figures

List of Tables

List of Acronyms

ARTA Assistive Robot for Transport of Adults.

ARTY Assistive Robot for Transport of Youngsters.

CAN Controller Area Network.

GP Gaussian Process.

GRU Gated Recurrent Unit.

IIHS Inference of Intention Helpfulness Score.

IMU Inertial Measurement Unit.

LAD Learning Assistance by Demonstration.

LbD Learning by Demonstration.

LUPI Learning Using Privileged Information.

MSE Mean Squared Error.

OECD Organisation for Economic Cooperation and Development.

OIESGP Online Infinite Echo-State Gaussian Process.

PCA Principal Component Analysis.

PID Proportional, Integral and Derivative.

ROS Robot Operating System.

RT-GENE Real-Time eye Gaze Estimation in Natural Environments.

SLAM Simultaneous Localisation and Mapping.

SVM Support Vector Machine.

SW Smart Wheelchair.

URDF Unified Robot Description Format.

USB Universal Serial Bus.

VR Virtual Reality.

Chapter 1

Introduction

Motor and cognitive disabilities can turn ordinary tasks into significant challenges. A wheelchair user, for example, might encounter difficulties when navigating a cluttered environment or narrow pathways. Similarly, other struggles are presented to bearers of visual, audio and neurological disabilities. This distinct reality becomes particularly hard when the disability is not congenital, but acquired during lifetime, through an accident, disease or ageing. In these cases, the adapting period can be long and arduous.

To ease this transition, professionals like nurses, caretakers and physiotherapists can be called to aid. Using their training and tacit knowledge, these professionals can significantly improve the quality of life of disability bearers. This solution, however, might be prohibitively expensive in many cases. Furthermore, with the world population ageing at a fast pace and the already insufficient number of nurses and caretakers (European Commission 2012), this tends to become an ever more scarce resource. The World Health Organisation estimates a shortage of over 16 million healthcare workers by 2030 (World Health Organization (WHO) 2016). Similarly, the United States projects a shortage of registered nurses by the same year, even with an estimated increase of 7% in this workforce (American Association of Colleges of Nursing 2020). In some emerging countries, like Brazil, the problem is stark. Data collected by the OECD shows that the life expectancy of the country (75 years) has increased considerably over the last decades, approaching that of more developed economies (80.8 years on average for OECD countries). Conversely, the number of nurses remains much lower in comparison: 1.5 against 9.1 for every 1000 people (OECD 2017).

A possible way to overcome this situation is to use robotic systems to support and augment the services provided by health workers. This area of technology is known as *assistive robotics* and may be broadly defined as the

Figure 1.1: Example of an assistive robotics application, where a robot helps a person with reduced mobility get dressed. Excerpt from (Zhang, Cully, and Demiris 2019)

field of robotics devoted to "give aid or support to human users" (Feil-Seifer and Mataric 2005). These systems, which have already existed for some years, have been gradually evolving towards the goal of improving the quality of life of those who need assistance (Robinson, MacDonald, and Broadbent 2014). Devices as varied as robotic feeders (Brose et al. 2010), smart wheelchairs (Leaman and La 2017), robotic white canes (Lock, Cielniak, and Bellotto 2017) and robots that provide cognitive stimulation for the elderly (Coşar et al. 2020), gain more relevance each day. Assistive robots, such as the dressing assistant illustrated in Figure 1.1, have the potential to considerably reduce the burden on caretaker staff (Mitzner et al. 2014).

The present reality, however, is that assistive robots struggle to reach full dissemination among end-users, remaining mostly confined to laboratories and controlled environments. A major reason for this is simply that generating proper assistance is very challenging. By its nature, assistance depends on the goal, the user, the user's state, the environment, etc. (Soh and Demiris 2013). Besides, even the same disease or similar traumas can affect the human body in different ways, leading to the need for personalised assistance. Robots, on the other hand, are traditionally designed to accomplish specific and repetitive tasks. Thus, to make assistive robots available to a greater number of people, a challenge that persists is making these robots adaptable to the individual needs of people.

1.1 Smart wheelchairs

Among many prolific areas for assistive robotic technologies, the topic related to powered wheelchairs is of distinct relevance. In the year 2000, the United Kingdom alone had 11,350 users of these devices, with the number steadily growing each year (Sanderson and Place 2001). Although a more recent estimate is lacking, the figure should be much higher by now, as the country's national health system currently reports 1.2 million wheelchair users (National Health Service England 2020), and the prices of powered wheelchairs have been reducing over the years.

Usage of powered wheelchairs has been shown to lead to improvements in mobility and quality of life of patients, besides a reduction in pain/discomfort levels (Davies, De Souza, and Frank 2003). But despite the benefits brought by this technology, prescribers often feel worried about the safety of their patients, especially if they are children or elderly (Evans et al. 2007; Mortenson, Clarke, and Best 2013), as accidents involving this equipment are not uncommon (Frank et al. 2000). Moreover, usage of a powered wheelchair may be challenging for many people, with up to 40% of patients finding it difficult or impossible to manoeuvre them (Fehr, Langbein, and Skaar 2000).

To mitigate these problems, a powered wheelchair might be augmented with sensors, enabling it to perceive its surroundings. Then, this newly added information may be used by a central 'intelligent' unit to aid the user during wheelchair navigation. This aid might come in different forms, such as: safer or faster driving, help in navigating crowded spaces, door traversal, cognitive workload reduction, etc. In any case, this enhanced device is then called a robotic or smart wheelchair, and in (Simpson, LoPresti, and Cooper 2008) the authors estimate that over 60% of wheelchair users could benefit from using this technology.

The most common architecture of smart wheelchair rely on the user input (which may come from a joystick, a sip-and-puff device, a touch screen, etc.), to perform collision avoidance, obstacle avoidance or autonomous navigation to a given point on a map. The algorithms used to perform these actions, however, are usually fixed and user-agnostic; that is, there is no customisation to an individual's specific needs. Although helpful to a significant part of the reduced mobility community, this approach may not be as relevant to another group of wheelchair users. For example, people suffering from Parkinson's disease, which can lead to severe tremors and inability to walk, should require a different kind of assistance from those that had a spinal cord injury without any upper-body impairment. And users suffering from a cerebrovascular accident, cerebral palsy, amyotrophic lateral sclerosis, multiple sclerosis, etc., may yet again have different needs. Even within the same disease, there are cases where some form of customisation of the assistive policy is likely to be needed. Consider, for example, the case of people suffering from multiple sclerosis and who use a powered wheelchair as their mobility devices. Among a range of possible symptoms, these users can suffer from muscle weakness and/or muscle spasms (Compston and Coles 2008). Then, it is evident that the appropriate assistive policy will be different for users experiencing each of the symptoms. In all these cases, a user-agnostic system can lead to more frustration than support.

While in theory it should be possible to develop hard-coded assistive policies to aid with specific impairments, this becomes impractical when considering the myriad of diseases and symptoms that may affect a person. Instead, it seems like a better approach to learn the assistive policy directly by observing how suitable professionals, like physiotherapists and caretakers, do it. This concept of using expert demonstrations to teach a robot how to execute a task is termed robot Learning by Demonstration.

1.2 Learning by Demonstration

A common problem in robotics is the process of mapping a robot's observed state corresponding action that it should take. This mapping, or policy, enables a robot to autonomously (at least partially) operate in its environment, instead of being teleoperated. However, manually engineering these policies can be very challenging, because of the wide variability in the number of states the robot can encounter. Another challenge is that, for some particular states, it may be difficult to clearly specify what actions should be taken.

As a result, many researchers have turned to machine learning (Bishop 2006) for policy development, with robot Learning by Demonstration (LbD) (Argall et al. 2009) being a popular approach. LbD can be seen as a subset of supervised learning: "In Supervised Learning the agent is presented with labelled training data and learns an approximation to the function which produced the data. Within LbD, this training dataset is composed of example executions of the task by a demonstration teacher" (Argall et al. 2009). Hence, these example executions form a dataset of action-state pairs, which can be used by LbD algorithms to extract the underlying policy behind the task being demonstrated. Through this approach, LbD presents a simple way of transferring expert knowledge to a robot on how to execute a particular task. An example application is shown in Figure 1.2.

A sub-field of LbD is Learning Assistance by Demonstration (LAD) (Soh and

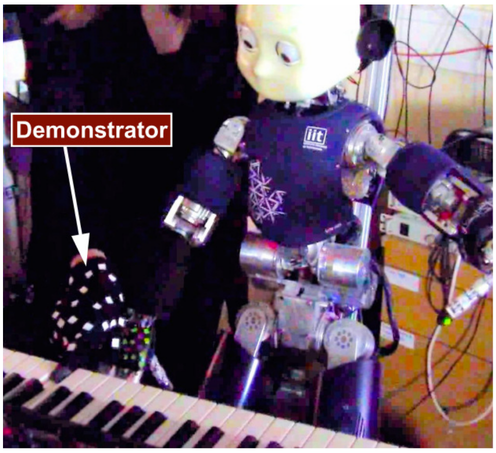


Figure 1.2: Example of robot Learning by Demonstration, where a teacher shows a robot how to play the keyboard. Excerpt from (Zambelli 2018).

Demiris 2015a), which is concerned with using demonstrations to teach a robot how to provide physical assistance. In this case, the expert is someone who is familiar with the specific disability of the person being helped, and also knows, through training or experience, how to best support that person. This expert, however, does not need to have knowledge in robotics or machine learning. Therefore, healthcare workers, such as nurses, caretakers or physiotherapists, should be suitable professionals to provide personalised help demonstrations. The main benefit of this approach is that it allows the robot to learn assistive policies that are customised to the needs of each individual, whatever their disability, symptoms or preferences.

1.3 Research problems

As discussed, non-roboticists can use LAD to teach robots how to help other people (users). For this, one only needs to record demonstrations of the triadic interaction between teacher-robot-user, and then feed the recordings to a pre-defined learning algorithm. Hence, the procedure can easily be repeated for multiple users, thus enabling the generation of *personalised* assistance. In the case of smart wheelchairs, personalisation is a crucial characteristic, as users can be impaired in innumerable ways and often need customised assistive solutions. More specifically, we consider the cases of people that, besides needing a powered wheelchair for mobility, also present hand-control impairments that make it difficult to handle a joystick, e.g., tremors, spasms, muscle weakness. This combination is a common outcome for a variety of diseases, such as cerebral palsy, Parkinson's disease and multiple sclerosis (Tortora and Derrickson 2016).

We believe that LAD can play a crucial role in making smart wheelchair technology more adaptable, and thus suitable to a greater number of people. Through this thesis, we shall explore this hypothesis in more detail, posing it as our principal research topic. Successful investigation of the topic, however, requires attention to some problems.

How can a human assistant intuitively provide demonstrations of navigation support to powered wheelchair users? The first step in the LAD paradigm is recording demonstrations of the assistive policy being executed by the expert. In the application considered, however, a mobile robot is involved, which complicates the demonstration task. One approach would be to have the assistant walk by the wheelchair, following the driver and helping with the joystick control when assistance was needed. But in this case, the robot would have no way of differentiating between the driver's input and the target signal, coming from the assistant. Instead, the assistant should have access to a dedicated joystick, which can remotely control the wheelchair. Furthermore, if the assistant can be comfortably sited in a base station, instead of constantly following the wheelchair, even better. That is the approach employed by previous work in this field (see Section 2.3.2) and also the approach we follow. However, this raises the issue of context understanding. When the assistant is observing the driving action from a remote location, spatial awareness and inference of intention can be compromised. To compensate, special interfaces might be used to aid the assistant in intuitively interpreting the data registered by the robot's sensors. We tackle this problem in Chapter 3.

How can personalised assistive policies be learned by observing the triadic interaction between a robotic wheelchair, an impaired driver and an expert human assistant? Once recordings of assistive demonstrations are available, they can be used to teach the robotic wheelchair how to automate the support provided by the human expert. In principle, any supervised learning algorithm could be used at this stage. But the application at hand poses several challenges in terms of machine learning, such as: the limited amount of data available for training, the high dimensionality of input data and the uncertainty in the quality of the demonstrations. To contend with these challenges, careful consideration is needed when designing the teleoperation platform, the learning algorithms and the methods used to evaluate assistive performance. We address this problem in Chapter 4.

How to improve the generalisation capabilities of LAD models? Although LAD has been used with robotic wheelchairs before, one key aspect was not thoroughly investigated by prior work: are the assistive policies learnt from demonstrations on a training course capable of generalising to physically different scenarios? Good generalisation is crucial property for a practical application of LAD, but our tests showed that it cannot be assumed. To increase generalisation performance, it is necessary to develop new and improved learning models. But comparing the assistive benefits of different models is not an easy task, because confounding variables on experimental data, such as drivers' level of concentration and learning effects, can severely impact performance. To remove these confounding variables and facilitate the development and testing of better LAD models, one can resort to repeatable simulations - the option we explore in Chapter 5.

Combining the topics discussed above, we can formulate the main research problem to be addressed by this thesis as:

> Use demonstrations of the triadic interaction between a robotic wheelchair, a driver with hand-control impairments and a remote assistant to generate personalised assistive policies that improve navigation performance.

We emphasise, however, that assessing the performance or applicability of LAD with *real* wheelchair users is not within the scope of this work. Instead, we aim to develop the best teleoperation platform, algorithms, models and training procedures that turn LAD into a strong, viable, option for generating personalised navigation support. Accordingly, our pilot user studies and simulation environment are designed and used to keep in check our assumptions about the features of LAD and the dynamics of the triadic interaction which it concerns. Tests with the target population are foreseen as a next step in this line of research.

When robots are endowed with LAD, they become more flexible and therefore better able to provide personalised support to people. Hence, by bringing human assistants into the learning loop of assistive robots, we expect these robots to finally escape laboratories and reach wider adoption among end-users.

1.4 Contributions

The main contribution of this thesis is a 'full-stack' (from multimodal teleoperation hardware to learning algorithms that improve generalisation) body of research on improving the operationalisation of the LAD methodology. This is broken down into the following original contributions:

- **Develop a custom teleoperation platform which allows an assistant to remotely help a wheelchair driver**. The platform uses multimodal interfaces - which are based on Simultaneous Localisation and Mapping (SLAM), haptic-pairing and eye-gaze estimation - to support the assistant on the task of remote inference of intention. Furthermore, the platform allows the assistant to observe the driver's operation either through a computer monitor or using a virtual reality headset, for an increased sense of presence and improved depth perception. The platform also implements a new ask-for-help paradigm, for determining when assistance should be provided. Compared to prior approaches (Soh and Demiris 2015a), this new paradigm simplifies the learning problem of LAD, while also allowing the driver to retain more control of navigation. In all this, we contribute both the hardware setup and the software needed to synchronise and process incoming data in real-time. The teleoperation platform is left as a legacy and thoroughly documented, allowing other research teams to recreate similar setups.

- **Create a computer simulation that can reproduce the full triadic interaction of LAD for smart wheelchairs**. The software can not only simulate the dynamics of wheelchair driving and teleoperation, but also simulate the behaviours of the human driver or/and the human assistant. This last option, where all three agents are simulated, is paramount for creating repeatable interactions, which are needed for quickly comparing different assistive models without the interference of confounding variables. Furthermore, the simulator enables the procedural generation of a range of disabilities, which impose varying levels of hand-control impairments. These impairments make navigation harder by taking clean input commands from human or simulated driver and mapping them to noisy and distorted ones. This software will be made open-source and freely available, to accelerate this line of research.

- **Study uncharted characteristics of LAD**. LAD deals with an unique paradigm of LbD, where the robot has to learn from the triadic interaction between expert, robot and person being assisted, instead of the more simple dyadic interaction between expert and robot. And although LAD has been studied before, many of the features and limitations of this novel learning paradigm are still unexplored. To help cover this knowledge gap, this thesis investigates some important characteristics of LAD. For example, we make a distinction between predictive performance, which only compares model predictions to the ground truth signal provided by the assistant, and assistive performance, which measures the actual benefits generated for the driver. For the latter, we employ diverse metrics to assess multiple dimensions of assistance: time to complete a lap in a *test* obstacle course, deviation from the optimal path, number of interventions needed by the assistant, and amount of time spent clearing collisions. We also assess if LAD models are robust to variations in the distribution of input data, i.e., if they can be fitted to work with different hand-control impairments without needing adaptations in model architecture. Tests were also carried out to analyse if the generated assistive policies are indeed personalised to the disabilities and specific needs of individuals, instead of generic input-output mappings that could be achieved by simpler means. We also conducted two pilot user studies: one to assess the impact that the multimodal interfaces of our teleoperation platform have at improving the assistant's capability of inferring the driver's intention; and another to evaluate if the learnt assistive policies are capable of generalising to physically different scenarios.

- **Test and improve generalisation to physically different scenarios**. Previous work that used LAD for robotic wheelchairs (Soh and Demiris 2013; Soh and Demiris 2015a; Kucukyilmaz and Demiris 2015; Kucukyilmaz and Demiris 2018) only tested model performance in the same physical location where the demonstrations of assistance were conducted. We show that this can lead to assistive models that overfit to the spatial features of the training environment, and hence would not be very useful for drivers in their daily lives (i.e., outside of the training environment). Through our simulator and pilot user study, we observed that creating models that can generalise well enough to operate in unseen locations is a significant challenge. To overcome it, we operate in multiple fronts: use our custom teleoperation platform to avoid the correspondence problem in LbD (Argall et al. 2009); use dedicated validation sets to prevent overfitting; explore different techniques for reducing the dimensionality of input data (sub-sampling, Principal Component Analysis (PCA), Autoencoders, MaxPooling); experiment with different families of machine learning algorithms (linear regression, Support Vector Machines, Gaussian Processes, multiple types of neural networks architectures) seeking increased generalisation performance; and investigate alternative model training procedures (multimodal deep learning (Ngiam et al. 2011), Learning Using Privileged Information (Vapnik and Vashist

2009)). Our final machine learning solution, which we call Generalised Learning Assistance by Demonstration (GLAD), is carefully documented and will be made open-source and freely available.

1.5 Thesis outline

The remainder of this document is structured as follows:

- Chapter 2 brings forward related work in which this thesis builds upon. This includes general work in smart wheelchairs, robot Learning by Demonstration in multiple domains and Learning Assistance by Demonstration for smart wheelchairs.

- Chapter 3 describes a custom teleoperation platform that was developed to enable a remote assistant to provide help to a disabled driver. This platform makes use of multimodal interfaces to aid the assistant in interpreting the raw data registered by the wheelchair's sensors. The chapter closes by describing a user study conducted to assess the impact that these interfaces have on the assistant's ability to infer the driver's intention. This chapter is an extension of the work presented in (Schettino and Demiris 2019).

- Chapter 4 explains how assistive policies can be learnt directly from demonstrations of helping a particular driver. Then, design considerations regarding how to implement LAD are discussed, while also comparing our decisions against those of previous work. Lastly, the chapter examines a user study conducted with a special data collection procedure, used to test the generalisation capabilities of LAD. This chapter is an extension of the work presented in (Schettino and Demiris 2020).

- Chapter 5 opens by describing a custom simulation environment developed to reproduce the full triadic interaction of LAD for smart wheelchairs. Using this simulator, it was possible to conduct multiple runs of experiments, studying important characteristics of LAD while collecting more statistically significant results. These results were then used to inform how to improve generalisation in LAD. The chapter also describes the custom neural network architecture used and the steps taken to arrive at it.

- Lastly, Chapter 6 presents finishing remarks and formally addresses the main research problem of this thesis. Limitations and possible lines of future work are also discussed.

This outline is illustrated in Figure 1.3, which emphasises the linear progression of chapters while building up to address this thesis' main research problem. The central topics discussed in each chapter are also highlighted.

Figure 1.3: Thesis outline. Figures are shown in full size at their respective chapters.

Chapter 2

Background

Before addressing the research problems raised in Section 1.3, it is important to first understand the background work in which this thesis builds upon. Hence, this chapter surveys the scientific literature on the topics of smart wheelchairs, estimation of intention, Learning by Demonstration and simulators, always under the lens of assistive robotics. We note, however, that our goal here is not to present an extensive literature review on any of these topics, but rather to explain, illustrate and clarify important concepts that will be used later on, and also to reinforce the motivation for our work. In order to understand current trends of the aforementioned fields and what technologies are presently available, more focus is given to works of the past decade.

2.1 Smart Wheelchairs

The invention of the first powered wheelchair is disputed, but it has been commercially available since at least the late 40's (Woods and Watson 2003). The standard self-propelled wheelchair demands significant physical effort and may not be a viable solution, depending on the user's disability. Hence, the technological shift brought by the powered wheelchair was the solution that eventually allowed many to finally achieve their mobility independence.

For people with quadriplegia or severe upper-body impairments, who cannot use a standard joystick to control the wheelchair, some alternative input methods exist (Winkler et al. 2016; Sinyukov et al. 2014; Bastos-Filho et al. 2014; Wästlund et al. 2015). Commercially available options include headrest array, sip-and-puff and chin joystick. However, in these cases navigation can be more difficult (Erdogan and Argall 2017), due to the input commands requiring more dexterity, being discrete instead of continuous, having lower dimensionality, presenting higher latency, or a combination of these.

Even with the traditional joystick, operation of a powered wheelchair can be challenging. A study (Torkia et al. 2015) with 12 users (14 years of driving experience on average) identified that drivers typically had difficulties with common navigation tasks, such as going through doorways, moving backwards and avoiding obstacles. The study was conducted with users that had a range of primary diagnoses (including multiple sclerosis, spinal-cord injury, rheumatoid arthritis, etc.), thus indicating that the problem is not specific to a class of patients. The authors also notice that the difficulties reported are similar to the ones experienced by novice users, suggesting that the skills necessary to overcome these difficulties might be too hard to acquire even after years of experience.

In another study (Kairy et al. 2014), interviews were conducted with 12 powered wheelchair users and 4 caregivers. It was concluded that many of the challenges encountered by these people could be overcome by using features commonly present in Smart Wheelchair (SW) prototypes, such as obstacle avoidance, path following and target following. When asked, half of the participants said that they would use a Smart Wheelchair (SW) if the technology was available for them. Similar observations were made in (Padir 2015), where over a hundred interviews were conducted with powered wheelchair users, therapists, manufactures and other stakeholders.

Research on SWs date as far back as 1986, with the first known prototype being described in (Madarasz et al. 1986). Although commercial products have not yet truly incorporated benefits deriving from this research (Leaman and La 2017), work on the area has been extensive and covered a variety of topics over the years. Here we discuss those topics that are most relevant to this thesis.

2.1.1 Assistance paradigms

Perhaps the most explored subject in SW research has been the different assistance paradigms that are possible when it comes to navigation support. In general, there is a conflict related to the amount of assistance that should be offered to the user. On one hand, more system autonomy may yield better performance when quantitative metrics, such as lap-completion time and the number of collisions, are considered (Erdogan and Argall 2017). On the other hand, drivers usually prefer to maintain control over the actions they are physically fit for (Kairy et al. 2014; Viswanathan et al. 2017). Therefore, removing user autonomy may lead to frustration and lower performance on qualitative metrics, such as comfort and enjoyment of the system (Kim et al. 2012).

Essentially, there are four fundamental types of assistance paradigms (including no assistance):

1. **No assistance:** this is the regular case for a powered wheelchair, where the driver is in full-control at all times.

2. **Full system autonomy:** in this case, drivers are usually presented with a map of their current environment and allowed to select a point on this map. After this, the system takes control and autonomously plan and navigate to the desired location without user intervention, while simultaneously avoiding unmapped obstacles.

3. **Shared control:** an intermediate point between the first two. Both driver and robot continuously provide control signals, and these are blended in one of two forms:

    (a) **Collision avoidance:** the robot does not alter the direction of movement. When approaching an obstacle, only the linear speed is capped. The maximum value is given by the robot's local obstacle avoidance system and continuously updated. If the driver was to attempt a collision against an object, the wheelchair would come to a halt.

    (b) **Obstacle avoidance:** when approaching an object, the angular speed of the wheelchair progressively change from the one given by the user to the one given by the robot's local obstacle avoidance system. The closer the SW gets to an object, less control the driver will have. The desired linear speed is usually not changed. If the driver was to attempt a collision against an object, the wheelchair would automatically deviate from it, without stopping.

Many works can be found on the literature where a SW prototype is developed and then a pairwise comparison of performance between two of the modes above is performed - one of them usually being 'no assistance'. However, until recently, no work had systematically explored all the paradigms.

In (Erdogan and Argall 2017), the authors compare the four variants above and an extra one, where the user signal is blended with automatic navigation to a nearby high-level goal (such as crossing a doorway). The goal is automatically determined from the SW's sensors input. In a quite complete study, the comparison is done regarding both performance metrics (task completion time, distance to obstacles and others) and subjective metrics (questionnaire); including different kinds of input devices (joystick and headrest array); independently considering trials performed in different days, to assess learning behaviours; and including both able-bodied and spinal-cord injury subjects. Among many findings, it was reported that: 1) more system autonomy tends to increase performance metrics, although, statistically, none of the options was significantly better; 2) users prefer not to relinquish control; 3) richer input methods (joystick) benefit less from system autonomy than simpler ones (headrest array); 4) only a few differences were observed between subjects with and without spinal-cord injuries; 5) there was high variability in the chosen preferred method.

Another study (Viswanathan et al. 2017), also comparatively evaluated different assistance paradigms, but this time only assessing it with older adults with cognitive impairments. Some methodology differences were present, however: only obstacle avoidance, collision avoidance and full autonomy modes were compared (due to subjects' cognitive impairments, *no assistance* was deemed unsafe); participants tested the system in different scenarios, such as hallway navigation, entering an elevator and back-in parking; no performance metrics, only qualitative feedback from semi-structured interviews were collected; longer assessment, over two weeks; vibration or audio feedback was used to indicate when assistance was being offered; and finally, it was a Wizard-of-Oz study. That is, the SW prototype did not have real autonomous assistive capabilities - one of the authors was remotely operating the wheelchair to simulate assistance when needed, but drivers were unaware of it. Some of the main findings of this study were: 1) participants wanted to be kept in the control loop as much as possible; 2) participants desired different levels of system autonomy depending on the context (during back-in parking, for example, more assistance was considered beneficial) and on their cognitive state at the moment; 3) vibration feedback was in general appreciated, but some users wanted richer feedback, while others wanted less information about the system operation; 4) 9 out of 10 participants found that some level of assistance was useful; 5) none of the methods was significantly preferred over the others, but obstacle avoidance was more likely to be picked.

Besides the four main assistance paradigms described above, some variants have been proposed. In (Sanders 2017) the author presents a shared control approach where the blending factor is determined not only by the proximity to obstacles but also by safety and tiredness factors. The safety factor is determined by the wheelchair speed - with lower speeds the operation is deemed safer and the user can get closer to obstacles without automatic assistance intervention. The tiredness factor is determined by how long the user has been driving throughout the day - longer times are assumed to lead to tiredness and lack of attention, therefore the assistance becomes more conservative. In

(Narayanan, Spalanzani, and Babel 2016), instead, short term goal prediction and humans personal space were used to adjust the blending factor for shared control. In (Goil, Derry, and Argall 2013), the variability in travelled paths during driving demonstrations is used to determine path-specific blending factors; a concept that has also be used for robot-assisted manipulation tasks (Abi-Farraj et al. 2017).

An expansion of the *full system autonomy* approach was explored in (Leishman et al. 2014). Instead of showing users a fixed map, or a list of actions, they were presented with contextual assistance options. For this, a camera is attached to the front of the SW and the video is displayed to the user on a screen. Image processing techniques are used to detect specific features, such as doorways and corridors, and rectangles are superimposed over these features on the live image. Thus, the user can manually operate the wheelchair, without any assistance, but also can easily switch to automatic control when desired, simply by clicking the rectangles on the touch-screen.

The blending approach for shared control is formally defined and discussed in (Dragan and Srinivasa 2013b), while (Dragan and Srinivasa 2013a) describes its use specifically for assistive teleoperation. The concept of switching between different levels of autonomy is commonly known as adjustable autonomy and is thoroughly studied in (Mostafa, Ahmad, and Mustapha 2017). Finally, the overall theme of autonomy for rehabilitation robotics is reviewed in (Argall 2018), including a discussion on smart wheelchairs.

2.1.2 Haptic feedback

Current literature on SWs indicates that providing feedback about the assistance that is being offered is considered a key feature by powered wheelchair users (Padir 2015). Despite this, many modern prototypes do not offer it in any form. Absence of this feature may have adverse effects as a consequence, such as drivers not understanding why, when or how assistance is being offered. In turn, this can contribute to an overall frustration or even rejection of the robotic wheelchair system (Padir 2015).

To provide this feedback to the user, some options are available, such as visual, audio and vibration cues (Zolotas, Elsdon, and Demiris 2018; Zolotas and Demiris 2019; Wang et al. 2011; Viswanathan et al. 2017). But an option that seems particularly interesting, and has been more explored, is the use of haptic interfaces (Kuchenbecker 2018), such as joysticks with force-feedback capability. These devices are able to exert on the joystick handle a programmable force that can be updated thousands of times a second, thus giving the user a sensation of real-time interaction.

To our knowledge, the first work to examine the usage of haptic joysticks to control a SW was (Brienza and Angelo 1996). There, it was explained how the exerted force feedback should be inversely proportional to the distance between the wheelchair and the nearest obstacle; and in opposite direction. Furthermore, this force might be exerted in two different modes, which can be related to previously discussed collision and obstacle avoidance. In passive (collision avoidance) mode, the applied force merely attempts to diminish the wheelchair's linear velocity. This is done by increasing the force in the joystick towards its central position, which would stop the wheelchair. In the active (obstacle avoidance) mode, the attempt is to maintain the user's desired linear speed. Thus, the force feedback actuates to change the angular speed, by laterally pulling the joystick towards the position that would make the wheelchair move to the closest obstacle-free area. In either case, however, the user might overrule this feedback force and drive into or near the obstacle, if they so desire.

Since this seminal work, many different variants were explored. More recently, in (Hadj-Abdelkader, Bourhis, and Cherki 2012), a SW was equipped with a laser-scanner and a haptic joystick. Ten able-bodied subjects were asked to drive through three predefined courses of varying difficulty, using both passive and active modes and also no assistance. It was observed that the force feedback assistance, especially in active mode, tended to reduce lap-completion time and number of collisions, without significantly increasing the cognitive workload. In (K. Narayanan et al. 2016), a vision-haptic system was used to improve safety in corridor-following. (Vander Poorten et al. 2012) describes an attempt to improve the haptic guidance technique by taking into account the non-holonomic nature of wheelchairs and planning circular, obstacle-free, paths; but only preliminary tests with the method were conducted.

Despite decades of research on the area, only a few works conducted tests with disabled people. (Morère et al. 2015) recently explored this, albeit only using a SW simulator, instead of a real prototype. Five children with motor and cognitive disabilities, all regular users of powered wheelchairs, were asked to drive through a course with obstacles, both with and without force feedback activated. It was found that, in general, the force feedback was beneficial to the navigation task, although for some users it had negative effects (the children felt like they had to fight against the wheelchair to control it).

A review on the usage of haptics to improve task performance by people with disabilities is presented in (Jafari, Adams, and Tavakoli 2016), with a section specifically devoted to SWs. (Petermeijer et al. 2015) surveys the effects that haptic support systems have on car drivers, finding that "In general, it can be concluded that a warning or guidance system provides benefits for the driver in terms of an improved performance, reduced reaction time, and reduced mental workload". Related works are also covered in (Hadj-Abdelkader, Bourhis, and Cherki 2012) and (Hadj-Abdelkader, Cherki, and Bourhis 2015).

2.1.3 Customisation and metrics

An interesting question to be asked is: why, after more than 30 years of research in SWs, almost none of the technological advances made it into a commercial product (Leaman and La 2017)? The question becomes especially intriguing when one considers the potential market size (Padir 2015; Fehr, Langbein, and Skaar 2000).

In (Leaman, La, and Nguyen 2016), the authors argue that one of the explanations for this is that most of the prototypes previously developed "only serve a small sub-population of SW users". Indeed, considering the high variability in limitations that might be imposed by a myriad of disabilities, it is hard to imagine that a single system architecture could be appropriate for all users. As a result, many recent works have discussed the importance of developing systems that can be customised for the final user.

For example, in (Leaman and La 2017) it was noted that there is not a single best input method for SW control. Instead, the best choice depends on each user individuals capabilities, needs and, ultimately, preferences. Likewise, the most appropriate assistance paradigm is usually dependent both on the chosen input method and on the user's preference (Erdogan and Argall 2017). (Padir 2015) further suggests that, besides input devices and assistance paradigms, customisation should also be allowed in feedback modes and levels. Finally, (Viswanathan et al. 2017) notes that the motor and cognitive states of some drivers might change over time, even in the course of a day. Therefore, the authors argue that these drivers should be able to choose the control option that best suits them at a given moment.

In (Chang et al. 2017) a machine learning approach to user customisation is presented, considering 'full system autonomy'. When the user selects a point on the map, multiple possibles paths to the destination are generated, instead of just one. The user is then asked to select one of these paths. After some iterations of this, the system can learn the user's preferences and automatically follow it, without asking.

In summary, the recent literature shows that many research groups have independently realised that there is not a 'one size fits all' solution. Consequently, they find customisation to be a key feature for SWs (Kairy et al. 2014; Padir 2015; Erdogan and Argall 2017; Leaman and La 2017; Viswanathan et al.

2017).

Because of the variability in users' conditions, assessing how well an assistive device is performing can be difficult, since competing goals might be present. In the case of SWs, common metrics include, but are not limited to: task completion time, number of collisions, users' physical effort, cognitive workload and agreement to suggested assistance. The most appropriate metric is usually dependent on the task being performed, but, in general, a combination of these can be adopted. However, one should note that performance measurements do not directly translate to user satisfaction (Javdani et al. 2018; Kim et al. 2012; Gopinath, Jain, and Argall 2017), ergo it is important to also collect users' subjective feedback. A review of the many metrics that have previously been used in the field of assistive navigation is presented in (Leishman et al. 2014).

The smart wheelchair literature contains numerous other works (Qiang Zeng, Burdet, and Chee Leong Teo 2009; Montesano et al. 2010; Carlson and Demiris 2012; Burhanpurkar et al. 2017), and an extensive review is presented in (Leaman and La 2017). Besides covering different topics on the field, it also shows that this line of research is still very active, with 39 prototypes developed in a decade, from all over the world. Other reviews can be found in (Simpson 2005; Desai, Mantha, and Phalle 2017; Wahyufitriyani, Susmartini, and Priadythama 2016).

2.2 Estimating intention

A recurrent theme in shared control and assistive robotics is estimation of intention. It is an important topic because, in order to correctly assist the human, the robot must share their goal, which is not always obvious to infer. For instance, in (Dragan and Srinivasa 2013b) a study was conducted where people used their body motions to teleoperate a robot while attempting to make it grasp one of two objects in a table. Because the control interface is not perfect, the robot can share control of the task, attempting to make it easier for the human. If the robot knew a priori which of the objects the person meant to grasp, it could take over control and autonomously execute the task. But since this information is not available, it has to first estimate the person's goal and only then provide assistance.

This 'predict-then-act' approach, however, can lead to a situation where the robotic assistance is idle during most of the time, due to not being able to confidently infer the human's goal. To cope with this, (Javdani, Srinivasa, and Bagnell 2015; Javdani et al. 2018) propose an algorithm which represents the human's goals as latent states in a Partially Observable Markov Decision

Process (Kaelbling, Littman, and Cassandra 1998), enabling the provision of assistance even when confidence on the human's intention is low. This happens in two forms: if the same assistive action is beneficial for reaching all goals, that action is taken. On the other hand, when the optimal assistive action for reaching different goals diverge, the robot delegates more control to the person and use their input to update the belief space over the set of possible goals. As the confidence over a specific goal is increased, the robot can regain more control and assist the person to reach it. When additional cues about the human's goal are available, such as eye-gaze, they can also be used to improve the intention estimation process (Admoni and Srinivasa 2016).

An issue with these approaches, however, is that they assume the existence of a finite set of goals, which is known a priori. Yet, for many applications in assistive robotics both of these assumptions can be broken, i.e., the set of possible goals may not be known beforehand and there may be an infinite number of goals (continuous goals space). To handle these cases, alternative solutions exist. (Koert et al. 2019) describes an application where the set of goals is not known beforehand but is assumed to be discrete and finite. In this case, an incremental Gaussian Mixture Model (Engel and Heinen 2010) can be used to continuously update the set of possible goals. The case of a continuous goal space can be handled by doing away with explicitly estimating the human's intention and instead reactively providing assistance. In (Reddy, Dragan, and Levine 2018), reinforcement learning (Kober, Bagnell, and Peters 2013) and a task reward function is used to learn a personalised map between a person's input and the optimal action. In (Du et al. 2020), it is proposed to assist the human simply by increasing their controllability of the environment at each moment, thus removing the need to explicitly infer their intention.

There also exist works specifically on estimating the intention of wheelchair pilots. An early example is (Carlson and Demiris 2008), where a multiple hypothesis method (Demiris and Khadhouri 2006) is employed to distinguish between which doorway (including none) the driver wants to go through. In this method, the human's possible actions are represented by inverse models. Then, the robot's current observed state is compared against the states predicted by the inverse models, thus leading to a belief space over the possible goals. In a more recent work (Narayanan, Spalanzani, and Babel 2016), it was shown how the problem of uncertainty over the intention estimation can be mitigated by only inferring short-term goals. These goals are sub-optimal and agnostic to the long-term intention of the user, but they are also continuously updated. If the goal is incorrectly predicted, it is discarded and a new prediction is made. If, on the other hand, the driver's input is in agreement with the predicted goal, assistance from an appropriate motion planner can be provided.

A review on intention detection, arbitration and feedback for shared control in physical human-robot interaction is presented in (Losey et al. 2018).

2.3 Learning by Demonstration

One of the main purposes of assistive robotics is to replace, completely or partially, the assistance that can be offered by healthcare professionals. This, however, is not an easy task. The support that nurses, caretakers and physiotherapists can provide is not only of high quality but also context-dependent and adaptive. To fully capture this type of support, with all its nuances, into hard-coded assistive policies would be an incredible engineering feat.

An alternative approach is to use robot LbD[1] (Argall et al. 2009). In this case, a human expert, or teacher, is asked to provide demonstrations of how to perform a specific task, while being recorded. The data is then used with a machine learning algorithm[2] to teach the robot how to autonomously replicate the execution of that task. If learning is appropriate, the robot should also be able to generalise to reasonable variations of the same task.

As an example, in (Bojarski et al. 2016; Bojarski et al. 2017) it was shown that a car can learn to autonomously drive directly by observing a human doing it. As input, the learning model only had access to the images of three fixed cameras positioned behind the car's windshield, and the driver's steering wheel angle was used as a target signal. Using 72 hours of driving demonstrations, the model learned how to autonomously steer the car, even in environments not seen during the data collection phase. The main benefit of this data-driven approach is that the human's implicit knowledge about the dynamics of the driving task, such as traffic laws and the physics of car motion, do not have to be hard-coded in any way. Instead, they are indirectly captured through the driving demonstrations. Using a Deep Learning (Lecun, Bengio, and Hinton 2015) architecture, the learning task was further simplified by directly using the images as input, removing the need of hand-crafted features that are often task-dependent (e.g. lanes or road outline detection in this case). This approach was also employed by (Zhang et al. 2017), which used LbD and camera images as input to teach a robot how to perform common manipulation tasks, such

[1] Also known as 'Learning from Demonstration' or 'Programming by Demonstration'.

[2] Here we only discuss works related to a subset of LbD commonly know as Imitation Learning, or Behavioural Cloning, which uses supervised learning (Bishop 2006). Another approach is the use of Inverse Reinforcement Learning (Abbeel and Ng 2004), which we refrain from discussing here to limit the scope of this section only to the research that is most relevant for this thesis.

as grasping or pushing an object. In this case, the teacher used a virtual reality teleoperation platform, which improves the quality of demonstrations by removing the correspondence problem (Argall et al. 2009). In turn, this reduces the amount of data needed for appropriate generalisation.

Within assistive robotics, a fruitful field for the use of LbD is physical rehabilitation training. In this case, a short movement usually has to be repeated by the patient several times, while supervised or assisted by a physiotherapist. To maximise the amount of training that the patient can receive, their initial interaction with the physiotherapist can be recorded and used to teach a robot to mimic this cooperation. This was implemented in (Najafi, Adams, and Tavakoli 2017), where, during an initial training phase, patient and therapist jointly held the handle of a rehabilitation device and performed a pre-defined movement, useful for restoring the patient's physical capabilities. After this, the same device can be used unattended, helping the user to execute the same movement and providing varying levels of assistance, according to the therapist's demonstration. Similar work is presented in (Ewerton et al. 2018), but for teaching someone the execution of a new movement. Here, the teacher's execution of the movement can be independently recorded and used by the robot to learn the appropriate motion. When a non-expert attempts to mimic the exercise, the robot may provide visual or haptic assistance, if a sub-optimal execution is detected.

The case of rehabilitation/movement training, however, is somewhat simplified by the fact that the same movement is always repeated. Hence, the intention of the assisted person is always known beforehand - to execute said movement. But this is not the case for many other applications in assistive robotics. (Losey et al. 2020) discusses the interesting application of a wheelchair-mounted robotic arm. In this case, the wheelchair user has to control the 6D pose of the arm's end-effector using only the joystick, a 2D input device[3]. Although this might be done by using switches to alternate between the control of different degrees of freedom, it is slow and unintuitive for the user. Hence, the authors propose to use LbD to learn a mapping between high-dimensional robot actions to low-dimensional latent input actions. For this, kinaesthetic teaching is first used to show the robot how to perform a given task. Then, a Variational Autoencoder (Kingma and Welling 2013) is employed to reduce the six degrees of freedom of the arm's end-effector to the 2D latent dimensions allowed by the joystick. To make the mapping more usable and intuitive, the latent space is conditioned on the robot's state, which allows for different actions to be performed by the same input - e.g., from the initial state, pushing up on the joystick will move up the arm, but if the gripper is holding a glass the same input will make the end-effector tilt. In another work (Koert et al. 2019), a human and a robot share the same workspace and complement each other actions to achieve a common goal. To make the collaboration more efficient and safer for the human, the robot must be capable of estimating the intention and movements of the user and act appropriately to avoid collisions. For this, first kinaesthetic teaching is used to teach the robot how to execute their portion of the task. The learned behaviour is extracted into Probabilistic Motion Primitives (Paraschos et al. 2018). Then, at each time step, the intention of the person is estimated, and if a potential collision between the robot's and the human's (predicted) planed paths is detected, the motion primitive is adapted online.

2.3.1 LbD for smart wheelchairs

Using driving demonstrations to aid in the derivation of assistive policies for smart wheelchairs has also been explored. In (Goil, Derry, and Argall 2013), for example, the variability in angular speed commands issued by drivers when travelling through different paths is used to determine, at each point in time, how much weight should be given to autonomous driving in a shared control scheme. For this, several driving experts are first asked to navigate a given route. Then, the variation in the driving paths is used as an indicator of how much flexibility the disabled user should have in that location. If there is little variability, such as should occur on a narrow doorway crossing scenario, it indicates that there is not much space for erratic manoeuvres, and therefore the blending factor should favour more system autonomy. In an ample hallway navigation, on the other hand, more variation is expected from the demonstrations and, consequently, the user gets more flexibility when driving on that area. In (Matsubara et al. 2015), instead, driving demonstrations are used to learn how to predict a driver's long-term goal in a known map. Having a known map and an estimated goal, path planning and autonomous driving can be obtained by using off-the-shelf techniques available in mobile robotics.

Other works used the expert demonstrations to directly learn desired navigation behaviours. In (Chow and Xu 2006), an expert performs driving demonstrations while being recorded. From the recorded data, a lookup table is formed, combining sensor readings and the corresponding control signal. Afterwards, any sensor reading from the robot can be matched against the closest point on this table, and the corresponding control action can be used to enhance or replace the input from the disabled driver. In (Poon et al. 2017), the

[3] For comparison of performance, in their user study a video-game joystick was used instead of a wheelchair's one, thus also allowing direct control with six degrees of freedom.

authors further motivate the usage of LbD by arguing that expert demonstrations might implicitly encode desired driving behaviours for specific scenarios, which would otherwise be hard to capture or model. In this sense, they propose short-term intention estimation combined with local autonomous assistance. The location-customised assistance is obtained by extracting path primitives from expert driving demonstrations. The intention prediction is made based on the wheelchair location and the user's joystick input.

In common, all these works use *driving* demonstrations from *expert* drivers in an attempt to generate policies for helping someone to navigate better. However, should the assisted person operate very differently from the expert, due to an incapacitating disability for example, it can be infeasible to match the expert and non-expert inputs, thus reducing the usefulness of the recorded demonstrations for intention estimation or local path planning.

2.3.2 Learning Assistance by Demonstration

As discussed in (Soh and Demiris 2013), the main difficulty in assisting with wheelchair navigation does not lie on learning how to drive, but on understanding when and how to properly assist. This is a fundamental concept and through its light it can be seen that, perhaps, the appropriate treatment is not precisely LbD, but a subclass of it, Learning Assistance by Demonstration (LAD). The term was coined in (Soh and Demiris 2013), where a clear definition of the approach is given: "LAD augments LbD by focusing on the assistive element. Current LbD systems are task-centred in that they focus on deriving a policy for completing the demonstrated task with or without a human-in-the-loop. [...] In contrast, LAD is a direct approach. Instead of deriving a policy for 'how-to-drive', we extract a policy for 'how-to-help-a-user-drive' ".

The concept was further developed and tested in (Soh and Demiris 2015a). There, able-bodied subjects controlled a SW with a haptic joystick. An assistant was provided with a second haptic joystick, connected to a computer that in turn was remotely connected to the wheelchair, as shown in Figure 2.1. During the training phase, the user is left in charge of most of the decisions, as should be. If, however, the assistant detects that help is needed, they can take control of driving by moving their joystick and applying a corrective force. The machine then observes this demonstration and can iteratively learn how to reproduce it. Since the corrective actions correspond specifically to that user and disability, this system should, therefore, lead to a robot capable of providing personalised assistance.

Subjects in the study were able-bodied, but a disability was simulated, making it harder for them to do sharp right-turns. Testing in a population

Figure 2.1: Example of LAD being used in a smart wheelchair application. The driver attempts to navigate a difficult scene while being observed by the assistant, which provides a helping control signal as needed. The interaction is recorded and then used to train a model. If learning is successful, the model should be able to autonomously replicate the demonstrated assistive behaviour. Excerpt from (Kucukyilmaz and Demiris 2015).

with characteristics likely closer to those of final users, makes it possible to explore in greater detail the potential benefits that the proposed architecture can offer. Regarding the learning model, it was based on Gaussian Process (GP) and composed of two separate sub-modules: a classifier for determining 'when to help' and a regressor for learning 'how to help'. Furthermore, it was proposed an online learning scheme, based on incoming sequential data. Then, to cope with the imposed time restrictions, it was developed a sparse and iterative GP variant, to reduce the computational requirements. This was called an Online Infinite Echo-State Gaussian Process (OIESGP) (Soh and Demiris 2015b). Finally, an online mixture of experts, with OIESGP as the expert units, was used to increase learning performance.

While the works in (Soh and Demiris 2013) and (Soh and Demiris 2015a) provide a great foundation and initial evaluation of LAD models applied to SWs, they also have limitations. In these works, both training and testing of the system were done on the same environment, following the same track. Therefore, it is not possible to determine if the learned policy was indeed relevant to the users' (simulated) disability or if it was a case of overfitting. It is possible, for example, that assistance was just being replicated whenever the user drove to the same spot where a sharp right-turn was necessary. This limitation was also noted by the authors.

In (Kucukyilmaz and Demiris 2015) and (Kucukyilmaz and Demiris 2018), a similar architecture was explored, albeit with some key methodology differences. Most notably, there was no *when-to-help* module. Instead, assistance was continuously provided and it was left for the machine to learn when the magnitude of this assistance should be zero, or close to it. Hence, only a GP regressor was used. Furthermore, it was made an option for offline learning, instead of online. With the time limitations relaxed, it was possible to afford the regular full GP variant. A drawback, compared to (Soh and Demiris 2015a), is that there was no simulated disability. This may prevent able-bodied subjects from perceiving the potential benefits of the proposed system. On the other hand, there was more exploration of the generalisation issue. Here, training was done on a track and testing was done both on the same track and on a different track on the same environment. The authors concluded that the learned assistive model presented good generalisation capabilities.

A complicating factor for all of (Soh and Demiris 2013; Soh and Demiris 2015a; Kucukyilmaz and Demiris 2015; Kucukyilmaz and Demiris 2018), is that the assistant was able to watch the driver's performance, either directly or through a camera. While this provides the assistant with important contextual information about the person's navigation, and whether or not assistance is required, it also inserts an information discrepancy. The assistant had access to (visual) data that was only available to them, not to the robot. If this data composes an important factor in deciding when and how to help, this information unbalance can severely cap the maximum attainable performance of the learning algorithm.

2.4 Simulators for assistive robotics

A major complication in assistive robotics is related to the difficulty in performing experiments and collecting data, which are needed to evaluate different algorithms and methods. This happens because the experiments involve people, potentially vulnerable, and thus require extra safety precautions. As a consequence, data collection becomes more expensive and time-consuming, which in turn slows down innovation, given that new ideas cannot be tested as quickly. To circumvent this, many works have turned to simulations.

For example, in (Morère et al. 2015) a SW simulator was used to test the effectiveness of combining haptic feedback and obstacle avoidance. In (Devigne et al. 2017), a simulator was used to test a simplified obstacle avoidance algorithm, based on the readings of low-cost sonar sensors. Interestingly, instead of monitors or virtual reality headsets, they used an immersive room-scale virtual environment and had the users sitting on a real wheelchair. This had the effect of improving the sense of presence and the familiarity with the device. A similar setup was used in (Di Gironimo et al. 2013), where they tested the usability of different joysticks for controlling a robotic arm mounted on a powered wheelchair. In the most simple case, the simulator can even be used only for rehabilitation purposes, allowing users to safely train manoeuvres in a virtual platform before moving to the physical device (John et al. 2018).

In common, it can be said that all these simulators were developed for use *by* humans. As discussed, this has the benefit of allowing quicker and safer testing of ideas. However, sometimes it is also constructive to simulate *the* human. For example, in (Reddy, Dragan, and Levine 2018) it was explored a model-free reinforcement learning algorithm for shared control. Using a simulated navigation environment and a simulated impaired driver allowed the authors to extensively test their method and make the necessary improvements, before moving on to user studies. This calculated and frugal approach can lead to major gains in performance because hundreds of simulations can be run in a day, while user studies, even in simulation, rarely employ more than a few dozen subjects. Moreover, simulated humans are not susceptible to 'human-factor', such as mood, tiredness, lack of concentration, etc., which unfortunately can significantly increase experimental noise.

Perhaps with these benefits in mind, a general-purpose simulator for assistive robotics was recently developed (Erickson et al. 2020). It allows the simulation of six different manipulation tasks for helping with activities of daily living: itch scratching, bed bathing, drinking water, feeding, dressing and arm manipulation. Additionally, four different types of robots can be employed. The simulator is mostly tailored to reinforcement learning applications and the human is usually simulated in a passive pose, i.e. no active movement but still constrained by realistic joint movements. Alternatively, co-optimisation, where both robot and human share the same reward function, can be used for simulating active movement by the person [4]. In (Erickson, Gu, and Kemp 2020), the simulator is taken one step further, allowing people to control the virtual human by using virtual reality headsets and controllers. Similar to what was done in (Reddy, Dragan, and Levine 2018), this enables researchers to smoothly transition through simulated humans, to real people using a virtual environment, to finally safely testing their assistive robots in a physical platform.

[4]The authors note, however, that "co-optimization does not guarantee that the simulated human controller learns realistic human motions"

2.5 Discussion

Some interesting observations can be drawn from the literature discussed in this chapter. First, regarding the different possible assistive paradigms for SWs. It was shown that multiple variations of these paradigms have been previously explored, varying from interfering very little with the user control signal to completely taking over and autonomously navigating to a given point. The motivation for this broad spectrum of research originates from the high variability in users' conditions. Some drivers preserve good motor command of their hands and will usually prefer to retain more control over their mobility device (Kairy et al. 2014; Viswanathan et al. 2017). For others, this is not possible, and thus more system autonomy is needed.

This high variability in users' conditions leads to the emerging topic of customisation for SWs. As discussed, many research groups have independently noted the importance of developing systems that can be customised to meet the needs and preferences of individual users, instead of grouping drivers under the same umbrella of 'needs assistance'. As a consequence, customizable systems come as a strong design recommendation to give SWs better acceptance outside of laboratories (Kairy et al. 2014; Padir 2015; Erdogan and Argall 2017; Leaman and La 2017; Viswanathan et al. 2017). Another feature commonly requested by potential SW users is for the wheelchair to better inform the driver about the assistance that is being provided (Padir 2015). The preferred type and amount of feedback once again vary between users, but some works indicate that richer feedback modes, such as the ones provided by haptic joysticks, tend to be preferred (Kuchenbecker 2018).

The literature has also shown that measuring the assistive performance of an SW is a multifaceted problem, and as such different metrics should appropriately be used (Leishman et al. 2014). However, while these metrics are important to give developers insights into what is working or not, it has been observed that they do not directly translate to user satisfaction (Javdani et al. 2018; Kim et al. 2012; Gopinath, Jain, and Argall 2017), which has the biggest impact on the acceptability of a system. Hence, one of the most important tools for evaluating an assistive system should be to directly collect feedback from its users (Erdogan and Argall 2017).

It was discussed how estimation of intention is a central topic in assistive robotics. This problem is manifested in many ways and, equivalently, many approaches exist to handle it (Losey et al. 2018). The case of a goal space which is continuous and not known beforehand is particularly challenging, especially in terms of the uncertainty that it can pose over the estimation. In these cases, it is usually preferred to avoid the predict-then-act approach of assistance and bypass the explicit estimation of intention step. Instead, help can be provided reactively, being adaptable to the robot's state and the user input (Reddy, Dragan, and Levine 2018; Du et al. 2020).

We also exposed the difficulty in doing experiments for assistive robotics and showed that many researchers have turned to simulations in order to test ideas more quickly and safer. Both toy (Reddy, Dragan, and Levine 2018) and general-purpose (Erickson et al. 2020) simulators for assistive robotics have already been developed, which allow research teams to collect more statistically powerful data. Particularly considering SWs, the literature shows that some simulators have also been developed (Morère et al. 2015; Devigne et al. 2017; Di Gironimo et al. 2013). But their purpose is to be used *by* humans, not being capable of simulating the drivers' input. More specifically, we are not aware of any open-source simulator that is capable of reproducing the interaction between a driver and its SW, let alone the triadic interaction between driver-SW-assistant. We believe that such a simulator would be an important contribution to accelerate research in the field of robotic wheelchairs. Other desirable characteristics of this simulator would be: being capable of simulating different types and levels of hand-control disabilities; allowing the testing of different assistive paradigms; simulating different environments, of varying difficulty of navigation; and allowing humans to take up the role of the simulated humans, as was done in (Erickson, Gu, and Kemp 2020).

Finally, we come to the topic of Learning by Demonstration. LbD offers a way to extract the tacit knowledge of healthcare workers and use it to autonomously provide assistance. The variety of applications in which this approach was explored shows the popularity and success of the method: physical rehabilitation (Najafi, Adams, and Tavakoli 2017), motion training (Ewerton et al. 2018), assisted teleoperation (Losey et al. 2020), workspace sharing (Koert et al. 2019). The use of LbD to generate assistive policies for SW navigation has also been investigated (Chow and Xu 2006; Goil, Derry, and Argall 2013; Poon et al. 2017). But all these works had limitations, as they: were not capable of generalising to different environments; could only extract a blending factor for shared control; or had to employ a 'predict-then-act' technique.

Conversely, Learning Assistance by Demonstration proposes a more direct approach to the same problem. Instead of learning how to drive, then deciding when to help, then inferring the intention of the user, and only then providing assistance; use the demonstrations of assistance to directly learn how and when to help the user (Soh and Demiris 2013; Soh and Demiris 2015a; Kucukyilmaz and Demiris 2015; Kucukyilmaz and Demiris 2018). However, although the

LAD concept has been tested before, some outstanding issues still need to be investigated. Foremost, the matter of generalisation to different environments. Some of the works that employed LAD for SWs used the same obstacle course and the same trajectory where training data was collected to test the assistive performance of the learned model. Other works used a different trajectory, but still on the same obstacle course. These, however, are not accurate ways to test performance. In a realistic setting, after training is complete the driver will use the SW and assistive model in scenarios completely different from the training one. Possibly this change would not have a significant impact on performance, but it has to be evaluated before LAD can be considered a viable alternative for assisted wheelchair navigation. Other topics also demand deeper investigation:

- is LAD appropriate for multiple disability types (previous work only considered a single type)?
- is LAD capable of providing *personalised* assistive solutions?
- is the generated assistance consistently beneficial across multiple metrics?

These questions show the existence of problems that were not investigated by previous research. Throughout this thesis, we will describe work conducted on the development of new systems, algorithms and approaches, which allowed us to better address these problems.

Chapter 3

Enabling Remote Assistance

For any robot, independent of hardware or software, the possibility of failure under particular situations will always exist. In the field of assistive robotics, however, an undesired state may be entered not because the robot failed, but because the human is incapable of properly operating it. If this robot is of vital importance for its human operator, such as a mobility device, there should exist a way of recovering from this undesired state. One possible way of achieving this recovery, is to ask another person for help.

Specifically in the case of smart wheelchairs, if the driver is impaired by some sort of hand-control disability, they could ask an assistant to help them navigate through a particularly difficult path or to guide them back to safety. Having the assistant constantly present alongside the driver might not always be an option, however. An alternative, possibly more promising approach, is to have a *remote* assistant, which is only summoned when assistance is needed. For this, the assistant must have access to a teleoperation platform. This platform should serve two purposes: inform the assistant about the driver's current state, actions and possible goals (e.g., going through a doorway, or clearing a collision); and allow the assistant to remotely control the wheelchair. With this system in place, the driver can navigate more confidently and safely, despite their difficulty in controlling the wheelchair. Furthermore, as will be discussed in Chapter 4, if these triadic interactions between robot, driver and assistant are recorded, they can later be used to train a model to autonomously reproduce the demonstrated assistive behaviour. As will be explained in that chapter, in this case special considerations must be taken when designing the teleoperation platform, in order to facilitate model learning.

In this chapter, we describe a custom teleoperation platform that was developed considering all of the above requirements. We start by giving a high-level overview of the platform in Section 3.1. Then, the individual components (hardware, custom algorithms, interfaces, etc.) that compose the platform are detailed in Section 3.2 and the software integration is discussed in Section 3.3. Following, Section 3.4 describes a user study conducted to assess the impact that different interfaces have on the assistant's capability of inferring the driver's immediate goal. Finally, conclusions are presented in Section 3.5.

Parts of this chapter form an extended version of the work presented in (Schettino and Demiris 2019).

3.1 System overview

An overview of our teleoperation platform is shown in Figure 3.1. A powered wheelchair is modified to include a computer and sensors for gathering information from both the local environment and from the driver. This robotic wheelchair is used as a mobility device by a person with driving difficulties, due to motor impairments on the hand. At the other end, an assistant observes the situation through the data gathered by the robot's sensors and provides alternative driving signals as needed.

Because the raw sensor data is of high dimensionality and not naturally understandable[1], different interfaces are used to aid the assistant in interpreting it. The first interface used is based on the movement and localisation of the wheelchair relative to nearby obstacles and goals. The laser-scanners update at a high rate compared to the wheelchair movement, thus making it possible to build online a map of the environment being navigated, or to localise the wheelchair in a previously built map. This map, with a virtual model of the wheelchair accordingly placed on it, is displayed to the assistant. Using their natural scene and context understanding abilities (e.g., looking at a map and recognising that it represents a corridor and then discerning what are the reasonably viable actions that can be taken in such an environment), the assistant can make inferences about the intention of the driver and act to help them. An example of how this interface is displayed is shown in Figure 3.3.

The second interface uses a web-camera facing the driver to estimate their eye-gaze direction. This is presented to the assistant by moving the head of a virtual human model, which is constructed on top of the virtual wheelchair to represent the driver. The goal is that, by knowing where the driver is looking, the assistant is better informed to infer the driver's intention. The third interface uses haptic-pairing to give the assistant an instant feel of how the driver is attempting to control the wheelchair. Two identical joysticks are employed and force-feedback is used to make the assistant's joystick follow the driver's one. A safety mechanism uses the laser data to ensure that the wheelchair will not accidentally collide against nearby obstacles.

3.2 Individual components

This section describes in more detail the individual components that form the teleoperation platform outlined in Section 3.1.

3.2.1 Robotic wheelchairs

Two different models of robotic wheelchairs can be used interchangeably with the teleoperation platform: Assistive Robot for Transport of Adults (ARTA) and Assistive Robot for Transport of Youngsters (ARTY). They are both pictured in Figure 3.2 and their architectures are described in more details in (Carlson and Demiris 2008) and (Soh and Demiris 2012), respectively. Despite their size difference, the wheelchairs are similar from an electronic and operational point of view. ARTA is equipped with two short-range (4 meters) laser-scanners on its front-left and front-right extremities and a long-range (30 meters) one on its back. Its controller interface is mainly analogue and an Arduino is used to enable a computer to send velocity commands to the wheelchair. ARTY is equipped with three short-range laser scanners on its front, back-right and

[1]In our platform we only consider laser-scanners as the sensors used to capture information from the environment. An alternative approach would be to use forward facing cameras, which output information that is naturally understandable by a remote human assistant. However, this would greatly increase the dimensionality of the data - while a laser scanner may typically use 360 channels to cover a full 360 degrees view around the wheelchair, a low-resolution camera (640x480) would need more than 800 times that amount of information to cover a much smaller field of view. This dimensionality increase can lead to two problems: first, in an application where the data has to be transmitted to a remote location, it can lead to latency issues, which in turn deteriorates the quality of the assistance being provided. Second, the ultimate goal of our application is to use this data to learn an assistive policy that can automate the actions of the human assistant; and when the dimensionality of the input data increases, the machine learning task is rendered more difficult, due to the increased chance of overfitting.

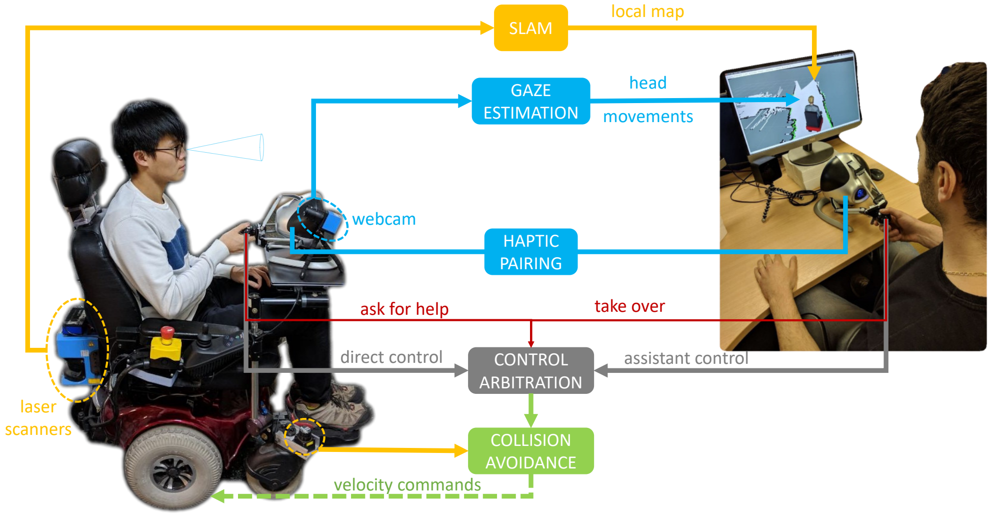


Figure 3.1: System overview. A remote assistant provides help to a wheelchair driver using our custom teleoperation platform. In order to intuitively make use of the data registered by the wheelchair's sensors, the assistant uses multimodal interfaces: SLAM or localisation to handle the laser-scan data, haptic-pairing to immediately sense the driver's actions, and eye-gaze estimation to better infer the intention of the driver.

back-left extremities. Its main controller exposes a CAN bus interface through which velocity commands can be sent to the wheelchair. For this, a CAN-USB adapter is used to connect the base computer to the wheelchair.

Both ARTA and ARTY are also equipped with standard joysticks - which have their 2D positions read by the computer - a power distribution system - fed by the wheelchair's batteries and used to power the laser-scanners - and a safety-stop button. Additionally, they both have an Inertial Measurement Unit (IMU) fixed to their chassis, which can be used for providing or improving odometry estimation. ARTY can further improve its odometry estimation by using built-in wheel-encoders. The base computer uses Robot Operating System (ROS) (Quigley et al. 2009) to connect the software nodes responsible for implementing different functionalities, such as processing sensor data, communicating with an external computer or sending velocity commands to the wheelchair. Aside from the low-level interface with the wheelchairs' electronics, the software used to control ARTA and ARTY are identical.

Although both smart wheelchairs already existed for some years, during the course of this thesis work had to be carried to improve and adapt them; especially ARTY, which was initially nonfunctional. Part of the low-level electronics responsible for power distribution had to be redesigned and rewired. The on-board computer was also replaced for a modern laptop, using more

(a) Adults wheelchair, ARTA (b) Kids wheelchair, ARTA

Figure 3.2: Different smart wheelchairs that can be used with our teleoperation platform. Despite being different from physical and electronic points of view, a wrapping software layer is used to guarantee that, from a high-level perspective, the wheelchairs can be used interchangeably.

recent versions of ROS and Ubuntu. In turn, this required firmware repairs to reestablish CAN bus communication. On the software side, significant effort was put into simplifying and merging the code-base for both wheelchairs, besides updating it to use open-source third-party ROS packages whenever possible, thus reducing maintenance requirements. We also created code for automatically generating a high-level description - known as URDF[2] files - of the robots' models, including joints and sensors positions. As input, this code only takes simple files with structured text, listing dimensions of the wheelchairs which are easily measurable. This development was later fundamental for the creation of our simulator - see Section 5.1.

3.2.2 SLAM and localisation

Different interfaces are used to allow a remote assistant to make sense of the sensor data registered by the robotic wheelchair. The first interface explored is based on the movement and localisation of the wheelchair relative to adjacent obstacles and potential goals. This can happen under two different scenarios, SLAM or localisation.

For SLAM (Durrant-Whyte and Bailey 2006), a map of the environment is built online, by leveraging the high update-rate of the laser-scanners to detect relative movement of spatial features like gaps and corners. To implement

[2] http://wiki.ros.org/urdf

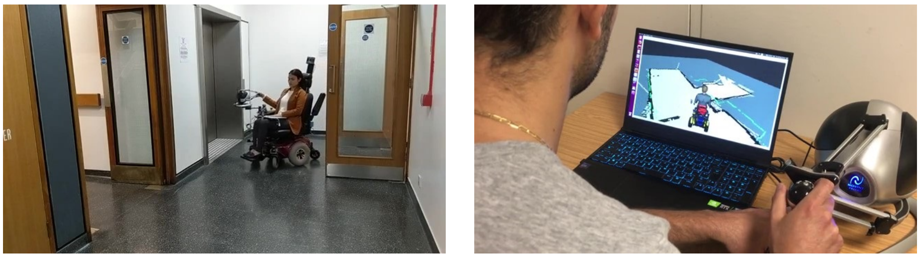


Figure 3.3: Example of how SLAM or map and localisation can be used by the assistant to help the driver. Because humans are naturally capable of scene and context understanding, this type of interface is useful in informing the assistant which are the possible near-term goals of the driver (e.g. going through a door), as well as discarding infeasible options (e.g. going through a wall).

this in an efficient and robust way, the hector_mapping[3] (Kohlbrecher et al. 2011) ROS package is used. For localisation, where the robot is moving in a location mapped beforehand, the Adaptive Monte Carlo Localisation (Fox 2001) approach, as implemented by the amcl[4] ROS package, is used. In this case, hector_mapping is kept only for providing an odometry estimation based on the laser readings.

In either case, SLAM or localisation, the result is a 2D map and a transform for the relative wheelchair position. This information is presented to the remote assistant by displaying to them a map of the environment, with virtual human and wheelchair models correctly positioned on it, as shown in Figure 3.3.

3.2.3 Haptic-pairing

An important component of a powered wheelchair is the joystick. On a regular wheelchair and most robotic ones, it is simply used as a mean for the driver to control movement. However, as discussed in Section 2.1.2, when a force-feedback device is available, it can also be used to present a haptic-feeling to the driver. This can be used to augment the driver's perception of potential obstacles, thus increasing safety, or to intuitively explain the assistance being offered, thus reducing frustration with unexpected system behaviour. Alternatively, on an assistive-teleoperation scheme, it might be used to give the remote assistant a haptic-feeling of the driver's input, vice-versa, or a combination of both.

In the proposed architecture, two Novint Falcon joysticks featuring programmable force-feedback are used. They were selected due to being able to exert relatively high forces and working under a high cycle-rate, which is important for conveying a real-time haptic-feeling. To better integrate this device

[3] http://wiki.ros.org/hector_mapping
[4] http://wiki.ros.org/amcl

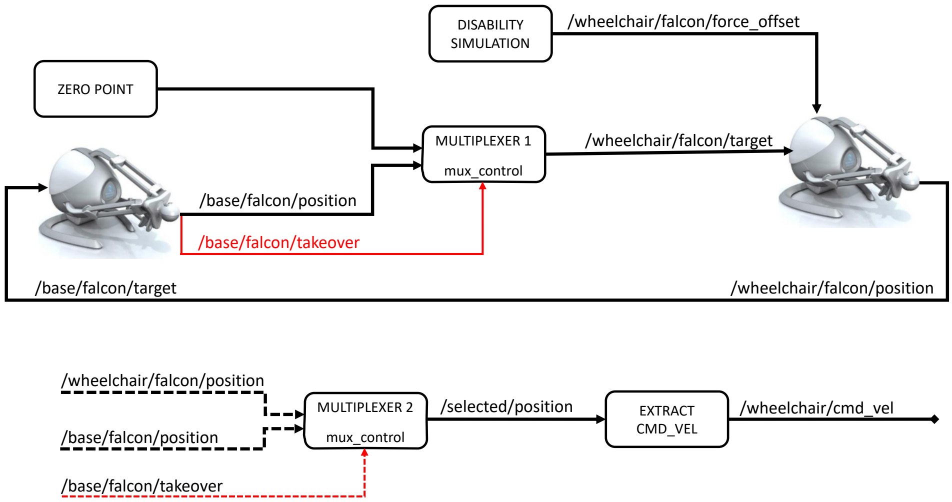


Figure 3.4: Schematic of how two identical force-feedback joysticks are linked by haptic-pairing, while controlling the wheelchair. Our platform enables different modes of operation: the assistant can take over control when they believe help is needed (this is the option shown), or the driver can click their joystick to request support. In any case, the assistant can always feel the driver's movements, and the driver receives feedback when help is provided.

to the ROS ecosystem, we built custom software on top of the available driver, aiming at both easy configuration and low-level control of the device. This software allows one to use a PID controller, with configurable parameters, to have the joystick follow a target 3D position. Furthermore, a variable position and/or force offset can be applied, which we use on the driver's joystick to simulate a hand-control disability - see Section 3.4.

In our system, this interface is used by constantly reading the position of the driver's joystick and using it as a target for the assistant's joystick, which then feels as if holding the same joystick. If the assistant detects that intervention is needed, they can press a button to take over control of the operation. In this case, the direction of haptic pairing is reversed, and the assistant's movements are used as feedback to the driver, thus explaining the assistance being provided. A schematic illustrating how teleoperation occurs under this scheme is shown in Figure 3.4. Alternatively, an 'ask-for-help' configuration can be implemented, where the assistant is constantly providing an assistive signal and the driver decides when to release control. This configuration is used and described in more details in Chapter 4.

3.2.4 Virtual reality

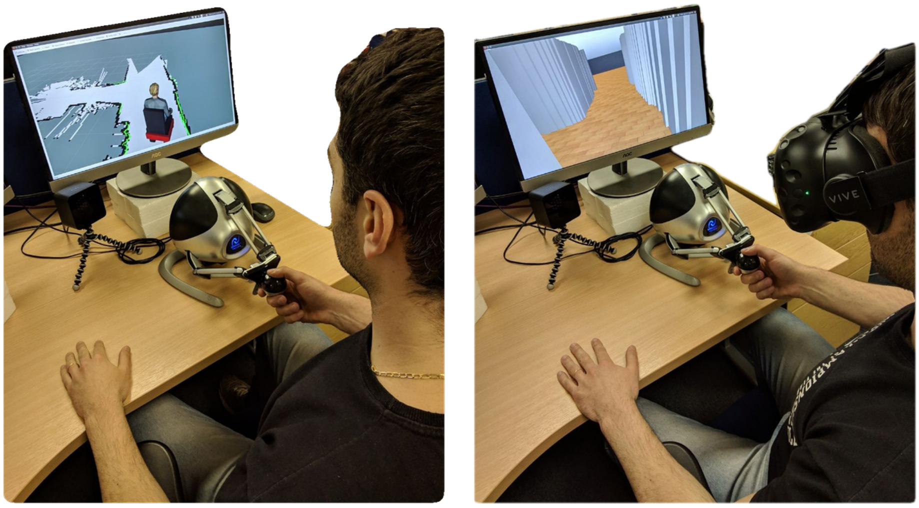
One of the goals of our work is that the demonstrations of assistance provided through our teleoperation platform are recorded and then used to train a model that could autonomously help the driver. Hence, as initially discussed in Chapter 2 and expanded on in Chapter 4, it is important to keep a consistency between the data available for the assistant and the data available for model learning. For this reason, the teleoperation platform developed does not include any form of direct- or camera-view of the wheelchair. Instead, the assistant can only observe the driver and their actions through the readings of the sensors present on the wheelchair. However, the raw data registered by the sensors is generally hard to understand for a human, and this can deteriorate the quality of the generated assistance.

To cope with this, it is possible to make use of Virtual Reality (VR) to, instead, provide the assistant with an interpreted or augmented version of the sensor data. As an example of this, instead of showing the assistant a list with 360 numbers representing the laser-scanner readings around the robot, it is better to display a 2D map representing the observed obstacles and free space. If using a regular computer monitor, this kind of visualisation can be readily achieved using RViz[5], the standard robotics visualisation tool for ROS. An example scene illustrating this approach is shown in Figure 3.5a.

If a virtual reality headset is available, the representation can be augmented by converting the map into an immersive 3D environment, which can be more familiar and easier to interpret for the assistant. Additionally, these devices can confer to the user an improved depth perception and sense of presence (Alshaer, Regenbrecht, and O'Hare 2017). In this setting, the remote assistant takes the point of view of the driver, as if actually sitting on the wheelchair and holding the same joystick. In our work, the HTC Vive[6] VR headset is used for this.

The implementation of this system, however, is challenging because it requires the integration of multiple software nodes, as shown in Figure 3.6. And, as a complicating factor, the Vive and some of these software do not officially support Linux, while ROS(1) cannot run on Windows. To work around this, beta ports of the Windows software and the Vive driver to Linux were employed. The laptop on the smart wheelchair sends laser-scan data to the remote assistant's computer over WiFi, using the ROS infrastructure. The VR headset is connected to this computer and the Unity[7] game engine is used

[5] https://wiki.ros.org/rviz
[6] https://www.vive.com/uk
[7] https://unity3d.com



(a) Virtual scene viewed in a monitor (b) Virtual scene viewed in VR headset

Figure 3.5: Rendering of the driver's environment and operation to the remote assistant. Our platform allows the use of either computer monitors or virtual reality headsets for this purpose, with the rendering correctly adjusted for each case. In both figures the driver is at the same physical location, thus contrasting the two visualisation options.

to create the 3D images. However, Unity cannot directly read ROS messages. Hence, the ROS#[8] plugin is used as an interface, converting laser-scan messages from ROS to a JSON format that can be read in Unity. The laser-scanners, however, only capture readings in a 2D plane. To generate the 3D views that will be rendered at the Vive, our software places a beam on top of each individual laser reading and extends it from the virtual floor to the virtual ceiling. Because of the fine angular resolution of the laser-scanner, when the beams are displayed together they render the effect of a virtual wall. The result is exemplified in Figure 3.5b. To improve depth perception and sensation of movement, the virtual floor is textured with a common wooden floor pattern. Additionally, by looking down, the assistant can see the wheelchair and human models, which have familiar dimensions, and thus can be used for comparatively estimating the size and distance of obstacles.

VR headsets promise many benefits, like an improved sense of reality and social presence, improved depth perception and natural manipulation of point of view (Alshaer, Regenbrecht, and O'Hare 2017; Thalmann, Lee, and Thalmann 2016). However, the technology itself still faces challenges, such as low resolution,

[8] `https://github.com/siemens/ros-sharp`

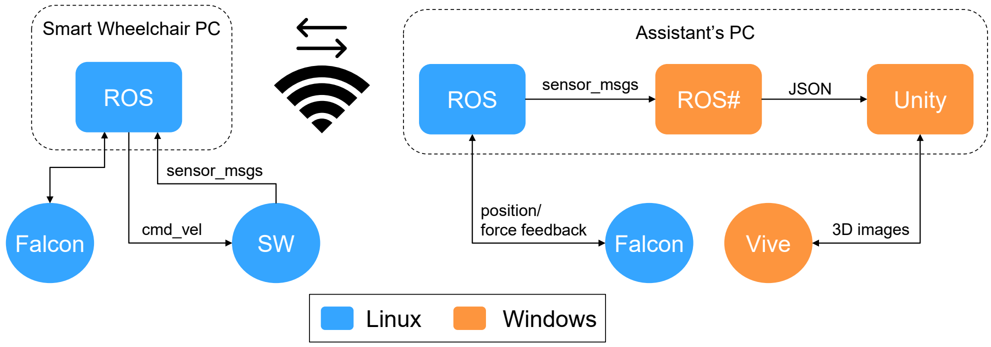


Figure 3.6: Schematic of how a VR headset is integrated to our teleoperation platform. The wheelchair laptop collects sensor data, such as laser-scans, and transmits it to the remote assistant's computer through WiFi using the ROS infrastructure. ROS# is used to interface different data formats, and custom software running on the game engine Unity transforms the raw sensor data into 3D images that can finally be rendered on the HTC Vive.

small field of view, and latency between display updates. Furthermore, usage of these devices can lead to cybersickness (John et al. 2018), where people become uncomfortable or nauseated because they can *see* a moving scene but cannot *feel* it. Hence, we took the decision of developing the virtual reality scene in parallel for both settings, a computer monitor and a VR headset; but the monitor setting was employed in our user studies.

3.2.5 Gaze estimation

For many drivers, the very reason for using an assistive robot would be the difficulty in naturally controlling a powered wheelchair. This difficulty might stem from motor disabilities leading to symptoms like muscle weakness or tremors of the hand. In these cases, the haptic-pairing interface might be inappropriate for communicating the driver's intention, due to containing only incomplete or noisy information. Hence, additional or alternative interfaces might prove beneficial.

One possible such interface is based on the estimation of eye-gaze direction (Zhang et al. 2019). This approach has been used before for predicting user intention in a shared autonomy for manipulation setting (Admoni and Srinivasa 2016). For navigation, the idea is that drivers should normally gaze at their goal location before reaching it, thus potentially communicating to a remote assistant their intention in a natural way. For this, a backwards-facing camera is used to capture a video feed of the driver and the images are processed online by the Real-Time eye Gaze Estimation in Natural Environments (RT-GENE) framework (Fischer, Chang, and Demiris 2018). RT-GENE first extracts image patches of the face and eyes of the driver. Based on these patches, the head pose and eye-gaze angles are estimated using a set of deep neural networks. This framework can run in real-time and with low latency on the wheelchair's base computer, thus reducing the dimensionality of the data transmitted over the network from a full image (640x480 pixels) to a single degree of freedom.

Once the gaze orientation of the driver is detected, this information has to be rendered to the remote assistant in an intuitive way. This can be done by integrating a human model into the virtual environment and turning its head accordingly. For this, we first found and downloaded a simple human model with movable joints [9]. It was important to use a model with low polygon-count in order to not overload the system with visualisation displays and collision checks during simulation (see Section 5.1). Then, Blender[10] and MeshLab[11], two 3D modelling software, were used to adjust the model's pose to a seated person and to fix all joints except for the head one. With the model and its texture ready for use with ROS, a script was created to read the yaw component of the RT-GENE gaze prediction and convert it to a JointState message[12]. Finally, the robot_state_publisher[13] node is used to parse the model's URDF and accordingly turn the head joint in the visualisation. An example scene illustrating this approach is shown in Figure 3.7, and Figure 3.8 exemplifies how the gaze information can be used to disambiguate when inferring the intention of the driver.

3.2.6 Obstacle and collision avoidance

The joystick position of the agent in control at any given time is used to drive the wheelchair. But this information is first passed through a safety layer, which is responsible for ensuring that the wheelchair will not collide with static or moving obstacles. A safety strategy can be chosen between collision or obstacle avoidance. In the former case, only the magnitude of the wheelchair's speed will be capped when an imminent collision is detected, bringing it to a halt if needed. In the latter, the angular velocities are adjusted instead, to move the wheelchair around the obstacle, without having to stop it.

These assistive paradigms are usually implemented as derivations from navigation algorithms traditionally used in the field of mobile robotics. For obstacle avoidance, for example, a common approach is to first use the laser

[9] "Lowpoly man", create by TiZeta and available at http://www.blendswap.com/blends/view/66412 under a CC BY 3.0 license. The model was modified for this work.

[10] https://www.blender.org

[11] https://www.meshlab.net

[12] http://docs.ros.org/en/api/sensor_msgs/html/msg/JointState.html

[13] http://wiki.ros.org/robot_state_publisher

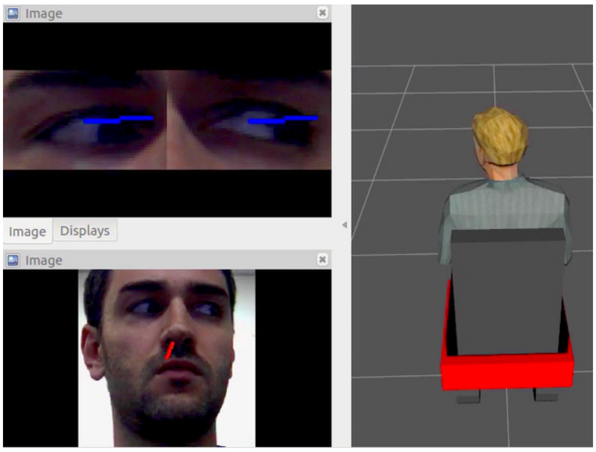


Figure 3.7: Illustration of how the estimated gaze direction is mapped to head movements of a virtual human model. RT-GENE uses a webcam facing the driver to estimate their head pose (red line) and gaze direction (blue lines). Our software combines both information to extract a final yaw angle, which is used to turn the head of a virtual mannequin.

Figure 3.8: Example of how the gaze estimation can be used to help the assistant disambiguate when inferring the intention of the driver. When navigating a corridor, it is not always clear if the driver intends to continue on it or enter a room. The haptic-paired joysticks are used to support the assistant to make a decision, but when the control signal noisy or distorted due to a disability, this information might not be enough. In these cases, gaze estimation can improve the quality of the assistance being provided. See also Figure 4.4, for a related result.

readings to build a local costmap around the robot. This costmap is then used by a collision checking algorithm, such as the Dynamic Window Approach (Fox, Burgard, and Thrun 1997), which projects into future time steps the relative position of the wheelchair. If an imminent collision is detected, the angular velocity of the robot is gradually adjusted and a new course is simulated. The process is repeated until a collision-free path is encountered.

In ROS, this can be implemented by using a combination of software from the navigation[14] (Marder-Eppstein et al. 2010) and nav2d[15] stacks. By changing a single parameter, nav2d allows switching between collision and obstacle avoidance. If a global map is available, autonomous navigation can also be performed.

3.3 Software architecture

In the previous sections, an overview of our teleoperation system was given, and the purpose and operation of individual components was detailed. Now, we discuss how these components interact, from a software perspective. The goal here is documentation for reproducibility. That is, detailing the different components of our software base and their connections, to allow other research teams to build a teleoperation platform functionally equivalent to ours.

A diagram depicting the software architecture is shown in Figure 3.9. First, one can notice how the deployment of software nodes must be distributed among two computers: the laptop on the wheelchair and the remote PC, which is used by the assistant. Communication between these computers happen over WiFi[16] and is transparently managed by ROS. ROS also manages the connection of most software nodes within each computer, using a 'topics' based form of communication in a producer/consumer framework. For this, a 'roscore' is needed, which is a dedicated node responsible for allowing other ROS nodes to register and find topics of interest. We set the roscore to be on the wheelchair laptop, guaranteeing that the driver can safely control the vehicle even if communication to the remote computer is lost.

A topics based form of communication also supports increased modularity, since only the *type* of the data being transmitted becomes relevant, not its origin. Hence, if a software node has to be replaced, it can be done without breaking the rest of the code base, as long as the replacement node keeps

[14] http://wiki.ros.org/navigation
[15] http://wiki.ros.org/nav2d
[16] Although, if a reliable and fast enough connection is available, this communication need not be limited to local area networks.

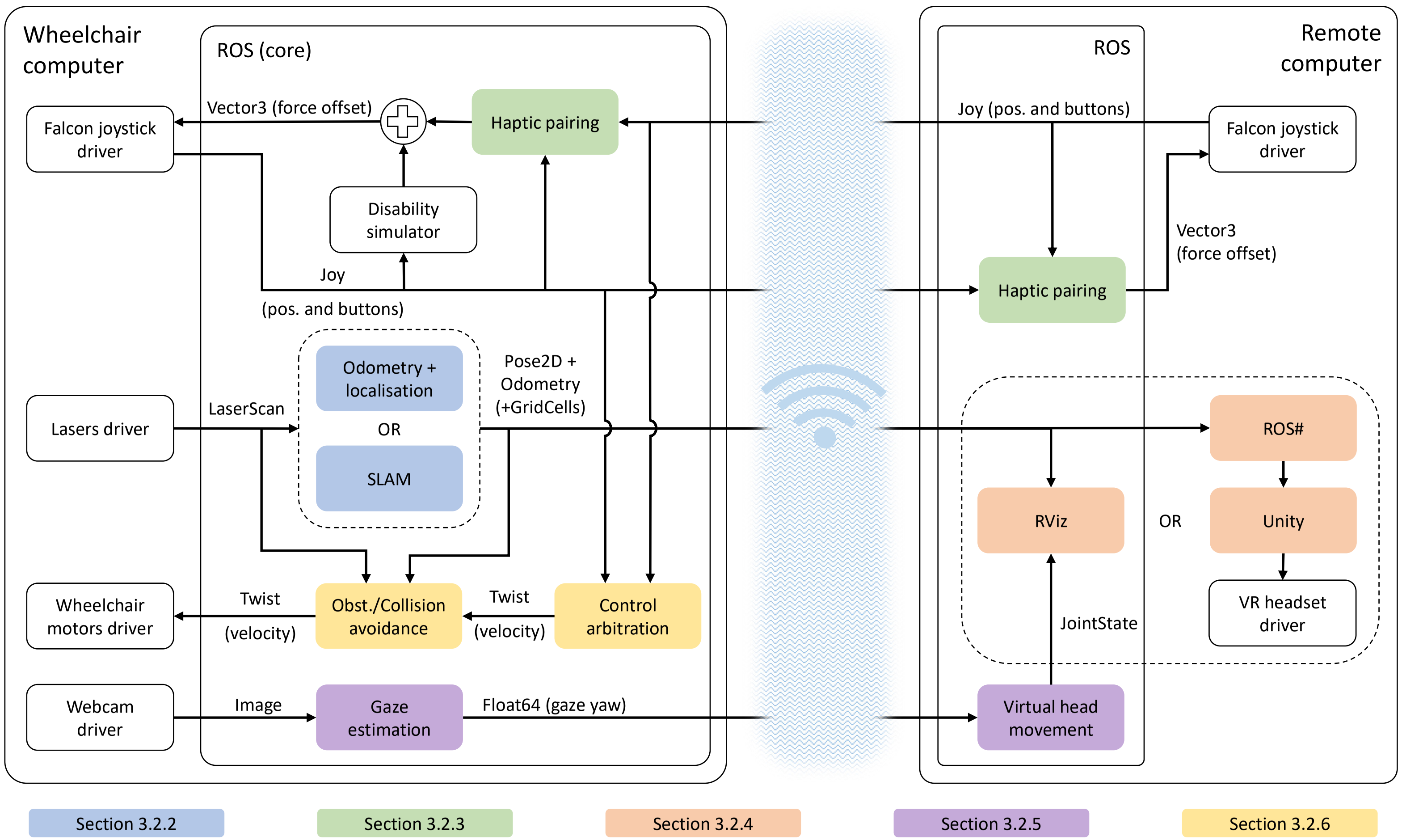


Figure 3.9: Software architecture for the teleoperation platform: shows how individual nodes are connected together and which message types are used. The colour code indicates where to find more detailed information about each piece of software.

reading and writing the same message types. Our system takes advantage of this by implementing operating modes that can be used interchangeably - as highlighted by the 'OR' blocks in Figure 3.9. Hence, if driving is taking place in uncharted environment, SLAM can be employed and the map information is sent to the remote computer for construction of the virtual scene. Or, if a map of the driving area is available beforehand, just the localisation information needs to be transmitted, saving bandwidth. The choice between alternatives is easily configured through software switches.

Regarding bandwidth, the communication bottleneck for this teleoperation platform is the WiFi bridge, especially because the haptic-pairing nodes require a high rate of data flow in both directions to operate smoothly. This bottleneck highlights the importance of using the gaze estimation software, which reduces the webcam data from a full picture to a single number, saving bandwidth for the transmission of haptic and odometry data.

One node in the diagram of Figure 3.9 that has not yet been discussed is the 'Disability simulator'. That is because this is an optional and generic node, which is implemented differently across our studies (Sections 3.4 and 4.2). But the end effect is always the same: an adjustable force offset being exerted on the driver's joystick.

Although it may seem difficult to deploy this web of nodes, ROS greatly simplifies the task by making use of 'launch' files. These are simple xml files listing which nodes to start and how to configure them. For more fine-grained control, we use three of such files: *wheelchair.launch* for turning on the wheelchair and the associated hardware, starting roscore and initiating the SLAM, gaze estimation and haptic-pairing algorithms on driver's side; *base.launch* for starting the assistant's Falcon joystick and haptic-pairing algorithm and launching the virtual reality setup; and *control.launch*, which starts the disability simulator, control arbitration and collision avoidance nodes, enabling the wheelchair to start moving.

3.4 Impact on inference of intention

The different interfaces used in the teleoperation system are made available in order to facilitate the assistant's job. To provide useful and timely help, the assistant must be able to accurately infer the driver's intention. However, without a direct view of the wheelchair, this task is rendered more difficult. Hence, the teleoperation platform was developed following the intuition that providing more information to the assistant would facilitate the remote inference of intention. Nonetheless, it is important to test to what degree this intuition holds, and under which circumstances.

This section discusses a user study conducted to explore the impact that the three different modalities of interfaces, map and localisation, haptic-pairing and eye-gaze estimation, have on remote inference of intention. Using the platform previously described, subjects were asked to infer the intention of a driver suffering from intense tremors on the hand.

3.4.1 Experimental setup

An experiment was devised in which drivers were asked to sequentially reach different goal locations in a room. Subjects in a different room used a combination of the available interfaces and tried to infer the intention of the driver. They had the competing goals of being as accurate and as fast as possible. In order to reduce the complexity of the experiment, it was divided into two stages. This allowed us to increase the number of subjects tested and also to mitigate random effects imposed by a driver-assistant pair independent variable.

The first stage consisted of recording driving data in an obstacle course, as illustrated in Figure 3.10. For this, we first had to build a map of the room using SLAM. Then, two drivers were asked to reach nine different goal positions on this map, while all the relevant data was recorded - laser-scanner readings, localisation on the map, joystick position and a video of their faces. The goal positions were marked by objects on the floor and drivers were instructed to always approach them at a right angle.

We are interested in the cases of people who might have difficulty in controlling a powered wheelchair, but asking our target population to use an untested system at this point would be exposing them to an unnecessary burden. Instead, we decided to work with able-bodied drivers and leverage the force-feedback joystick to impose on them a simulated disability. The chosen disability was a tremor of the hand, similar to what people suffering from Parkinson Disease may feel. This particular form of tremor (typically with a 4-8 Hz frequency) is more intense when the arms are at a resting position and less intense during the execution of a movement. This was simulated by imposing on the drivers' joystick a circular force-offset of varying intensity.

The actual testing happened on the second stage. Subjects were presented with a map displayed on a screen and the associated goal positions, as shown in Figure 3.11. When the recorded data was played back, the wheelchair model would start moving on the map and subjects were asked to estimate as accurately and as fast as possible which was the goal position of the driver. For this task, they would be aided by of one of the following combinations of interfaces: 1) only map and localisation, 2) map and localisation + haptic-pairing, 3) map

Figure 3.10: Shows a user driving in obstacle course while data is recorded. The driver was impaired by a simulated disability, which used the force-feedback joystick to impose a tremor on their hand. Two users participated in the data collection, being tasked with repeatedly reaching pre-defined goal locations.

and localisation + gaze-estimation and 4) map and localisation + haptic-pairing + gaze-estimation. For simplicity, from here on these combinations will be respectively called 'map only', 'haptic', 'gaze' and 'haptic and gaze'.

Every subject was tested with all four combinations, performing five trials with each. To compensate for learning effects, the order of which the combinations were presented was randomised and also balanced. The target for each trial was also randomly selected. However, an initial analysis showed that some of the goals were harder to infer than others (for example, goals 5 and 6 are more difficult than 1). If, by chance, for the same subject, one of the combinations was selected to have a higher or lower number of these difficult goals among the five trials, the comparison of performance against the other combinations would be unfair. Conversely, using the same targets for all combinations is not feasible, as subjects would quickly memorise it, leading to improved performance on the latter combinations tested. To avoid these problems, among the five trials for each combination, three goals were always repeated and only these three are used in the analysis of results. For example, during the four combinations, one subject could experience the following sequence of goals: (1, **2**, 5, **4**, **9**); (**4**, 7, **2**, 3, **9**); (6, 5, **4**, **2**, **9**); (**9**, 8, **2**, **4**, 1). In such case, analysis of results for this subject would be done considering only goals (**2**, **4**, **9**). Subjects were not informed of this and investigation with a few people prior to starting the experiment showed that they were not able to notice it.

Regarding the experimental protocol, subjects were first given an information sheet about the experiment and had their questions answered. Then, they were asked to sign a consent form about the data collected and fill in a survey

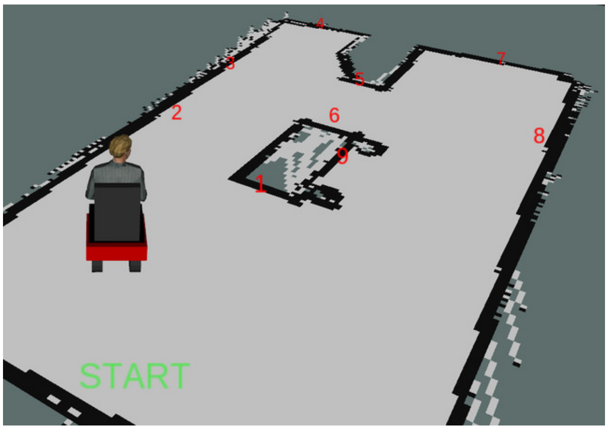


Figure 3.11: Virtual scene that was displayed to the subjects of our user study. Twelve subjects participated and were exposed to a combination of the interfaces available in our teleoperation platform, while attempting to infer the intention of the driver as fast and as accurately as possible.

with demographic questions and queries about whether or not they had prior experience with any of the interfaces used. Following, instructions about the experiment were given and a demo run would be shown. Each trial lasted about 30 seconds and would be stopped when the participant pressed the 'take over control' button on their joystick. At this point, the inferred goal and the time taken for inference would be recorded. At the end of all trials, subjects were asked to answer a questionnaire about their preferences and perceptions of the interfaces and finally asked for general feedback. Overall, the experiment took about 25 minutes per subject.

3.4.2 Metrics

The goal of this study is to understand the impact that the proposed interfaces have on assistants' performance, in terms of inferring the intention of a remote driver. In the field of human-computer interfaces, Fitts's law (Fitts 1954) has been employed to compare the performance of different input devices (mouse, joystick or directional movement keys) at the task of pointing to target objects (Card, English, and Burr 1978). Under Fitts's law, an *index of difficulty* (ID) metric is proposed to quantify the difficulty of a target selection task. Then, the *movement time* (MT) for the given task is also measured. Finally, the *index of performance* (IP) is calculated as $IP = ID/MT$. Assuming a linear relationship between task difficulty and movement time, the latter can be estimated as $MT = a + b * ID$. To characterise the performance of an interface over a range of task difficulties, the parameter a is often interpreted as a delay and ignored when considering average performance. Then, the inverse of the fitting slope, $1/b$, is interpreted as the IP, and can be used to compare the performance of different interfaces.

Despite its extensive usage in the design of human-computer interfaces and graphical user interfaces, Fitts's law may not be appropriate for our study for two reasons. First, Fitts's law was conceived to model the act of *pointing* to a target. Although since its conception the law has been employed to model a variety of different tasks, it is not clear whether it could also be useful for the task of remote inference of intention. Second, to fit the parameter b we would need to separately grade the difficulty of each of the considered targets, and it is not immediately obvious how this could be done for our experiment.

Instead, we resort to the more basic measurements of accuracy and speed (how quickly the assistant can make a decision). However, accuracy or speed alone cannot fully characterise the performance of an interface, as both factors play a role. To combine accuracy and speed, the Inverse Efficiency Score (IES) (Townsend and Ashby 1978) can be used. The metric is defined as $IES = RT/Accuracy$; with RT being the average response time of the correct actions taken by the subject. Although this metric has been widely used in cognitive and human-performance research, its application to our study presents two problems. First, the response time for guessing the intention of the driver when they are moving to different goals cannot be directly compared. This is exemplified in Figure 3.12: the driver's goal takes much longer to become clear when they are moving towards 'C' than 'A'. Second, given the difficulty of the task considered, it is possible that some subjects would achieve an accuracy of zero in our experiment. This would lead to the undesirable effect of evaluating IES to infinity.

To remedy both problems, we adapt the IES to the formulation shown in Equation (3.1). Here, *Accuracy* is defined as the number of goals correctly inferred by the assistant divided by the total amount of goals considered (three); and the fraction of time ($FracTime$) is calculated as the ratio between the time that the assistant takes to make an inference and the total time that the driver took to reach their goal. The resulting metric, which we call Inference of Intention Helpfulness Score (IIHS), is bounded between zero an one, with higher values indicating more helpful interventions. As an intuition on the meaning of the IIHS metric, a few scenarios are evaluated: a wrong guess of the goal provides no help for the driver and, accordingly, IIHS would be evaluated to zero in this case. Similarly, a zero score is obtained if the goal is guessed correctly but when the driver is already at the goal. On the other hand, a correctly inferred goal will be more helpful the earlier the guess is made, thus increasing the score, up to a maximum of one. This intuition is also illustrated

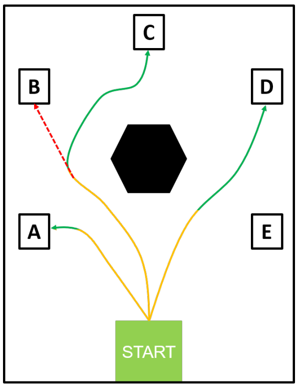


Figure 3.12: Illustration of how assistance can be helpful in different levels. Here, different lines represent navigation to different goals, with the colour changing the moment the assistant infers the intention of the driver. If the inference is wrong, such as the arrow pointing to goal B, the assistance provided is never helpful. If the inference is correct but happens when the driver is already very close to the goal (arrow pointing to A), the assistance is also not very helpful. On the other hand, a correct and timely inference (arrow pointing to D) can make navigation faster and safer for the driver. This intuition is captured by our IIHS metric.

in Figure 3.12.

$$IIHS = Accuracy * (1 - FracTime) \tag{3.1}$$

3.4.3 Results and discussion

In total, 12 people participated in the experiment (10 men and 2 women). Of these, 6 participants reported having prior experience with a powered wheelchair (5 men and 1 woman). Ages ranged from 22 to 32 years old, approximately evenly distributed. One of the participants was left-handed.

Performance measurements Averages (along with standard deviations) for accuracy, fraction of time taken for inference and IIHS are shown in Table 3.1. Figure 3.13 shows the average IIHS score per user for the different combinations of interfaces, and users are grouped by their response to the survey question of having or not prior experience with a powered wheelchair. Filled circles represent users' preferences. For a different perspective, Figure 3.14 displays how the performance of individual subjects varied with the addition of multimodal interfaces to the baseline case of 'map only'.

Questionnaire responses After the experiment, subjects were asked which was their preferred combination of interfaces. The results can be seen in

Table 3.1: Results of our user study. *Average (standard deviation)* values are reported for each of the metrics considered. Users' subjective preferences for each of the interfaces were also recorded.

	Accuracy	Frac. time	IIHS	Pref.
Map only	0.917 (0.28)	0.778 (0.12)	0.185 (0.11)	0%
Haptic	0.917 (0.28)	0.795 (0.10)	0.174 (0.10)	33%
Gaze	0.861 (0.35)	0.789 (0.12)	0.166 (0.12)	42%
Hap. and gaze	0.833 (0.38)	0.774 (0.12)	0.162 (0.11)	25%

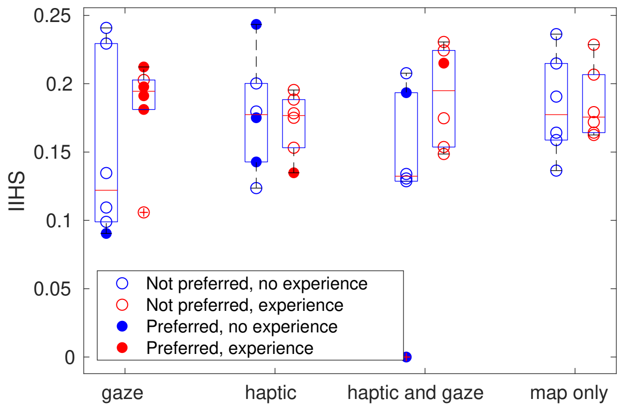


Figure 3.13: Average performance of subjects in terms of IIHS. Subjects grouped by their response to having or not prior experience with powered wheelchairs. Full circles represent subjects' preferred combinations of interfaces. People with prior experience in operating a power wheelchair tend to benefit more the use of multimodal interfaces.

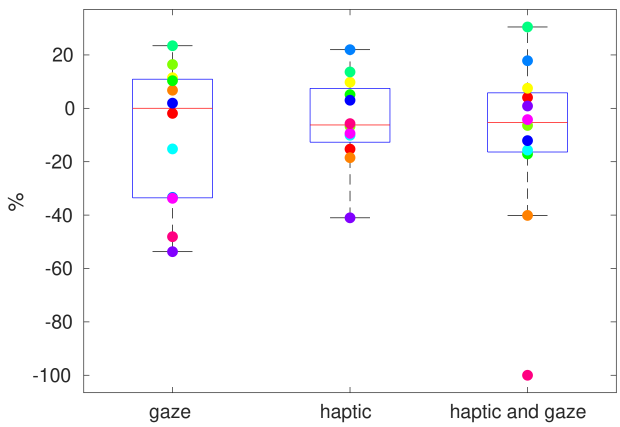


Figure 3.14: Each subject of the study is represented by a colour in this plot, which shows how much performance improved by adding extra interfaces on top of 'map only'. The vertical axis indicates the percentage of improvement in terms of the IIHS score. Although no combination of interfaces can clearly outperform the others *across* subjects, most people experienced a positive impact when using at least one of the additional interfaces.

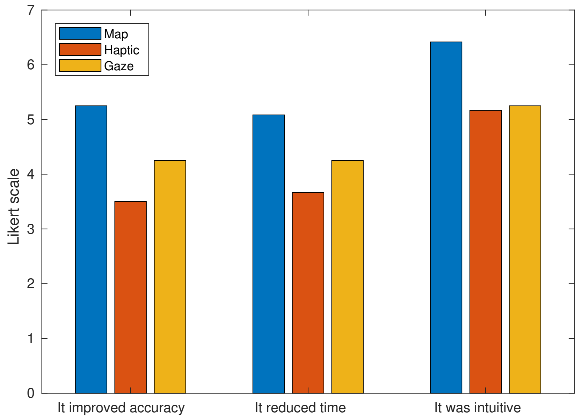


Figure 3.15: Users' subjective evaluation of each interface separately. While the 'map' interface consistently outperforms the others, when used alone it was not preferred by any of the subjects (see Table 3.1. This again shows the value in including multimodal interfaces to facilitate the task of inference of intention.

Table 3.1. Additionally, regarding the interfaces individually, subjects were asked to rate on a 7-point Likert Scale how much they agreed with each of the following statements: "It was intuitive to use", "It helped me predict goals more accurately", "It helped me predict goals faster". The average scores are shown in Figure 3.15.

Discussion The experimental results provide interesting insights into how subjects react to the incorporation of different modalities of interfaces. First, from Figure 3.13, it can be noticed that no combination of interfaces clearly outperforms the others. However, distinct patterns are observed for the two groups of subjects.

For the experienced group, adding multimodal interfaces tends to improve performance, with 'haptic and gaze' being the best scoring combination. Additionally, 4 out of 6 subjects chose their preferred combination of interfaces the same as their best performing one, yielding a 67% predicting accuracy for the preference question. The inexperienced group, on the other hand, exhibits a larger variance in performance between subjects, and it is not clear that adding extra interfaces is helpful. Indeed, 'haptic and gaze' was the worst-performing combination for this group. Furthermore, the preference question only achieved a 17% predicting accuracy for this group and the preferred combination even had an average performance worst than the average of the other options.

For the experienced group, the increase in performance upon addition of interfaces seems natural, since subjects would have used their driving experiences as expectations for the remote driver behaviour, and having extra sources of information helped them confirm these expectations. This agrees with prior work on cognitive studies, which show that humans use their own internal models when predicting the actions of others (Demiris, Aziz-Zadeh, and Bonaiuto 2014). In contrast, by not having this prior expectation of behaviour, the extra interfaces become less useful for the inexperienced group, and might even act as a distracting factor, as mentioned by some subjects during the general feedback session. This helps to explain the low predicting accuracy of the preference question for this group.

Nevertheless, even when analysing both groups together, as in Figure 3.14, it is observed that 9 out of the 12 subjects experienced improvements with at least one of the combined interfaces, as opposed to ‘map only’ (indicated by a colour being above the zero line for one or more of the three categories in that figure). From this figure, it is interesting to notice how much subjects’ performance can vary between different interfaces, even though the correct goals to be inferred were always the same. The explanation for this might be related to an apparent contradiction in the questionnaire responses. While Table 3.1 shows that ‘map only’ wasn’t preferred by any subject, Figure 3.15 shows that, when analysed individually, it consistently outperforms haptic and gaze, which *on average* achieved lower scores in all three topics of the questionnaire. This happened because, as reported during the general feedback session, many subjects, especially from the inexperienced group, had a strong rejection against either haptic or gaze information, usually due to finding it distracting (intense tremor) or misleading (drivers looking at closer obstacles instead of the goals). This suggests that people might naturally have different inclinations towards using visual or tactile senses for the given task. Thus, our study indicates that adding extra interfaces can improve performance, but the appropriate combination is largely dependent on the assistant’s level of experience and natural preferences.

Regarding an apparent low overall score for all interfaces, one should observe that the IIHS metric is naturally skewed, due to a zero score being given upon an inaccurate guess, independent of the time taken for it. However, estimations with relatively low scores can still be very beneficial - an IIHS of 0.5, for example, means that a correct estimation was made when the driver was only halfway through its path, which would be very helpful. This also means that the effective dynamic range for the metric is narrow, and as such even small improvements on the score are valuable.

3.4.4 Limitations

Limitations of interfaces This study allowed us to observe some constraints into the maximum impact that the haptic-pairing and gaze-estimation interfaces can have in the IIHS. First, the haptic interface has a clear potential in reducing the time needed for an estimation, given that the movement of the joystick precedes the wheelchair movement, due to inertia. However, this benefit is limited, in the sense that it is only indicative of the desired goal when the driver is already close to the end of the path.

Something similar can be said about the gaze interface. We found that, in a small room, drivers only gaze the target shortly before approaching it. Most time is spent looking at nearby obstacles, trying to stay away from them. This was especially true in our experiment because the obstacle course had some relatively narrow passages and the simulated disability imposed extra difficulty in navigation. In different conditions, like an open area with fewer obstacles, the gaze-behaviour of drivers might have been different, more goal-oriented, which would make this interface more helpful.

Study limitations Post hoc observations also allowed us to notice limitations on the experiment itself, which should be overcome in a future study. Most importantly, we observed the need for a more extensive training phase. A single demo run proved to not be sufficient for people to get completely familiarised with the different interfaces and in which way they could aid at the given task. Since people are capable of adapting to new sources of information at different rates, this led to a wide variance in the results, which should be avoidable if a longer training session is present. This was corroborated again at the general feedback session, where some subjects reported learning *during* the experiment how to better use the interfaces.

Other minor drawbacks of the experiment were: 1) the camera facing the drivers could have been better positioned, to avoid glare from the ceiling lamps, which can reduce the performance of the gaze-estimation framework; 2) the intensity of the simulated disability was found by a medical doctor to be exaggerated in comparison to real patients of Parkinson Disease - it could have been slightly reduced and the frequency slightly increased. Performing these adjustments before data recording might increase the potential benefits of gaze and haptic interfaces, respectively.

3.5 Conclusions

This chapter had the goal of addressing the following research problem: "How can a human assistant intuitively provide demonstrations of navigation support to powered wheelchair users?". We started by arguing that a teleoperation platform is the most straightforward way of achieving this and outlined the requirements for such platform. Then, we presented our custom solution, highlighting our contributions in hardware integration and the software developed to synchronise and process all the data generated by and reaching the wheelchair.

One of the major challenges in using teleoperation to provide navigation assistance to another person is that, being remote, it may be hard to correctly understand the intention of the driver. To overcome this difficulty, our platform makes use of virtual reality and multimodal interfaces, allowing the assistant to more intuitively and timely interpret the raw data registered by the wheelchair's sensors. The different interfaces are based on map and localisation, haptic-pairing and gaze-estimation, and can all be used simultaneously or individually by a remote assistant. Furthermore, virtual reality environments were developed for visualisation in both computer monitors and VR headsets, each with their benefits and drawbacks.

Our platform also incorporates features to improve the driver's experience. First, the readings from the laser-scanners can be used to perform autonomous collision or obstacle avoidance, to make navigation safer. This remains in place even when the remote assistant is controlling the wheelchair, to avoid collisions against unnoticed obstacles. Alternatively, in a previously mapped scenario, autonomous navigation is also possible. To offer the driver more immediate feedback about the help provided by the remote assistant, their joysticks are haptically-paired at all times.

To assess the potential benefits that our platform can have in facilitating the demonstration of assistance, a user study was conducted. The goal was to explore the impact that the different combinations of interfaces have on the assistant's capability of remote inference of intention. Results indicate that the added interfaces representing the driver's state can improve performance on the task in terms of the IIHS metric; in particular for users familiar with the operation of powered wheelchairs. However, it also indicates that the most appropriate choice of interfaces is highly dependent on users' natural preferences. In this sense, it seems like the best solution is to offer remote assistants the possibility of choosing their preferred combination of interfaces.

Chapter 4

Learning from Triadic Interactions

While smart wheelchair technology is deemed to be helpful for many users of powered wheelchairs (Simpson, LoPresti, and Cooper 2008), the drivers that struggle with hand-control disabilities are certainly among those that could benefit the most. Unfortunately, there are a myriad of conditions (cerebral palsy, Parkinson's disease, amyotrophic lateral sclerosis, Erb's palsy, etc.) that can lead to this type of impairment. The symptoms include loss of dexterity, tremors, muscle weakness and spasms, and can affect each person in a unique way. As a consequence, it is difficult to derive hard-coded policies that can satisfyingly assist with all these disabilities. Instead, better results should be obtained by deriving personalised assistive policies, tailored to the needs of each driver.

When robots need to execute custom tasks that are difficult to be programmatically designed, but can readily be performed by humans, Learning by Demonstration (LbD) (Argall et al. 2009) can usually be leveraged. If the task is related to assisting someone, it then falls within Learning Assistance by Demonstration (LAD) (Soh and Demiris 2013), a subset of LbD. In the context of robotic wheelchairs, one can leverage the tacit knowledge of health-care workers, who are familiar with the driver and their condition, and ask them to provide demonstrations of how to help that specific person. Then, a learning model can be used to approximate a mapping function between input data (sensor readings and driver commands) and the provided demonstrations of assistance. If learning is successful, the wheelchair should now be able to autonomously help the driver.

This chapter describes how LAD can be used to learn an assistive policy by observing the triadic interaction between a driver, a robotic wheelchair and a remote assistant. First, in Section 4.1, we present a general overview of the LAD framework, formally define the problem setting, and delve into the design decisions that were made in our research, while contrasting it against previous work. Following, Section 4.2 describes a user study conducted to investigate the matter of model generalisation to unseen environments, and analyse how different combinations of machine learning models and dimensionality reduction algorithms affect this. The chapter closes with concluding thoughts in Section 4.3.

Parts of this chapter form an extended version of the work presented in (Schettino and Demiris 2020).

4.1 Learning Assistance by Demonstration

As previously discussed, Learning Assistance by Demonstration (LAD) is concerned with triadic interactions where the demonstrations of a human agent are used to teach a robot how to assist another human. The concept has previously been explored with robotic wheelchairs to provide customised assistance to individuals with driving difficulties. In this setting, an initial training stage is used to collect data from both a driver and from a remote person providing assistance demonstrations. This data is then used to fit a machine learning model, which should be able to generalise and substitute the assistant when training is over; at least partially.

This section first presents the general framework for the application of LAD in the context of robotic wheelchairs and the problem setting is formally defined. Afterwards, we discuss design choices that further specify our LAD implementation, while contrasting it against previous work.

4.1.1 Framework overview

An overview of the LAD framework, as per our implementation, is shown in Figure 4.1. During a training stage, the driver is tasked with following a designated route. Meanwhile, sensors on the wheelchair are used to record both driver input and environment information. Driver input can be comprised of both direct control commands, which are registered from the joystick position, and indirect cues, such as direction of eye-gaze. Indirect cues are those exerted by the driver and that are not used to directly control the wheelchair, but that might help to inform the driver's intention or state. Other examples could be heart or blinking rates, as indicators of stress or tiredness. Environment information is useful for limiting the range of reasonable goals for the driver,

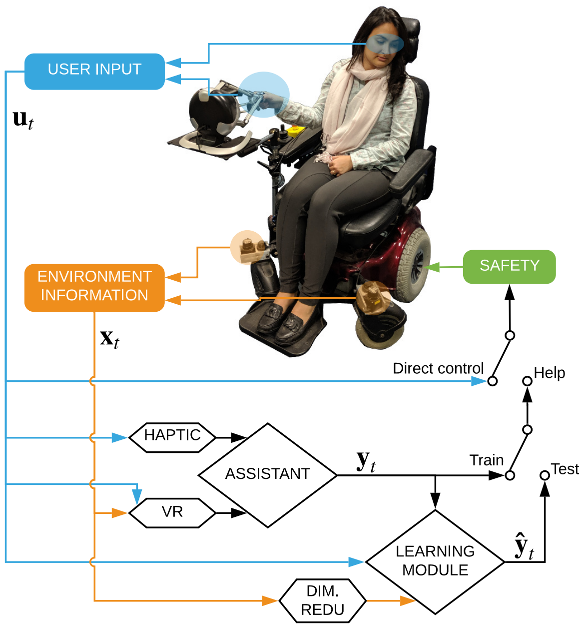


Figure 4.1: System architecture used for Learning Assistance by Demonstration in smart wheelchairs. A remote human assistant provides teleoperated demonstrations of how to help a person with driving difficulties. To infer the driver's intention in real-time, the assistant is aided by haptic and virtual reality interfaces, allowing easier interpretation of the multimodal information registered by the wheelchair. The assistive demonstrations are used to teach a model how to autonomously help the driver.

i.e. they cannot drive through a wall. Thus, this type of information is invaluable in the task of inference of intention. In our system, laser-scanners are used to collect environment information, but cameras or sonars could also be incorporated.

While training is happening, both driver input and environment information are fed in real-time to a remote assistant. As described in Chapter 3, the assistant uses haptic and virtual reality interfaces for more timely and intuitively interpreting the incoming data. The assistant is not aware of the driver's goals and has to continuously infer their intention, adjusting as needed. Based on this inference, the assistant provides a separate driving signal, which should be better, given that they are not hindered by a hand-control impairment. This assistive signal is also recorded.

After training is complete, the recorded data is fed to a learning module, which can be any type of supervised learning algorithm. This modules' task is to learn a mapping between the input (environment information and driver's control signal) and the target signal (assistant's control signal). When the environment information is of high dimensionality, as is the case with the laser-scanners, a dedicated dimensionality reduction technique may be applied to it before it is fed to the learning module. After the learning module is trained on the recorded data, it can be used to automate the actions of the assistant, and the driver should then be able to use it in their daily lives.

Both during training and after it, the driver can easily switch the assistive signal on or off through the push of a button in their joystick (this signal is not used for learning). Independent of the origin of the control signal, it goes through a safety layer before being sent to the wheelchair's low-level controller. This safety mechanism uses the laser-scanners to detect imminent collisions. If the operation is safe, the control signal goes through this layer unmodified. Otherwise, the wheelchair is slowed down or brought to a halt; but the direction of movement is unmodified.

4.1.2 Problem formulation

To formalise the LAD for robotic wheelchairs problem setting, the reader is again referred to Figure 4.1. Data composed of environment information $\mathbf{x}_t$ and driver input $\mathbf{u}_t$ is captured by the wheelchair's sensors and transmitted to the assistant. Aided by the haptic and virtual reality modules for intuitively making use of the information, the assistant continuously provide demonstrations $\mathbf{y}_t$ of how to assist that specific driver. If the assistant's actions are modelled by a mapping function $\mathbf{y}_t = f(\mathbf{u}_t, \mathbf{x}_t) + \theta_t : \mathcal{U}, \mathcal{X} \rightarrow \mathcal{A}$, where θ represents noise due to errors in inference of intention and other random factors, the machine task can then be formally defined as learning a regression function $\hat{\mathbf{y}}_t = \hat{f}(\mathbf{u}_t, \mathbf{x}_t)$ such that $\hat{\mathbf{y}}_t \approx \mathbf{y}_t$ for the entire set $\{\mathcal{U}, \mathcal{X}\}$.

As will be seen in Section 4.2, it is particularly important to examine if the approximation holds for the elements of $\mathcal{X}$ not seen during training.

4.1.3 Design considerations

Although a general overview of the basic LAD framework has already been given, other design decisions are needed to further specify its implementation. These decisions are important because they result in a significant impact on applicability and model performance.

For a more principled approach, before making these design decisions for our custom implementation of LAD for robotic wheelchairs, we first looked at the relevant literature. The most prominent LAD works are (Soh and Demiris 2015a) and (Kucukyilmaz and Demiris 2018). Although similar in terms of the final application, they present some key design differences, which are summarised in Table 4.1. In our implementation, we found that a combination of these design

Table 4.1: Comparison of key design decisions between the most relevant LAD works. In our implementation we used a combination of these choices and also explored new alternatives, focusing on improving generalisation and assistive performance.

	(Soh and Demiris 2015a)	(Kucukyilmaz and Demiris 2018)
Subjects condition	Simulated disability	No disability
When to help	Dedicated learning module	Continuously
Machine learning technique	Custom GP variant	Vanilla GP
Online vs offline learning	Online	Offline
Testing generalisation	Training and testing done on the same track	Testing done on a different track but on the same course
Performance assessment	Predictive and assistive	Predictive
Assistant's view	Direct view	Camera view

choices, coupled with new additions, yielded the best results. Following, we elaborate observations on this.

Subjects condition we understand that the LAD scheme is most beneficial to those users that, besides needing a wheelchair, also have difficulty in controlling it. Thus, simulated disabilities were used in all the user studies we conducted. Although not a perfect representation of hand-control disabilities, the simulations are a reasonable proxy, and the best alternative until the platform is proven effective and tests with target users are possible.

When to help previous work in LAD for robotic wheelchairs employed different strategies for sharing control of navigation between driver and assistant. In (Soh and Demiris 2015a), to provide help the assistant would take over control of the wheelchair whenever they thought help was needed. From a machine learning perspective, this increases complexity, because now the model needs to learn not only how, but also when to help the driver. Accordingly, the authors implemented a dedicated learning module to classify when to activate autonomous assistance. As a second complicating factor, this approach takes control away from the driver, as their commands can be overridden at any time. This not only may have an impact on user satisfaction (Kim et al. 2012), but is also potentially dangerous, if the intention of the driver is not correctly estimated by the machine.

A different approach was taken in (Kucukyilmaz and Demiris 2018), where navigation was continuously shared between driver and assistant through means of haptic pairing. While this does make the machine learning task easier (no need to learn when to help), it also has the downside of constantly distracting the driver with spurious small scale assistive signals.

During initial trial runs in our user studies, however, drivers reported feeling uncomfortable with either approach (assistant takeover or continuously shared navigation). The reason was that they had to unwillingly relegate (partial) control of the vehicle, at times being driven to points contrary to their goals. This agrees with smart wheelchairs literature, where lack of control is reported as a potential rejection factor for users, even when the system performs well (Kairy et al. 2014; Viswanathan et al. 2017).

To improve this, we explored a new paradigm for *when-to-help*, which we call **ask for help**. This simply means letting the driver decide when to release control of the wheelchair and allow the assistant to take over, thus asking for help. During training, the assistant is asked to continuously provide assistive demonstrations, thus building the input-target (driver-assistant signals) data pairs needed for LAD. And even when asking for help, the driver's commands are still transmitted to the assistant, being used to inform them of the driver's intention. Whether in training or testing, the driver asks for help simply by pressing a button on their joystick, and support will correspondingly come from the human assistant or the learnt assistive policy.

With this new setup, the driver always knows when they are relinquishing control and can quickly regain it if the provided assistance disagrees with their intention, thus alleviating the lack of control problem. This is consistent with previous literature of assistive robotics, which shows that people prefer to retain control even at the cost of lower performance or higher workload (Kim et al. 2012; Gopinath, Jain, and Argall 2017). Furthermore, in a LAD scenario, this paradigm has the added benefits of both eliminating the need to learn when-to-help and precluding spurious assistive signals from constantly interfering with the driver normal operation.

Machine learning technique both works in Table 4.1 used GPs as the base machine learning technique, but they varied in terms of the kernels used, usage or not of mixture models and online vs offline learning. Since they have previously been shown to be effective, GPs are a good candidate for implementing the Learning Module. However, we also explored alternative families of learning algorithms, such as Support Vector Machines, Random Forests and Neural Networks, see Section 4.2.3 and Section 5.3. This exploration is important since the alternatives can lead to great variations in model performance when generalisation issues are considered.

Online vs offline learning online learning is an interesting feature for LAD, as it gives the assistant an in-training perspective about the assistive policy being learned, highlighting what needs to be improved. However, online learning algorithms also impose more severe computational limits and, in general, can achieve worse predictive performance than their unbounded offline counterparts. In our work, we only explored offline learning.

Testing generalisation our view is that neither work in Table 4.1 delved deep enough into exploring the generalisation capabilities of the learnt assistive policies (see Section 2.3.2). Conversely, from the user point of view, this matter is of the highest importance, as the learnt model will have to be applied in scenarios entirely different from the training one. Thus, we propose to present drivers with different environments for training and testing - see Section 4.2. Furthermore, it is important to remember that the main goal of LAD is to offer users assistance that is customised to their specific needs. Therefore, it is also imperative to test if the proposed architecture works similarly well for different kinds of impairments - see Section 5.4.5.

Performance assessment while it is interesting to observe how well an assistive behaviour demonstrated by a human expert can be replicated by a LAD model, ultimately, the important factor to be assessed is: what benefits does this model bring to the driver? The work in (Kucukyilmaz and Demiris 2018) only explored the first aspect of performance: "our purpose is to provide a comparison between human-provided guidance and robotic assistance policies, not to show therapeutic benefits of the proposed technique". However, achieving a good predictive performance does not guarantee that good assistive performance will be obtained. Accordingly, they observe that in their study "Lap completion time was significantly increased when the user operated with robotic assistance", thus defeating the purpose of providing autonomous assistance. Conversely, (Soh and Demiris 2015a) assessed both predictive and assistive performance, using lap completion time as a metric to compare *no assistance* vs *human assistance* vs *autonomous assistance*. This is the approach we also follow in our user studies, additionally including new metrics when appropriate - see Section 5.4.1.

Assistant's view previous work on LAD always had the assistant presented with some type of camera or direct view of the wheelchair operation during training. And, at first, it may seem that to best help the driver the assistant should be provided with as much information as possible. Yet, this may lead to a discrepancy between the information available for the assistant and the information available for learning, in an issue commonly known as *correspondence problem* (Argall et al. 2009). Since the help provided by the assistant is used as a target for the learning algorithm, there must be an accurate correspondence between the input data supplied to both agents, or the learning task may turn infeasible. To make this point more concrete, an example is provided: consider that, during a training run, the assistant sees an obstacle from their camera view and uses their joystick to help the driver. If this obstacle is not detectable by the robot, due to being above the laser-scanners' plane, for example, the match between input and output is broken for the machine. Even though the assistance provided is correct, the agents were presented with different input data. From the machine point of view, assistance was given when no obvious obstacle or limitation was present. Nevertheless, it will attempt to learn from this example. This may lead to an incorrect assistive policy being derived, based on a random feature extracted from the raw sensor data[1].

Therefore, this situation presents conflicting goals: while it is important to offer the assistant enough information to enable proper help, it is also important that the same information is provided to both assistant and machine so that learning performance is maximised. To deal with this problem, our teleoperation platform (see Chapter 3) was specifically designed to only present to the assistant the same information that is also available to the robot. That is, there is no third-person camera view of the driver's environment, no direct sight of the wheelchair and no video of the driver themselves, for example. Instead, the assistant only has access to the data registered by the wheelchair's sensors. Using a teleoperation platform in this way enables us to avoid the potential correspondence problem; an approach that has also been used in other LbD work (Zhang et al. 2017). Additionally, the virtual reality and haptic interfaces are used to make the sensor data more easily understandable for the assistant, thus enabling them to provide proper help for the driver. Particularly, the

[1]False examples are a common problem in machine learning applications and many techniques can be used to deal with it. However, it is important to remember that the training stage for this particular application involves the continuous effort of two human agents, one of them disabled. Therefore, the number of examples available for training is more likely to be in the order of dozens, instead of hundreds or thousands that most applications have at disposal. Ergo, in this case, making the most of each example provided could be the difference between failure and success in learning.

VR interface can be used to augment the laser-scanners' readings and give the assistant an increased feeling of presence on the remote scene. The assistant also has better visualisation of the wheelchair's surroundings, not having to worry about blind-spots or having to have the wheelchair constrained to their field of sight.

4.2 Testing generalisation

A basic premise of LAD is that, after collecting training data, the learned policy should be able to help the person using it in their daily lives. For robotic wheelchairs, this translates to the assistive policy being generic enough to still be useful outside of the training space. Nevertheless, the validity of this premise had not been investigated before.

Furthermore, some characteristics of the data naturally available for this scenario pose considerable generalisation concerns for machine learning models. First, there is a need for a triadic interaction between two human agents and a robot, which limits the number of data samples that can be collected in a reasonable amount of time. Second, the data needed to represent environment information with acceptable accuracy is of high dimensionality. These two factors combined can potentially lead to models that overfit the assistive policy to spatial features that are only present in the training course. If performance is then tested on the same course, it might yield apparently good results without being truly useful for the driver during normal use (outside of the training space).

To remedy this, this section describes a study undertaken to evaluate the assistive potential and the generalisation capability of LAD for robotic wheelchairs. Using the teleoperation platform from Chapter 3, data was simultaneously collected from a driver exposed to a simulated disability and from an expert providing assistive demonstrations. The setting described in Section 4.1 was used to fit a variety of learning models to this data. Then, the generalisation capability of these models was assessed by using a new obstacle course for testing. Seeking to improve performance, different combinations of dimensionality reduction techniques and machine learning models were investigated. Finally, track-completion time was used as a metric to contrast the assistive benefits of human and machine-generated help.

4.2.1 Data collection

To address the possible overfitting concerns previously mentioned, special data gathering and testing procedures were devised, as illustrated in Figure 4.2. In a

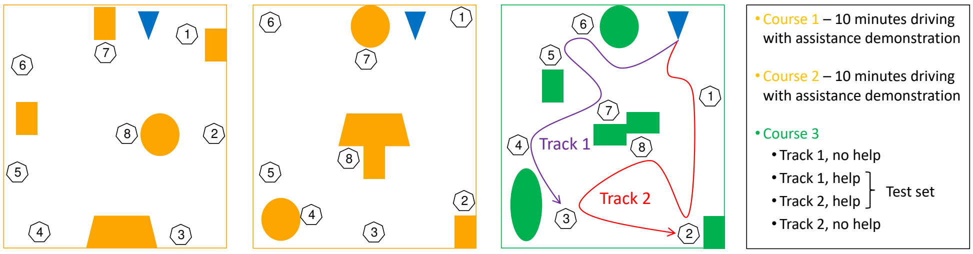


Figure 4.2: Illustration of the experimental procedure used. In orange, two different obstacle courses with marked goal positions are used for gathering a training dataset. The driver is asked to continuously drive to randomly selected goal numbers, while environment and driver information are recorded, along with the provided assistive demonstrations. To test the generalisation capability of the learning model, a third, unseen, course is used. Here, finite tracks are employed and track completion time is used as a metric to test assistive performance.

controlled and wheelchair-safe environment, three physically different scenarios, or obstacle courses, were built. The first two courses were used to compose a training dataset. It is important to have distinct courses composing the training data because we are specifically seeking methods that can generalise to a variety of environments and situations, instead of overfitting to local spatial features (for example, whenever approaching obstacle X, provide an assistive signal to turn left). Each course was marked with predefined target locations and an endless list of goals was generated by randomly sampling the target numbers. The driver was then asked to follow the list, driving to the goals in sequential order for approximately 10 minutes. During this, the assistant was continuously providing assistive driving commands, which were recorded alongside the laser-scan data and the driver's gaze estimation, command velocities and ask for help signal.

The third obstacle course was used only for testing performances. In this course, the driver was given two finite lists with 10 goals each and asked to finish the tracks as fast as possible. The same kind of data was recorded and this formed the test set. To assess the assistive potential of human help, the tracks were also run without the assistant being available. To correct for learning effects, the order in which the assistant was available or not was switched between the tracks.

The data collected was used to train machine learning models and, afterwards, the same procedure was repeated on the third obstacle course. Except that the assistive signals were being automatically generated by the machine, in real-time. This was done to evaluate the assistive performance of the learned model and happened on a separate day, a month later, thus mitigating possible learning effects by the driver.

We want to test the application of the LAD concept in cases where it is likely to be used - with drivers that would normally have difficulty in manoeuvring a wheelchair without assistance. While the ideal scenario would be to test the system directly with our target population, it would also be ill-advised to expose disabled subjects to a still experimental system. Thus, as in previous work on the field (Soh and Demiris 2013; Soh and Demiris 2015a), using a simulated hand-control disability was the compromise found. For easier and fairer comparison against these works, the same disability was used, which consists of difficulty in executing right-turn motions with the joystick. This kind of symptom could be manifested, for example, in people suffering from Erb's palsy, which can lead to a loss of supination power on the forearm (Tortora and Derrickson 2016). The disability was simulated using the force-feedback capability of the joystick, by automatically exerting an opposing force whenever the driver tried to perform right turns.

The target signal for our learning algorithm is formed by the velocity commands issued by the assistant, but only its angular component. This is because, since the simulated disability only affects the driver's steering capability, interfering with their speed control is unlikely to be beneficial. Thus, our control arbitration software was modified to take the linear component of the driver's command velocity and directly bypass it to the robot, ignoring the linear component of the assistant's signal even when the driver asks for help. Accordingly, the assistant's linear velocity signal is not recorded.

Since our goal is to learn *personalised* assistive policies, throughout the experiment a single driver-assistant pair was used. One could argue that increasing the number of subjects would lead to more statistically powerful results. But we note that this would significantly increase the complexity of data collection and likely yield similar outcomes. That is because the wheelchair-driving behaviours of most people in an indoor environment are relatively similar (restricted space, little room for path deviations). The major divergences in driving profiles emerge from the different limitations imposed by distinct disabilities. Thus, to observe statistically significant effects we would have conduct a study analysing multiple disabilities and greatly increase the number of subjects, which is beyond our resources at this stage. Instead, a simpler approach was taken. Here we maintain the single driver-assistant pair, doing a sanity-check that the LAD approach can indeed work with human drivers, and assessing if the learnt assistive policy can generalise to unseen environments. In Chapter 5 we resort to simulations to investigate LAD's capability of adapting to multiple disability types.

4.2.2 Data preprocessing

After recording the data, the entire dataset was time synchronised to 4 Hz using a nearest-sample approach; except for the control signal, which used a nearest-previous-sample approach due to its naturally low-changing frequency. Each sample is composed of the target signal and 363 features - 360 readings from the laser-scans, the estimated yaw angle of the driver's eye-gaze and the driver's linear and angular command velocities. An initial analysis revealed that some of the samples on the training set were corrupted due to problems with gaze-estimation software, and thus had to be discarded. In the end, the training dataset was composed of approximately 3000 samples.

A common problem with laser-scan samples is that many of the readings can be corrupted with invalid values. In our system, this happens if the reading is out of the sensor's maximum range (an Inf value is returned) or if the reading falls within the footprint of the wheelchair (a NaN value is returned). To cope with this and avoid discarding samples, the NaN values are first converted to Inf and then all readings are inverted. This results in readings close to the robot, perhaps the most relevant ones, having a high value and invalid readings having a zero value.

The relatively high number of features, due to the 360 channels on the scan samples, can lead to overfitting problems. Therefore, it is prudent to test the effects of applying different dimensionality reduction algorithms to this data. The simplest method for this is spatial subsampling, which was applied by all previous work on the field. Maintaining a uniform angular distance between readings of the circular scan, tests were performed keeping 256, 64, 16 and 4 readings from each sample. Additionally, tests were also performed keeping the full scan sample and completely removing it.

However, due to the natural characteristics of the readings generated by a moving laser scanner, this simplistic feature selection mechanism can deteriorate the quality of input data. As a more advanced approach, we propose the use of Autoencoders (Vincent et al. 2008). We experimented with this using unsupervised training and up to four fully-connected layers, resulting in encodings of sizes 256, 64, 16 or 4. For all layers, logistic sigmoid was used as the encoding and decoding transfer functions and both weight and sparsity regularisation was applied. As an alternative, PCA (Abdi and Williams 2010) was also considered, keeping enough principal components to explain 99, 90, 75 and 50% of the observed variance of the scan data. Post-analysis showed that this corresponded to respectively 229, 59, 14 and 5 components.

4.2.3 Learning algorithms

Finally, we explored the impact that different machine learning algorithms have on the generalisation capability of LAD. All previous work on the field (Soh and Demiris 2013; Soh and Demiris 2015a; Kucukyilmaz and Demiris 2015; Kucukyilmaz and Demiris 2018) used variants of GP (Rasmussen and Williams 2006) as learning models. When the number of features is not very high, GPs can be an interesting option due to their ability to directly capture uncertainty in the input space, being trainable with relatively small datasets and allowing one to readily insert prior knowledge into training. However, the choice of the kernel function is important. While Squared Exponential kernel has previously been used (Kucukyilmaz and Demiris 2018), we noticed that it can be overly smooth to capture the variations observed in the target signal, shown in Figure 4.3. Hence, we opted for also evaluating performance with the Matérn 3/2 kernel (Rasmussen and Williams 2006). To better deal with the cases where the number of features is high, we further tested this kernel with Automatic Relevance Determination (Rasmussen and Williams 2006). For completeness, we also compared GPs to different families of common machine learning models, namely: Linear Regression (LR), Support Vector Machine (SVM), Random Forest and Gradient Boosting (Boost) (the acronyms are used as headers in Table 4.2).

Due to their simplicity and training speed, linear regression models (Bishop 2006) are sometimes used to construct baseline performance for function approximation. Notwithstanding, these models are prone to overfitting, especially when the dimensionality of the input space is high. Thus, besides vanilla linear regression, in this work we also experimented with Lasso and Ridge regularisation, to mitigate overfitting. In these cases, cross-validation was used to determine appropriate values for the regularisation coefficient.

Support Vector Machine (SVM) (Wang 2005), or more specifically Support Vector Regression for the regression variation, have model regularisation builtin to their optimisation functions, and also allow for non-linear regression through the use of kernel transformations. For our tests, we experimented with linear, Gaussian and polynomial kernels. Cross-validation was used to select suitable values for the insensitive band, box constraint, kernel scale and polynomial order (for the polynomial kernel) parameters.

A common way to improve generalisation performance and robustness of prediction is to employ ensemble methods. Here we consider two types of such methods, both based on decision trees. First we experiment with Random Forests (Breiman 2001), which independently trains multiple trees and then aggregates their predictions to arrive at the final model output. Each tree is fit on a bootstrap replica of the training data (re-sample the dataset with replacement until each replica has the same size as the original dataset). Additionally, at each split, each tree in the ensemble randomly selects only a subset of the input features to decide how to divide the decision space. This technique is normally used with 'strong' learners, such as deep decision trees, to reduce the variance of the ensemble. On the other hand, Gradient Boosting (Hastie, Tibshirani, and Friedman 2009) algorithms are normally used with 'weak' learners, such as shallow decision trees, to reduce the bias of the ensemble. In this case, the trees are trained sequentially, each on a modified version of the dataset. The modification happens (in the regression case) by extracting the difference (Mean Squared Error (MSE)) between the target signal and the current prediction of the ensemble, which is made by aggregating all previously fit trees. For both Random Forest and Gradient Boosting models, cross-validation optimisation was used to choose appropriate values for the number of learning cycles used and the minimum leaf size of each tree on the ensemble. For Gradient Boosting the learning rate of shrinkage is also optimised.

For hyperparameter tuning, 5-fold cross-validation with MSE as the loss function was always used and Bayesian optimisation applied for more efficient search of minima. Additionally, the maximum number of objective function evaluations and maximum training time was the same for all models. All the dimensionality reduction techniques and machine learning models were implemented and trained with Matlab and, where omitted, default parameters were used.

The moments where assistance is requested by the driver can present different characteristics from the rest of the driving time, where no help is needed. For the simulated disability studied here, for example, the driver does not need any help when going straight, backwards or doing left turns. And when doing right turns, the sensor data will tend to be localised on a particular region of the input space. For instance, it is likely that the driver's angular velocity will be zero or slightly negative, their gaze direction should be to the right and the laser-scan readings should indicate an open space to the right. Since this region of the input space is more relevant to the application at hand, our learning algorithms should accordingly give more importance to it. Nevertheless, the samples corresponding to when no assistance was requested - which are the majority of the recorded data - can still be useful for the learning algorithm. To conciliate these, a sample-weighting mechanism was implemented and used with all learning models. This means that the prediction loss of samples where assistance was requested contribute more heavily to the

Table 4.2: Validation loss (RMSE) for different combinations of dimensionality reduction techniques and machine learning models. By using a more appropriate kernel for the GPs model and a more advanced technique for reducing the laser-scan data, we managed to improve performance when compared against the combination used by the latest work on the field.

	LR	LR Lasso	LR Ridge	GP SqrExp	GP Matérn3/2	GP ARD Matérn3/2	SVM Lin	SVM Gauss	SVM Poly	Random Forest	Boost	Average
FullScan	0.4143	0.3209	0.6483	0.4155	0.4137	0.4405	0.3941	0.4255	0.4207	0.3956	0.3466	0.4214
SubSamp256	0.3893	0.4840	0.3353	0.4541	0.4445	0.4390	0.4125	0.3931	0.3444	0.3595	0.3277	0.3985
SubSamp64	0.3799	0.3697	0.3792	0.3947	0.3925	0.3829	0.3907	0.5306	0.4843	0.3566	0.3473	0.4007
SubSamp16	0.3659	0.3647	0.3651	0.3788	0.4162	0.4593	0.3662	0.4297	0.4227	0.3430	0.3587	0.3882
SubSamp4	0.3329	0.3330	0.3329	0.3600	0.3528	0.3370	0.3357	0.4820	0.3330	0.3418	0.3361	0.3525
NoScan	0.3214	0.3214	0.3214	0.3085	0.3058	0.3048	0.3291	0.3337	0.3824	0.3794	0.3321	0.3309
PCA99	0.3831	0.3983	0.4536	0.7630	0.4771	0.4587	0.4115	0.4391	0.3598	0.5585	0.3506	0.4594
PCA90	0.3639	0.3590	0.3793	0.3922	0.3890	0.3912	0.3951	0.4486	0.3507	0.4013	0.3460	0.3833
PCA75	0.3659	0.3660	0.3659	0.3511	0.3554	0.3589	0.3825	0.6660	0.4661	0.4531	0.3823	0.4103
PCA50	0.3513	0.3516	0.3514	0.3556	0.3975	0.4029	0.3429	0.7010	0.3332	0.3546	0.3373	0.3890
Autoenc256	0.3824	0.5003	0.3494	0.3550	0.3832	0.3921	0.4119	0.3847	0.4341	0.3834	0.3392	0.3923
Autoenc64	0.3396	0.3288	0.3375	0.3389	0.3282	0.3358	0.3366	0.3202	0.3335	0.3282	0.3332	0.3328
Autoenc16	0.3425	0.3218	0.3368	0.2924	0.2914	0.2942	0.3285	0.3296	0.3265	0.3366	0.3289	0.3208
Autoenc4	0.3410	0.3214	0.3214	0.3172	0.3081	0.3276	0.3342	0.3218	0.3124	0.3442	0.3313	0.3255
Average	0.3624	0.3672	0.3770	0.3912	0.3754	0.3804	0.3694	0.4433	0.3788	0.3811	0.3427	0.3790

learning process. After experimenting with a few different values (1, 2, 5, 10, 50, 100), a 10-1 ratio between the weights of assistance/no assistance samples yielded good performance. When measuring predictive performance for the validation/test sets, only the samples where assistance was requested were considered.

4.2.4 Results

Table 4.2 shows Root Mean Squared Error losses for all combinations of dimensionality reduction techniques and models discussed in the previous section. Highlighted in green is the combination used by the latest related work (Kucukyilmaz and Demiris 2018) and in orange is the best observed performance. Using data collected from the test tracks when help was being requested, Figure 4.3 compares the responses predicted by these combinations against the target signal (angular velocities provided by the assistant). To evaluate assistive performance, the best observed combination was used for making real-time predictions on the wheelchair. The driver that participated in the data-gathering experiment was asked to return and was allowed as much time as wanted to familiarise themselves with the new driving mode. Afterwards, the same testing procedure on the third course was repeated; except with assistance being automatically generated by the machine. Lap completion times for all runs on both test tracks are shown in Table 4.3.

Discussion

As can be seen from Table 4.2, a Gaussian Process model using the Matérn 3/2 kernel combined with Autoencoders for reducing the dimension of laser

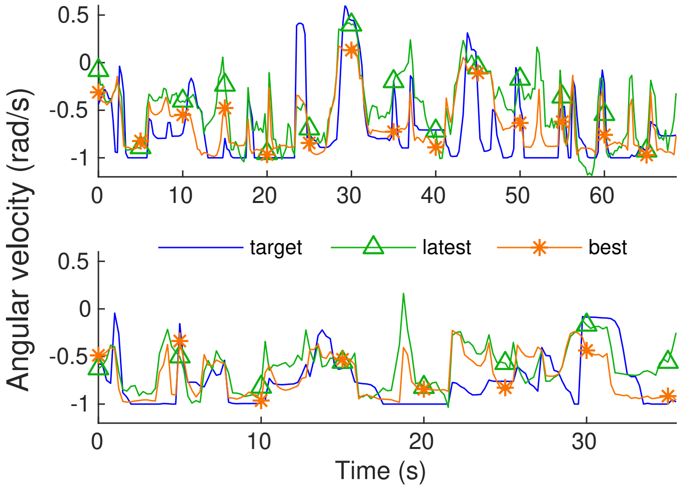


Figure 4.3: Comparison of human provided assistive signal against machine predictions for both test tracks. The improved combination (orange) of machine learning and dimensionality reduction algorithms makes the model better capable of tracking the target signal[2], when compared against the combination used by the latest work on the field (green). For RMSE values refer to Table 4.2.

Table 4.3: Time taken by the driver to complete a lap on the test tracks and the improvements obtained when assistance was available. Despite variations in the driver's attitude between the two days, the learnt assistive policy is still capable of helping the driver overcome the imposed disability.

	1st day		2nd day	
	No help	Improvement due to human help	No help	Improvement due to machine help
Track 1	281 s	-0.3%	388 s	15.5%
Track 2	279 s	3.2%	379 s	4.0%

data down to 16 features attained the best prediction performance, with a 23% improvement against the combination used by the latest work on the field. Although not shown here, we also observed moderate improvements due to the inclusion of gaze-estimation as a predictor and to the use of different weights for samples were assistance was being requested.

From Table 4.3 we first notice a difference in lap completion times between the two days when data was collected (the two 'No help' columns), despite the same tracks and disability being used. This seems to be related to the driver presenting different physical and/or mental states between both days. While on the first day the driver presented a more aggressive driving behaviour, on the second day they seemed more subdued by the limitation imposed by the disability. Although not anticipated, these variations in motor function and attitude are common with many real disabilities (Bruguerolle and Simon 2002; Lamers et al. 2012; Bellamy et al. 1991).

To correct for this variation, the improvements in track completion times are calculated on a day-by-day basis. Following this analysis, we observed that the help provided by the assistant did not lead to significant improvements in lap completion times. Contributing to this is the possibility that, by overcoming the simulated disability, the driver was already navigating near their best possible performance, hence not allowing much room for improvement. Additionally, when the assistant needs to remotely infer the intention of the driver, their assistive signal naturally becomes more erratic, thus reducing its usability.

This unpredictability in the assistant behaviour also leads to a more noisy target function, therefore increasing the difficulty of the machine learning task. Notwithstanding, we observed that the policy learned by our model was able to generate helpful assistive commands in real-time. On the second day of the experiment, when the driver exhibited a slower driving behaviour and thus could potentially take more advantage of the custom assistance, machine help led to an average improvement in track completion time of 9.8%. We note that, by training with multiple demonstration samples, the learning module can filter-out part of the inference of intention noise, thus generating a more consistent assistive policy, which benefits the driver.

From this experiment, it was also interesting to observe how the LAD model correctly learned to use the information from all available sources. In Figure 4.4, we show a scene where the driver reached a point where she could turn left or right. Before deciding, she stopped at an open space and looked both ways, without moving the joystick. Because the laser-scan and joystick inputs were

[2] For a better result in prediction performance see Figure 5.9.

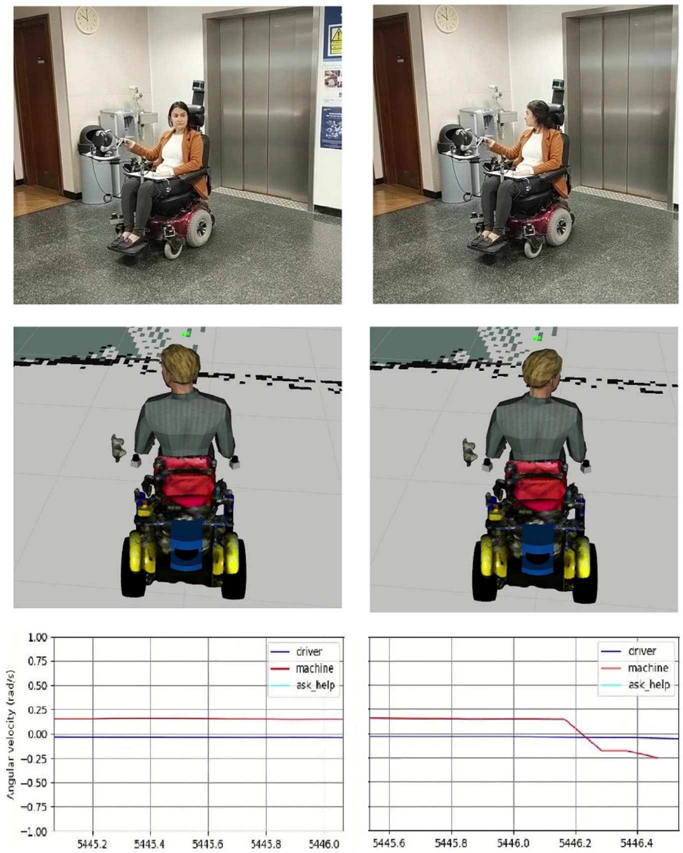


Figure 4.4: Example of how our model uses the multimodal inputs to predict assistance. In this scene, the driver reaches an open space and holds her joystick in position while deciding if going left or right. On the bottom plots, the blue lines represent the driver's angular velocity input, held at zero, and the red lines are the model's prediction of assistance. As can be seen, when the joystick input is not informative, the model automatically leverages the gaze estimation for predicting assistance.

not useful in informing the intention of the driver at this particular moment, the model relies at the gaze input for predicting what assistance should be offered. This behaviour was not explicitly programmed or enforced in any way. Instead, it naturally emerges from our implementation of LAD.

Limitations

A limitation of this study is related to the disability simulation. In an attempt to more realistically mimic a real human disability, we did not constrain the velocity commands that were sent to the robot and merely imposed opposing forces at the joystick during right turns. However, due to hardware limitations of the joystick, the driver is able to overcome the simulated disability, should they exert enough force. This made our experiment more susceptible to the interference of "human factors", such as differences in attitude, which complicates analysis. Notwithstanding, we posit that with real disabilities factors like tiredness, distraction and learning should play similar roles. As a way to reduce the experimental noise caused by these factors, in Chapter 5 we also explored the usage of simulated drivers and assistants.

4.3 Conclusions

This chapter had the goal of addressing the following research problem: "How can personalised assistive policies be learned by observing the triadic interaction between a robotic wheelchair, an impaired driver and an expert human assistant?". For this, we first delved into the concept of Learning Assistance by Demonstration (LAD), a subset of Learning by Demonstration. It was shown how this technique may be used to provide autonomous and personalised assistance to wheelchair drivers.

After formally defining the problem and presenting the general framework for LAD, we discussed some design decisions that differentiate our research from previous work on this field. For instance, a new paradigm for 'when-to-help' the driver was explored. Switching to 'ask for help' instead of assistant takeover or continuously shared navigation, as was done in previous work, guaranteed more control for the driver, which tends to improve user satisfaction (Kairy et al. 2014; Viswanathan et al. 2017). Additionally, this new paradigm was beneficial in alleviating the machine learning problem, by removing the need to learn when to help the driver. We also explored the usage of new machine learning techniques, used more extensive evaluations of performance[3] and used a teleoperation platform that facilitates learning by avoiding the correspondence problem in LbD (Argall et al. 2009).

Afterwards, we examined the generalisation capabilities of LAD. A basic premise of this technique is that LAD policies should be able to generalise to physically different scenarios, so that the driver can use the autonomous assistance in their daily lives, not only on the training course; but this had not been thoroughly tested before. To investigate the validity of this premise, we conducted a user study with a special data collection procedure. Two physically different scenarios were used for collecting expert demonstrations of how to assist a driver with a simulated disability. The data was then used to fit a model and the learnt assistive policy was tested on a separate scenario, not seen during training. To reduce the possibility of model overfitting, dedicated

[3] This will be shown on Chapter 5

data-preprocessing and dimensionality reduction techniques were employed. Furthermore, various families of supervised learning algorithms were tested, to assess their generalisation performance. Using Autoencoders to reduce the dimension of laser-scan data and Gaussian Process as the learning model, a 23% improvement in prediction performance was achieved, against the combination used by the latest work on the field. Using this model to assist a driver exposed to a simulated disability, it was observed a 9.8% reduction in track-completion time, compared to driving without assistance. Therefore, it was observed that generalisation to physically different scenarios is possible for LAD and that the use of more sophisticated learning models and dimensionality reduction techniques can help in this sense.

Chapter 5

Improving Generalisation

Chapter 4 showed that Learning Assistance by Demonstration (LAD) can be used to improve the driving performance of wheelchair users with hand-control disabilities. However, the complex nature of experiments involving two people and a robot make it difficult to fully assess the benefits provided by this approach. This is because, besides being long and laborious, the experiments are not well suited for within-subjects studies, due to learning effects significantly impacting drivers' performance. Consequently, this also hinders the development and testing of new techniques that could improve assistive results.

To cope with this, in this chapter we resort to simulations to perform a more systematic evaluation of the effectiveness of LAD. Through repeated runs, we can assess how variations in data collection, data preprocessing, model architecture and training procedure affect assistive performance. These results are used to advance the development of LAD techniques, with a special focus on improving model generalisation to unseen scenarios.

The chapter is organised as follows: Section 5.1 describes a custom simulator developed which is capable of reproducing the wheelchair assisted-navigation experience and also of recreating the full triadic interaction between driver, assistant and robot. Section 5.2 details the data preprocessing techniques that were used. Section 5.3 discusses many approaches that were considered in terms of model architecture and model training, before arriving at the option that we found as best suitable for enhanced generalisation performance. The hyperparameter optimisation procedure used to arrive at this model architecture is also explained. Section 5.4 describes a series of experiments conducted to analyse different aspects of LAD performance, including generalisation capability, assistive performance and personalisation. A discussion of the results obtained and a statement of limitations is also presented. Finally, conclusions are presented in Section 5.5.

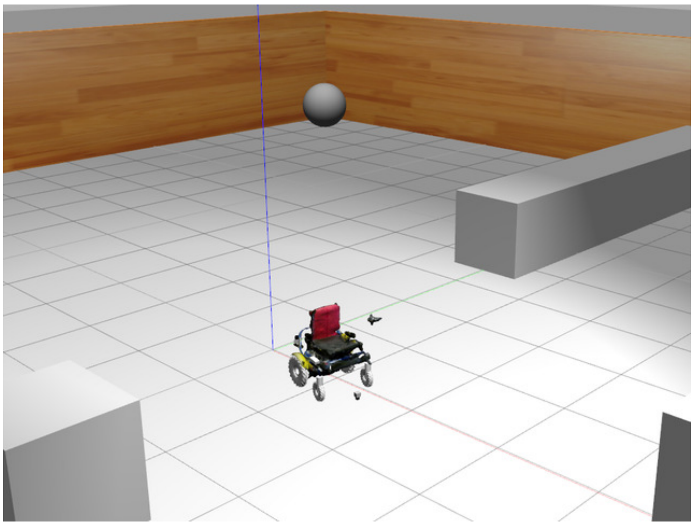


Figure 5.1: Custom simulation environment developed. The driving dynamics and sensors' characteristics are designed to resemble the real wheelchair as close as possible. Using path planning and local control, both driver and assistant behaviours can be simulated. Alternatively, humans can take up their roles using simple joysticks.

5.1 Simulations

Having a simulator of the real wheelchairs allows quick experimentation with changes in software, techniques and algorithms, which greatly reduces development time. Furthermore, if the same ROS interface is used for both the real and simulated robots, higher-level applications can first be tested in simulation, until functionally correct implementations are achieved. Then, by altering a single software switch, everything can be seamlessly transferred to the real robot for testing performance. The simulator also allows one to explore scenarios that would be too complex or dangerous to test using the real wheelchairs. Lastly, simulations can also be used to collect artificial data for use with machine learning algorithms.

To leverage all these benefits, a custom simulator was developed using Gazebo[1], which handles both the ROS interface and the physics simulation, including collision and inertia. The same properties that are used to describe the real robot are also used in the simulation, which makes it easy to implement changes on both sides and also makes the simulated behaviour more realistic. An example scene with the children's wheelchair in a fictitious world is shown in Figure 5.1.

The wheelchair model is based on differential driving and take as input

[1] http://gazebosim.org

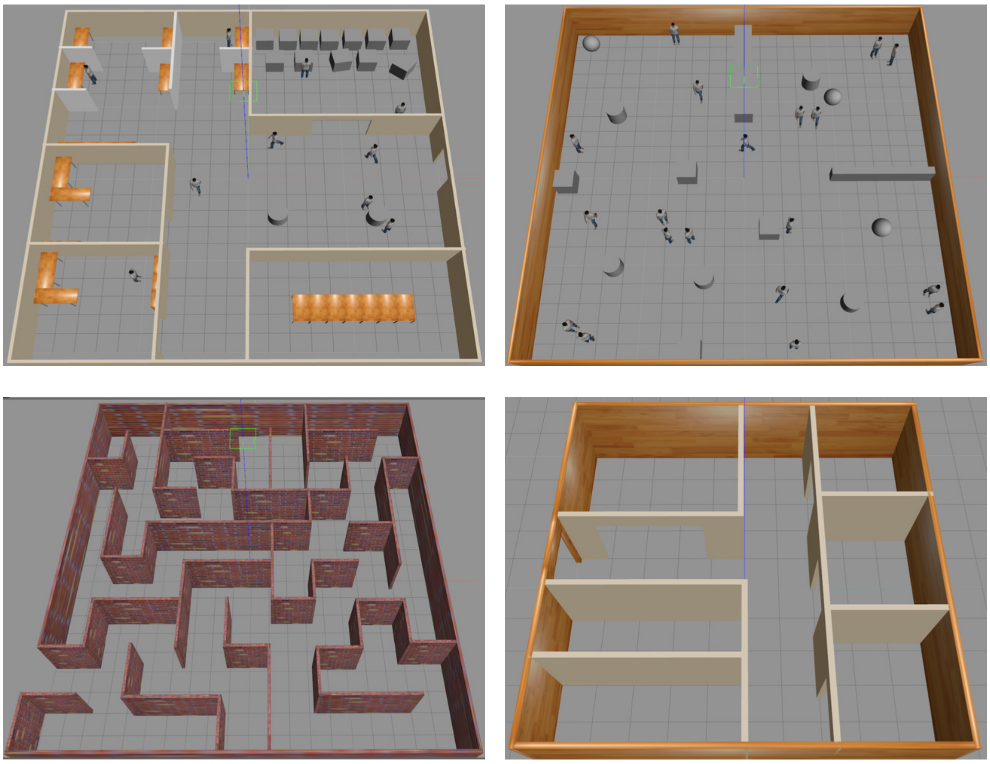


Figure 5.2: Examples of different simulated environments created for testing LAD and collecting artificial data. The increased variability in the training data can lead to improved generalisation performance.

command velocities, like the real wheelchair. The characteristics and noise present in sensor readings, such as those from the IMU and laser-scanners, are modelled based on the nominal values defined by the manufacturers. Calibration of the relative sensor positions was done using an optical motion capture system based on infra-red cameras[2]. The graphics for the model were created using 3D scanning with a Microsoft Kinect.

In order to increase the variability of the data that can be collected in simulation, and also to better test the robustness of assistive models, multiple scenarios were created. These scenarios, four of which are depicted in Figure 5.2, were designed as abstract generalisations of common indoor environments.

5.1.1 Simulating triadic interactions

When the map of a scenario is available, localisation, path planning and local control can be combined to simulate an optimal (non-disabled) driver. For this, 2D maps of the environment are first created by a human driver, which uses a

[2] optitrack.com

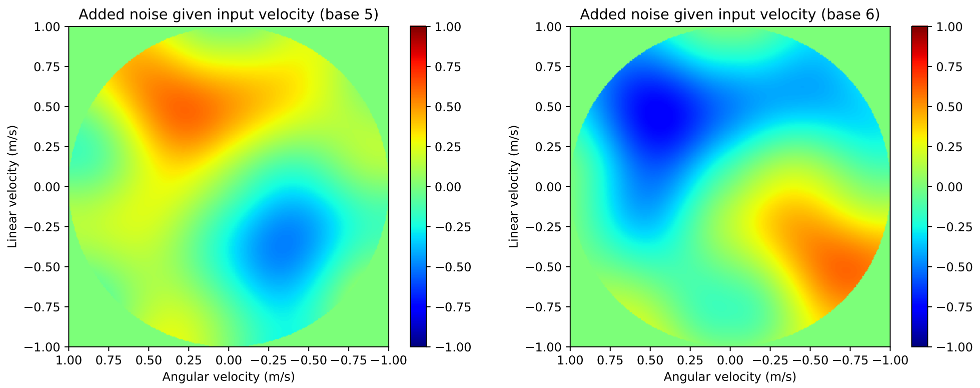


Figure 5.3: Examples of how distortion maps can be procedurally generated using additive Perlin Noise. The colour scale shows how much angular velocity distortion is added for a given input command. The distortions are used to simulate different hand-control disabilities: on the left, a driver that tends to overshoot when doing forward-left turns; on the right, a driver that will pull to the right when attempting to drive straight.

joystick to explore the simulated scenario with SLAM activated. Having the map, localisation and path planning can be used to create waypoints reaching pre-defined target locations. Finally, local control is used to autonomously guide the wheelchair through these waypoints, while using the Dynamic Window Approach (Fox, Burgard, and Thrun 1997) to avoid static or moving obstacles along the path. The parameters of the SLAM, localisation, path planning and local control algorithms are tuned specifically for this wheelchair, to ensure smooth control.

Once the optimal driver is available, the motor disability on the hand can be simulated by mapping the optimal control signals to distorted ones. This is done by using Perlin Noise (Perlin 1985). Perlin noise is a technique developed in the 1980s for the Disney movie *Tron*, to compose natural-looking computer-generated textures (e.g. fire, smoke, etc.). The textures generated are multidimensional, pseudo-random, noise maps, where the level and interplay of details can be controlled by parameters of the algorithm. We use the Perlin maps to inject a varying distortion in the driver's angular velocity, given their linear and angular velocity input. Examples of these distortion maps are shown in Figure 5.3.

There exists a computationally faster version of Perlin Noise, known as Simplex Noise, which we use in this work through the implementation of the python library noise[3]. The function *pnoise2* (because we have a 2D input) uses

[3] https://pypi.org/project/noise/

three parameters to control the characteristics of the output noise map[4]:

- *octaves*: determines how many levels of details the output map should have. In a terrain generation application, for example, the first level could determine the presence of mountains, the second would determine the presence of boulders in those mountains and the third rocks. Because the input space for our application is relatively small, only $\pm 1\mathbb{R}^2$ for linear and angular velocities, a single octave was used to avoid spikes in the distortion map.

- *lacunarity*: determines the frequency of each octave relative to the previous one (how many boulder in each mountain). We use the default value of 2.

- *persistence*: determines the amplitude of each octave relative to the previous one (size of boulders relative to mountains), thus defining how much they contribute to the overall shape of the map. We use the default value of 0.5.

Because Perlin noise is only pseudo-random, the same output can be consistently obtained for a given input. If a different distortion map is needed, the input-output relation can be deterministically changed simply by using a different seed (known as *base* in the Perlin algorithm). This allows one to procedurally generate different disabilities and easily switch between them.

In addition to the Perlin distortion, white noise is also added to the driver's input signal, thus emulating both the deterministic and random parts of the disability. Besides varying the Perlin base, the relative levels of the deterministic and random distortions can easily be adjusted, even *during* the simulation.

The assistant is simulated simply by using the optimal driver's control signals - that is, before the distortion. This is possible because local control is employed to follow the designated trajectories. Thus, even when the disabled driver deviates from the optimal path, the corresponding assistive signal being generated is still correct. Hence, in a LAD application, as the driver follows the designated trajectories a correspondence set is built between disability and optimal signals, respectively input and target.

As an alternative, the assistant and/or the driver can be directly controlled by humans, using regular video-game joysticks. The alternative mode is activated by changing a single parameter in the simulator and allows for human

[4]The explanation and examples are adapted from `https://medium.com/@yvanscher/playing-with-perlin-noise-generating-realistic-archipelagos-b59f004d8401`

takeover even when the simulation is already running. The simulated hand-control disability and trajectories to be followed by the driver are kept, and model assistance can be turned on and off with the push of a button. This alternative mode is very useful for quickly testing different assistive policies developed and gathering the impressions of drivers, without exposing them to potentially dangerous situations.

The main benefit of working with simulated drivers and assistant is the ability to perform repeated runs, thus collecting more statistically powerful results. This allows one to assess what tends to work better. When running experiments with the real wheelchair, one is very limited in the number of interactions that can be recorded, due to constraints in the time it takes, space available for training, etc. But more importantly, drivers are very susceptible to factors like tiredness, mood, concentration, and learning effects. These factors generate a lot of experimental noise, potentially concealing trends in data that could distinguish good and poor assistive models. While we understand that simulations do not perfectly represent the driver-assistant interaction, they give great insight into what *can* work or not.

5.2 Data preprocessing

During training, either in simulation or on the real wheelchair, all data is recorded using rosbags[5], the standard format used by ROS. This includes the laser-scan data and the command velocities from the driver and the assistant[6]. Because these different sources generate data at different rates and asynchronously, some work has to be done before they can form a full dataset. We first define a sampling rate of 10 Hz, equivalent to the slowest data source, the laser-scanners. Then, a nearest-sample approach is used to synchronise all sources. During inference time, a mechanism is used to buffer incoming data and synchronise it to the same sampling rate.

As discussed in Chapter 4, a common problem with laser readings is the corruption with NaN and Inf values. Furthermore, the data is of high dimensionality, with up to 720 channels in every sample to cover a 360 degrees scan. Here we tackle both problems simultaneously, by employing a 'valid average' approach. This is done by grouping neighbour channels and taking the average of the valid values encountered. While this reduces the granularity of the laser-scan data, it also facilitates the task of machine learning. Our tests showed that

[5] `http://wiki.ros.org/rosbag`

[6] During testing, the wheelchair positions on the map and the planned paths are also recorded to measure performance - see Section 5.4.1

reducing the number of channels to 72 worked well. If any group of neighbour channels contains only invalid readings, their corresponding average is replaced by the laser-scanner nominal maximum rage. Afterwards, the readings are inverted and normalised to the 0-1 range. This leads to obstacles near the wheelchair yielding readings close to 1 and far away obstacles readings close to 0. Although this is a non-linear transformation, we observed that it helps model training by converging to better solutions.

Command velocities from driver and assistant are derived from the 2D position of their joysticks and naturally have low dimensionality and bounded limits, without invalid readings. Hence, they are directly used in model training, without any preprocessing.

5.3 Model architecture and training

In the user study of Section 4.2, an Autoencoder was used to reduce the dimensionality of the laser-scan input, which was then combined with the driver's velocity input before being fed to a Gaussian Process model to predict the target angular velocity. Here, after considering several alternatives, we opted for a model fully based on deep neural networks (Lecun, Bengio, and Hinton 2015), as it achieved the best performance in our tests. This section discusses all the approaches we explored (Section 5.3.1) in terms of model architecture and model training, and the hyperparameter optimisation procedure we employed (Section 5.3.2) before arriving at our final model architecture (Section 5.3.3).

5.3.1 Approaches explored

Developing an assistive model that can help disabled drivers in a meaningful way is not a trivial task. Before arriving at our final model architecture and training method, many alternatives had to be explored. For completeness, here we briefly discuss our exploration experience, hoping that the insights may be useful for continuing research in this field.

- **Separate inputs**: before testing LAD with all the input data available, tests were done using just the laser input and just the velocity input separately. Although in the end it became clear the benefits of using the full input, this step was very useful for finding the best architecture to process each part of the input. Furthermore, using just the velocity input turned out to be a very good initial baseline for the work done with simulated drivers and simulated assistants, given that the control signals of the former are derived from the latter.

- **Multi Layer Perceptron versus Convolutional Neural Network**: previous work in LAD for smart wheelchairs performed heavy sub-sampling of laser-scan channels, which were then treated as independent inputs. However, similar to images, there is a clear correlation between the information on neighbour channels. In neural networks, this correlation can usually be exploited for improved learning performance by using convolutional layers (Zeiler and Fergus 2014), instead of densely connected ones. In this work, we explored both alternatives and found that convolutional layers yield better performance, as long as the appropriate data preprocessing steps are conducted.

- **Laser-scan preprocessing**: Section 5.3 discusses all the preprocessing steps that are applied to the laser-scan input. Of these, the change that yielded the greatest performance gains was using the 'valid average' approach, instead of pure sub-sampling. We also explored reducing the number of laser channels to different values (16, 36, 144, 256, 720) before settling at 72. Other changes considered were whether or not to invert the readings and whether or not to normalise the final values.

- **Recurrent Neural Networks**: we started our research using memoryless models, as was done by previous work. However, exploiting the temporal information of the inputs, especially the velocity commands, turned out to be crucial for increasing the model's capacity to infer the driver's intention. Before settling at 2 seconds recurrent windows, other values (1, 4, 10, 20) were also tested. We also explored using different layer types: vanilla recurrent layers (Pascanu, Mikolov, and Bengio 2013), Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Unit (GRU) (Chung et al. 2014), with GRU clearly outperforming the others. Additionally, we also investigated merging the laser-scan and velocity inputs before the recurrent layers, thus forming a single recurrent path; but this led to a small drop in performance.

- **Other machine learning algorithms**: besides deep neural networks, we also experimented with linear models, testing Lasso, Ridge and Elastic-Net (Zou and Hastie 2005) regularisation. These simple models generated a good baseline for comparing performance. As done in previous work (Soh and Demiris 2015a; Kucukyilmaz and Demiris 2018), we also explored using Gaussian Processes with various kernels, but they could not outperform the neural network models.

- **Autoencoders**: for the user study in Chapter 4, we used Autoencoders to reduce the dimensionality of the laser-scan input. Here, we tested this option again, by pre-training an Autoencoder to reconstruct the laser-scan data, disregarding the temporal information. We considered both densely connected and convolutional Autoencoders, with variations in the number and size of layers. However, we could not observe any significant improvement over directly using uninitialised convolutional layers.

- **Multi-headed convolutional networks**: instead of using a single convolutional path for processing the laser-scan data, we also considered having multiple paths that were merged afterwards. The difference between these shallow paths was the kernel size of the first layer. The hope was that this would allow the network to capture both coarse and fine-grained features of the environment from the beginning. This, however, heavily increased the network's memory consumption without leading to meaningful improvements.

- **Residual connections and batch normalisation**: following recent trends with ResNets (He et al. 2016), we investigated implementing residual connections between our convolutional layers. However, the architectures we could afford (data-wise) were not deep enough to benefit from this modification. Similarly, using Batch Normalisation (Ioffe and Szegedy 2015) did not yield any gains for our application.

- **Network architecture from (Valente, Joly, and De La Fortelle 2019)**: this work used a CNN-LSTM architecture to perform odometry estimation from 2D laser-scan data collected from autonomous vehicles. Given the partial similarity of that work to our research, we performed tests replicating their neural network architecture but did not achieve good results. This may be explained by several factors that separate both works: there is a significant difference in the amount of data available for training; different laser-scanners were used to collect the data (which lead to different resolution, number of channels and quality of the laser-scans); outdoor versus indoor applications; and different end-goals (estimating current movement versus the desired movement).

Two other approaches explored are more involved, and thus are described in more details in the following paragraphs.

Multimodal inputs Combining inputs of different modalities, such as velocity and laser-scan inputs, is a common problem in machine learning. Aside from simply merging the data sources, many strategies are available to deal with multimodal inputs (Baltrusaitis, Ahuja, and Morency 2019; Zambelli, Cully, and Demiris 2020). We experimented with the approach proposed in (Ngiam et al. 2011). The main idea behind this method is to augment the training data by separately zeroing out each of the input modalities and then using an Autoencoder to reconstruct the full input. The encoder is then extracted and used as the input layer for a full classification or regression algorithm. The goal is that, by reconstructing missing inputs, the encoder layers are forced to learn representations that better capture the correlation between different modalities.

For our application, this was tested by taking the initial training set and effectively tripling its size by merging three datasets: original training data, training data with the velocity input zeroed-out and training data with the laser-scan input zeroed-out. After pre-training the Autoencoder on the augmented dataset, the encoder layers were used to handle the full input and then connected to the recurrent and fully connected layers. The end result, however, was sub-optimal. The main reason was that the Autoencoder could not converge to satisfactory reconstruction results. We believe that this happened because of the large difference between the dimensionality of the two input modalities. While it should in principle be possible to partially reconstruct the angular velocity input (2 dimensions) from the laser-scan data (72 dimensions), the other way around is impractical. And without good reconstruction results the method is turned ineffective. We attempted to mitigate this issue by further reducing the dimensionality of the laser-scans (4, 8 and 16 dimensions), but then the information becomes too coarse to correctly represent the wheelchair's surroundings.

Learning Using Privileged Information Our work with simulations and using path planning for mimicking the behaviour of a human assistant resulted in an interesting idea: could we simultaneously learn from both the path planning and the human control signals? The motivation for this is that, as was shown in Chapter 3, human assistants are capable of inferring the intention of a remote driver, which is a very challenging task from a machine learning perspective. On the other hand, if the driver's intention was known *a priori*, the path planning and local control algorithms are capable of generating optimal driving signals, which could then be used as the target. Hence, if possible, it would be interesting to use the human assistant signal solely as another input feature in the training set, and the path planning signal as the target to be learned. The problem of this approach, of course, is that we are interested in

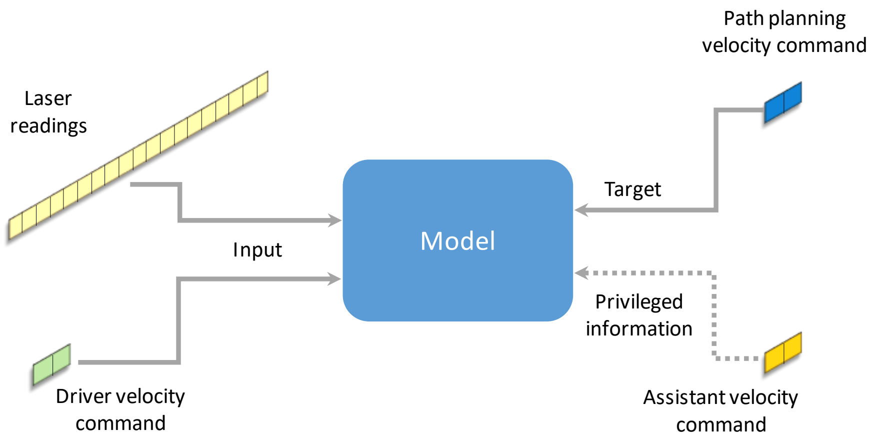


Figure 5.4: Schematic of how Learning Using Privileged Information can be used in the context of LAD for smart wheelchairs. Despite being a theoretically sound approach for combining two sources of demonstration of assistance, we could not observe gains in performance when using it.

an autonomous assistive system and, as such, the human assistant signal is not available during inference time.

This problem of dealing with features that are only available during training and not during inference has previously been studied. The concept was first formalised by Vapnik et al. in the context of Support Vector Machine (SVM) (Boser, Guyon, and Vapnik 1992), and termed Learning Using Privileged Information (LUPI) (Vapnik and Vashist 2009; Pechyony and Vapnik 2010; Vapnik and Izmailov 2015). The theory behind it, however, is very rooted in the SVM framework, and not easily extensible to other learning algorithms. Nevertheless, other works were capable of reformulating the LUPI theory as an importance weighting mechanism (Sharmanska, Quadrianto, and Lampert 2014; Lapin, Hein, and Schiele 2013). In particular, more weight is given to the samples that are easier to learn, whereas the demand on hard or impossible to predict samples is relaxed.

Weight samples can be directly incorporated into neural networks, so we used this approach to test if LUPI can be effective in improving LAD performance. The weight samples were derived from the similarity between the human assistant, which was unaware of the driver's goals, and path planning signals. The moments where the assistant could more easily infer the intention of the driver would naturally lead to assistive control signals that are more similar to the path planning generated signals, thus increasing the weights in these easier and more important to learn samples. A schematic of this implementation is shown in Figure 5.4.

To our surprise, however, at least in simulation this approach did not lead to any significant improvements in terms of predictive performance, assistive performance or data efficiency (a claimed feature of LUPI). We do not have a strong hypothesis of why this is the case. It could be that in the case of human drivers, where the correlation between driver and target signal is reduced, LUPI would have a stronger effect in differentiating the easy and hard samples. However, without the benefit of multiple repeated runs, we could not verify if this is indeed the case.

Alternatives exist to the classic LUPI paradigm, which still allow the use of features that are only available during training. A neighbouring concept to the LUPI reformulation as importance weighting is curriculum learning (Bengio et al. 2009), which presents easier to learn samples *first* during the training process. Another option would be to present the privileged information as a multimodal feature and then use the process described in (Ngiam et al. 2011) to pre-train an Autoencoder. This would force the encoder layers to learn a representation of the privileged feature based on the non-privileged ones. Then, at inference time, the missing feature could just be zeroed-out. A more direct approach would be to use a dedicated secondary model to learn how to predict the missing information based on the non-privileged features, as was done in (Chen et al. 2019). Lastly, a more advanced port of the LUPI paradigm to deep learning architectures was presented in (Lambert, Sener, and Savarese 2018). There, it is proposed the use of heteroscedastic dropout in convolutional and recurrent networks, with the variance of the dropout layers being a function of the privileged information. During training, this leads to model uncertainty being controlled by the privileged information. During test/inference time, as usual, dropout is not utilised and thus the privileged information is not needed. We could not investigate the use of these alternatives, but the approach of (Lambert, Sener, and Savarese 2018) seems particularly promising. We live its study with LAD as a future line of research.

5.3.2 Hyperparameter optimisation

To arrive at our final network architecture, a series of optimisation steps were used. First, a heuristic approach was employed, attempting to observe obvious trends in the variation of performance for large jumps in hyperparameter selection. For example, we would perform 10 runs of training on the same architecture, varying only the number of dense layers at the end from one to five, and observing if there were clear improvements favouring one value or another. This served only to filter down the possible values of hyperparameters to reasonable ones. This process was also guided by general heuristics found in the neural network literature.

Once reasonable values were selected, further filtering was performed using automatic hyperparameter optimisation. Given the large number of possible combinations of hyperparameters and the non-negligible training time, grid search was not a viable approach. Instead, the Hyperband algorithm (Li et al. 2018) was employed, using the Keras Tuner framework [7]. Hyperband will select random combinations of hyperparameters and start many trials in parallel. After a given number of epochs, it will stop half of the trials and only continue training on the better-performing ones. The halving is repeated until only the best trial is kept and this one is trained until early-stopping. Then the whole process is restarted with new combinations of hyperparameters, running repeatedly for a given time budget. Because, unlike grid or random search, most trials are stopped early on during the training process, more computational resources can be allocated to quickly finish the potentially good trials. This allows more combinations of hyperparameters to be explored in a given time budget.

But even with Hyperband, it was not possible to perform an exhaustive search over the hyperparameter space in a reasonable amount of time. Instead, the goal of this strategy is to find trends in the selection of hyperparameters that lead to improved model performance. For this, a graphical tool like HiPlot [8] can be very useful. A custom script [9] was created to translate the results of Keras Tuner to a file that can be interpreted by HiPlot. A sample result can be seen in Figure 5.5. By selecting only the trials that had worse or better performance, one can see the variation in the distribution of specific hyperparameters, thus guiding the selection process.

Finally, once a more reasonable amount of hyperparameter combinations was left to test, Bayesian Optimisation (Snoek, Larochelle, and Adams 2012) was employed to select the best combination of hyperparameters. This works by using Bayesian uncertainty to balance exploration and exploitation of the hyperparameter space, instead of randomly searching it.

5.3.3 Final model architecture

A schematic of the final model architecture we used is shown in Figure 5.6. After being preprocessed, the laser-scan readings go through a series of 1D convolutional and MaxPooling layers pairs. Despite consuming more memory, we observed that employing a relatively large kernel size on the first convolutional

[7] https://keras-team.github.io/keras-tuner
[8] https://github.com/facebookresearch/hiplot
[9] https://github.com/vbschettino/keras-tuner-hiplot

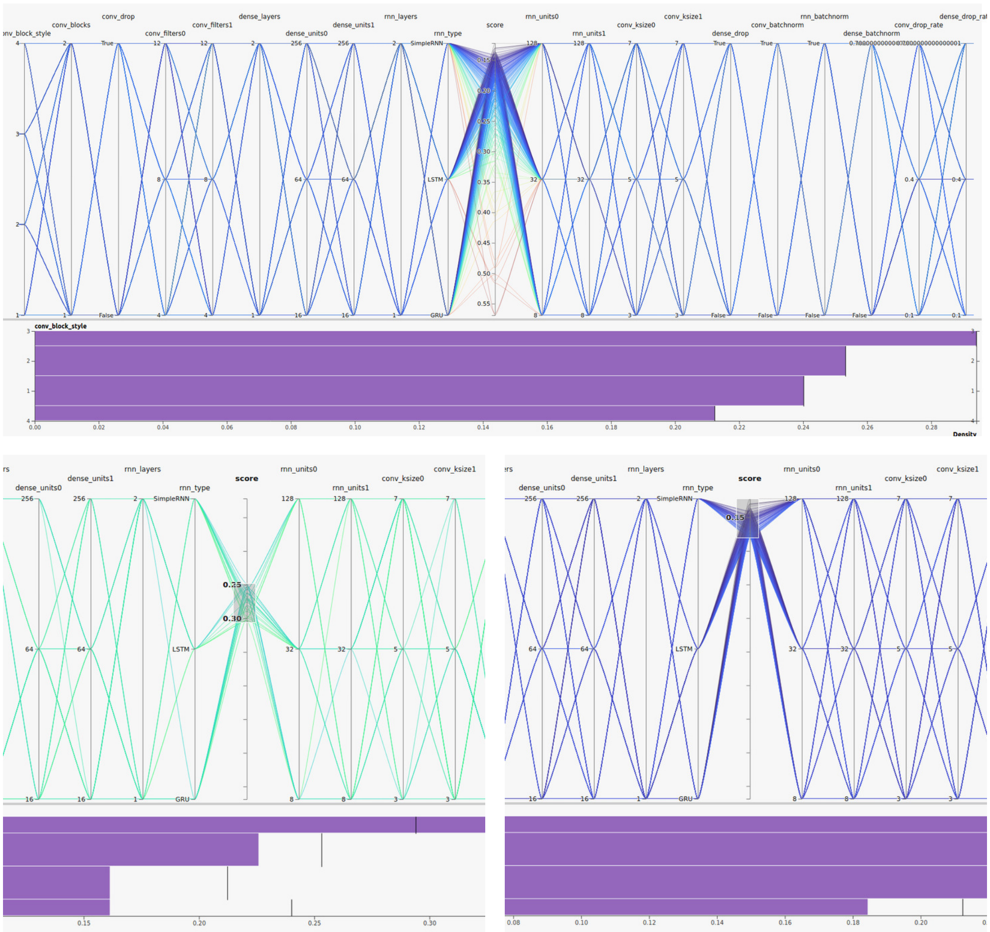


Figure 5.5: Insight into the hyperparameters selection procedure. *Top:* each line on this plot represent one trial of training on the full dataset, with a different combination of hyperparameters. Darker colours indicate better performance. The bar plot shows the distributions of values for a specific hyperparameter. *Bottom:* by selecting only a subset of trials according to the end score obtained, we can observe how the distribution of values shifts for that hyperparameter.

layer was beneficial to the model performance, because this allows the model to capture coarse features of the environment, such as door openings, corners, and long walls on corridors. As the scanner readings go deeper inside the model, the kernel size is reduced and the number of convolutional filters increased, allowing the model to capture more fine-grained details of the environment.

After being reduced to 4 channels due to pooling, and flattened, the scanner readings go through a recurrent layer (GRU (Chung et al. 2014)), which allows the model to capture changes in the environment through time, as the driver navigates. The recurrent unit work with a buffered window of the past 20 scans, which, at 10 Hz, corresponds to two seconds. The driver input commands are composed of only linear and angular velocity components and thus are fed directly to a separate recurrent path, with the same time-window size. Both input paths are then merged, before being fed to a series of interleaved fully connected and dropout (Srivastava et al. 2014) layers, to predict the target angular velocity.

TensorFlow (Abadi et al. 2016) was used for building and training the model. Adam (Kingma and Ba 2015) was used as the optimiser for the backpropagation algorithm and MSE as the loss function. ReLu was used as the activation function for the convolutional and fully connected layers. The recurrent layers used hyperbolic tangent as the main activation function and sigmoid for the recurrent step. A batch size of 32 samples was found to yield the best training performance. Early stopping, using a separate driving scenario as the validation set, was employed to avoid overfitting.

5.4 Experiments

In this section we explore a series of experiments conducted to analyse different aspects of the assistance that can be generated by LAD. The main goals of the experiments are:

- to understand the impact that different training and validation procedures have on generalisation performance (Section 5.4.2);
- to evaluate assistive performance in a realistic setting (Section 5.4.3);
- to test if LAD can generate useful assistive models for multiple types of impairments (Section 5.4.4);
- to observe if personalised assistive models are being learnt (Section 5.4.5).

These topics are separately investigated in the following subsections, after briefly introducing the metrics used in the experiments.

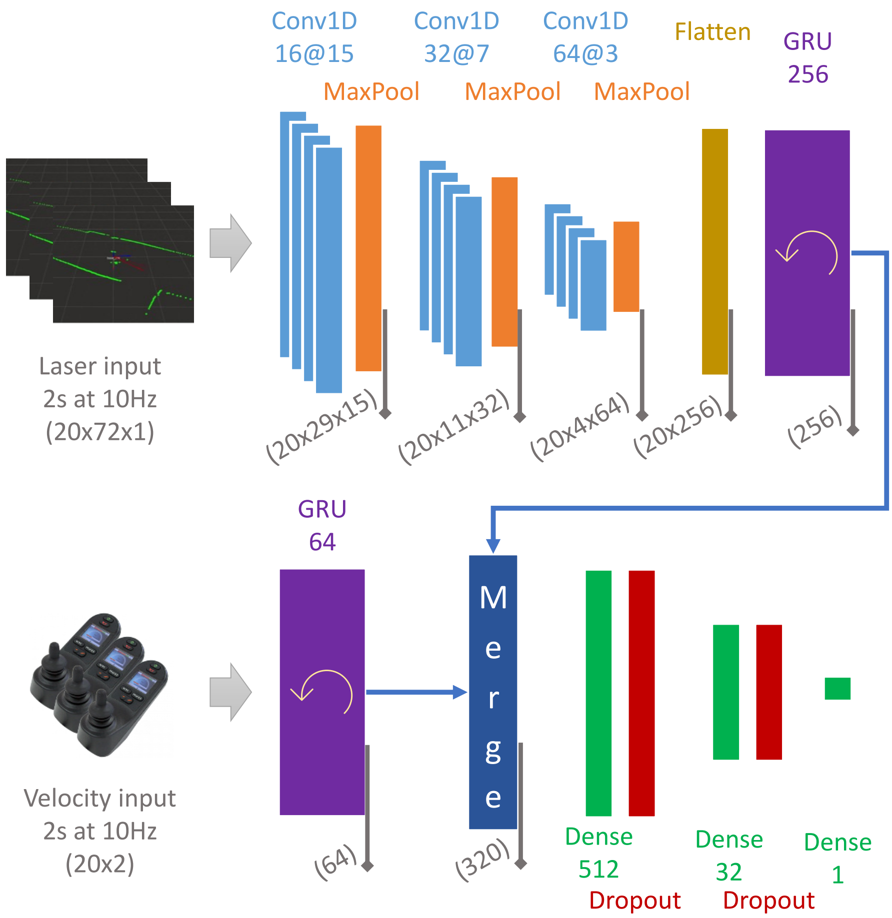


Figure 5.6: Schematic of model architecture. The 72 channels of laser reading are fed to a series of 1D convolutional and MaxPooling layers before being flattened and fed to a recurrent layer. The driver's commands are directly fed to a separate recurrent path. Finally, both paths are merged before reaching the fully connected and dropout layers.

5.4.1 Metrics

MSE is used as the main metric to compare the predictive performance of different models. The error is considered between the optimal assistive signal recorded on a test course and the model prediction. However, simply obtaining a lower predictive loss does not necessarily lead to better assistive performance. For example, a model that is always slightly off when assisting to drive straight in open areas, but accurate when going through doorways, is arguably better than the opposite. Nevertheless, this might not show in the total predictive error, especially if the dataset used for training is mainly composed of driving straight (which is usually the case). Hence, a distinction is made here between predictive and assistive performance, with the former being useful for quickly comparing different models, and the latter being the measurement of true interest for drivers.

Regarding assistive performance, it is important to evaluate different dimensions of the help that is being provided. For example, an assistive model that improves navigation speed but also leads to constant collisions against obstacles might not be advantageous for a driver. For this reason, here we consider four different metrics to evaluate assistive performance.

- Time to complete a lap: total time needed to complete a lap in the dedicated test course.
- Average distance from planned path: the distance between the wheelchair's centre position to the closest point on the current planned path is measured every 0.1 seconds, and then averaged for each completed lap.
- Relative amount of time spent clearing collisions: escape sequences are a feature used by the local control software employed (*move_base*) to 'escape' a dangerous situation for the robot. This happens when the robot collides with an object or an imminent collision is detected. In these cases, the robot will manoeuvre backwards in a straight line for a fixed distance at a constant low velocity, attempting to clear the collision. This metric reports the relative amount of time spent performing these escape sequences, compared to the total lap time.
- Number of instructor interventions needed: because of the narrow pathways that have to be navigated and the erroneous control signals given by the simulated disabled driver, the wheelchair might sometimes get stuck against an obstacle, temporarily or permanently. But data collection is always monitored by a human instructor. If the robot gets stuck for more than 10 seconds, the instructor takes over control of the wheelchair and

manually navigates it to the closest point on the planned path after the obstacle. This metric reports the number of interventions needed per lap.

All predictive performance results represent an average of 10 training runs with random seed initialisation, and all assistive performance results are averaged from 5 laps on the test course. These number were chosen to express the maximum variance in the results that could be afforded, given our available time budget for simulations and model training.

5.4.2 Generalisation

In this section, we further explore the issue of model generalisation to unseen environments. First, we contrast the procedures used for testing performance employed in (Soh and Demiris 2015a) and (Kucukyilmaz and Demiris 2018) against our approach, as described in Section 4.2. For this, we start by collecting data in a training scenario and then build three different validation sets, all with the same size. The first set is composed of a second lap in the same obstacle course and following the same trajectory, as was done in (Soh and Demiris 2015a). The second set is composed of a run on the same course where training data was recorded, but following a different trajectory, as was done in (Kucukyilmaz and Demiris 2018). The third validation set is formed from a run on a dedicated course, physically different from the one used for gathering training data. We then fit our LAD model to the training data and, as training progresses, check the predictive loss against the different validation sets. The result is shown in Figure 5.7.

As can be seen, the first two validation sets indicate good performance, but in reality the model is operating in an overfitting state. In a realistic setting, where the wheelchair has to be used in environments not seen during training, this assistive model would not perform well. This happens because, as fitting progresses, the model tends to learn more and more from the spatial features observed (through the laser-scanners) in the training environment. But during test/inference time, the spatial features are completely different, and the model becomes unable to offer correct assistive signals.

This issue, however, is not merely one of *measuring* performance. Using an appropriate validation set is crucial for proper model regularisation with early-stopping, especially when dealing with deep neural networks. Hence, this experiment shows that a simple adjustment in the dataset collection procedure is capable of significantly *improving* performance.

Next, we assess the impact that using different courses to compose the training data has on generalisation. Figure 5.8 shows the predictive performances

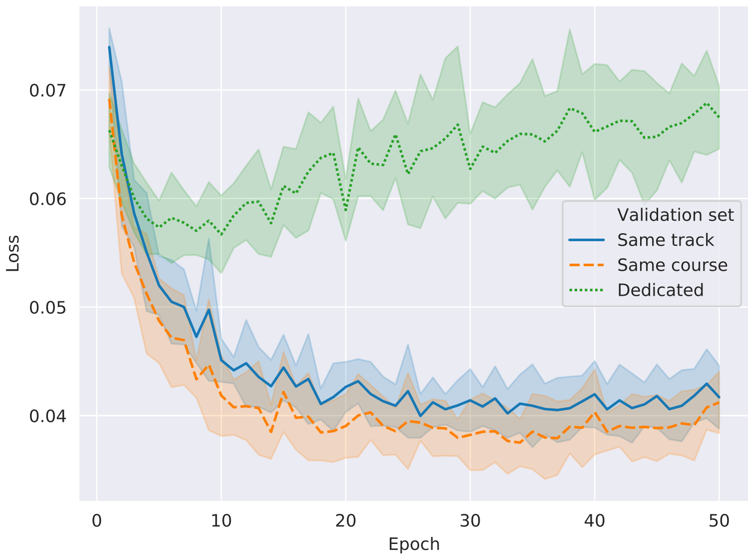


Figure 5.7: Impact that using a dedicated course as validation set has on model evaluation and training. Here, a single model is trained, but at the end of each epoch, performance is tested against three distinct validation sets. As can be seen, only the validation set recorded on a dedicated and unseen course (green line) can indicate that the model is operating in a overfitting state. Without this, previous work (blue and orange lines) could not properly know when to stop training. Hence, having a dedicated validation course allows one to properly use early stopping and improve generalisation performance.

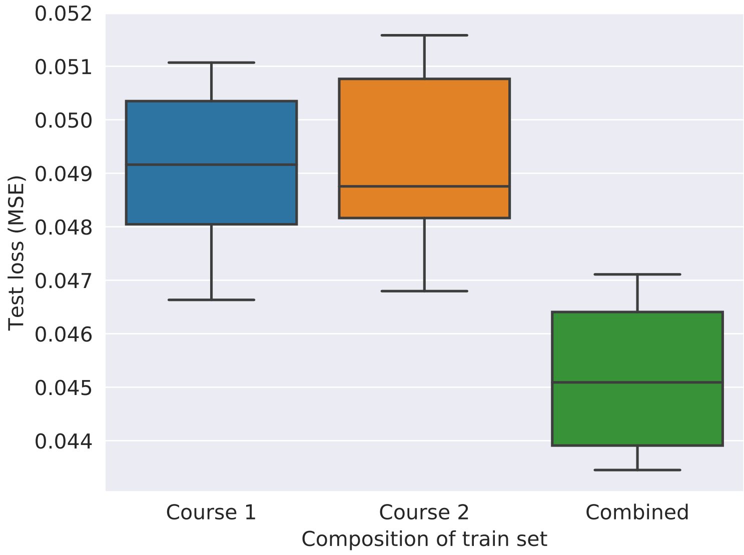


Figure 5.8: Impact that using multiple courses for collecting training data has on model performance. In all cases the same amount of data is available for training, but 'combined' uses demonstrations recorded in physically different scenarios, instead of repeated laps on the same course. As can be seen, this leads to improved performance when predicting assistance on an unseen environment.

of our assistive model when it is trained with different datasets. All datasets have the same size but, in the first two cases, they are formed from repeated laps on the same physical course. This approach, which was used by previous work in LAD for smart wheelchairs, again promotes an overfitting behaviour. Instead, we combine one lap from each course, which forces the model to learn more generic representations of the spatial features of the environment. As expected, this approach leads to better results during testing, which happens on a dedicated course. This shows another case of a simple change in the data collection procedure that leads to an improved generalisation performance.

Lastly, Figure 5.9 exhibits a plot showing the input, target and predicted velocity signals, recorded while the simulated driver navigated a dedicated test course, not seen during training. The assistive model employed here used the best procedures just described for data collection and preprocessing, architecture optimisation and fitting. Notice how the predicted angular velocity correctly tracks the trends of the target signal, while filtering out most of the noise and distortion from the input.

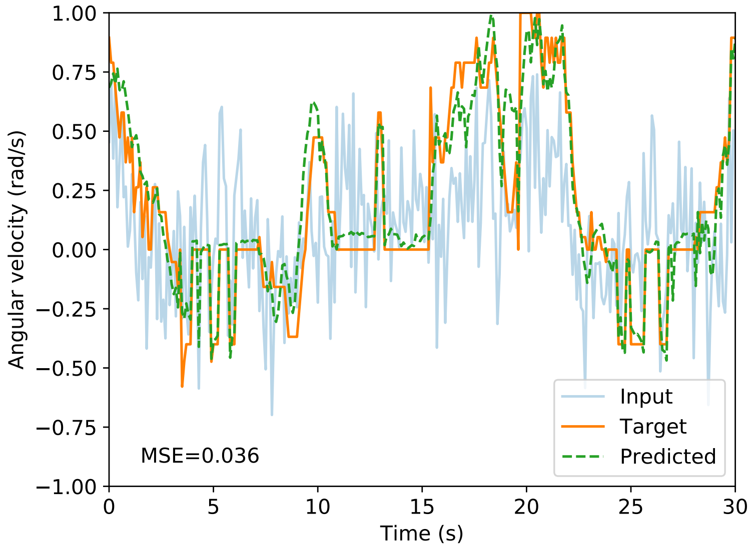


Figure 5.9: Example plot showing input, target and predicted velocity signals on a dedicated test course. Input is scaled to the ±1 range. Notice how the model prediction can correctly follow trends of target signal, while filtering-out noise and distortion on the driver's input.

5.4.3 Assistive performance

Predictive performance, as discussed in the last section, is useful for comparing different models and training procedures, especially when the goal is to enhance generalisation capabilities. However, from the driver's point of view, the most important aspect is to test how much aid in navigation the final model can offer. Hence, in this section, we discuss the benefits that can be provided by LAD considering different assistive aspects.

Table 5.1 reports driving performances for three different simulated drivers. 'Optimal', 'Disability' and 'LAD' respectively correspond, to a driver without any disability, a driver with a disability, and a driver with the same disability but aided by the LAD model. Driving performance is then reported using the metrics described in Section 5.4.1. As can be observed, the imposition of impairment has a significant negative impact on all measures. The disability makes driving more erroneous, which increase the average distance from the planned path. And it can also lead to under or overshooting turns, which increases the chance of collision against obstacles. In turn, this leads to a surge in the number of backwards movements needed and in the number of times the wheelchair gets stuck. All these contribute to an average increment of 82% in the time it takes for the driver to complete a lap around the test course.

However, by providing this driver with autonomous assistance learned from

Table 5.1: Assistive performance results. Different metrics (see Section 5.4.1) are considered and results are reported as *average (standard deviation)*. 'Optimal recovery' stands for how much of the gap between 'Optimal' and 'Disability' was recovered by using LAD. Model assistance has a significant positive impact across all metrics.

	Optimal	Disability	LAD	LAD improvement	Optimal recovery
Time to complete lap (s)	179.3 (4.8)	325.7 (23.8)	247.9 (11.6)	23.9%	53.2%
Avg. distance from path (cm)	8.7 (0.3)	23.0 (1.6)	16.2 (1.1)	29.4%	47.5%
Rel. time clearing collisions	2.0% (0.8%)	23.3% (1.4%)	3.6% (0.8%)	84.6%	92.4%
Instructor interventions	0.0 (0.0)	3.2 (1.2)	0.2 (0.4)	93.8%	93.8%

a LAD model, they are capable to improve performance in all aspects. In two of the metrics, number of human interventions needed and relative amount of time spent on escape sequences, the driver is capable of nearly recovering their optimal performance (last column of Table 5.1). While the recover of optimal performance is not as pronounced for the other two metrics, the results are still significant. To put the benefits in context, one can imagine being a wheelchair user, and then suddenly being able to navigate everywhere 24% faster, while having a far smaller chance of collisions.

Furthermore, we noticed that the gains in lap completion times are mostly hindered by difficulties in inferring the driver's intent. When inference is wrong, the driver might end up having to execute an in-place rotation, which delays their navigation. However, one should note that this problem is more severe in simulation than it would be for a human. That is because our platform allows the driver to quickly turn model assistance on and off. Hence, if the person notices that the model is providing incorrect assistance, they can briefly switch it off, gaining full control of navigation. In the simulation, we could not program this human perception capability and thus left model assistance always turned on.

5.4.4 Robustness

The biggest motivation for using LAD is that it promises a simple way of generating custom assistive models that can help with different disabilities. Hence, testing that this promise holds true is paramount; but it had not been done before. Previous work in this field (Soh and Demiris 2013; Soh and Demiris 2015a) only carried out experiments with a single disability type - difficulty in performing right turns. We also experimented with this disability in Chapter 4, and implemented a different kind of impairment in Chapter 3 (a tremor of varying intensity). However, the approach we proposed in Section 5.1,

Table 5.2: Assistive performance results for a different simulated disability. While results are inferior in this case, LAD can still significantly help the driver, without requiring any adaptations in model architecture or hyperparameter optimisation.

	Optimal	Disability	LAD	LAD improvement	Optimal recovery
Time to complete lap (s)	179.3 (4.8)	298.7 (26.9)	252.8 (9.8)	15.3%	38.4%
Avg. distance from path (cm)	8.7 (0.3)	20.8 (0.2)	15.0 (1.3)	27.8%	48.0%
Rel. time clearing collisions	2.0% (0.8%)	21.8% (5.2%)	5.4% (1.8%)	75.2%	82.8%
Instructor interventions	0.0 (0.0)	2.0 (0.6)	0.8 (0.4)	60.0%	60.0%

of procedurally creating different disabilities using Perlin noise, allows us to test if LAD is indeed robust and applicable to multiple kinds of impairments.

We start by repeating the experiment of Section 5.4.3 for a second driver, which is afflicted by a different disability (distortion map on the left in Figure 5.3). The results are shown in Table 5.2. As can be seen, the LAD model is again capable of significantly helping the driver across all metrics, not being necessary to fine-tune hyperparameters or to change the model architecture. We note, however, that a smaller part of the gap to optimal performance was recovered in this case. This shows another characteristic of LAD: being a data-centric approach, the same level of performance improvement cannot be expected across all disability types.

We carry on and repeat the experiment for another three drivers. The results, summarised in Figure 5.10, show that in all cases LAD can provide useful assistance, indicating the robustness of the approach.

5.4.5 Personalisation

In this section, we analyse if the assistive models learned can offer *personalised* assistance. The goal here is to understand if LAD is indeed capable of offering customised assistive solutions, targeted at individuals and their unique disabilities; or if it is merely offering generic help, and thus could be replaced by simpler solutions, like low-pass filtering of the input or obstacle avoidance.

For this, we took both simulated drivers from the last experiment and repeated their runs in the test course with model assistance enabled. Except that the wrong model was used - that is, driver 1 received assistance from the model learned for driver 2 and vice-versa. The results are shown in Table 5.3.

This simple experiment leads to interesting findings. As can be seen, using the wrong LAD model leads to severe degradation of performance. For most metrics, the deterioration is so extreme that it results in a negative improvement, that is, the driver would be better off without any assistance. This shows that

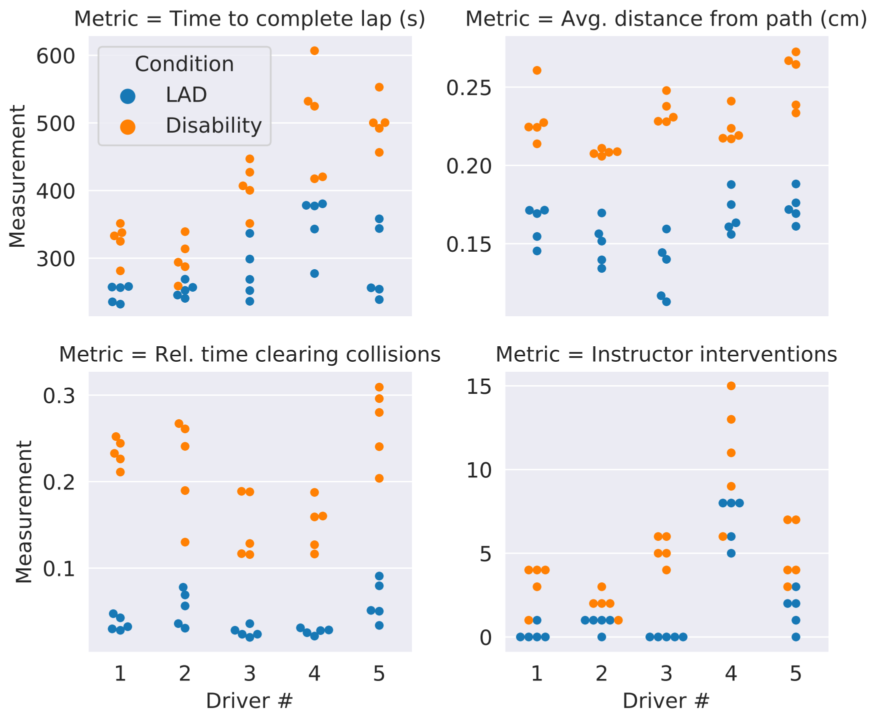


Figure 5.10: Shows performance metrics for five drivers, comparing with and without LAD assistance. Each driver is afflicted by a different disability and each dot represents a lap on the test course. As can be seen, enabling LAD consistently improves performance, across different metrics and different disabilities. This result shows that the LAD approach is robust and capable of generating useful assistive models for multiple disability types.

Table 5.3: Personalisation experiment. Shows the improvements achieved when using LAD (fourth column in Tables 5.1 and 5.2), considering different assistive metrics. Using the correct models for each driver lead to the results in the diagonals (bold), and swapping the models lead to the results off-diagonal. The significant drop in performance shows that LAD leads to the generation of *personalised* assistive models.

Time to complete lap (s)		
	Model 1	Model 2
Driver 1	**23.9%**	-35.5%
Driver 2	-31.2%	**15.3%**

Avg. distance from path (cm)		
	Model 1	Model 2
Driver 1	**29.4%**	-47.1%
Driver 2	-42.7%	**27.8%**

Rel. time clearing collisions		
	Model 1	Model 2
Driver 1	**84.6%**	58.9%
Driver 2	42.3%	**75.2%**

Instructor interventions		
	Model 1	Model 2
Driver 1	**93.8%**	-31.2%
Driver 2	-10.0%	**60.0%**

indeed the models being learnt are personalised; tailored to the needs of an individual driver.

On the other hand, from a subjective observation of the runs with the wrong models, we could notice that the model assistance is most impacted in terms of inferring the intention of the driver. For example, the autonomous assistance would frequently navigate into rooms that were not part of the planned path, which rarely occurred with the correct models in place. But, it is also possible to notice that the model still helped in avoiding collisions, especially when navigating narrow paths, like doorways. This is reflected in the results, where the only metric that did not exhibit a negative improvement was 'Relative time clearing collisions', which directly correlates to the number of collisions. This observation is not surprising when one considers the nature of the problem at hand. The disability only affects part of the model input, the velocity commands. The other part, the laser-scanner readings, is only dependent on the environment. While the velocity commands are most useful to aid in inferring the intent of the driver, the laser readings are valuable in *limiting* the set of plausible desired commands. In other words, while the velocity input indicates whether the driver wants to turn left or right, the scan input determines that they definitely do not want to continue straight, because they would crash into a wall. This experiment's result suggests that the assistive models for both drivers have independently learnt this obstacle avoidance behaviour. This indicates a potential for the employment of transfer learning (Pan and Yang 2010; Tan et al. 2018), which conceivably could speed up or improve training performance. We leave this line of research to be explored as future work.

5.4.6 Discussion

The experiments conducted uncovered many interesting findings. First, it was shown that measuring performance using dedicated validation and test courses is not only more realistic, but it also leads to an improvement in performance. When contrasted against the approach taken by previous work, it was shown that, without this, the model can report good results while actually operating in an overfitting state. It was also shown the importance of increasing the variability of the training data, and the impact that it has on improving generalisation performance. Additionally, we made a distinction between predictive and assistive performance and used multiple metrics to demonstrate the comprehensive usefulness of our model. Lastly, the experiments showed that our model can be employed to aid with different disability types, which is the main argument in favour of using LAD. This is possible because, as shown, LAD leads to the learning of personalised assistive solutions, which are uniquely targeted at the disability of an individual.

Limitations The main limitation of this work is related to the simulated hand-control disability. While we put a lot of effort into being able to simulate generic disability types, we cannot guarantee that these faithfully correspond to the difficulties experienced by real wheelchair users. However, we did show that our model works for multiple disability types (as long as they do not change over time). And since there is nothing special about the Perlin Noise maps used for simulating the disabilities, we believe that the proposed approach could work with *any* form of disability, although performance may vary.

5.5 Conclusions

This chapter had the goal of addressing the following research problem: "How to improve the generalisation capabilities of LAD models?". To answer the question in a meaningful way, it was important to increase the amount of data collected, thus reducing the effect of experimental noise. But collecting data from triadic interactions is very laborious and time-consuming, and potentially wasteful of resources if the right methodology is not in place. Thus, we turned to simulations. A custom simulator was built that is capable to reproduce the full triadic interaction between a disabled driver, a smart wheelchair and a remote assistant. The platform allows one to easily and procedurally generate any number of disabilities, while effortlessly varying distortion and noise levels. Furthermore, the system is built in a way that enables people to seamlessly take up the roles of either or both driver and assistant, without leaving the simulation platform.

The simulator allowed us to collect more data, from repeated runs and in a variety of environments. In turn, this enabled the exploration of various data preprocessing techniques, model architectures and training procedures, which led to an improved generalisation capability; crucial for providing assistance in a realistic setting.

Finally, many experiments were conducted to cement the comprehension of LAD and its characteristics. First, it was shown the importance of building appropriate training and validation sets, not only to correctly measure performance, but also to improve it. Then, multiple metrics were developed and used to understand how the LAD system can help *the driver*. This is important because simple fitting metrics, such as MSE loss, do not directly translate to assistive performance. Moreover, it was shown that the proposed LAD model is capable of generalising to different environments and is applicable to multiple impairments. Lastly, it was demonstrated that the learnt models are indeed capable of offering *personalised* assistive solutions, tailored to match the specific needs of different disabilities.

Chapter 6

Conclusions

At the beginning of this thesis we set out to address the following research problem: "Use demonstrations of the triadic interaction between a robotic wheelchair, a driver with hand-control impairments and a remote assistant to generate personalised assistive policies that improve navigation performance". In the process, this work managed to advance the edge of scientific research in a few fronts.

First, we developed a custom teleoperation platform for smart wheelchairs, with the goal of enabling a remote assistant to more easily and accurately provide help to a wheelchair driver. It was discussed how, in a machine learning setting, this situation presents a conflict regarding what information should be presented to the assistant. On one hand, the assistant should have as much information as possible, to increase the quality of the help that is being provided. On the other hand, it is important to avoid an information discrepancy problem, where the assistant makes decisions based on information that is not available for the robot to learn from (e.g., the assistant can see obstacles in the path of the driver that are not observable by the robot). If, by attempting to solve the information problem, both assistant and robot are fed with data of high dimensionality (e.g. using multiple cameras in front of the robot), we then incur into the 'curse of dimensionality' (Bishop 2006) problem, where more data is required to achieve satisfactory learning performance. To solve both issues simultaneously, our teleoperation platform only presents to the assistant data that is also available for the robot to learn from, and then use multimodal interfaces to make the interpretation of this data more intuitive. Using this approach, our platform assures that the quality of the demonstrations of assistance can remain high, while avoiding the information discrepancy and curse of dimensionality problems. We also conducted a user study and found that the assistant's capacity to infer the intention of the driver, which is directly correlated to the quality of the demonstrations provided, is significantly impacted by the multimodal interfaces used.

Following, we discussed how Learning Assistance by Demonstration (LAD) can be used to generate assistive policies by observing the triadic interaction between smart wheelchair, driver and remote assistant. Even though the technique had already been investigated in previous work, we found that several design changes in our approach led to improved performance. In particular, we described a new paradigm for 'when to help', which empowers the driver to only receive assistance when they request it. Furthermore, we performed the first experiment with LAD in a more realistic setting, where the model assistance is tested in an environment that was not seen during training. It was shown that in this setting LAD is still capable of generating helpful driving assistance. To improve generalisation performance, we explored numerous techniques in data preprocessing, model training and machine learning algorithms, ranging from simple linear models to highly optimised recurrent and convolutional neural networks.

Lastly, we resorted to simulations to more meaningfully assess the capabilities of LAD. A custom simulator was developed and will be made open-source. This simulator can reproduce the full triadic interaction considered in our studies. It can also procedurally generate different types and levels of hand-control impairments and simulate different environments, with varying levels of difficulty of navigation. Furthermore, it also allows people to take up the roles of simulated driver and/or assistant by using simple video-game joysticks. Using this simulator we were capable of collecting more statistically powerful results and assess the impact that different data collection procedures have on model training. Additionally, the simulations allowed us to validate claimed features of LAD that had not been demonstrated before: robustness and adaptation to different disabilities, and providing personalised assistive solutions.

Putting it all together, we have shown that LAD is indeed a viable way to exploit "demonstrations of assistance [...] to improve the navigation performance of powered wheelchair users with hand-control disabilities". However, in order to achieve satisfactory performance, several modifications and improvements had to be implemented against previous work in this field. These changes and improvements were thoroughly detailed in this thesis.

6.1 Limitations

Throughout this work we strove to show and improve the applicability of LAD to assisted navigation for robotic wheelchairs. Nevertheless, there are limitations to our work. Most significant is that only simulated hand-control disabilities were explored. The reason for this is that, upon commencing this research, we did not feel confident to expose the target population to a still experimental system. Thus, using simulated impairments was a natural step in our research plan - but it also limits the extent to which conclusions can (currently) be drawn about the assistive benefits of our proposed system.

From the machine learning side, this work could be improved by collecting more driving data and using a more diverse set of drivers. From a personalisation point of view, perhaps more training data would not make much difference, given that a person's driving profile tends to remain constant, even through changing environments. Furthermore, asking someone with driving difficulties to provide dozens of hours of demonstration data is not realistic. But from a perspective of developing learning algorithms that can better generalise to different disabilities, collecting more training data could be beneficial. Unfortunately, we were limited in this sense by practical matters, such as the amount time, manpower and resources needed for conducting studies in this field.

Lastly, related to the previous point, in Chapter 5 we resorted to using simulated drivers to collect artificial data, which was then used to design better learning algorithms, seeking improved assistive performance. However, this new machine learning architecture has yet to be tested with real drivers, to assess if the assistive improvements hold in that case.

6.2 Further work

Moving forward, we envision four main lines of work to continue this research. First and foremost, we believe that the combination of LAD for smart wheelchairs is ready for initial tests with target users. We anticipate that there will always exist differences between the input of a real user with hand-control impairment and any disability that can be simulated. But these differences can only be reduced by having access to recordings of a real control signal. Furthermore, the data recorded from this interaction can be used to make further improvements to our model architecture and training procedure. But most importantly, the subjective feedback that can be collected from these users is invaluable in understanding what is working or not, and in informing future design decisions.

Second, our work with multimodal interfaces for teleoperation platforms can be extended to other domains where learning is not necessary, but robot control has to be intuitive and timely. Examples include robots that have to operate in inaccessible or hazardous environments and a conduct a one-time execution of a task. Exploring this direction, in (Girbes-Juan et al. 2020) we showed how the integration of haptic interfaces with assisted low-level control can improve performance in tasks that require fine-grained movements. Continuing this line of research, we envision the combination of different types of interfaces for dealing with different dimensions of a task. For instance, a motion capture system can be used to quickly and naturally teleoperate a robotic arm towards reaching a target object, and then a haptic interface is used for carefully grasping and manipulating the object.

Third, we are curious about the application of LAD to other domains of assistive robotics. Notably, a neighbour but quite different application is a wheelchair-mounted robotic arm (Losey et al. 2020). This extension can be invaluable to wheelchair users, particularly those suffering from severe motor impairments affecting the upper-body. We would like to explore if LAD can be used to offer the user a more intuitive control of the robotic arm, since the joystick input cannot be directly mapped to the six degrees of freedom of the arm's end effector. In this case, an expert would provide assistive demonstrations by performing kinaesthetic teaching, showing how to help the user reach objects of interest. The demonstrations would then be used by a learning algorithm to create a new mapping between a low-dimensional input and a higher-dimensional output, which should also be contextually adaptive.

Lastly, back to the problem of navigation support for smart wheelchairs, it would be interesting to address the case of motor impairments that continuously evolve. If the rate of change in the user's control characteristic is too high, LAD might not be the most appropriate solution. That is because, as shown in Chapter 5, LAD leads to the learning of customised assistive models. When the disability changes, the model performance degrades. Perhaps this case would be better dealt with by employing machine learning techniques that allow for continuous learning, such as human-in-the-loop reinforcement learning (Tjomsland, Shafti, and Faisal 2019) or self-supervised learning (Nava et al. 2019).

6.3 Final remarks

Developing robots that can provide physical assistance to people is an important and exciting line of research. A significant challenge of the field, however, is ensuring that these robots are flexible enough to work in diverse environments and to meet the needs of different people. To this end, in this thesis we proposed to use LAD, a technique that enables non-roboticists to intuitively generate customised assistive policies, which can then be employed by robots to autonomously help people. While there are still challenges ahead in order to deploy LAD outside of laboratories, this thesis demonstrated that the approach is flexible in terms of personalisation and generalisation, two of the key components towards making assistive robots more widely available for end-users.

References

Abadi, Martín et al. (Mar. 2016). "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems". In: *arXiv preprint* (cit. on p. 107).

Abbeel, Pieter and Andrew Y. Ng (2004). "Apprenticeship learning via inverse reinforcement learning". In: *Twenty-first international conference on Machine learning - ICML '04*. New York, New York, USA: ACM Press, p. 8. DOI: 10.1145/1015330.1015430 (cit. on p. 36).

Abdi, Hervé and Lynne J. Williams (July 2010). "Principal component analysis". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2.4, pp. 433–459. DOI: 10.1002/wics.101 (cit. on p. 84).

Abi-Farraj, Firas 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 (cit. on p. 31).

Admoni, H. and S. Srinivasa (Sept. 2016). "Predicting user intent through eye gaze for shared autonomy". In: *AAAI Fall Symposium Series: Shared Autonomy in Research and Practice*, pp. 298–303. DOI: 10.1177/0278364913478447 (cit. on pp. 35, 56).

Alshaer, Abdulaziz, Holger Regenbrecht, and David O'Hare (Jan. 2017). "Immersion factors affecting perception and behaviour in a virtual reality power wheelchair simulator". In: *Applied Ergonomics* 58, pp. 1–12. DOI: 10.1016/J.APERGO.2016.05.003 (cit. on pp. 54, 55).

American Association of Colleges of Nursing (2020). *Nursing shortage* (cit. on p. 15).

Argall, Brenna D. (May 2018). "Autonomy in Rehabilitation Robotics: An Intersection". In: *Annual Review of Control, Robotics, and Autonomous Systems* 1.1, pp. 441–463. DOI: 10.1146/annurev-control-061417-041727 (cit. on p. 31).

Argall, Brenna D. et al. (May 2009). "A survey of robot learning from demonstration". In: *Robotics and Autonomous Systems* 57.5, pp. 469–483. DOI: 10.1016/j.robot.2008.10.024 (cit. on pp. 18, 23, 36, 37, 73, 80, 91).

Baltrusaitis, Tadas, Chaitanya Ahuja, and Louis-Philippe Morency (Feb. 2019). "Multimodal Machine Learning: A Survey and Taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2, pp. 423–443. DOI: 10.1109/TPAMI.2018.2798607 (cit. on p. 102).

Bastos-Filho, Teodiano Freire et al. (2014). "Towards a new modality-independent interface for a robotic wheelchair". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22.3. DOI: 10.1109/TNSRE.2013.2265237 (cit. on p. 27).

Bellamy, N et al. (Apr. 1991). "Circadian rhythm in pain, stiffness, and manual dexterity in rheumatoid arthritis: relation between discomfort and disability." In: *Annals of the rheumatic diseases* 50.4, pp. 243–8. DOI: 10.1136/ard.50.4.243 (cit. on p. 89).

Bengio, Yoshua et al. (2009). "Curriculum learning". In: *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*. New York, New York, USA: ACM Press, pp. 1–8. DOI: 10.1145/1553374.1553380 (cit. on p. 104).

Bishop, Christopher M (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, p. 738 (cit. on pp. 18, 36, 85, 121).

Bojarski, Mariusz et al. (Apr. 2016). "End to End Learning for Self-Driving Cars". In: *CoRR* (cit. on p. 36).

Bojarski, Mariusz et al. (Apr. 2017). "Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car". In: *CoRR* abs/1704.0 (cit. on p. 36).

Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik (1992). "Training algorithm for optimal margin classifiers". In: *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*. New York, New York, USA: Publ by ACM, pp. 144–152. DOI: 10.1145/130385.130401 (cit. on p. 103).

Breiman, Leo (2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32. DOI: 10.1023/A:1010933404324 (cit. on p. 85).

Brienza, David M. and Jennifer Angelo (Jan. 1996). "A force feedback joystick and control algorithm for wheelchair obstacle avoidance". In: *Disability and Rehabilitation* 18.3, pp. 123–129. DOI: 10.3109/09638289609166029 (cit. on p. 32).

Brose, Steven W. et al. (June 2010). "The Role of Assistive Robotics in the Lives of Persons with Disability". In: *American Journal of Physical Medicine & Rehabilitation* 89.6, pp. 509–521. DOI: 10.1097/PHM.0b013e3181cf569b (cit. on p. 16).

Bruguerolle, Bernard and Nicolas Simon (July 2002). “Biologic Rhythms and Parkinson’s Disease: A Chronopharmacologic Approach to Considering Fluctuations in Function”. In: *Clinical Neuropharmacology* 25.4, pp. 194–201. DOI: 10.1097/00002826-200207000-00002 (cit. on p. 89).

Burhanpurkar, Maya et al. (July 2017). “Cheap or Robust? The practical realization of self-driving wheelchair technology”. In: *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE, pp. 1079–1086. DOI: 10.1109/ICORR.2017.8009393 (cit. on p. 34).

Card, Stuart K., William K. English, and Betty J. Burr (1978). “Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a crt”. In: *Ergonomics* 21.8, pp. 601–613. DOI: 10.1080/00140137808931762 (cit. on p. 64).

Carlson, T. and Y. Demiris (June 2012). “Collaborative Control for a Robotic Wheelchair: Evaluation of Performance, Attention, and Workload”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.3, pp. 876–888. DOI: 10.1109/TSMCB.2011.2181833 (cit. on p. 34).

Carlson, Tom and Yiannis Demiris (May 2008). “Human-wheelchair collaboration through prediction of intention and adaptive assistance”. In: *IEEE International Conference on Robotics and Automation*. IEEE, pp. 3926–3931. DOI: 10.1109/ROBOT.2008.4543814 (cit. on pp. 35, 49).

Chang, Yizhe et al. (Oct. 2017). “A Shared Autonomy Approach for Wheelchair Navigation Based on Learned User Preferences”. In: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. IEEE, pp. 1490–1499. DOI: 10.1109/ICCVW.2017.176 (cit. on p. 33).

Chen, Yuying et al. (2019). “Gaze Training by Modulated Dropout Improves Imitation Learning”. In: *IEEE International Conference on Intelligent Robots and Systems*. DOI: 10.1109/IROS40897.2019.8967843 (cit. on p. 104).

Chow, H.N. and Yangsheng Xu (June 2006). “Learning Human Navigational Skill for Smart Wheelchair in a Static Cluttered Route”. In: *IEEE Transactions on Industrial Electronics* 53.4, pp. 1350–1361. DOI: 10.1109/TIE.2006.878296 (cit. on pp. 38, 44).

Chung, Junyoung et al. (Dec. 2014). “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”. In: *NIPS 2014 Deep Learning and Representation Learning Workshop* (cit. on pp. 100, 107).

Compston, Alastair and Alasdair Coles (Oct. 2008). “Multiple sclerosis”. In: *The Lancet* 372.9648, pp. 1502–1517. DOI: 10.1016/S0140-6736(08)61620-7 (cit. on p. 18).

Coşar, Serhan et al. (July 2020). “ENRICHME: Perception and Interaction of an Assistive Robot for the Elderly at Home”. In: *International Journal* *of Social Robotics* 12.3, pp. 779–805. DOI: 10.1007/s12369-019-00614-y (cit. on p. 16).

Davies, A., L. H. De Souza, and A. O. Frank (2003). “Changes in the quality of life in severely disabled people following provision of powered indoor/outdoor chairs”. In: *Disability and Rehabilitation* 25.6, pp. 286–290. DOI: 10.1080/0963828021000043734 (cit. on p. 17).

Demiris, Yiannis, Lisa Aziz-Zadeh, and James Bonaiuto (2014). “Information processing in the mirror neuron system in primates and machines”. In: *Neuroinformatics*. DOI: 10.1007/s12021-013-9200-7 (cit. on p. 69).

Demiris, Yiannis and Bassam Khadhouri (May 2006). “Hierarchical attentive multiple models for execution and recognition of actions”. In: *Robotics and Autonomous Systems* 54.5, pp. 361–369. DOI: 10.1016/J.ROBOT.2006.02.003 (cit. on p. 35).

Desai, Sumit, S. S. Mantha, and V. M. Phalle (Jan. 2017). “Advances in smart wheelchair technology”. In: *2017 International Conference on Nascent Technologies in Engineering (ICNTE)*. IEEE, pp. 1–7. DOI: 10.1109/ICNTE.2017.7947914 (cit. on p. 34).

Devigne, Louise et al. (July 2017). “Design of an immersive simulator for assisted power wheelchair driving”. In: *IEEE International Conference on Rehabilitation Robotics*. IEEE, pp. 995–1000. DOI: 10.1109/ICORR.2017.8009379 (cit. on pp. 41, 44).

Di Gironimo, G. et al. (July 2013). “A virtual reality approach for usability assessment: case study on a wheelchair-mounted robot manipulator”. In: *Engineering with Computers* 29.3, pp. 359–373. DOI: 10.1007/s00366-012-0274-x (cit. on pp. 42, 44).

Dragan, Anca D. and Siddhartha S. Srinivasa (2013a). “Formalizing Assistive Teleoperation”. In: *Robotics*. The MIT Press, pp. 73–80. DOI: 10.7551/mitpress/9816.003.0015 (cit. on p. 31).

— (June 2013b). “A policy-blending formalism for shared control”. In: *The International Journal of Robotics Research* 32.7, pp. 790–805. DOI: 10.1177/0278364913490324 (cit. on pp. 31, 34).

Du, Yuqing et al. (June 2020). “AvE: Assistance via Empowerment”. In: *Advances in Neural Information Processing Systems 33 (2020)* (cit. on pp. 35, 44).

Durrant-Whyte, H. and T. Bailey (June 2006). “Simultaneous localization and mapping: part I”. In: *IEEE Robotics & Automation Magazine* 13.2, pp. 99–110. DOI: 10.1109/MRA.2006.1638022 (cit. on p. 51).

Engel, Paulo Martins and Milton Roberto Heinen (Oct. 2010). “Incremental Learning of Multivariate Gaussian Mixture Models”. In: *Brazilian Sym-*

posium on Artificial Intelligence (SBIA 2010). Vol. 6404 LNAI. Springer, Berlin, Heidelberg, pp. 82–91. DOI: 10.1007/978-3-642-16138-4{_}9 (cit. on p. 35).

Erdogan, Ahmetcan and Brenna D. Argall (Aug. 2017). "The effect of robotic wheelchair control paradigm and interface on user performance, effort and preference: An experimental assessment". In: *Robotics and Autonomous Systems* 94, pp. 282–297. DOI: 10.1016/j.robot.2017.04.013 (cit. on pp. 27–29, 33, 43).

Erickson, Zackory, Yijun Gu, and Charles C Kemp (2020). "Assistive VR Gym: Interactions with Real People to Improve Virtual Assistive Robots". In: *arXiv* (cit. on pp. 42, 44).

Erickson, Zackory et al. (May 2020). "Assistive Gym: A Physics Simulation Framework for Assistive Robotics". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 10169–10176. DOI: 10.1109/ICRA40945.2020.9197411 (cit. on pp. 42, 44).

European Commission (2012). *Commission Staff Working Document on an Action Plan for the EU Health Workforce* (cit. on p. 15).

Evans, Subhadra et al. (Jan. 2007). "Young people's experiences using electric powered indoor – outdoor wheelchairs (EPIOCs): Potential for enhancing users' development?" In: *Disability and Rehabilitation* 29.16, pp. 1281–1294. DOI: 10.1080/09638280600964406 (cit. on p. 17).

Ewerton, Marco et al. (May 2018). "Assisting Movement Training and Execution With Visual and Haptic Feedback". In: *Frontiers in Neurorobotics* 12. DOI: 10.3389/fnbot.2018.00024 (cit. on pp. 37, 44).

Fehr, L, W E Langbein, and S B Skaar (2000). "Adequacy of power wheelchair control interfaces for persons with severe disabilities: a clinical survey." In: *Journal of rehabilitation research and development* 37.3, pp. 353–60 (cit. on pp. 17, 33).

Feil-Seifer, David and M.J. Mataric (2005). "Socially Assistive Robotics". In: *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.* IEEE, pp. 465–468. DOI: 10.1109/ICORR.2005.1501143 (cit. on p. 16).

Fischer, Tobias, Hyung Jin Chang, and Yiannis Demiris (2018). "RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments". In: *European Conference on Computer Vision*, pp. 339–357. DOI: 10.1007/978-3-030-01249-6{_}21 (cit. on p. 56).

Fitts, Paul M. (June 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 (cit. on p. 64).

Fox, D., W. Burgard, and S. Thrun (Mar. 1997). "The dynamic window approach to collision avoidance". In: *IEEE Robotics & Automation Magazine* 4.1, pp. 23–33. DOI: 10.1109/100.580977 (cit. on pp. 59, 96).

Fox, Dieter (2001). "KLD-sampling: Adaptive particle filters". In: *Proceedings of the 14th International Conference on Neural Information Processing Systems*, pp. 713–720 (cit. on p. 52).

Frank, A O et al. (Dec. 2000). "Introduction of a new NHS electric-powered indoor/outdoor chair (EPIOC) service: benefits, risks and implications for prescribers". In: *Clinical Rehabilitation* 14.6, pp. 665–673. DOI: 10.1191/0269215500cr376oa (cit. on p. 17).

Girbes-Juan, Vicent et al. (2020). "Haptic and Visual Feedback Assistance for Dual-Arm Robot Teleoperation in Surface Conditioning Tasks". In: *IEEE Transactions on Haptics*. DOI: 10.1109/TOH.2020.3004388 (cit. on p. 124).

Goil, Aditya, Matthew Derry, and Brenna D. Argall (June 2013). "Using machine learning to blend human and robot controls for assisted wheelchair navigation". In: *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*. IEEE, pp. 1–6. DOI: 10.1109/ICORR.2013.6650454 (cit. on pp. 31, 38, 44).

Gopinath, Deepak, Siddarth Jain, and Brenna D. Argall (Jan. 2017). "Human-in-the-Loop Optimization of Shared Autonomy in Assistive Robotics". In: *IEEE Robotics and Automation Letters* 2.1, pp. 247–254. DOI: 10.1109/LRA.2016.2593928 (cit. on pp. 34, 43, 78).

Hadj-Abdelkader, Amine, Brahim Cherki, and Guy Bourhis (June 2015). "Powered wheelchair driving using a 3D haptic device". In: *2015 International Conference on Virtual Rehabilitation (ICVR)*. IEEE, pp. 106–114. DOI: 10.1109/ICVR.2015.7358580 (cit. on p. 33).

Hadj-Abdelkader, Mohammed-Amine, Guy Bourhis, and Brahim Cherki (2012). "Haptic feedback control of a smart wheelchair". In: *Applied Bionics and Biomechanics* 9.2, pp. 181–192. DOI: 10.3233/ABB-2012-0067 (cit. on pp. 32, 33).

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY: Springer New York. DOI: 10.1007/978-0-387-84858-7 (cit. on p. 86).

He, Kaiming et al. (June 2016). "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2016-Decem. IEEE, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on p. 101).

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735 (cit. on p. 100).

Ioffe, Sergey and Christian Szegedy (Feb. 2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *32nd International Conference on Machine Learning, ICML 2015*. Vol. 1. International Machine Learning Society (IMLS), pp. 448–456 (cit. on p. 101).

Jafari, Nooshin, Kim D Adams, and Mahdi Tavakoli (June 2016). "Haptics to improve task performance in people with disabilities: A review of previous studies and a guide to future research with children with disabilities". In: *Journal of Rehabilitation and Assistive Technologies Engineering* 3. DOI: 10.1177/2055668316668147 (cit. on p. 32).

Javdani, Shervin, Siddhartha Srinivasa, and Andrew Bagnell (July 2015). "Shared Autonomy via Hindsight Optimization". In: *Robotics: Science and Systems XI*. Vol. 11. Robotics: Science and Systems Foundation. DOI: 10.15607/RSS.2015.XI.032 (cit. on p. 34).

Javdani, Shervin 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 (cit. on pp. 34, 43).

John, Nigel W. et al. (May 2018). "The Implementation and Validation of a Virtual Environment for Training Powered Wheelchair Manoeuvres". In: *IEEE Transactions on Visualization and Computer Graphics* 24.5, pp. 1867–1878. DOI: 10.1109/TVCG.2017.2700273 (cit. on pp. 42, 56).

K. Narayanan, Vishnu et al. (Aug. 2016). "Vision-based adaptive assistance and haptic guidance for safe wheelchair corridor following". In: *Computer Vision and Image Understanding* 149, pp. 171–185. DOI: 10.1016/j.cviu.2016.02.008 (cit. on p. 32).

Kaelbling, Leslie Pack, Michael L. Littman, and Anthony R. Cassandra (May 1998). "Planning and acting in partially observable stochastic domains". In: *Artificial Intelligence* 101.1-2, pp. 99–134. DOI: 10.1016/s0004-3702(98)00023-x (cit. on p. 35).

Kairy, Dahlia et al. (Feb. 2014). "Exploring Powered Wheelchair Users and Their Caregivers' Perspectives on Potential Intelligent Power Wheelchair Use: A Qualitative Study". In: *International Journal of Environmental Research and Public Health* 11.12, pp. 2244–2261. DOI: 10.3390/ijerph110202244 (cit. on pp. 28, 33, 43, 78, 91).

Kim, Dae Jin et al. (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*. DOI: 10.1109/TSMCA.2011.2159589 (cit. on pp. 28, 34, 43, 77, 78).

Kingma, Diederik P. and Jimmy Lei Ba (Dec. 2015). “Adam: A method for stochastic optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR (cit. on p. 107).

Kingma, Diederik P. and Max Welling (Dec. 2013). “Auto-Encoding Variational Bayes”. In: *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings* (cit. on p. 37).

Kober, Jens, J. Andrew Bagnell, and Jan Peters (Sept. 2013). “Reinforcement learning in robotics: A survey”. In: *International Journal of Robotics Research* 32.11, pp. 1238–1274. DOI: 10.1177/0278364913495721 (cit. on p. 35).

Koert, Dorothea et al. (2019). “Learning Intention Aware Online Adaptation of Movement Primitives”. In: *IEEE Robotics and Automation Letters*. Vol. 4. 4. IEEE, pp. 3719–3726. DOI: 10.1109/lra.2019.2928760 (cit. on pp. 35, 38, 44).

Kohlbrecher, Stefan et al. (Nov. 2011). “A flexible and scalable SLAM system with full 3D motion estimation”. In: *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. Kyoto: IEEE, pp. 155–160. DOI: 10.1109/SSRR.2011.6106777 (cit. on p. 52).

Kuchenbecker, Katherine J. (2018). “Haptics and Haptic Interfaces”. In: *Encyclopedia of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–9. DOI: 10.1007/978-3-642-41610-1{_}19-1 (cit. on pp. 31, 43).

Kucukyilmaz, Ayse and Yiannis Demiris (Aug. 2015). “One-shot assistance estimation from expert demonstrations for a shared control wheelchair system”. In: *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, pp. 438–443. DOI: 10.1109/ROMAN.2015.7333600 (cit. on pp. 23, 40, 41, 44, 85).

— (July 2018). “Learning Shared Control by Demonstration for Personalized Wheelchair Assistance”. In: *IEEE Transactions on Haptics* 11.3, pp. 431–442. DOI: 10.1109/TOH.2018.2804911 (cit. on pp. 23, 41, 44, 76–79, 85, 87, 100, 110).

Lambert, John, Ozan Sener, and Silvio Savarese (May 2018). “Deep Learning under Privileged Information Using Heteroscedastic Dropout”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8886–8895 (cit. on p. 104).

Lamers, Sanne M A et al. (Oct. 2012). "The impact of emotional well-being on long-term recovery and survival in physical illness: a meta-analysis." In: *Journal of behavioral medicine* 35.5, pp. 538–47. DOI: 10.1007/s10865-011-9379-8 (cit. on p. 89).

Lapin, Maksim, Matthias Hein, and Bernt Schiele (June 2013). "Learning Using Privileged Information: SVM+ and Weighted SVM". In: *Neural Networks* 53, pp. 95–108. DOI: 10.1016/j.neunet.2014.02.002 (cit. on p. 103).

Leaman, Jesse and Hung Manh La (Aug. 2017). "A Comprehensive Review of Smart Wheelchairs: Past, Present, and Future". In: *IEEE Transactions on Human-Machine Systems* 47.4, pp. 486–489. DOI: 10.1109/THMS.2017.2706727 (cit. on pp. 16, 28, 33, 34, 43).

Leaman, Jesse, Hung Manh La, and Luan Nguyen (Sept. 2016). "Development of a smart wheelchair for people with disabilities". In: *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, pp. 279–284. DOI: 10.1109/MFI.2016.7849501 (cit. on p. 33).

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton (May 2015). "Deep learning". In: *Nature* 521.7553, pp. 436–444. DOI: 10.1038/nature14539 (cit. on pp. 36, 99).

Leishman, Frederic et al. (Feb. 2014). "Driving Assistance by Deictic Control for a Smart Wheelchair: The Assessment Issue". In: *IEEE Transactions on Human-Machine Systems* 44.1, pp. 66–77. DOI: 10.1109/TSMC.2013.2287792 (cit. on pp. 31, 34, 43).

Li, Lisha et al. (June 2018). "Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization". In: *Journal of Machine Learning Research* 18, pp. 1–52 (cit. on p. 105).

Lock, Jacobus, Grzegorz Cielniak, and Nicola Bellotto (2017). "A portable navigation system with an adaptive multimodal interface for the blind". In: *AAAI 2017 Spring Symposium - Designing the User Experience of Machine Learning Systems* (cit. on p. 16).

Losey, Dylan P. et al. (Jan. 2018). "A Review of Intent Detection, Arbitration, and Communication Aspects of Shared Control for Physical Human–Robot Interaction". In: *Applied Mechanics Reviews* 70.1. DOI: 10.1115/1.4039145 (cit. on pp. 36, 43).

Losey, Dylan P. et al. (May 2020). "Controlling Assistive Robots with Learned Latent Actions". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 378–384. DOI: 10.1109/ICRA40945.2020.9197197 (cit. on pp. 37, 44, 124).

Madarasz, R. et al. (1986). "The design of an autonomous vehicle for the disabled". In: *IEEE Journal on Robotics and Automation* 2.3, pp. 117–126. DOI: 10.1109/JRA.1986.1087052 (cit. on p. 28).

Marder-Eppstein, Eitan et al. (May 2010). "The Office Marathon: Robust navigation in an indoor office environment". In: *2010 IEEE International Conference on Robotics and Automation*. Anchorage: IEEE, pp. 300–307. DOI: 10.1109/ROBOT.2010.5509725 (cit. on p. 59).

Matsubara, Takamitsu et al. (Aug. 2015). "Sequential intention estimation of a mobility aid user for intelligent navigational assistance". In: *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. Kobe: IEEE, pp. 444–449. DOI: 10.1109/ROMAN.2015.7333580 (cit. on p. 38).

Mitzner, Tracy L. et al. (2014). "Identifying the Potential for Robotics to Assist Older Adults in Different Living Environments". In: *International Journal of Social Robotics* 6.2, pp. 213–227. DOI: 10.1007/s12369-013-0218-7 (cit. on p. 16).

Montesano, Luis et al. (Apr. 2010). "Towards an Intelligent Wheelchair System for Users With Cerebral Palsy". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 18.2, pp. 193–202. DOI: 10.1109/TNSRE.2009.2039592 (cit. on p. 34).

Morère, Y. et al. (Oct. 2015). "Haptic control for powered wheelchair driving assistance". In: *IRBM* 36.5, pp. 293–304. DOI: 10.1016/j.irbm.2015.09.003 (cit. on pp. 32, 41, 44).

Mortenson, W. Ben, Laura Hurd Clarke, and Krista Best (2013). "Prescribers' experiences with powered mobility prescription among older adults". In: *American Journal of Occupational Therapy* 67.1, pp. 100–107. DOI: 10.5014/ajot.2013.006122 (cit. on p. 17).

Mostafa, Salama A., Mohd Sharifuddin Ahmad, and Aida Mustapha (2017). *Adjustable autonomy: a systematic literature review*. DOI: 10.1007/s10462-017-9560-8 (cit. on p. 31).

Najafi, Mohammad, Kim Adams, and Mahdi Tavakoli (July 2017). "Robotic learning from demonstration of therapist's time-varying assistance to a patient in trajectory-following tasks". In: *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE, pp. 888–894. DOI: 10.1109/ICORR.2017.8009361 (cit. on pp. 37, 44).

Narayanan, Vishnu K., Anne Spalanzani, and Marie Babel (Oct. 2016). "A semi-autonomous framework for human-aware and user intention driven wheelchair mobility assistance". In: *2016 IEEE/RSJ International Confer-*

ence on Intelligent Robots and Systems (IROS). IEEE, pp. 4700–4707. DOI: 10.1109/IROS.2016.7759691 (cit. on pp. 31, 35).

National Health Service England (2020). *Improving Wheelchair Services* (cit. on p. 17).

Nava, Mirko et al. (Apr. 2019). "Learning Long-Range Perception Using Self-Supervision from Short-Range Sensors and Odometry". In: *IEEE Robotics and Automation Letters* 4.2, pp. 1279–1286. DOI: 10.1109/LRA.2019.2894849 (cit. on p. 124).

Ngiam, Jiquan et al. (2011). "Multimodal deep learning". In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011* (cit. on pp. 23, 102, 104).

OECD (Nov. 2017). *Health at a Glance 2017*. Health at a Glance. OECD Publishing. DOI: 10.1787/health{_}glance-2017-en (cit. on p. 15).

Padir, Taskin (Aug. 2015). "Towards personalized smart wheelchairs: Lessons learned from discovery interviews". In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Vol. 2015-Novem. IEEE, pp. 5016–5019. DOI: 10.1109/EMBC.2015.7319518 (cit. on pp. 28, 31, 33, 43).

Pan, Sinno Jialin and Qiang Yang (2010). "A survey on transfer learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10, pp. 1345–1359. DOI: 10.1109/TKDE.2009.191 (cit. on p. 117).

Paraschos, Alexandros et al. (Mar. 2018). "Using probabilistic movement primitives in robotics". In: *Autonomous Robots* 42.3, pp. 529–551. DOI: 10.1007/s10514-017-9648-7 (cit. on p. 38).

Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio (2013). "On the difficulty of training recurrent neural networks". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. PMLR, pp. 1310–1318 (cit. on p. 100).

Pechyony, Dmitry and Vladimir Vapnik (2010). "On the Theory of Learning with Privileged Information". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 1894–1902 (cit. on p. 103).

Perlin, Ken (July 1985). "An image synthesizer". In: *ACM SIGGRAPH Computer Graphics* 19.3, pp. 287–296. DOI: 10.1145/325165.325247 (cit. on p. 96).

Petermeijer, Sebastiaan M. et al. (Oct. 2015). "The Effect of Haptic Support Systems on Driver Performance: A Literature Survey". In: *IEEE Transactions on Haptics* 8.4, pp. 467–479. DOI: 10.1109/TOH.2015.2437871 (cit. on p. 32).

Poon, James et al. (May 2017). "Local driving assistance from demonstration for mobility aids". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5935–5941. DOI: 10.1109/ICRA.2017.7989699 (cit. on pp. 38, 44).

Qiang Zeng, Etienne Burdet, and Chee Leong Teo (June 2009). "Evaluation of a Collaborative Wheelchair System in Cerebral Palsy and Traumatic Brain Injury Users". In: *Neurorehabilitation and Neural Repair* 23.5, pp. 494–504. DOI: 10.1177/1545968308323692 (cit. on p. 34).

Quigley, Morgan et al. (2009). "ROS: an open-source Robot Operating System". In: *ICRA workshop on open source system* 3.3.2 (cit. on p. 50).

Rasmussen, Carl Edward. and Christopher K. I. Williams (2006). *Gaussian processes for machine learning*. MIT Press, p. 248 (cit. on p. 85).

Reddy, Siddharth, Anca D. Dragan, and Sergey Levine (Feb. 2018). "Shared Autonomy via Deep Reinforcement Learning". In: *Robotics: Science and Systems (RSS)* (cit. on pp. 35, 42, 44).

Robinson, Hayley, Bruce MacDonald, and Elizabeth Broadbent (July 2014). "The Role of Healthcare Robots for Older People at Home: A Review". In: *International Journal of Social Robotics* 6.4, pp. 575–591. DOI: 10.1007/s12369-014-0242-2 (cit. on p. 16).

Sanders, David A. (Aug. 2017). "Using Self-Reliance Factors to Decide How to Share Control Between Human Powered Wheelchair Drivers and Ultrasonic Sensors". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.8, pp. 1221–1229. DOI: 10.1109/TNSRE.2016.2620988 (cit. on p. 30).

Sanderson, Diana and Michael Place (Feb. 2001). "Evaluation of the powered wheelchair and voucher schemes". In: *British Journal of Therapy and Rehabilitation* 8.2, pp. 46–51. DOI: 10.12968/bjtr.2001.8.2.13725 (cit. on p. 17).

Schettino, Vinicius and Yiannis Demiris (Nov. 2019). "Inference of user-intention in remote robot wheelchair assistance using multimodal interfaces". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4600–4606. DOI: 10.1109/IROS40897.2019.8968203 (cit. on pp. 24, 48).

— (May 2020). "Improving Generalisation in Learning Assistance by Demonstration for Smart Wheelchairs". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5474–5480. DOI: 10.1109/ICRA40945.2020.9197490 (cit. on pp. 24, 74).

Sharmanska, Viktoriia, Novi Quadrianto, and Christoph H. Lampert (Oct. 2014). "Learning to Transfer Privileged Information". In: (cit. on p. 103).

Simpson, Richard C. (2005). “Smart wheelchairs: A literature review”. In: *The Journal of Rehabilitation Research and Development* 42.4, p. 423. DOI: 10.1682/JRRD.2004.08.0101 (cit. on p. 34).

Simpson, Richard C., Edmund F LoPresti, and Rory A Cooper (Dec. 2008). “How many people would benefit from a smart wheelchair?” In: *Journal of rehabilitation research and development* 45.1, pp. 53–71. DOI: 10.1682/JRRD.2007.01.0015 (cit. on pp. 17, 73).

Sinyukov, Dmitry A. et al. (2014). “Augmenting a voice and facial expression control of a robotic wheelchair with assistive navigation”. In: *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*. Vol. 2014-Janua. January, pp. 1088–1094. DOI: 10.1109/SMC.2014.6974059 (cit. on p. 27).

Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams (June 2012). “Practical Bayesian optimization of machine learning algorithms”. In: *Advances in Neural Information Processing Systems*. Vol. 4, pp. 2951–2959 (cit. on p. 105).

Soh, Harold and Yiannis Demiris (2012). “Towards Early Mobility Independence : An Intelligent Paediatric Wheelchair with Case Studies”. In: *IROS Workshop on Progress, Challenges and Future Perspectives in Navigation and Manipulation Assistance for Robotic Wheelchairs* (cit. on p. 49).

— (Nov. 2013). “When and how to help: An iterative probabilistic model for learning assistance by demonstration”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3230–3236. DOI: 10.1109/IROS.2013.6696815 (cit. on pp. 16, 23, 39–41, 44, 73, 83, 85, 114).

— (Jan. 2015a). “Learning Assistance by Demonstration: Smart Mobility With Shared Control and Paired Haptic Controllers”. In: *Journal of Human-Robot Interaction* 4.3, p. 76. DOI: 10.5898/JHRI.4.3.Soh (cit. on pp. 18, 22, 23, 39–41, 44, 76, 77, 79, 83, 85, 100, 110, 114).

— (Mar. 2015b). “Spatio-Temporal Learning With the Online Finite and Infinite Echo-State Gaussian Processes”. In: *IEEE Transactions on Neural Networks and Learning Systems* 26.3, pp. 522–536. DOI: 10.1109/TNNLS.2014.2316291 (cit. on p. 40).

Srivastava, Nitish et al. (2014). “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958. DOI: 10.5555/2627435.2670313 (cit. on p. 107).

Tan, Chuanqi et al. (Oct. 2018). “A survey on deep transfer learning”. In: *Artificial Neural Networks and Machine Learning – ICANN 2018*. Vol. 11141

LNCS. Springer Verlag, pp. 270–279. DOI: 10.1007/978-3-030-01424-7{_}27 (cit. on p. 117).

Thalmann, Daniel, Jun Lee, and Nadia Magnenat Thalmann (2016). "An evaluation of spatial presence, social presence, and interactions with various 3D displays". In: *Proceedings of the 29th International Conference on Computer Animation and Social Agents - CASA '16*. New York, New York, USA: ACM Press, pp. 197–204. DOI: 10.1145/2915926.2915954 (cit. on p. 55).

Tjomsland, Jonas, Ali Shafti, and A. Aldo Faisal (Dec. 2019). "Human-Robot Collaboration via Deep Reinforcement Learning of Real-World Interactions". In: *NeurIPS Workshop on Robot Learning: Control and Interaction in the Real World* (cit. on p. 124).

Torkia, Caryne et al. (2015). "Power wheelchair driving challenges in the community: a users' perspective". In: *Disability and Rehabilitation: Assistive Technology* 10.3, pp. 211–215. DOI: 10.3109/17483107.2014.898159 (cit. on p. 28).

Tortora, Gerard J. and Bryan Derrickson (2016). *Principles of anatomy and physiology*. 15th. Wiley (cit. on pp. 19, 83).

Townsend, James T. and F. Gregory Ashby (1978). "Methods of Modeling Capacity in Simple Processing Systems". In: *COgnitive Theory*. First. Vol. 3. Psychology Press. DOI: 10.4324/9781315802473-14 (cit. on p. 65).

Valente, Michelle, Cyril Joly, and Arnaud De La Fortelle (June 2019). "An LSTM network for real-time odometry estimation". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. Institute of Electrical and Electronics Engineers Inc., pp. 1434–1440. DOI: 10.1109/IVS.2019.8814133 (cit. on p. 101).

Vander Poorten, E.B. et al. (May 2012). "Powered wheelchair navigation assistance through kinematically correct environmental haptic feedback". In: *2012 IEEE International Conference on Robotics and Automation*. IEEE, pp. 3706–3712. DOI: 10.1109/ICRA.2012.6225349 (cit. on p. 32).

Vapnik, Vladimir and Rauf Izmailov (2015). "Learning using privileged information: Similarity control and knowledge transfer". In: *Journal of Machine Learning Research* 16.61, pp. 2023–2049 (cit. on p. 103).

Vapnik, Vladimir and Akshay Vashist (July 2009). "A new learning paradigm: Learning using privileged information". In: *Neural Networks* 22.5-6, pp. 544–557. DOI: 10.1016/j.neunet.2009.06.042 (cit. on pp. 23, 103).

Vincent, Pascal et al. (2008). "Extracting and composing robust features with denoising autoencoders". In: *Proceedings of the 25th international conference on Machine learning - ICML '08*. New York, New York, USA: ACM Press, pp. 1096–1103. DOI: 10.1145/1390156.1390294 (cit. on p. 84).

Viswanathan, Pooja et al. (2017). "Intelligent wheelchair control strategies for older adults with cognitive impairment: user attitudes, needs, and preferences". In: *Autonomous Robots* 41.3, pp. 539–554. DOI: 10.1007/s10514-016-9568-y (cit. on pp. 28, 30, 31, 33, 43, 78, 91).

Wahyufitriyani, Cindy, Susy Susmartini, and Ilham Priadythama (2016). "Review of intelligent wheelchair technology control development in the last 12 years". In: *2016 2nd International Conference of Industrial, Mechanical, Electrical, and Chemical Engineering (ICIMECE)*. IEEE, pp. 201–206. DOI: 10.1109/ICIMECE.2016.7910458 (cit. on p. 34).

Wang, Lipo, ed. (2005). *Support Vector Machines: Theory and Applications*. Vol. 177. Studies in Fuzziness and Soft Computing. Berlin, Heidelberg: Springer Berlin Heidelberg. DOI: 10.1007/b95439 (cit. on p. 85).

Wang, Rosalie H. et al. (2011). "Usability testing of multimodal feedback interface and simulated collision-avoidance power wheelchair for long-term-care home residents with cognitive impairments". In: *The Journal of Rehabilitation Research and Development* 48.7, p. 801. DOI: 10.1682/JRRD.2010.08.0147 (cit. on p. 31).

Wästlund, Erik et al. (2015). "Evaluating gaze-driven power wheelchair with navigation support for persons with disabilities". In: *Journal of Rehabilitation Research and Development* 52.7, pp. 815–826. DOI: 10.1682/JRRD.2014.10.0228 (cit. on p. 27).

Winkler, Sandra L. et al. (Feb. 2016). "Innovative Power Wheelchair Control Interface: A Proof-of-Concept Study". In: *American Journal of Occupational Therapy* 70.2, 7002350010p1. DOI: 10.5014/ajot.2016.015750 (cit. on p. 27).

Woods, Brian and Nick Watson (Dec. 2003). "A Short History of Powered Wheelchairs". In: *Assistive Technology* 15.2, pp. 164–180. DOI: 10.1080/10400435.2003.10131900 (cit. on p. 27).

World Health Organization (WHO) (2016). *Global strategy on human resources for health: Workforce 2030*. Tech. rep. World Health Organization (WHO) (cit. on p. 15).

Zambelli, Martina (2018). "Online multimodal autonomous learning of robots internal models". PhD thesis. Imperial College London. DOI: https://doi.org/10.25560/73977 (cit. on p. 19).

Zambelli, Martina, Antoine Cully, and Yiannis Demiris (Jan. 2020). "Multimodal representation models for prediction and control from partial information". In: *Robotics and Autonomous Systems* 123, p. 103312. DOI: 10.1016/j.robot.2019.103312 (cit. on p. 102).

Zeiler, Matthew D and Rob Fergus (2014). "Visualizing and Understanding Convolutional Networks". In: *European Conference on Computer Vision*, pp. 818–833. DOI: 10.1007/978-3-319-10590-1{_}53 (cit. on p. 100).

Zhang, Fan, Antoine Cully, and Yiannis Demiris (2019). "Probabilistic Real-Time User Posture Tracking for Personalized Robot-Assisted Dressing". In: *IEEE Transactions on Robotics*, pp. 1–16. DOI: 10.1109/TRO.2019.2904461 (cit. on p. 16).

Zhang, Tianhao et al. (Oct. 2017). "Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation". In: *CoRR*, pp. 1–8. DOI: 10.1109/ICRA.2018.8461249 (cit. on pp. 36, 80).

Zhang, Xucong et al. (Jan. 2019). "MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.1, pp. 162–175. DOI: 10.1109/TPAMI.2017.2778103 (cit. on p. 56).

Zolotas, Mark and Yiannis Demiris (2019). "Towards Explainable Shared Control using Augmented Reality". In: *IEEE International Conference on Intelligent Robots and Systems*. DOI: 10.1109/IROS40897.2019.8968117 (cit. on p. 31).

Zolotas, Mark, Joshua Elsdon, and Yiannis Demiris (2018). "Head-Mounted Augmented Reality for Explainable Robotic Wheelchair Assistance". In: *IEEE International Conference on Intelligent Robots and Systems*. DOI: 10.1109/IROS.2018.8594002 (cit. on p. 31).

Zou, Hui and Trevor Hastie (Apr. 2005). "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2, pp. 301–320. DOI: 10.1111/j.1467-9868.2005.00503.x (cit. on p. 100).