MODULATING PHYSICAL INTERACTIONS IN HUMAN-ASSISTIVE TECHNOLOGIES

Siyao Hu

A DISSERTATION

in

Mechanical Engineering and Applied Mechanics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2020

Katherine J. Kuchenbecker, Supervisor of Dissertation
Adjunct Professor of Mechanical Engineering and Applied Mechanics

Jennifer Lukes, Graduate Group Chairperson
Professor of Mechanical Engineering and Applied Mechanics

Dissertation Committee:

Mark Yim, Professor of Mechanical Engineering and Applied Mechanics

Michelle J. Johnson, Assistant Professor of Physical Medicine and Rehabilitation

Katherine J. Kuchenbecker, Adjunct Professor of Mechanical Engineering and Applied Mechanics

MODULATING PHYSICAL INTERACTIONS IN HUMAN-ASSISTIVE TECHNOLOGIES

Acknowledgments

First and foremost, I would like to thank my PhD supervisor Dr. Katherine J. Kuchenbecker, for all the research support and guidance. I truly appreciate her mentorship, and I am grateful for the opportunities and freedom to explore research topics of interest and career options that she so kindly supported. Also thanks to her, I was able to make new friends and connections at different conferences and research institutes, and it has been an amazing journey for the last five years.

I would also like to thank my thesis committee, Dr. Michelle J. Johnson and Dr. Mark Yim, for their guidance and feedback on this research. I also thank Dr. Erin V. Vasudevan for her support and insight in our collaboration. Special thanks to the GRASP Lab and particularly Dr. Johnson for their support in the final part of this research. Also thanks to many members of the Penn Haptics Lab and Rehabilitation Robotics Lab: Alexis Block, Kevin Bui, Alex Burka, Naomi Fitter, Michael Sobrepera, and Eric Young. And I would like to thank the MEAM and GRASP Staff for their unfailing support.

Finally, I would like to thank my family members for their love and care over the years I'm especially filled with gratitude for my parents, Liang Hu and Xiufang Pan, for nurturing me through my childhood years and their continued confidence and encouragement. I would like to thank my wife Zihao, for always caring for me and giving me strength.

ABSTRACT

MODULATING PHYSICAL INTERACTIONS IN HUMAN-ASSISTIVE TECHNOLOGIES

Siyao Hu

Katherine J. Kuchenbecker

Many mechanical devices and robots operate in home environments, and they offer rich experiences and valuable functionalities for human users. When these devices interact physically with humans, additional care has to be taken in both hardware and software design to ensure that the robots provide safe and meaningful interactions. It is advantageous to have the robots be customizable so users could tinker them for their specific needs. There are many robot platforms that strive toward these goals, but the most successful robots in our world are either separated from humans (such as in factories and warehouses) or occupy the same space as humans but do not offer physical interactions (such as cleaning robots).

In this thesis, we envision a suite of assistive robotic devices that assist people in their daily, physical tasks. Specifically, we begin with a hybrid force display that combines a cable, a brake, and a motor, which offers safe and powerful force output with a large workspace. Virtual haptic elements, including free space, constant force, springs, and dampers, can be simulated by this device. We then adapt the hybrid mechanism and develop the Gait Propulsion Trainer (GPT) for stroke rehabilitation, where we aim to reduce propulsion asymmetry by applying resistance at the user's pelvis during unilateral stance gait phase. Sensors underneath the user's shoes and a wireless communication module are added to precisely control the timing of the resistance force. To address the effort of parameter tuning in determining the optimal training scheme, we then develop a learning-from-demonstration (LfD) framework where robot behavior can be obtained from data, thus bypassing some of the tuning effort while enabling customization and generalization for different task situations. This LfD framework is evaluated in simulation and in a user study, and results show improved objective performance and human perception of the robot. Finally, we apply the LfD framework in an upper-limb therapy setting, where the robot directly learns the force output from a therapist when supporting stroke survivors in various physical exercises. Six stroke survivors and an occupational therapist provided demonstrations and tested the autonomous robot behaviors in a user study, and we obtain preliminary insights toward making the robot more intuitive and more effective for both therapists and clients of different impairment levels. This thesis thus considers both hardware and software design for robotic platforms, and we explore both direct and indirect force modulation for human-assistive technologies.

Contents

List of Tables

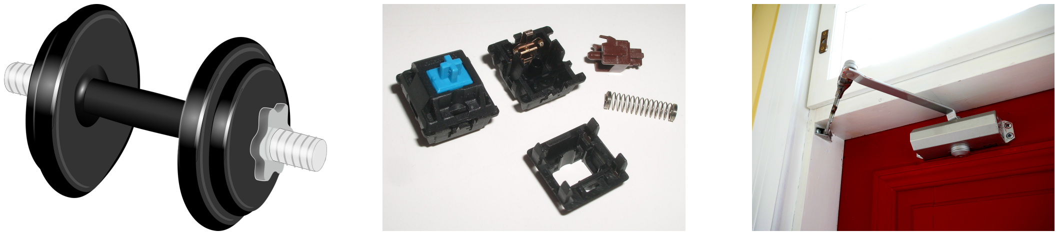
List of Figures

Chapter 1

Introduction

Many mechanical devices and robots are designed to operate in human environments, like homes and hospitals. In many cases, these devices interact physically with the humans in their environments, exchanging forces and torques during the process. For example, humans can push buttons on a gaming controller to issue commands to their avatars, while the controller can vibrate to activate the mechanoreceptors in the human hand to provide haptic feedback that signals virtual events. In another example, by carefully picking up and manipulating objects such as a coat or a spoon, a humanoid robot could assist an elderly person or a person with a disability to perform daily tasks like dressing and eating. If the devices or robots succeed in these assistive tasks, they help humans enjoy certain media and perform better in physical tasks.

Similar to how faster processors, smaller sensors, more accurate touch screens, an abundance of applications, etc., have made the smartphone an inseparable part of our lives, design considerations in both hardware and software that ensure safe, intuitive, and meaningful interactions between the devices and people could contribute to bringing robots into daily routines. Thus, we present typical models of physical interactions and motivate where the design considerations can come into play for robotic devices and control schemes for human-assistive technologies.



(a) A 3D render of a dumbbell.

(b) Components of a mechanical keyboard switch.

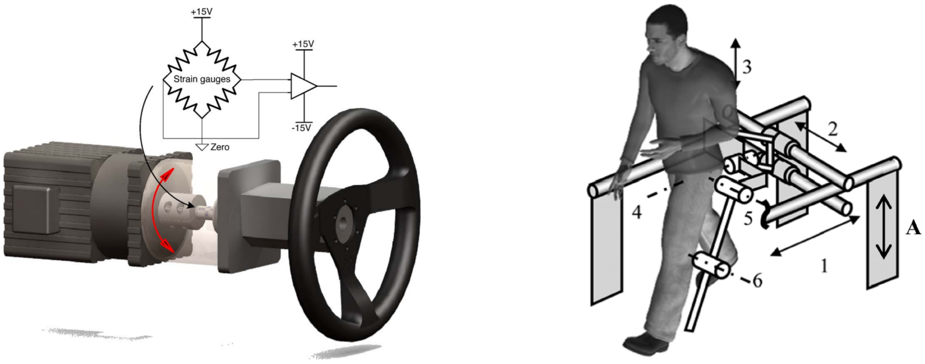
(c) A door closer. Image by Santeri Viinamäki / CC BY-SA.

Figure 1.1: Example devices that offer mechanical interactions.

1.1 Motivation

The simplest characterization of physical interaction could be a combination of three mechanical components: mass, spring, and damper. For example, a dumbbell (Fig. 1.1a) could effectively be modeled by a point mass, and humans could repeatedly lift dumbbells to exercise and become stronger. A keyboard switch (Fig. 1.1b) includes a mass (key stem) attached to a spring, which provides resistance as the key is pressed and returns the key to its original position once it is released. A door closer (Fig. 1.1c) consists of all three components (the rotational inertia of the door, a spring that pulls the door in the closing direction, and a pneumatic or hydraulic valve acting as a dashpot that slows down the door while it closes), and it is pervasively used so that humans don't have to manually close a door once it is opened. Assembled into different configurations, these three elements could model a large variety of physical interaction modes that are encountered in our daily life.

However, a mass-spring-damper system is passive, i.e., when there is no external force and no stored energy in the spring, such a system at rest will not start moving or change its output. Hence the electric motor, whose output torque is proportional to the current flowing through its coil, is commonly used to achieve active interaction. For instance, the electric motor fixed in the gaming controller has an asymmetric weight attached to its shaft, so that the controller vibrates in the user's hand when the motor turns. The person holding this controller can thus feel virtual events, although neither the motor nor the weight is directly in contact with the human hand. Given proper sensing and mechanical support structure in a driving simulator, a motor can be connected to the shaft where the steering wheel is



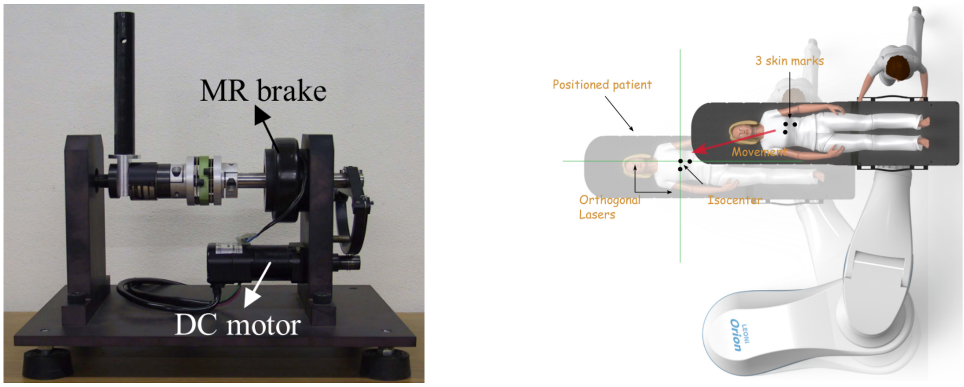
(a) Wheel assembly of the driving simulator in [52]. © 2011, IEEE.

(b) Illustration of the exoskeleton in [109]. © 2007, IEEE.

Figure 1.2: Example devices that utilize electric motors.

mounted and thus provide assistance or resistance to user inputs, such as Fig. 1.2a. Similar to the mechanical components, a few motors can be assembled into different configurations to support a multitude of assistive tasks, such as exoskeletons for physical therapy (see Fig. 1.2b).

Because of the active capabilities of the motors, it is necessary to regulate their torque outputs to ensure safety while they do work, especially in unstructured and dynamic environments that contain humans. Such regulation is often done in software, which assigns an input current to each motor in the system. Depending on the application, force/torque regulation can be done directly (where the device/robot exerts a prescribed force/torque output) or indirectly (where the robot arm follows its prescribed trajectory but alters its behavior when facing external disturbances). Examples of direct force/torque regulation include haptic devices simulating various physical elements where the force and/or torque is the output to the user (e.g., Fig. 1.3a) and robotic manipulators maintaining axial force in applications such as welding (e.g., [27]). Indirect force/torque regulation can be achieved by controlling the robot arm to appear as an impedance or an admittance. In the impedance approach, the robotic arm tracks a desired trajectory, but if one pushes against it, the interaction would feel like pushing on a passive mass-spring-damper system [36]. In the admittance approach, the pushing force would cause a velocity of the robot arm, as seen in many factory collaborative robots that assist lifting and positioning of heavy objects (e.g., 1.3b). In short, indirect methods do not track desired forces;



(a) Haptic device for virtual friction display with a magnetorheological brake [7]. © 2006, IEEE.

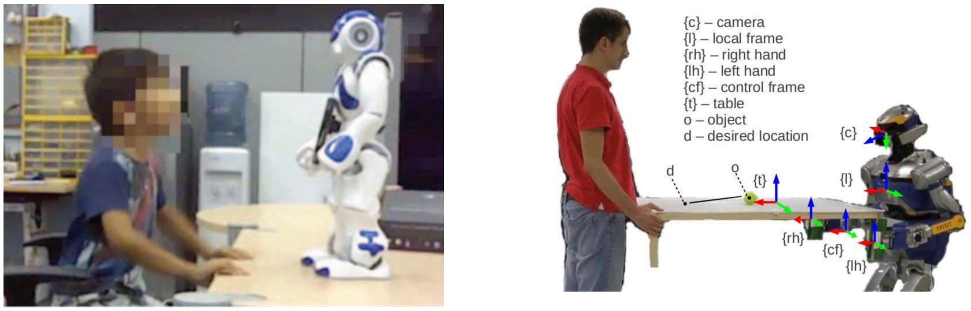
(b) Patient pre-positioning on the OrionPT robot using admittance control in [13]. © 2015, IEEE.

Figure 1.3: Example devices with force/torque regulation.

instead, the response from external inputs and disturbances are controlled. Finally, force/torque regulation can also be done using specific hardware. A common choice is to include electrical brakes in the system so that they can quickly dissipate energy and stop undesired motion, e.g., to enhance desirable properties like passivity in haptic devices (see Fig. 1.3a).

Humanoid robots may be a suitable tool for affording more complex and life-like interactions: one could imagine them behaving as social agents (such as providing companionship to children and the elderly, see Fig. 1.4a) as well as physical workers (such as actively helping humans carry and move heavy weights instead of only holding them in place, e.g., Fig. 1.4b). However, due to their complexity, the majority of robotic manipulators and humanoid robots in real-world applications need to be operated and programmed by professionals and can only work on a small fixed set of tasks and in static, known environments. But for assistive applications, where each user may have different needs of the robot and a different environment (like a person's home), it is important to let the users customize their robots for their needs without having to re-program it as soon as anything changes in the task or the environment. Thus many approaches allow the robot to be 'programmed' by demonstrations provided by users [9]; for instance, a person can hold a humanoid robot's arm and move it back and forth above the robot's shoulder, so that the robot learns to wave hello.

Importantly, in many of these programming by demonstration methods, force



(a) A child interacting with a Nao robot [6]. © 2019, IEEE.

(b) A humanoid robot carries an object with a human partner [3]. © 2014, IEEE.

Figure 1.4: Robots in assistive tasks.

regulation is not explicitly considered [105, 107, 74]: the kinematic trajectories are learned, and a generic positional controller is used to track them while the human is required to identify and adapt to robot motions (e.g., when the robot is waving its hand, humans step away to avoid being hit by the robot's arm). This omission could be largely due to the fact that kinematic motion is easier to learn, and in many of scenarios, the omission is perfectly allowed: for example, usually there isn't any physical interaction between a robot and a person waving at each other. But it could be problematic in other cases, especially in assistive tasks like joint manipulation and elderly care. Therefore, researchers also consider kinematic and force output behaviors at the same time, which would allow a waving robot to not hurt people even if they get in its way.

In this thesis, we present our work towards modulating human-robot physical interactions for assistive tasks, considering both hardware design and software integration. First, we develop a hybrid haptic device for high magnitude and safe force output, and we show that such a device offers unique opportunities for gait rehabilitation for people with stroke. In the second half of this thesis, we develop a learning from demonstration framework, which is instantiated on a humanoid robot and with indirect force regulation in a collaborative object-movement task. Then we use the framework in an occupational therapy setting and with direct force regulation, where stroke survivors and a therapist provided demonstrations and tested the robot in three different types of exercises.

1.2 Thesis Overview

After the broad introduction of various types of physical interactions and existing devices and robots that afford such interactions with humans, here we describe the components of this thesis, which contribute to methodologies for designing and controlling physical interactions for human-assistive technologies.

Our efforts begin in Chapter 2 with a hybrid force display, which combines an electric motor and a magnetic particle brake. Contrary to most previous motor-based or hybrid systems, we use the brake as the main actuator for passive, high-magnitude outputs, while the motor is used to compensate for the brake's inherent friction and provide active torque when necessary. Thus, in this chapter, we focus on hardware design and we use direct force regulation. A cable-based transmission enables a large workspace with a low moving mass. This chapter additionally includes characterization and control of a prototype, as well as experimental results of force outputs.

The hybrid force device from Chapter 2 presents a unique opportunity for rehabilitation purposes, particularly because the force output from the magnetic particle brake is powerful enough to affect human movement, and the output is passive and is therefore safe. Chapter 3 explores the effectiveness of the hybrid device in gait propulsion training for people with hemiparesis. In this chapter, design considerations for gait propulsion training are discussed, and the consequent adaptations of the prototype from Chapter 2, both in hardware and controller software, are described. A prototype of the gait propulsion trainer (GPT) was built, and extensive tests verified the time and magnitude characteristics of the prototype. A pilot study was conducted, and the results show that the GPT was effective in inducing larger propulsion forces and impulses in the targeted limb. Here we notice that there are a variety of parameters that could change the outcomes of a rehabilitation training scheme, and being able to customize the training for each individual user may be advantageous.

Thus, Chapter 4 develops a hierarchical learning from demonstration (LfD) framework for trajectories of the robot's end-effector, so that human users can easily teach their robots how to behave according to their specific needs. In this chapter, force output is regulated indirectly; we focus on developing a robust trajectory learning algorithm such that a generic (e.g., Cartesian-space PID) controller would lead to favorable robot behaviors for human users. With this hierarchical structure, each demonstration is annotated by its task parameters, and these parameters form the upper layer of the hierarchy that differentiates the demonstration (the base layer) during both training and testing time. Improvements of our approach are validated extensively in simulation, where the hierarchical framework could predict and improve performance during generalization as well as reduce computational cost. We also describe a user study where naive users tested a Willow Garage PR2 robot in a collaborative-object movement task; results show that our hierarchical LfD framework increased the efficiency of the task and led to more positive human perception toward the robot, compared to the state-of-the-art baseline.

Continuing the learning from demonstration approach, Chapter 5 uses the PR2 robot in a user study for an occupational therapy setting: the PR2 holds a small object with a stroke survivor and supports the person in various physical exercises. In addition, we invite an occupational therapist to provide the demonstrations with the stroke survivor in each study session. In this chapter, we explore the possibility of learning direct force regulation from demonstration, particularly for periodic exercises targeting range of motion; we also include one episodic exercise with roughly fixed start and goal positions, which uses the learning setup in Chapter 4. Results of this study show the potential advantages of learning force responses directly from demonstrations compared to the conventional method of tracking an 'optimal' trajectory, but the latter may be more helpful for people with lower motor functions.

Finally, this thesis concludes in Chapter 6 by summarizing the contributions of the presented work and discussing possible extensions and future work.

Chapter 2

One-Dimensional Haptic Force Display

Electric motors are used in the majority of haptic devices for their active torque outputs, which may require various transmission systems, like gears and capstans, to be amplified due to their limited magnitudes. Such transmission systems may introduce disadvantageous side effects such as friction, backlash, and higher moving mass, and there may still be stability issues due to discretization and time delay. Electric brakes, which generally have higher outputs and dissipate energy, are thus included in some designs for stronger forces and improved stability. Fig. 2.1 shows the specifications of three commercial product lines of electromagnetic brakes and three commercial product lines of DC motors (without gearboxes). It can be seen that brakes offer higher torque output and are more efficient than direct-drive DC motors in terms of volume, mass, and electric power.

This chapter describes our design where the brake is used as the main actuator and the motor as the secondary one, compensating for the brake's inherent friction and providing active motion. Thanks to the brake's high power, a simple spool and cable transmission is possible to allow a large workspace and a low moving mass. By directly controlling the torque outputs from the actuators, we created a hybrid device capable of simulating various interactions, such as virtual springs and dampers with high stiffness and viscosity. This work was first reported at the 2017 IEEE World Haptics Conference [39].

After discussing related work in Section 2.1, our prototype design is given in

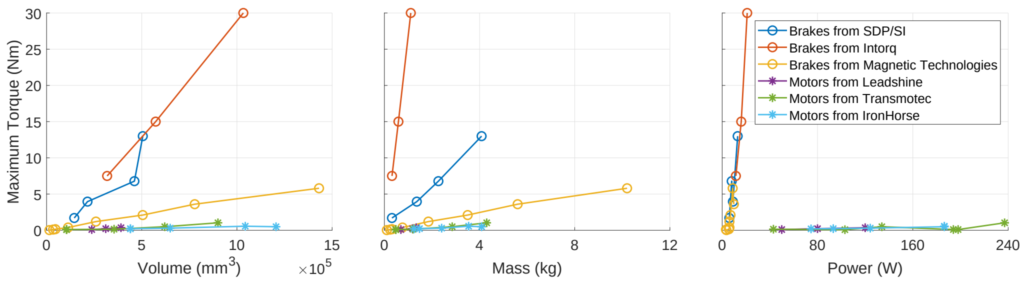


Figure 2.1: A comparison between selected electromagnetic brakes and direct-drive DC motors from representative commercial vendors[1]. Each product appears once on each of the three subplots, indicating how its maximum output torque compares to its volume, mass, and power consumption.

Section 2.2. The two actuators are characterized in Section 2.3, including their torque constants and dynamics parameters for the motor. Section 2.4 describes the controller given the actuator characteristics and position feedback from the sonar sensor, and results of the device simulating various haptic elements are given in Section 2.5. Section 2.6 summaries this chapter and discusses potential applications for this hybrid device design.

2.1 Related Work

Electric motors are commonly used in haptic devices (e.g., [78, 112, 100]). The PHANToM was the first design that combined low friction, low mass, stiff links, and relatively high force-generating capabilities (10 N), so that free space feels free and virtual constraints are not easily saturated [78]. Although carefully designed algorithms and controllers enable motors to realistically simulate virtual environments, hardware constraints, usually stemming from heat considerations, often greatly limit the maximum force on the user. For example, Ryu and Yoon achieved a virtual wall with high stiffness of 290 kN/m using a one-degree-of-freedom (1-DOF) haptic device driven by a DC motor, but the maximum exert-able force is 7 N because of the current limit [100]. This limitation allows easy saturation of the virtual environment, which is undesirable for haptics applications [78].

[1] Specifications sampled and obtained from the following sources (links to manufacturer/distributor websites): SDP/SI, Intorq, Magnetic Technologies, Leadshine, Transmotec, IronHorse.

One approach to outputting higher forces is to use brakes rather than motors. Blake and Gurocak measured an average sustainable torque of 620 N mm at the human metacarpophalangeal joint and 220 N mm at the proximal interphalangeal joint; consequently they designed compact magnetorheological (MR) brakes that can output up to 821 N mm, preventing saturation when a glove that incorporates such brakes is worn on the hand [15]. The MR glove was shown to successfully enable users to grasp objects and distinguish spring stiffness values in virtual reality [15]. However, the inherent friction of the MR brakes was also felt by the user in free space.

By combining a motor and a brake, one can take advantage of the high force output of the brake and eliminate its inherent friction using the motor. For example, An and Kwon designed a force feedback interface with a DC motor and an MR brake connected to one shaft to simulate friction [7]. In this study, the brake was used to display virtual rotational friction (up to 3.5 N m), and the motor was used to reduce the brake's inherent friction in free space (down to 0.004 N m) [7]. This system's handle is fixed on the shaft, and therefore its workspace is confined. This motor/brake mechanism can also be extended to include more degrees of freedom, e.g., a 2-DOF five-bar linkage device [59, 8]. Conti and Khatib included an additional spring between the shafts of the motor and the brake, enabling force display performance comparable to traditional haptic devices while reducing power consumption [25]. However, when the desired torque and the measured torque are opposite in direction, the motor has to provide the desired torque plus the torque required to counter the spring, which is not energy efficient [25].

One disadvantage of MR brakes is their residual off-state torque. Rossa et al. eliminated the residual torque by complex mechanical design, making the brakes unidirectional by overrunning clutches between a DC motor and two MR brakes [97]. The off-state torque can also be eliminated by better sensing and modeling of the brake, such as closed-loop magnetic induction control with an embedded Hall-effect sensor in the MR brake designed for use with a DC servomotor for virtual needle insertion [32]. Moreover, hysteresis of torque output due to magnetization of steel in the brake can also be eliminated by this closed-loop induction control [32].

Some applications for kinesthetic haptic displays require large workspaces and involve more of the human body than just the hand, such as sports training. For instance, large-scale indoor sport simulation can be achieved by tensioning ropes

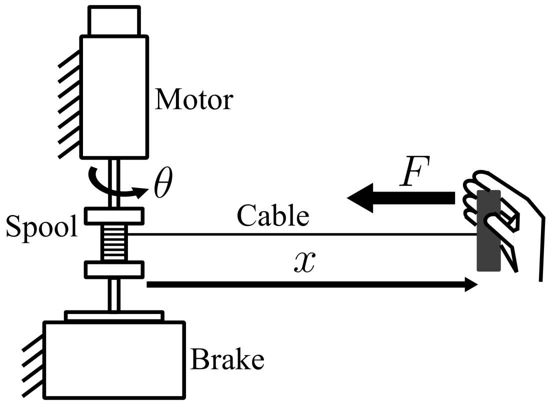


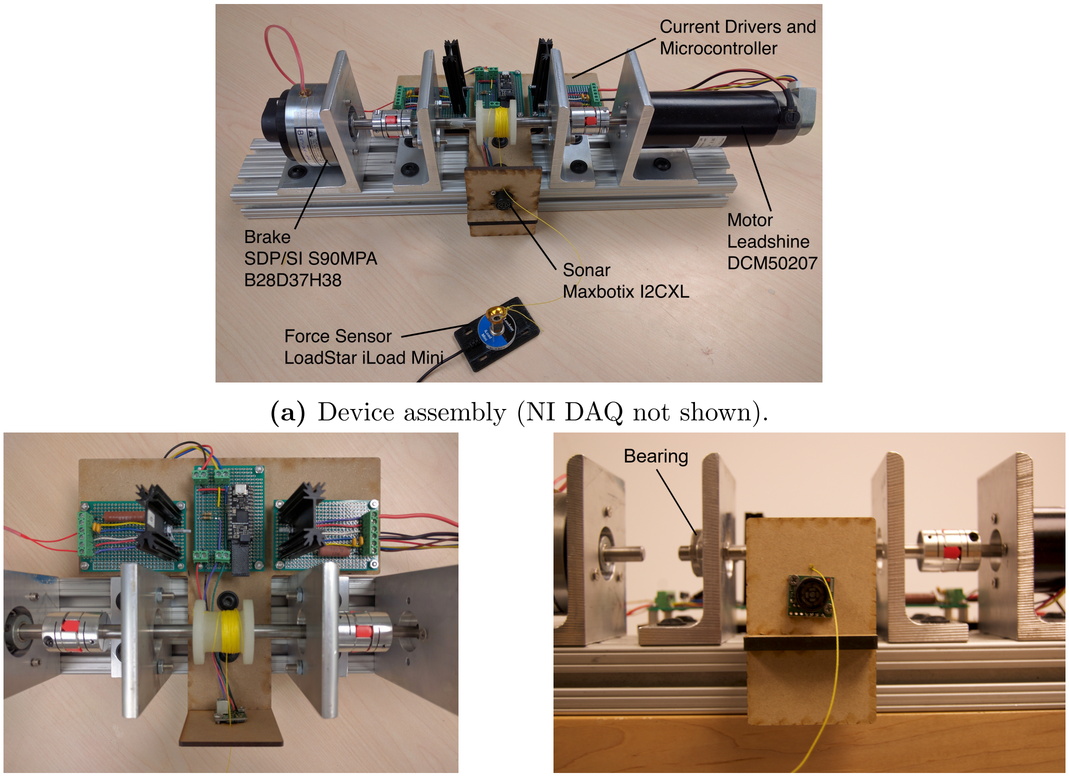
Figure 2.2: Schematic of the hybrid force display.

connecting motors and the sport equipment being used, for example oars in rowing [112] and rackets in tennis [111].

2.2 Prototype Design

We created a one-dimensional haptic force display that combines a brake and a motor and uses a cable on a spool to tether the user to the base, as shown in Fig. 2.2. When the user moves away from the base, the brake is activated to generate a force resisting the user's motion. When low forces are required, the motor can be activated to cancel the inherent friction of the brake. The motor can also reel in the unwound cable. By using a brake as the source of force output and a spooled cable to tether the user, this design has the potential to safely output high magnitude (about 133 N) unidirectional forces in a large workspace with relatively low inherent friction and moving mass.

A prototype of the force display has been built, as shown in Fig. 2.3. The core of the prototype consists of a brushed DC motor and a magnetic particle brake both connected to a steel shaft, where a 3D-printed spool is mounted. A cable is reeled onto the spool and tethers a force sensor (used only for verification) on the other end. The user holds the force sensor and can move in a one-dimensional workspace in front of the base. The brake can generate a resistance force only when the user moves away from the base. The motor cancels the inherent friction of the brake when the desired force output is low and reels the cable onto the spool when the


Current Drivers and Microcontroller
Brake SDP/SI S90MPA B28D37H38
Motor Leadshine DCM50207
Sonar Maxbotix I2CXL
Force Sensor LoadStar iLoad Mini
Bearing

(a) Device assembly (NI DAQ not shown).

(b) Top view. No bearings are used when both actuators are connected to the shaft.

(c) Front view. A bearing supports the shaft when only one actuator is connected.

Figure 2.3: Device assembly.

user moves back toward the base. Therefore, the device never does positive work on the user.

2.2.1 Actuator Selection

Magnetic brakes have been adopted for use in haptic devices, e.g., [7, 12]. A DC current flowing through the brake coil generates a magnetic field, forcing the enclosed magnetic particles to form chains connecting the shaft and the stationary housing. When the shaft rotates, a resisting torque is generated; its magnitude is proportional to the field strength and independent of angular velocity. The SDP/SI S90MPA-B28D37H38 magnetic particle brake that we selected has a maximum torque $\tau_{max} = 1.69\,\mathrm{N\,m}$ at $6\,\mathrm{W}$. With a spool radius $r = 1.27\,\mathrm{cm}$, the maximum brake output force is $F_{\max} = \frac{\tau_{\max}}{r} = 133.07\,\mathrm{N}$. The minimum torque output, or the brake's inherent friction, is $0.034\,\mathrm{N\,m}$, corresponding to $F_{\min} = 2.68\,\mathrm{N}$ for a spool of the same radius. If a higher output force is desired, brakes in the same product series can generate torques up to ten times greater than the one chosen in this prototype, with only twice the size (brake body diameter) and power consumption.

For this prototype, the motor torque needs to be large enough to overcome the inherent friction of the brake and also to accelerate the shaft fast enough to reel in the cable quickly. Rated at 120 W, the LeadShine DCM-50207 permanent magnet brushed DC motor has a maximum continuous torque of 0.35 N m. We do not use a gear head to avoid adding inertia, backlash, and friction that would contaminate the system's haptic output. A quadrature encoder with a resolution of 4000 counts per revolution is mounted onto the motor shaft.

2.2.2 Sensors and Drivers

A LoadStar iLoad Mini load cell is used to measure the cable tension during experiments, as shown in Fig. 2.3a. The force sensor has a range of 222.41 N and a diameter of 3.18 cm. The force sensor is sampled at 62.5 Hz for validation purposes and is not used in the controller.

A Maxbotix I2CXL sonar sensor is used to measure user position directly. The sensor reading has a resolution of 0.01 m, but its maximum update rate is 25 Hz. An ATmega32U4-based microcontroller is used to read from the sonar sensor. The same microcontroller also tracks the motor encoder counts, and it sends the sonar and encoder readings to the host computer via the serial port at 175 Hz.

Linear current drivers were built for both the brake and the motor using OPA548 operational amplifiers equipped with heatsinks. A custom program is used to read and record data and generate the necessary commands for the actuator drivers via a National Instruments USB-6216 DAQ.

2.2.3 Prototype Assembly

The two actuators, current drivers, and the sonar sensor are assembled on a custom base, as shown in Fig. 2.3a. The motor and the brake are fixed on aluminum angle brackets sitting at the two ends of the assembly. A shaft equipped with a spool is located at the center of the prototype. When the shaft is connected to only one of the actuators via a flexible shaft coupler, a self-aligning bearing can be mounted on another angle bracket at the opposite side of the connected actuator to support the shaft, as shown in Fig. 2.3c. The shaft can also be connected to both actuators simultaneously. Using two bearings would reduce the overhung load on the actuators, but we found the bearings introduced shaft misalignment that caused a position-dependent force output as well as vibration when the shaft would rotate. Therefore, we remove the bearings from the prototype when both actuators are connected to the center shaft. The angle brackets are secured on a double-slot 80/20 aluminum rail, enabling quick attachment/detachment of the actuators from the center shaft.

A 3 m long PowerPro Maxcuatro fishing line is wound on the spool, and the other end of the cable is tethered to the force sensor. Assuming the effective spool radius is constant when a cable is reeled and the cable is rigid and massless, the relationship between the torque exerted by the string on the spool (τ) and the tension measured by the force sensor (F) is $\tau = Fr$.

The sonar sensor is mounted at the front of the prototype facing away from the spool to measure the position of the end effector. On the same piece of medium-density fiberboard that holds the sonar sensor, there is a small hole at the same height as the lower side of the spool, guiding the cable as it reels and unreels.

2.3 System Characterization

We characterized the dynamics of the components before using the haptic force display. As a passive device, the rotary brake is mainly used to generate large torques. Therefore, its most important feature is the torque-current relationship. The brake's inertia is $3.84 \times 10^{-6}\,\mathrm{kg\,m^2}$ from the datasheet, more than 12 times less than that of the motor, which is $4.73 \times 10^{-5}\,\mathrm{kg\,m^2}$ from characterization detailed below. The motor's inherent friction also has an important role when designing a controller for the system. Details of the characterization results are shown in Table 2.1, which also includes for comparison the design by Rossa et al. that combines a motor and two brakes with overrunning clutches [97].

2.3.1 Brake Torque Calibration

Although velocity-independent, the brake's torque-current curve is only approximately linear, and the manufacturer's manual does not include an exact current-

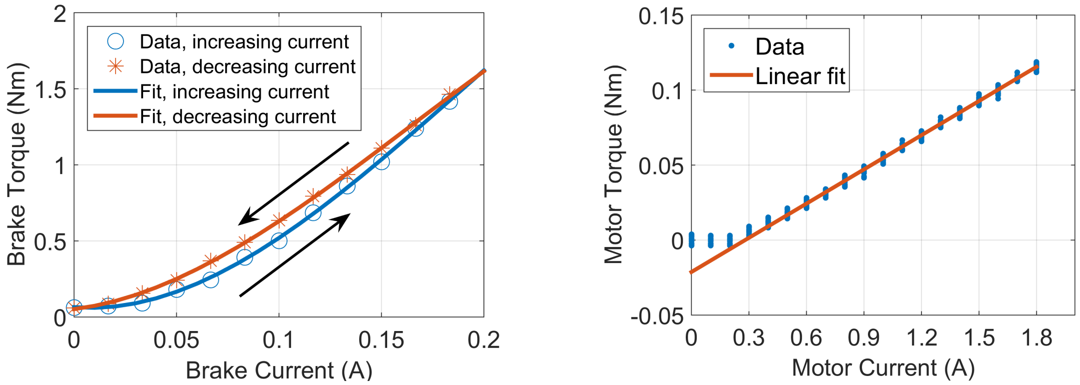
Table 2.1: Characteristics of actuators.

	Rated Electrical Power (W)	Rotor Inertia ($\mathrm{kg\,m^2}$)	Maximum Active Torque (N m)	Maximum Passive Torque (N m)	Minimum Friction Torque (N m)	Damping Coefficient (N m s)
LeadShine DCM50207 Motor	120	5.56×10^{-5}	0.35	0	0.021	7.45×10^{-5}
SDP/SI B28D37H38 Brake	6	3.84×10^{-6}	N/A	1.69	0.034	0
Hybrid Device	126	8.03×10^{-5}	0.35	1.69	0.051	7.45×10^{-5}
Dual Brake Design [97]	170	4.18×10^{-5}	0.2	5.5	0.034	7.32×10^{-4}

torque relationship. Therefore extra calibration of the brake was necessary.

The brake was secured on a table, and selected currents were applied in sequence. For each current value, the experimenter held the force sensor and moved away from the prototype to measure the cable tension F and thus the braking torque τ. The computer recorded readings from the force sensor and the encoder, and we calculated the mean of the force data when the shaft was rotating. Calibration results are shown in Fig. 2.4a. There are two curves because of hysteresis: when the current in the brake increases, torque output follows the lower curve, and when current decreases, torque output follows the upper curve. Third-order polynomial models were used to fit the increasing and decreasing curves shown in Fig. 2.4a.

When controlling the brake, we need to be able to calculate the current needed for the desired torque output, so a direct data fit was used to solve for the inverse



(a) Brake.

(b) Motor.

Figure 2.4: Torque-current relationship for both actuators.

functions:

$$\begin{aligned} i_{\text{inc}} =& -0.8713\text{ A} + (2.844\text{ A}/\sqrt{\text{Nm}})\tau^{\frac{1}{2}} \\ & - (10.48\text{ A}/\sqrt[3]{\text{Nm}})\tau^{\frac{1}{3}} + (8.651\text{ A}/\sqrt[4]{\text{Nm}})\tau^{\frac{1}{4}} \end{aligned} \tag{2.3.1}$$

$$\begin{aligned} i_{\text{dec}} =& -0.5317\text{ A} + (2.004\text{ A}/\sqrt{\text{Nm}})\tau^{\frac{1}{2}} \\ & - (6.792\text{ A}/\sqrt[3]{\text{Nm}})\tau^{\frac{1}{3}} + (5.458\text{ A}/\sqrt[4]{\text{Nm}})\tau^{\frac{1}{4}} \end{aligned} \tag{2.3.2}$$

2.3.2 Motor Characteristics

Large permanent magnet DC motors have a linear current-torque relationship and non-negligible inertia and friction. As detailed below, we calibrated our motor's torque constant in a similar manner to the brake, and we used an adaptive controller to model the motor's dynamics.

Torque Constant

The motor torque constant can be calculated using commanded current, i, and force measurement, F. For this experiment, the force sensor was clamped down while the motor exerted a series of torques on the shaft, therefore increasing cable tension. Data at zero current was used to tare the force sensor, and Fig. 2.4b shows the tared data. It can be seen that there is inherent friction in the motor, so that if the generated torque is less than the inherent friction, no force is exerted on the force sensor. A linear fit was performed on data with current greater than 0.5 A. The torque constant was estimated to be $K_t = 7.6 \times 10^{-2}\,\mathrm{N\,m\,A^{-1}}$ and the minimum friction torque to be $\tau_f = 0.021\,\mathrm{N\,m}$.

Inertia and Damping

An adaptive controller [26] was used to estimate the motor inertia and damping coefficient. The device follows a desired trajectory while updating the estimated parameter values in real time according to errors in shaft position and velocity. When there is no additional load on the motor shaft and we ignore static friction, the shaft dynamical model is:

$$J\ddot{\theta} + b\dot{\theta} + \mathrm{sgn}(\dot{\theta})\tau_f = \tau \tag{2.3.3}$$

where J is shaft inertia, b damping coefficient, τ_f kinetic friction, and τ generated motor torque. We define error $e = \theta_{\mathrm{des}} - \theta$ and use the following PD controller,

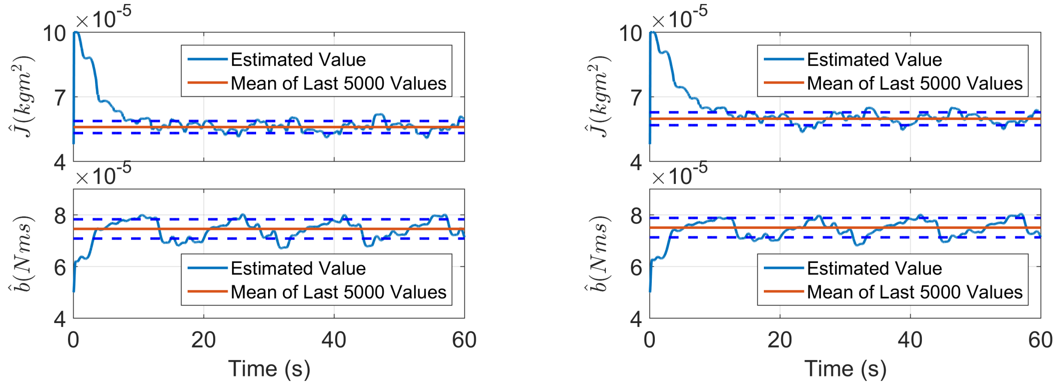
$$\tau = \hat{J}(\ddot{\theta}_{\mathrm{des}} + K_d\dot{e} + K_p e) + \hat{b}\dot{\theta} + \tau_f\mathrm{sgn}(\dot{\theta}) \tag{2.3.4}$$

where the hat operator indicates the estimated value of the parameter. Following [26], the update law for parameter estimation can be derived as follows:

$$\frac{d}{dt}\begin{bmatrix}\hat{J}\\ \hat{b}\end{bmatrix} = W^{\top}\hat{J}^{-1}B^{\top}PE \tag{2.3.5}$$

where $W = \begin{bmatrix}\ddot{\theta} & \dot{\theta}\end{bmatrix}$, $B = \begin{bmatrix}0 & 1\end{bmatrix}^{\top}$, and P is a positive-definite matrix. This adaptive controller ensures that the achieved trajectory converges to the desired one, and consequently the estimated parameter values also converge. For the motor with one degree of freedom, the term $\hat{J}^{-1}B^{\top}P$ acts like a gain that converts a two-dimensional error to a one-dimensional update value, and the two-dimensional term $W^{\top}$ scales the update value according to weights that are $\ddot{\theta}$ and $\dot{\theta}$.

If the desired trajectory is persistent excitation, the estimated parameter values will converge to their true values [26]. For a 1-DOF system like the motor, a sum of sinusoids can be used as the desired trajectory in experiments. We chose the



(a) Parameter estimation using only the motor.

(b) Parameter estimation using the motor with shaft coupler.

Figure 2.5: Parameter estimation using adaptive controller. Dotted lines indicate the ±5 % interval of the mean value.

following:

$$\theta_{\text{des}} = 25\sin(1\,\mathrm{s}^{-1}t) + 25\sin(\sqrt{2}\ \mathrm{s}^{-1}t) \tag{2.3.6}$$

Fig. 2.5 shows estimated parameter values over time from a 100 Hz controller for the motor alone and for the motor plus shaft coupler. We use the mean of the last 5000 data points as the final estimation, and it can be seen that the parameters approximately converge to a ±5% interval around these means. For the motor alone, the estimated inertia and damping coefficient are $\hat{J} = 5.60 \times 10^{-5}\,\mathrm{kg\,m^2}$ and $\hat{b} = 7.45 \times 10^{-5}\,\mathrm{N\,m\,s^{-1}}$, respectively. When the shaft coupler is mounted, the estimated inertia increases to $5.98 \times 10^{-5}\,\mathrm{kg\,m^2}$, while the estimated damping coefficient stays almost constant at $7.50 \times 10^{-5}\,\mathrm{N\,m\,s^{-1}}$, as expected.

2.4 Force Display Controller

To interact with the device, the user holds the force sensor mounted on a piece of 3D-printed plastic and moves in a one-dimensional workspace. The prototype can detect the motion of the user and activate one or both of the actuators to exert a force on the user or reel in the unwound cable when the user moves toward the base.

2.4.1 Output Force Control

When a non-zero force is desired, the necessary command current for the brake driver is determined from (2.3.1) or (2.3.2), enabling cable tensions up to about 130 N.

Another important criterion for an effective haptic device is that free space should feel free [78]. This requirement is inherently satisfied when the user moves back toward the base, because the tension generated from the slack cable's weight is negligible. However, due to inherent friction in the brake and the motor, cable tension when the user moves away from the base is not negligible if zero current is applied to both actuators. Thus, we use the motor to actively assist in friction cancellation in these situations. However, if the motor generates too high a torque, it will keep rotating after the user has stopped moving away, which makes the cable loose and the feeling inconsistent. Therefore, we only partially cancel the combined friction of the brake and motor.

2.4.2 Reeling Strategy

The sonar sensor is used to measure the user motion. When the user moves toward the base, the motor reels in the unwound cable, using the controller in (2.3.4). Since the second-order dynamics of this system tend to overshoot, especially if the user moves quickly toward the base and suddenly stops, it is necessary to avoid the situation where the motor reels too much cable and actively yanks on the end effector, as this behavior causes a spike in the force on the user and may not be safe. Therefore, we devised a threshold on the position discrepancy between the hand and the reel:

$$\delta = \left(\theta - \frac{x}{r}\right) \tag{2.4.1}$$

where θ is current shaft angle, x user position measured by the sonar sensor, and r radius of the spool. When the position error is below the position discrepancy threshold, δ_{th}, the motor does not reel in the cable even if the end effector is moving toward the base. The value for δ_{th} was experimentally determined to be 3 rad so that the shaft angle does not overshoot under moderately fast user speeds.

2.5 Experimental Results

After the force display controller was implemented, we conducted experiments while recording force and position data. The controller can generate a constant force output as well as simulate haptic elements like virtual springs and dampers. When suitable, each actuator is also used alone in experiments as a comparison to the combined device.

2.5.1 Constant Force Output

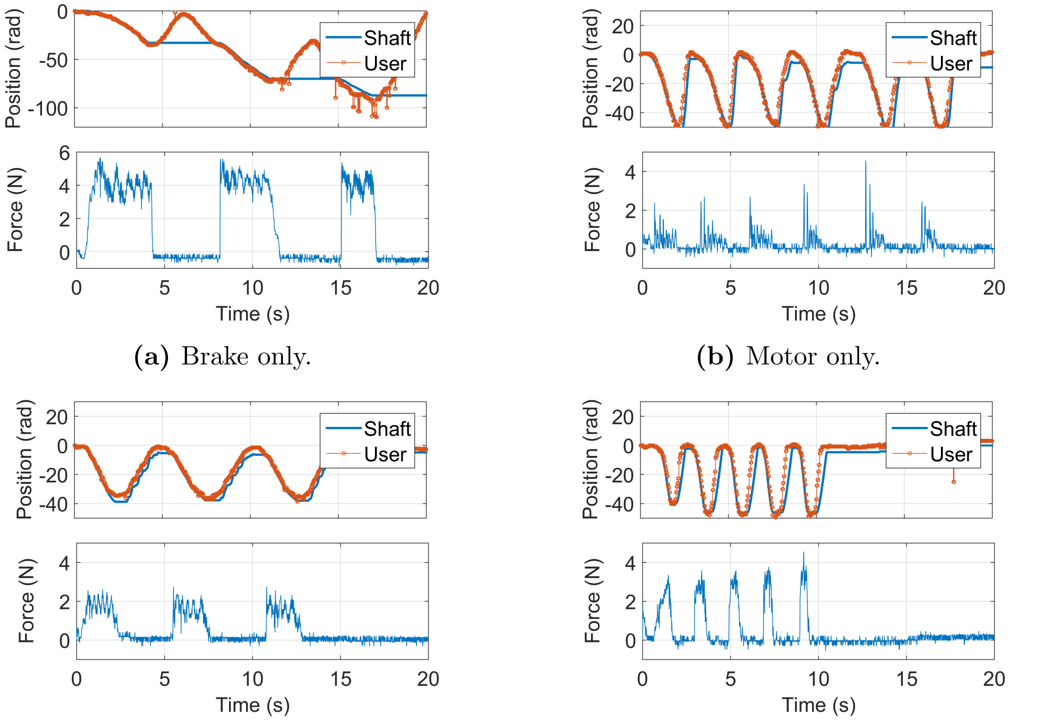
Zero Force

Fig. 2.6 shows the recorded position and force data when the desired force is zero. When the system has only the brake, it is not activated at all to minimize the resistance force. The user feels a force of about 4 N when moving away from the spool, as shown in Fig. 2.6a, generated from the inherent friction of the brake and the bearing. In addition, the brake does not have the ability to gather unwound cable when the user moves back toward the base. In contrast, the motor can spool the cable and exerts an average force of less than 1 N when the user is moving away, as seen in Fig. 2.6b. The larger spikes in this force recording are generated at the moments when the cable first became taut due to the high inertia of the motor.

For the combined device, the user feels a resistance force because of the brake's non-canceled inherent friction, as shown in Figs. 2.6d and 2.6c. Nevertheless the average force felt is about 2 N, canceling about 45% of the combined friction of the brake and motor without spikes in cable tension, and unwound cable can be spooled.

Constant Non-Zero Force

A 100 N force was chosen as the desired force output, and a 60 N force was chosen to test how the device behaves at an intermediate force value. Individual actuators were not tested in this scenario because the brake cannot reel in the cable and the motor cannot continuously generate large forces due to its maximum steady-state current of 3.94 A, which corresponds to an output force of 27.56 N. As shown in Fig. 2.7, the system is well calibrated and can accurately generate the desired force when the user moves away and reel the cable in at the appropriate times.



(a) Brake only.

(b) Motor only.

(c) Combined device with slow movement.

(d) Combined device with fast movement.

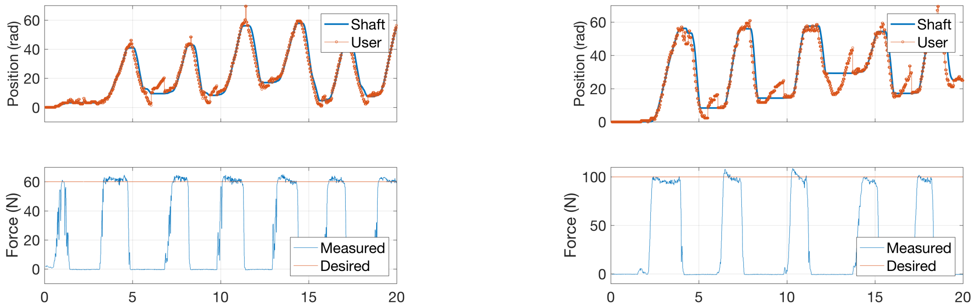
Figure 2.6: Data from free-space simulations. The shaft position was recorded by the encoder, and the user position was recorded by the sonar sensor and the converted to rad.

2.5.2 Haptic Elements

The device can also simulate other haptic elements, such as virtual springs and dampers, by controlling brake force output according to the user's position and velocity.

Virtual Springs

When simulating a virtual spring, the force output is proportional to the distance the user has moved away from the initial position. Figs. 2.8a and 2.8b show the results when the combined device simulates two strong virtual springs. The data points that form vertical lines were generated when the user was stopping, as the device can exert controlled forces only when the user moves away.



(a) 60 N force output.

(b) 100 N force output.

Figure 2.7: Data when simulating constant force.

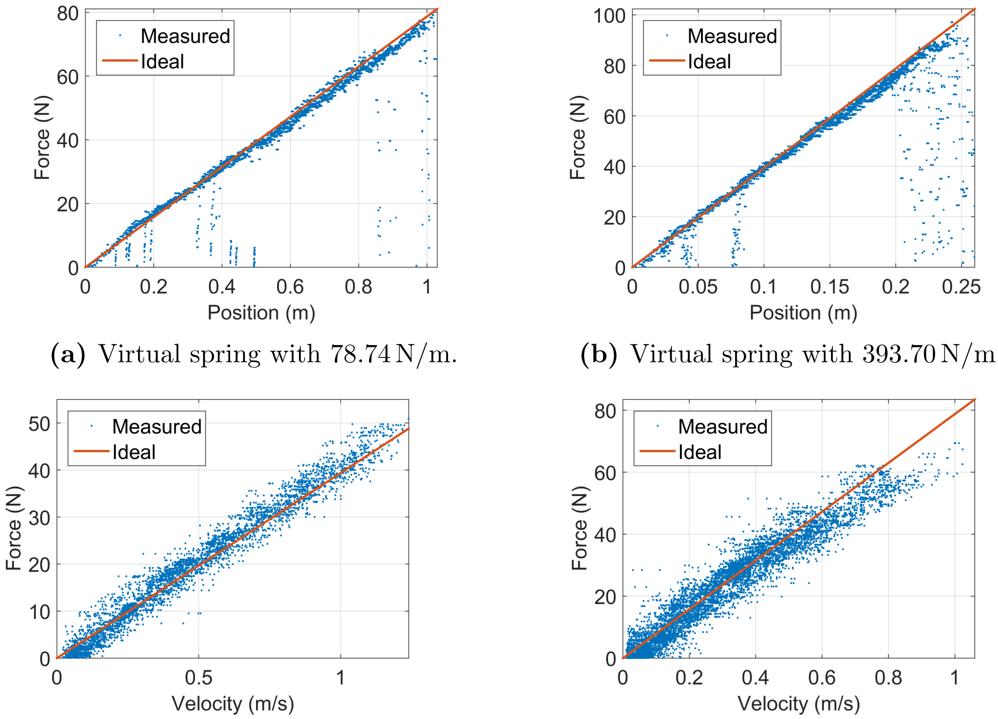
Virtual Dampers

When simulating a virtual damper, the force output is proportional to the velocity at which the user is moving. When the shaft rotationally accelerates, for example when the user starts moving after being stationary, the brake is activated using (2.3.1) given the desired force calculated from the shaft rotational speed and the virtual damping constant. When shaft rotation decelerates, for example when the user stops moving, a lower force output is required and therefore (2.3.2) is used. Figs. 2.8c and 2.8d show the results when the device simulates two strong virtual dampers.

2.6 Summary

We designed a haptic device consisting of a DC motor and a magnetic particle brake, with a cable tethering the user to the actuated shaft. This design enables high-range force display (2 N to 130 N) with a large workspace (3 m) and low moving mass (~0.50 kg). Due to the brake's passivity, the device is inherently stable and safe to use. On the other hand, the motor assists in friction cancellation and enables the device to reel in unwound cable. A prototype of the device was built to compare the performance of the combined device to that of each actuator alone.

When simulating free space, where the desired force output is zero, the device is essentially the motor with the additional inherent friction and inertia from the brake. Although the motor can partially cancel the friction and reduce the output force, the combined device makes free space feel less free. A well-designed programmable



(a) Virtual spring with 78.74 N/m.

(b) Virtual spring with 393.70 N/m.

(c) Virtual damper with 39.37 N/(m/s).

(d) Virtual damper with 78.74 N/(m/s).

Figure 2.8: Data for strong virtual haptic elements.

mechanism, such as a clutch, that detaches the brake from the shaft could make the free-space simulation more realistic but would add complexity and might introduce transient effects that can be felt.

Constant forces and virtual springs can be well simulated by the combined device due to the brake's activation properties. However, the force-velocity relationship when simulating virtual dampers is noisier, possibly because the system is switching between two discrete brake torque curves, (2.3.1) and (2.3.2), due to the noisy velocity estimation from differentiating the encoder readings. A smooth hysteresis model could alleviate this problem, and a brake with built-in magnetic field strength sensor could be used to achieve closed-loop control of the field strength and consequently the output torque, e.g. [32].

Regarding the hardware components, the magnetic particle brake and the DC motor we chose in the prototype provide good performance. However, the sonar sensor presents a few limitations. First, the sensor has limited range and update frequency, which means the user cannot move too far from the device and the controller performance is limited. In addition, because of the sonar sensor is most accurate when the detected object is straight in front of it, small displacements of the end-effector in the lateral and vertical directions may cause inaccuracies in readings and consequently errors in the cable reeling controller. Thus, a different type of sensor should be used to measure the distance in future iterations.

Finally, employing a cable to tether the user to the actuators makes the workspace of the device limited only to the range of the sonar sensor. In addition, the cable also makes the moving mass low because its density is negligible. The non-rigidity of the cable makes force output possible only when the user moves away from the base, an advantage particularly in applications such as rehabilitation, where avoiding positive work on the user is desired. This uni-directional output becomes a limitation in other applications that require large bidirectional forces; it could be solved, for example, by using a pulley system to tether the end-effector to the base on both sides, but that solution will limit the workspace and add extra mass and friction to the system.

Because of the lightweight linkage, large workspace, and safe high-magnitude force output, the combination of brake and motor has good potential for use in large-scale VR, physical therapy, and rehabilitation, which will be presented in the next chapter.

Chapter 3

Over-Ground Gait Propulsion Trainer for Stroke Rehabilitation

The majority of robotic rehabilitation devices use electric motors to provide assistance during training, guiding the impaired limb along desired paths or providing help when the person cannot exert enough power to finish particular motions. Therefore, system designers need to make sure enough assistance is added to help people in the task of interest. But at the same, there should not be too much assistance such that the user simply follows along without exerting any effort. Moreover, because the electric motors add energy to the system, they need to be carefully regulated so that they don't accidentally hurt people.

The hybrid mechanism in Chapter 2 thus offers a unique approach for rehabilitation, because the main force output is from the magnetic particle brake, which only dissipates energy. Having resistance force means that the user has to exert enough effort to finish tasks, and the resistance can be adjusted to accommodate different levels of motor function. Importantly, if the resistance is too high, the user will not be able to overcome it to finish the task at hand, but the interaction is safe because no energy is added by the system. With this consideration, we developed a Gait Propulsion Trainer (GPT) that targets unilateral propulsion force by applying periodic resistance force to the pelvis for people with hemiparesis from stroke. A device verification test and a pilot study show the effectiveness of the GPT in the intended application.

After discussing related work in Section 3.1, design considerations are discussed in Section 3.2 that motivate the use of the hybrid mechanism and various adaptions needed in this chapter. The GPT prototype is detailed in Section 3.3, including individual actuator control and the state-machine controller for the overall GPT. Section 3.4 describes the device verification test and the pilot study, and the results are presented in Section 3.5 and discussed in Section 3.6. Section 3.7 summaries this chapter and future steps that could validate this approach for gait rehabilitation.

3.1 Related Work

Mobility impairments are a frequent cause of stroke-related disability. Among ischemic stroke survivors 65 years and older, half reported hemiparesis (weakness on one side) persisting six months post-stroke, and 30% were unable to walk without assistance [53]. Difficulty with functional movements like gait can lead to physical deconditioning post-stroke, which contributes to poor cardiovascular fitness, muscular atrophy, and metabolic syndrome [44]. These effects can increase the risk of a second stroke or cardiovascular event [44]. Motor impairments are also associated with negative psychological and social outcomes such as depression and withdrawal [11].

There are several training strategies aimed at increasing walking function after stroke [17, 29]. However, many strategies appear to deliver comparable outcomes and may not result in meaningful gains in walking speed [28]. It is possible that treatments correcting a specific functionally limiting aspect of gait may be more effective than more global treatments that aim to correct a variety of functional deficits during walking practice (e.g., body-weight-supported treadmill stepping). Recently, propulsion asymmetry after stroke has been identified as a functionally limiting gait deficit that is correlated with walking speed [5, 10, 16, 117]. A stroke that results in hemiparesis is often accompanied by propulsion asymmetry, which occurs when the more affected (i.e., paretic) leg is unable to generate propulsion forces that match those on the less affected (i.e., non-paretic) leg. Propulsion asymmetry between the two legs is negatively correlated with walking speed and functional mobility; conversely, improvements in propulsion asymmetry with training are associated with improvements in walking speed [10, 38, 96]. These findings suggest that targeting propulsion symmetry during gait training is a promising approach for improving functional mobility after stroke. For a person with hemiparetic gait, propulsion must be increased in one leg (i.e., the paretic leg) while propulsion on the contralateral leg is either reduced or maintained. Since walking is a bilateral activity, it can be challenging to train propulsion in one leg only.

One way to accomplish unilateral propulsion training is through functional electrical stimulation (FES), which administers an electric pulse to stimulate muscle contraction at appropriate times in the gait cycle. Training walking with FES stimulating paretic ankle musculature results in increased paretic leg propulsion, reduced gait asymmetry, and increased walking speed after stroke [10, 38]. However, one possible disadvantage of using FES is that the pattern of muscle fiber recruitment is non-specific [33] and dissimilar to the pattern of recruitment found naturally in gait[35]. Gait rehabilitation that involves mechanical assistance or resistance training may have advantages over electrical stimulation training due to the natural order of muscle recruitment, particularly for people who are ambulatory and have some ability to voluntarily recruit leg muscles. Additionally, FES may be contraindicated in people with pacemakers and skin lesions; thus it is not an appropriate treatment for all.

Robotic devices can potentially be used for unilateral mechanical gait training because they can be configured for various output modes and different levels of interaction with the user [75]. For example, the LOwer extremity Powered ExoSkeleton (LOPES) allows both 'patient-in-charge' (e.g., for the unimpaired limb) and 'robot-in-charge' (e.g., for the impaired limb) modes during training [109]. In most existing approaches, robotic devices provide assistance to users during training, which allows patients to practice motions that self-generated effort cannot achieve. However, too much assistance may also reduce physical effort during motor training [43] and may encourage slacking [75]. In addition, the assistance provided by robotic devices typically relies on a desired trajectory, but commonly used normative movements such as minimum-jerk trajectories and trajectories recorded from unimpaired volunteers have not been proved to maximize training effect [75]. Although it is possible to infer the desired motion of the paretic limb from the unimpaired leg for gait training [108], the stability of this approach for stroke patients remains to be determined.

On the other hand, robotic resistance devices have been less extensively investigated. A few studies have found that non-robotic resistance exercises that require higher effort from the impaired limb could indeed help people with stroke improve motor function [113, 88, 82]. One study of robot-applied resistance found that

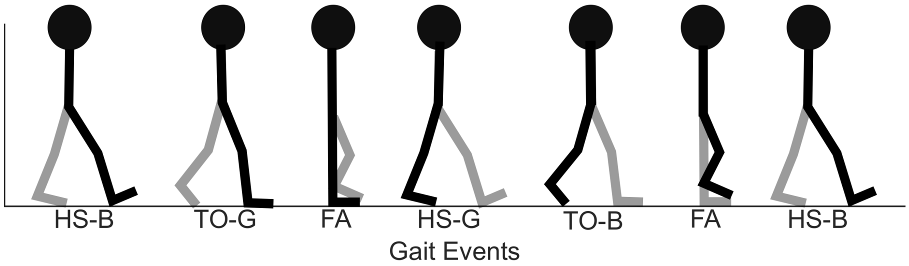


Figure 3.1: Illustration of a series of gait events. HS, TO, and FA represent heel-strike, toe-off, and feet-adjacent events, and B and G represent the black and gray legs that correspond to the events. Note that the horizontal distances between events are for illustration only and are not necessarily proportional to time.

velocity-dependent moments against hip and knee movements during swing phase in people with spinal cord injury resulted in enhanced knee flexion after the resistance was removed [60]. In another study, gait asymmetry in people with stroke improved more following a single period of walking with swing-phase resistance applied at the ankle, compared to swing-phase assistance [119]. However, another study using the same device did not find a significant difference in walking speeds between people receiving 6 weeks of assistance or resistance training [118], suggesting there is much room to explore the best resistance training schemes and how they can help with motor recovery.

3.2 Design Motivation

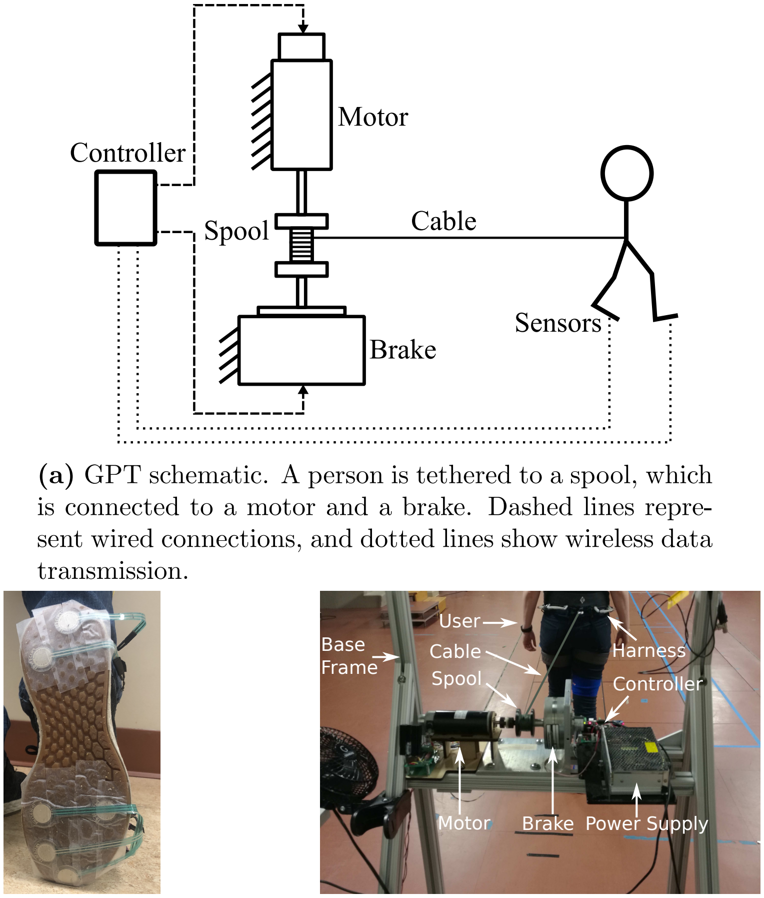
At the moment, the most effective resistance training for gait rehabilitation is undetermined. Thus, we aim to design a robotic device that can apply programmable stance-phase resistance targeting unilateral propulsion during walking training, while maintaining safety. Fig. 3.1 illustrates a series of gait events, and our current control scheme applies the stance-phase resistance from TO-G to TO-B (see the third design motivation below for details). The mechanism in Chapter 2 was adapted according to the following design motivations for this application.

First, maintaining safety means that any positive work done on the user (typically by motors in robotic devices) must be extensively regulated, or a protective mechanism must be in place in case of instability (e.g., the harness in [119] was used for safety but not for body weight support). Actuators that dissipate energy, such as brakes, generate resistance forces by acting against the user's motions, making them a good option for safe resistance intervention, and thus we use the hybrid mechanism in Chapter 2 for actuation.

Second, the requirement of stance-phase resistance eliminates the common approach of applying robotic disturbance at the ankle. The foot is stationary during the stance phase; thus the robotic disturbance would normally do little to no mechanical work, and any positive work could destabilize the user. Instead, we apply the resistance at the pelvis, which is propelled to advance forward in space by the leg in stance phase. We postulate that such resistance could lead to improved propulsion symmetry, similar to how step-length symmetry was shown to improve with resistance training [119]. Since we aim to apply resistance at the pelvis, we tether the user's pelvis to a stationary brake via a cable, so that resistance forces can be applied by the brake as the user walks away.

Third, for unilateral intervention during stance phase, the resistance force needs to turn on when one leg is in single limb support; that is, the resistance should begin to be applied when only the resisted (i.e., ipsilateral) leg is in contact with the ground (TO-G in Fig. 3.1), and the contralateral leg is in swing phase. It then needs to turn off when the resisted leg lifts off the ground ("toe-off") and initiates swing phase (TO-B in Fig. 3.1). In people without neurological damage, the duration of the period starting at the onset of single limb support to toe-off on one side is a little less than 50% of the gait cycle [91], or about 0.4−0.5 s, depending on how fast the person is walking. In people with stroke, this period can be variable: it may be longer than 0.5 s in some people because they walk slower, and it may be shorter in others because people with stroke spend less time in single limb stance on their weaker side. Regardless, accurate and fast gait phase detection is required. Given that we allow over-ground walking and apply resistance only to the pelvis, an orthosis may be cumbersome and introduce erroneous interaction with the user, while drastically increasing the cost of the system. In contrast, we add a lightweight and easy-to-use wearable device with wireless data transmission for gait-phase detection.

To the best of our knowledge, the GPT is the first device concerning robotic stance-phase resistance.



(a) GPT schematic. A person is tethered to a spool, which is connected to a motor and a brake. Dashed lines represent wired connections, and dotted lines show wireless data transmission.

(b) Event switches on a shoe sole.

(c) The stationary device. The cable is digitally enhanced to increase visibility.

Figure 3.2: Schematic and a prototype of the GPT.

3.3 Prototype Design

The overall design of the Gait Propulsion Trainer is shown in Fig. 3.2a. In this section, the GPT's wireless sensing system and wearable equipment is described. Then details about the GPT's stationary device and the controllers for the two actuators of the GPT are given. Finally, the state machine that coordinates the two actuators according to the user's stance phase and handle unexpected situations to guarantee user safety is explained.

3.3.1 Sensing System and Wearable Equipment

The sensing system uses twelve Motion Lab Systems MA-153 event switches to determine the user's gait phase. Each event switch reads 'on' when there is contact on its surface, and 'off' for no contact. Six event switches are attached to the bottom of each of the user's shoes: four underneath the metatarsal bones, and two underneath the heel, as shown in Fig. 3.2b. This setup is similar to the gait phase detection system in [89], but without a gyroscope and with two event switches wired in parallel instead of each force-sensing resistor for improved reliability. An ATmega32U4-based microcontroller reads the outputs from the event switches and wirelessly transmits them to the stationary device (detailed in the next subsection) at 100 Hz using an XBee module. The microcontroller and wires are attached to the subject's thigh using soft straps. Since the user will always walk forward during training and only return to the starting point after finishing each trial, here we did not include the sonar sensor from the prototype in Chapter 2.

The user also wears a climbing harness below his or her waist. The back of the harness can be attached to a cable, which we use to tether the user to the stationary device and apply the resistance force to the pelvis.

3.3.2 Stationary Device

As shown in Fig. 3.2c, the stationary device consists of an aluminum frame to which the other components are attached. Its height is adjustable to accommodate users of different heights. This setup is similar to the hybrid force display in Chapter 2, but a brake with higher rated torque output is chosen to provide the necessary resistance in gait training: the SDP/SI 24 V S90MPA-B45D75S rotary magnetic-particle brake has a maximum output torque of 6.78 N m, and its inherent friction (minimum output) is 0.11 N m. A 3D-printed spool with a diameter of 3.18 cm is mounted on the shaft of the brake, and the cable that tethers the waist-worn harness is reeled on the spool. The same motor in Chapter 2, Leadshine 30 V DCM50207, is used in the GPT.

An ATmega32U4-based microcontroller wirelessly receives event switch readings from the sensing system and determines the user's gait events on each side, including stance, toe-off (stance-to-swing transition), swing, and heel-strike (swing-to-stance transition), as defined in [89]. These gait phases then determine the actuator outputs, as detailed in Section 3.3.4. The microcontroller also reads the motor encoder values. During experiments, the microcontroller sends sensor readings and control signals to a PC for recording at approximately 30 Hz.

3.3.3 Actuator Control

Motor Control

When the person walks away from the stationary device, we use the motor to generate an assisting torque that partially cancels the inherent friction of the rotary brake. This friction compensation reduces the resistance force applied on the user during unimpaired limb stance, to allow unilateral training. With the GPT prototype, we increased the motor torque from 0 until a magnitude such that when the brake is not activated, the shaft would start turning with minimal pulling on the cable and immediately stop turning once the pulling force is removed. We achieved a friction reduction of 0.06 N m (about 54% of the brake's inherent friction).

When the user reaches the end of a trial and the GPT needs to reset, we use the motor to reel in the loose cable. We use the following controller for reeling speed:

$$\begin{aligned}\tau_m &= \tau_{m,0} + K(v - v_{\text{des}}) \\ &= k_t i_m,\end{aligned} \tag{3.3.1}$$

where τ_m is the motor torque, $\tau_{m,0} = 0.12\,\mathrm{N\,m}$ is the feedforward term (chosen to be slightly larger than the brake's inherent friction), $K = 1.27 \times 10^{-4}\,\mathrm{N\,m\,s}$ is the proportional gain, v is the current reeling speed, $v_{\text{des}} = 25.13\,\mathrm{s}^{-1}$ is the desired reeling speed, i_m is the motor current, and $k_t = 0.08\,\mathrm{N\,m\,A}^{-1}$ is the motor torque constant. Unlike Chapter 2, reeling only takes place after each trial, and therefore no additional calibration is done for the motor's dynamics model.

Brake Control for Cable Tension

We use the rotary brake to generate a pelvic resistance force during training. As the subject walks away from the stationary device, the cable unreels and causes the spool and the brake shaft to rotate.

The brake was calibrated in a similar procedure as in Chapter 2, and the results are shown in Fig. 3.3, where we ignore the hysteresis effect and only acquire one

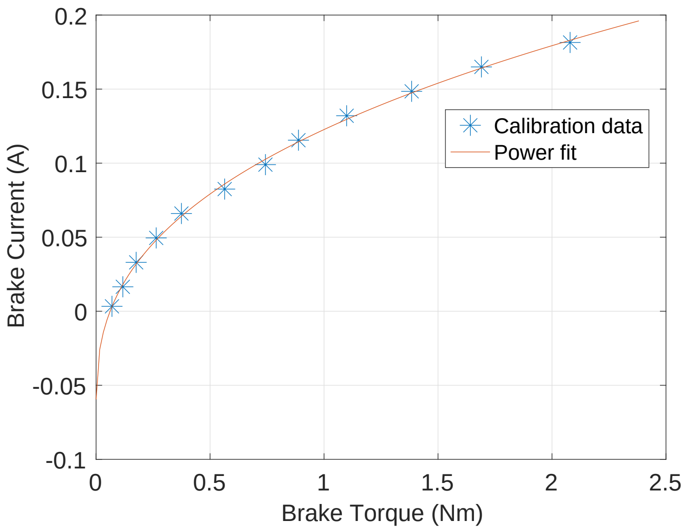


Figure 3.3: Torque-current relationship of the rotary magnetic brake.

curve because the GPT will only output a constant force during use. We obtained the following power model from the data:

$$\begin{aligned} i = &- 0.05951\ \text{A} + (0.01538\ \text{A}/\sqrt{\text{Nm}})\tau^{\frac{1}{2}} \\ &+ (0.2644\ \text{A}/\sqrt[3]{\text{Nm}})\tau^{\frac{1}{3}} - (0.09759\ \text{A}/\sqrt[4]{\text{Nm}})\tau^{\frac{1}{4}} \end{aligned} \tag{3.3.2}$$

where i is the current through the brake and τ is the braking torque. Note that the power fit gives negative current values when desired torque outputs are close to zero because the inherent friction of the brake is not entirely canceled by the motor. Negative current generates the same resistance force as positive current in such a brake. Therefore, we put a lower bound of 0 A on the brake current command.

When the control signal for the brake driver increases instantly from zero to a high value, the resultant torque tends to behave like static friction: an initial load torque higher than the steady-state torque is needed to start the shaft rotation. To avoid this unwanted effect when the brake is first actuated, we set the brake current to linearly rise from zero to the desired level within a specified time interval, as shown in the top panel in Fig. 3.4. In experiments we use a default ramp-up time of $t_r = 60\,\text{ms}$.

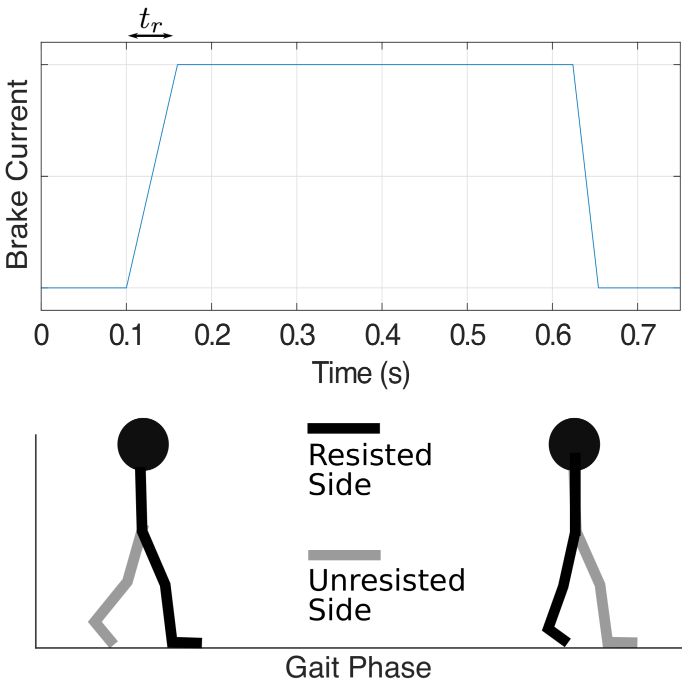


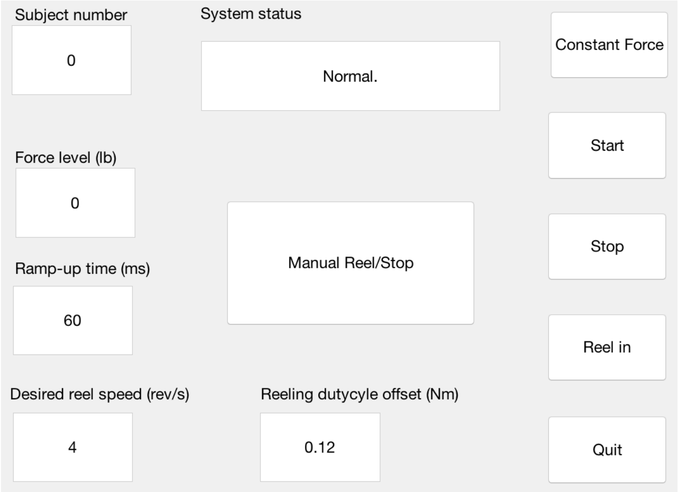
Figure 3.4: Illustration of the brake current in an example step with corresponding gait phases as the user walks, where t_r (typically 60 ms) represents the ramp-up time when the brake is first actuated.

3.3.4 State-Machine Controller

We implemented a state-machine controller for the GPT to determine the desired output for the two actuators. To ensure safe operation, we include error-handling states that control how the GPT behaves when the event switches miss readings and when the motor turns unexpectedly.

Operator Interface and State Machine

An operator controls the system via a graphical user interface (GUI) in Matlab running on a PC, as shown in Fig. 3.5. Subject number, desired resistance force level, ramp-up time (t_r), reeling speed (v_{des}), and reeling duty cycle offset ($\tau_{m,0}$) are specified in the text input fields. The 'System status' field shows pre-defined messages that indicate the operation modes on the GPT and error messages, if any. Buttons in the GUI correspond to some of the transition signals in the state-machine controller shown in Fig. 3.6.



Subject number
0

System status
Normal.

Force level (lb)
0

Ramp-up time (ms)
60

Desired reel speed (rev/s)
4

Reeling dutycycle offset (Nm)
0.12

Figure 3.5: Matlab GUI. When the program starts, the operator enters the subject number, the desired force level, and other relevant parameters if necessary. The operator then controls the GPT via the push buttons and monitors the system status.

As shown in Fig. 3.6, when the program starts, the GPT enters the 'System Idle' state where both the brake and the motor are off. After the operator clicks the 'Start' button, the GPT zeros the motor encoder reading and enters the 'Non-Resisting' state, where the brake is off. In this state, the motor is activated in the forward direction with the assisting torque so the inherent friction of the rotary brake is partially canceled; the motor remains activated in the forward direction until the 'Stop' button is clicked.

When the resisted foot is in stance phase and the unresisted foot is in swing (single limb stance; arrow marked 'Ron URoff'), the GPT enters the 'Resisting' state where the brake is activated to generate the desired resistance force on the user. When the unresisted leg is in stance phase and the resisted foot leaves the ground (arrow marked 'Roff URon'), the GPT returns to the 'Non-Resisting' state. The desired resistance forces and the corresponding gait phases are illustrated in Fig. 3.4 for transitions between the 'Non-Resisting' and 'Resisting' states.

After the operator clicks the 'Stop' button in the GUI, the controller returns to the 'System Idle' state. All data collected during the walking trial, including time-stamped event switch and encoder readings and the control signal for the rotary brake, are then saved to the PC. After the user is detached from the cable, the operator can then click the 'Reel' button for the GPT to enter the 'Reeling' state,

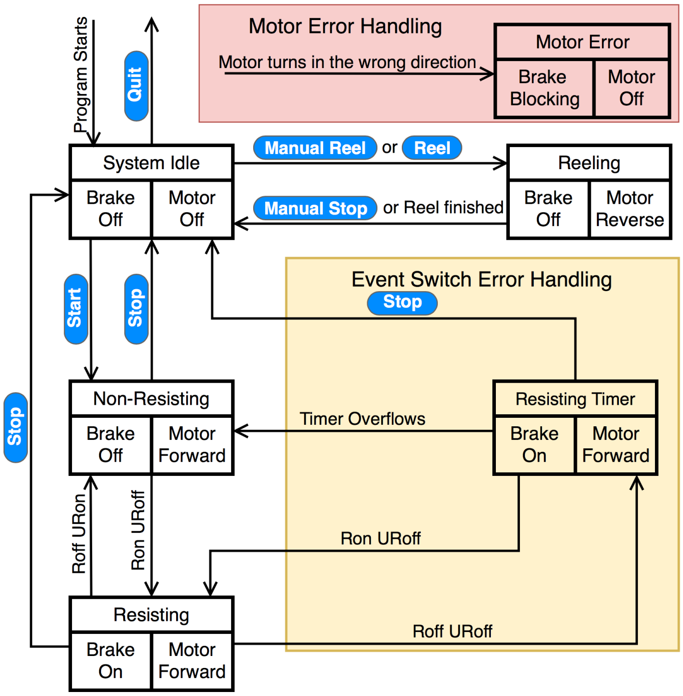


Figure 3.6: State-machine controller for the GPT. Each state has a name (e.g., 'Non-Resisting') and descriptions of the actuators (e.g., 'Brake Off' and 'Motor Forward'). 'Ron' represents when any of the event switches on the resisted side reads 'on', i.e., when the resisted leg is in stance phase. 'Roff' represents when all these event switches read 'off', i.e., when the resisted leg is in swing phase. 'URon' and 'URoff' similarly represent the unresisted side.

where the rotary brake remains off and the motor automatically rotates the spool in the reverse direction until the loose cable is fully reeled. If necessary, the operator can also use the 'Manual Reel/Stop' button to reel the loose cable for an arbitrary length. At the same time, the user can walk or sit in a wheelchair and come back to the starting point near the stationary device.

Error Handling States

To ensure safety, we added error handling states that determine how the GPT should behave if a sensor or actuator misbehaves.

One possible type of error is with the event switches under the user's shoes. Under normal conditions, the event switches should read on and off periodically, indicating the gait phases as the user walks forward. However, if the event switches are not securely attached to the correct locations under the shoes, there may be time periods where all event switches read no contact, suggesting that neither of the user's feet is in contact with the ground. However, this scenario should never happen, considering the user should be walking and not running or jumping when training with the GPT. Thus, to avoid sudden changes in the brake force when the event switches misread, especially when no event switches are pressed during the 'Resisting' state (arrow marked 'Roff URoff'), an extra 'Resisting Timer' state imposes a short delay before turning off the brake resistance force, as shown in the 'Event Switch Error Handling' box in Fig. 3.6. Note that this error handling state and its transitions are enclosed in the yellow region, indicating that this error is recoverable and the GPT can continue normal functioning if all event switches still read no contact or if expected event switch readings are acquired within the short delay.

Secondly, a rare but potentially dangerous failure of the GPT is when the DC motor turns unexpectedly, particularly in the reeling direction, which may be caused by incorrect motor installation or driver breakdown. Thus readings from the quadrature encoder on the motor are used to detect this failure, and the 'Motor Error Handling' box shown in Fig. 3.6 enables the GPT to react quickly and ensure that the user is never actively pulled: when triggered, the GPT turns off the motor output immediately and activates the magnetic brake to generate a high magnitude torque and completely block the shaft from turning. Note that this error handling component is enclosed in the light red region to represent that the 'Motor Error' state has the highest priority in the state machine and can be entered from every other state. The operator must completely turn off the system once the GPT enters the 'Motor Error' state.

3.4 Evaluation Methods

In order to evaluate the performance of the GPT prototype, we performed two preliminary hardware tests. In the first test, we verified GPT characteristics, including resistance force control and the time delay between corresponding gait events and GPT resistance actuation. In the second test, one healthy participant tested the device, and the propulsion forces were measured by force plates.

In these two tests, the GPT was placed at one end of a 10 m over-ground walkway

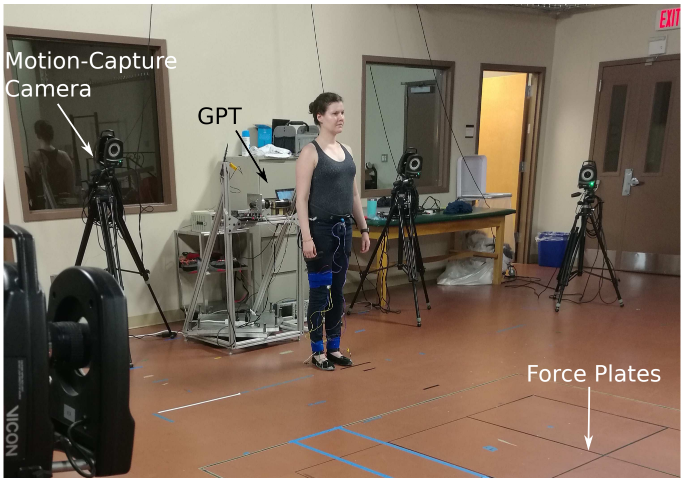


Figure 3.7: A user in front of the GPT ready to start training. She will walk over the in-ground force plates while feeling periodic resistance through the harness, and her kinematics will be recorded by the motion-capture system.

that has four AMTI Force and Motion embedded in-ground force plates. As shown in Fig. 3.7, the kinematics of the subject was measured using a Vicon optical motion-analysis system.

3.4.1 Verification Test

Because this test was for objective hardware performance, an experimenter, instead of a consented participant, started near the GPT and walked to the end of the walkway while wearing the GPT harness in each trial.

The participant wore reflective markers bilaterally on the pelvis (iliac crest), hip (greater trochanter), mid-knee line, ankle (lateral malleolus), and toe (5th metatarsal). The Vicon motion capture system recorded 3D marker trajectories at 100 Hz. The recorded marker trajectories were used to determine the intended GPT behavior based on the state-machine controller in Fig. 3.6.

A one-dimensional iLoad Mini force sensor rated up to 222.41 N was fixed to the back of the harness worn by the subject, and the cable was tied to the load side of the sensor via the sensor's hook. This in-line configuration allowed the sensor to accurately record the cable's tension during the test. In addition, the remote sensing system was slightly modified so that it generated two signals indicating the states of Ron/Roff and URon/URoff. These signals were recorded, time-stamped, and synced with the marker trajectory data by the Vicon analog-to-digital conversion system at 1000 Hz.

For resistance force control in this test, we chose a typical peak propulsion force magnitude of 140 N as the baseline (BL), and we tested GPT resistance force values of 0% BL, 20% BL, 30% BL, 40% BL, 50% BL, and 60% BL. The 0% BL condition was included to determine the residual inherent friction from the brake, and the other conditions sought to verify the full functionality of the GPT. We tested five trials for each condition.

Outcome measures in this device verification test include statistics about the magnitude of the GPT resistance force and the time delays between the various important events in the system, such as actual and detected heel strike and toe off, as well as resistance force onset and termination.

3.4.2 Pilot Study

The second hardware test served as a pilot study for how effective the GPT is in unilateral propulsion training. In this study, one healthy adult male participant (30 years) repeatedly walked along the over-ground walkway. The participant also wore reflective markers for the Vicon motion-capture system. Instead of the force sensor in the device verification study, we used the embedded in-ground force plates to measure 3D ground reaction forces, and the participant in the second test was not told their locations. The participant was instructed to walk naturally at a comfortable pace and to try to maintain the same pace throughout the experiment.

The conditions for this study are detailed in Table 3.1. In addition to the device verification procedure, the pre-training condition was added to calculate an average peak anterior ground reaction force, which we used as the baseline for the subsequent conditions. Fig. 3.8 shows the anterior-posterior ground reaction force (AP-GRF) normalized by body weight from an example step. AP-GRF in the negative (posterior) direction is the braking force, and AP-GRF in the positive (anterior) direction is the propulsion force, i.e., the reaction force from the participant pushing against the ground to propel the body forward in mid-late stance. For this subject, the baseline was 133.3 N. The post-training condition was added to find out whether walking with the GPT causes any aftereffect.

Secondly, the number of trials in each condition depended on how the participant

Table 3.1: Pilot study procedure. For number of trials, the letter designations T, R, and UR represent the numbers of total, good for resisted leg, and good for unresisted leg trials.

Condition	Subject Wearing GPT	Intended GPT Resistance Level	Number of Trials
Pre-training	No	Not applicable	6T 6R 4UR
0% BL	Yes	0 N	5T 4R 4UR
20% BL	Yes	26.7 N	5T 4R 4UR
30% BL	Yes	40.0 N	7T 4R 4UR
40% BL	Yes	53.3 N	5T 4R 4UR
50% BL	Yes	66.7 N	5T 5R 5UR
60% BL	Yes	93.3 N	6T 6R 5UR
Post-training	Yes	0 N	7T 4R 4UR

stepped on the force-plates. At the beginning of each trial, the experimenter started the participant in a position that was predicted to result in good force plate steps, which were steps squarely on at least one of the force plates and not near the edges. If one or more good force plate steps occurred with a particular limb in a trial, this trial was considered a good trial for that limb. Thus a trial could be good for both limbs or only one limb, or it could be a bad trial (i.e., no good force plate steps on either side), depending on how the steps were located. During the pilot study, each experiment condition was repeated until at least four or five good trials for each limb were obtained, and then the next experiment condition was tested. The numbers of total and good trials for each limb are shown in the fourth column of Table 3.1.

Outcome measures in this pilot study include peak propulsion forces and propulsion impulses for each leg. We also calculate propulsion impulse symmetry from average propulsion impulses across good resisted leg and unresisted leg steps within each condition (Table 3.1). Peak propulsion force is shown by the red circular marker in Fig. 3.8. Propulsion impulse, p, is calculated as the time integral of the anterior portion of the AP-GRF for each leg, which is the area of the shaded region

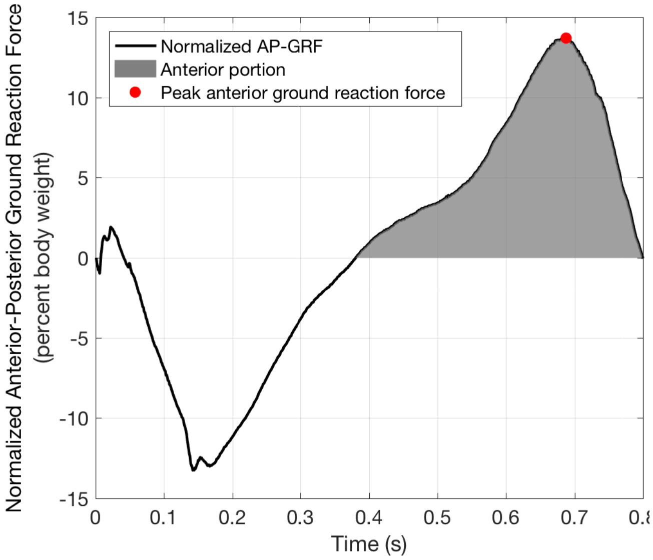


Figure 3.8: Anterior-posterior ground reaction force from one example step.

in Fig. 3.8. Propulsion impulse asymmetry, θ, is calculated as

$$\theta = \frac{\bar{p}_{\mathrm{r}} - \bar{p}_{\mathrm{nr}}}{\bar{p}_{\mathrm{r}} + \bar{p}_{\mathrm{nr}}}, \tag{3.4.1}$$

where $\bar{p}_{\mathrm{r}}$ and $\bar{p}_{\mathrm{nr}}$ represent the average propulsion impulses for the resisted and the unresisted legs from all steps during each condition. A larger value for θ indicates more asymmetric propulsion impulses from the two legs.

Given the small sample sizes in each condition, we performed the one-tailed nonparametric Fisher–Pitman permutation test for independent samples [94, 51] to evaluate performance differences.

3.5 Results

3.5.1 Device Verification

Data collected from an example trial is shown in Fig. 3.9. As the experimenter walked away from the GPT, data from the event switches beneath the soles of the shoes closely approximated the heel strike and toe off events on both legs, following the actual events extracted from Vicon trajectory data using the coordinate-based

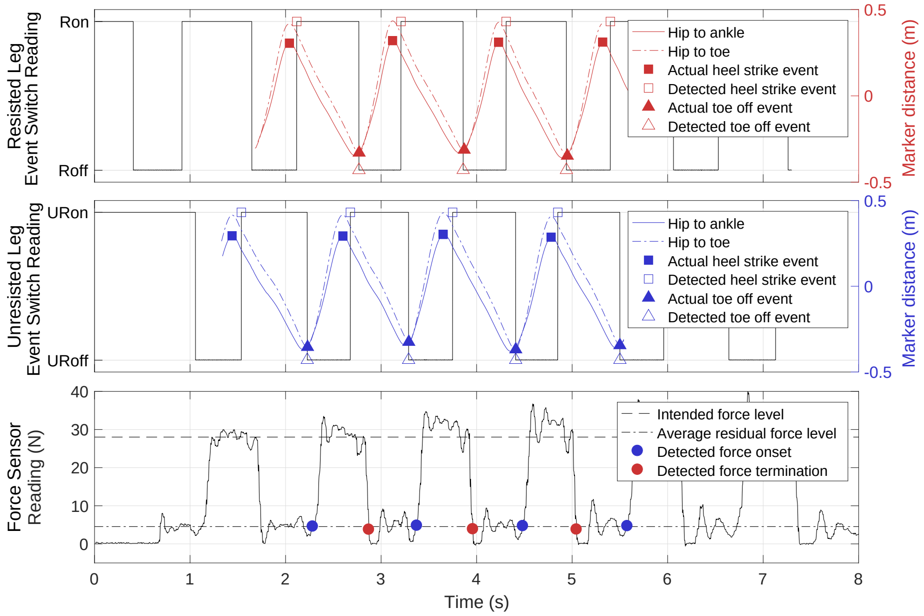


Figure 3.9: Data from an example trial in the verification test. The top panel shows readings from the event switches on the resisted leg, where 'Ron' and 'Roff' represent the resisted foot being in stance or swing phase. Horizontal distances between markers (hip to ankle and hip to toe) were extracted from Vicon trajectory data, where a positive value means that the hip marker is behind the ankle or the toe marker, and vice versa. Therefore, we use the maximum values of hip-to-ankle distance as actual heel strike events and the minimum values of hip-to-toe distance as actual toe off events, an approach validated in [121]. The center panel shows the same data for the unresisted leg. The bottom panel shows force output recorded by the force sensor. We use when the force value first exceeded the average residual force level (4.5 N) after the brake was activated (unfilled blue triangles) as the onset of resistance force, and we use when the force value first became lower than the average residual force level after the brake was deactivated (unfilled red triangles) as the termination of resistance force. The average residual force level is the 0% BL level in Table 3.3.

algorithm in [121]. In addition, Fig. 3.9 shows that the resistance force behaved as expected, turning on after the unresisted leg entered swing phase (URoff) and turning off after the resisted leg entered swing phase (Roff). The residual force when the unresisted leg was in stance phase (after detected force termination and before detected force onset) was generally small in magnitude. We believe it is caused by the inertia of the shaft and the part of the brake's inherent friction that was not canceled by the motor.

To evaluate how fast the GPT reacted to the user's motion, we calculated time

Table 3.2: Time delays between corresponding events in the device verification test.

Event	Time Delay ($\mu \pm \sigma$)
Actual and detected heel strike	$0.082\,\mathrm{s} \pm 0.019\,\mathrm{s}$
Actual and detected toe off	$-0.005\,\mathrm{s} \pm 0.013\,\mathrm{s}$
Intended and detected force onset	$0.051\,\mathrm{s} \pm 0.022\,\mathrm{s}$
Intended and detected force termination	$0.113\,\mathrm{s} \pm 0.020\,\mathrm{s}$

delays between four corresponding pairs of events as listed below; the marker descriptions in parentheses refer to Fig. 3.9:

- Actual heel strike (filled squares) and detected heel strike (unfilled squares),
- Actual toe off (filled triangles) and detected toe off (unfilled triangles),
- Intended resistance force onset (filled blue triangles) and detected resistance force onset (blue circles), and
- Intended resistance force termination (filled red triangles) and detected resistance force termination (red circles).

The results are shown in Table 3.2. Note that the actual and detected toe-off events had a negative delay on average, because the event switches attached to the toe area were not at the very front of the shoe, so they would switch from on to off slightly before the actual toe-off event. The delays for force termination were slightly larger than those for force onset, because it took some time for tension to release in the cable after the brake was deactivated. In addition, the delays for force onset and termination were larger than the delays for toe-off, because the detected toe off events had to be wirelessly transmitted to and processed by the stationary device, which took extra time.

Table 3.3 shows GPT resistance force magnitudes at different levels of intended force output. Since the brake was never activated in the 0% BL condition, we used all data when the subject was walking away from the GPT to calculate the statistics. In the other conditions, we used data when the brake was activated, as illustrated by the segments between the blue and red circles in the bottom panel of Fig. 3.9.

Table 3.3: Force magnitudes in the device verification test.

Condition	Intended GPT Resistance Level	Actual Force Output ($\mu \pm \sigma$)
0% BL	0 N	4.5 N ± 1.1 N
20% BL	28 N	29.6 N ± 2.8 N
30% BL	42 N	47.5 N ± 3.7 N
40% BL	56 N	62.5 N ± 3.8 N
50% BL	70 N	77.7 N ± 3.8 N
60% BL	84 N	85.6 N ± 4.1 N

The resistance force in the 0% BL condition has a small magnitude of 4.5 N, and the GPT resistance force control was relatively accurate in the 20% BL and 60% BL conditions. However, in the 30% BL, 40% BL, and 50% BL conditions, the actual force output was approximately 10% larger than the intended level, indicating some level of inaccuracy in our initial brake calibration.

3.5.2 Pilot Study

Fig. 3.10 shows the normalized AP-GRF recorded from force plates as the pilot study participant walked in each condition. Force magnitude is normalized by the participant's body weight, and time is normalized by total stance phase time, i.e., 0 for heel strike and 100 for toe off.

Normalized peak propulsion forces from both legs during each condition are shown in boxplots in Fig. 3.11. With higher GPT resistance, peak propulsion forces increased with the resisted leg while remaining relatively constant with the unresisted leg. Significantly larger average peak propulsion forces were found using the one-tailed nonparametric Fisher–Pitman permutation test in the resisted leg compared to the unresisted leg in the 30% BL ($p = 0.014$), 40% BL ($p = 0.014$), 50% BL ($p = 0.004$), and 60% BL ($p = 0.002$) conditions.

Normalized propulsion impulses are shown in Fig. 3.12. With higher GPT resistance, propulsion impulses from both legs increased, although the increase in the resisted leg was almost always bigger. Significantly larger average propulsion impulses

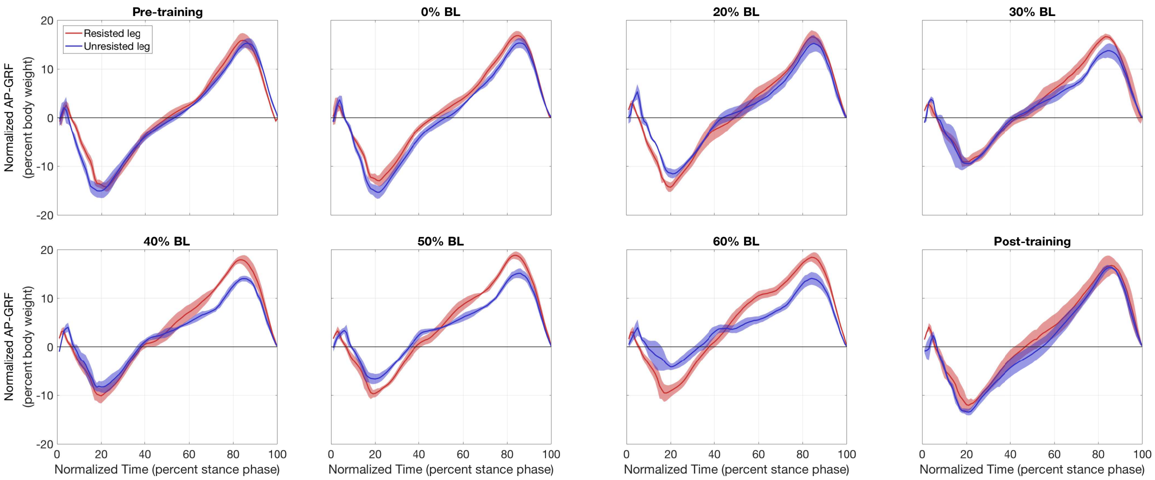


Figure 3.10: Normalized AP-GRF from the resisted and unresisted legs compared for each condition in the pilot study (see Table 3.1). Solid lines represent the mean and shaded regions represent the standard deviation for each time step.

were found using the one-tailed nonparametric Fisher–Pitman permutation test in the resisted leg compared to the unresisted leg in the pre-training ($p = 0.038$), 0% BL ($p < 0.001$), 40% BL ($p = 0.014$), 50% BL ($p = 0.008$), and 60% BL ($p = 0.002$) conditions.

Propulsion impulse asymmetry is shown in Fig. 3.13 for each condition, as calculated from eq. (3.4.1). Note that since θ is calculated using average propulsion impulses of each condition, no error bar is included. The participant started with slightly asymmetric impulses in the baseline condition. In general, higher GPT resistance level caused larger propulsion impulse asymmetry, although the relationship is not monotonic. Compared to the baseline condition, the impulse asymmetry level more than doubled in the post-training condition.

3.6 Discussion

The device verification test shows that the GPT reacted quickly to the user's gait events and turned the resistance force on and off with correct timing. The magnitudes of the resistance forces were well controlled, as shown by their small standard deviations, but the GPT's force output should be re-calibrated to make sure the applied resistance matches the intended magnitude. This re-calibration can be easily

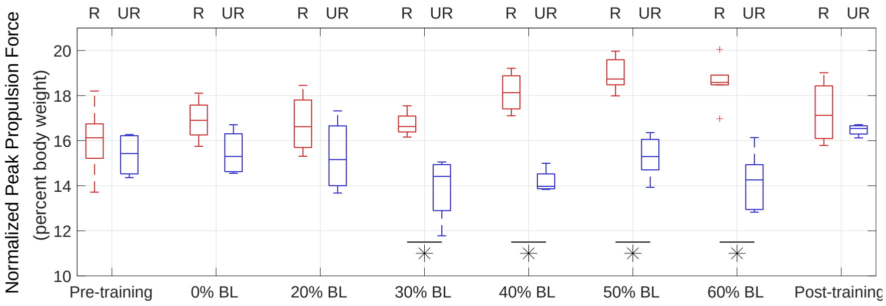


Figure 3.11: Normalized peak propulsion forces during the pilot study. 'R' and 'UR' represent the resisted and the unresisted legs. The central lines indicate the median, and the bottom and top edges of the boxes indicate the 25th and 75th percentiles, respectively. A star and a horizontal line below boxplots show that the mean of data from R is significantly higher than the mean of data from UR.

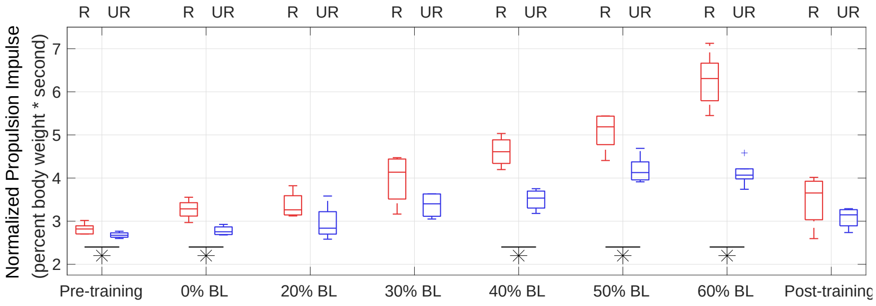


Figure 3.12: Propulsion impulses during the pilot study.

performed and will not affect any other aspect of the GPT's function.

The pilot study results show the effectiveness of the GPT in training unilateral propulsion. Unilateral resistance intervention caused the pilot participant to exhibit greater peak propulsion forces and propulsion impulses from the resisted leg than the unresisted leg during training. In addition, compared to the baseline condition, the propulsion asymmetry more than doubled in the post-training condition. However, we didn't find significant differences of peak propulsion forces or propulsion impulses in the post-training condition.

We hypothesized that propulsion impulses from the unresisted leg would not change during training, which was not the case in the pilot study. One possible explanation is that as GPT resistance increased, the participant walked more carefully and slowly (see Fig. 3.14), and times spent in anterior portion of stance phase in each step for both legs were longer (see Fig. 3.15). As a result, the propulsion

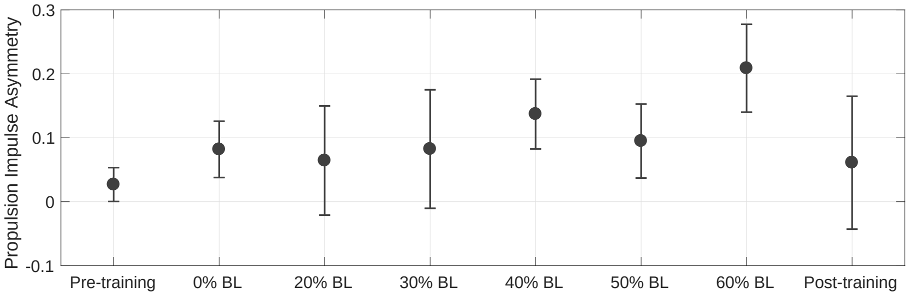


Figure 3.13: Propulsion impulse asymmetry levels during the pilot study. The error bars indicate ±1 propagated standard deviation from eq. (3.4.1).

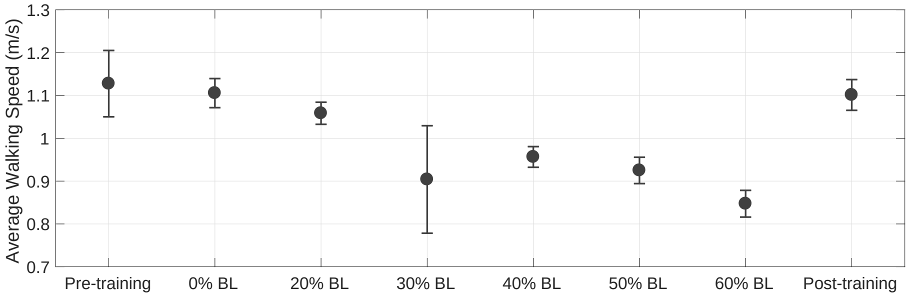


Figure 3.14: Average walking speed during the pilot study. The error bars indicate ±1 standard deviation.

impulse (the integral of propulsion force with respect to time) of the unresisted leg increased despite the peak resistance force magnitudes only slightly increased or even decreased in certain conditions (see Fig. 3.11).

Although propulsion asymmetry generally increased with higher GPT resistance, the relationship was not monotonic. A possible indication from this fact is that walking with the highest GPT resistance force tolerable may not yield the most favorable results. It is likely that the appropriate magnitude of GPT resistance to produce optimal gains in propulsion impulse symmetry is unique to the individual.

One side effect to note is that the braking force seemed to decrease, especially in the unresisted leg in the 50% and 60% BL conditions, as shown in Fig. 3.10. In the current control scheme, the GPT resistance would be applied until resisted side toe-off, so the brake helps with breaking on the resisted side, which may have caused the reduction in braking forces. Nonetheless, the participant did not experience any discomfort or danger because of the reduced braking force in our pilot study.

In addition, this pilot study was performed with a healthy young adult. Ad-

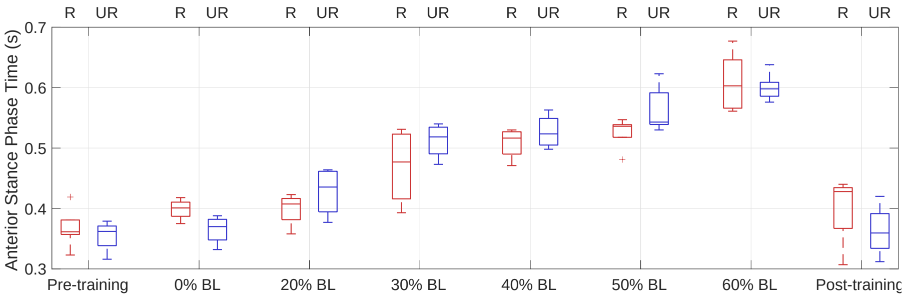


Figure 3.15: Times of anterior stance phase for each leg during the pilot study.

ditional evaluation with more participants could obtain more evidence regarding increased propulsion impulse from the unresisted leg as well as the relationship between GPT resistance and the resultant impulse propulsion asymmetry. More importantly, we postulate that for people with stroke who have propulsion impulse asymmetry in the baseline condition (hemiparesis from stroke often causes smaller propulsion forces on the paretic leg compared to the non-paretic leg), applying GPT resistance during paretic leg stance phase would lead to increased paretic side propulsion impulse during training and consequently improved propulsion impulse symmetry post-training. Clinical evaluations would be able to test our hypothesis.

3.7 Summary

In this chapter, we designed and evaluated a gait rehabilitation device based on the hybrid mechanism in Chapter 2. A wireless gait phase detection system was added, which enables unilateral training for people with hemiparesis, and the passive, high-magnitude force output from the brake and the cable transmission offer the unique resistance training during stance phase. Efforts in the controller aim to make the GPT as safe as possible even when sensors or actuators malfunction.

Note that although the GPT does not use a sonar sensor to measure the user's position, we still require the user to walk straight away from the GPT and remain in front of the device, as in Chapter 2. This is because if the user walks at an angle α, as illustrated in Fig. 3.16, the cable tension will be $T_{\text{actual}} = \frac{\tau}{r\cos\alpha}$, where $\tau = T_{\text{des}}r$ is the brake's output torque and r is the radius of the spool, so that the tension would be higher than the desired value ($T_{\text{actual}} = {}^{T_{\text{des}}}/_{\cos\alpha}$). However, this

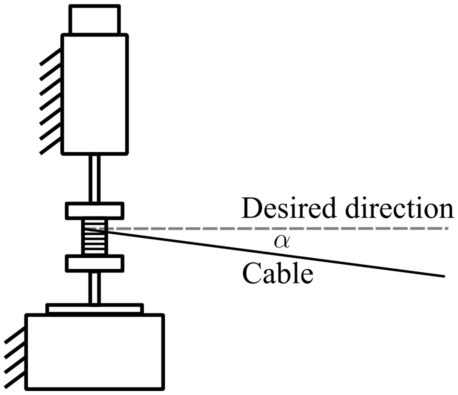


Figure 3.16: Top view of the GPT when the user walks at an angle from the desired direction.

effect (the angle α) could be eliminated by placing the brake-spool-motor assembly vertically, in which case the cable will be perpendicular to the shaft and the spool axis as long as the spool is set at the height of the subject's pelvis.

The verification test and the pilot study show the effectiveness of the GPT to induce increased propulsion forces from the resisted leg in a timely manner, and we anticipate significant changes after repeated training with the GPT. Such efforts are ongoing with our collaborators at Stony Brook University and will be published when ready. Overall, given its effectiveness, low cost, and ease of use, the GPT may be an advantageous choice for people with hemiparesis to train at home.

On the other hand, there is room for clarification and improvement in the current design. For example, we hypothesized that the GPT would not affect the propulsion forces on the unresisted leg, which was not the case in the pilot study with the healthy subject. Second, it is still unclear how much resistance and how much repetition (the dosage of rehabilitation) is optimal with the GPT. For both of these concerns, more data is required to fully uncover the underlying interaction between GPT resistance and rehabilitation outcome.

Finally, in addition to the dosage, there are a few parameters, such as rising time and the timing of resistance force onset and termination, that could affect the performance of rehabilitation. A systematic user study with a large number of participants may offer more insights into the design of these parameters, but it may also be advantageous if they could be automatically obtained and customized from a few observations or demonstrations from the user about their impairment level or preference. Therefore, in the next two chapters we focus on customization of robot behaviors for physical assistive tasks.

Chapter 4

Hierarchical Structure of Learning from Demonstration

Thanks to their high number of degrees of freedom, humanoid robots and robotic manipulators can offer richer and more complex interactions with humans than simpler mechatronics systems. Moreover, they are equipped with better sensing and computation, which enable them to obtain data, process it, and then update their behavior accordingly in near realtime. As we have discussed in Chapter 3, customizing robot behavior from observations of the human user could drastically simplify the design process, and humanoid robots with superior sensing and computation become a good platform for this customization purpose.

Thus we developed a hierarchical structure of task-parameterized learning from demonstration (LfD), where users can intuitively teach the robot what to do for their task at hand. In the learning process, demonstrations for each task situation are annotated by a set of task parameters, and such parameters help select only relevant training data at test time and therefore improve generalization performance. Other advantages of the hierarchical structure, compared to the conventional approach that encodes all demonstrations together, includes the ability to encode a wider range of task situations in the same framework and reduced computational load when adding or removing demonstrations, as shown in a simple simulated task and in a human-subject study where naive users collaborated with a Willow Garage PR2 and moved an object together. In this chapter, we focus on designing the hierarchical structure so that the robot would learn from limited user demonstrations and adapt well to new test situations, and thus we use an indirect Cartesian PID controller for regulating physical interactions on the PR2. This work was first reported in Hindawi Applied Bionics and Biomechanics, special issue Analysis of Human Behavior for Robot Design and Control [40].

After discussing related work in Section 4.1, we give a brief introduction to the task-parameterized Gaussian mixture model (TP-GMM), which we use as the basis of our hierarchical structure, and we illustrate some peculiar behaviors from the model in Section 4.2. Then we detail our hierarchical structure in Section 4.3, and we validate it in a simulated trajectory-retrieval task compared to the vanilla TP-GMM. Section 4.4 describes a human-subject study with the PR2, where results show that users can differentiate between LfD models and that our framework provide advantages in both quantitative and qualitative metrics. Section 4.5 summarizes this chapter.

4.1 Related Work

As opposed to having an operator devise control policies and reprogram the robot for every new situation it encounters, learning from demonstration (LfD, also known as programming by demonstration, PbD) provides a direct method for robots to learn and replicate human behaviors [14, 9]. LfD control policies are learned from demonstrations in which a human teacher controls the robot to accomplish the task. Various learning algorithms are suitable for encoding interactions recorded during demonstrations, such as hidden Markov models (HMMs) [23] and hidden semi-Markov models (HSMMs) [120]. By extending the HSMM framework, Rozo et al. enabled the robot to be proactive if the partner does not follow the demonstrations, which were encoded by the observed temporal patterns and sequential information [99]. Dynamic motion primitives (DMPs) [42] provide another framework for interaction encoding, for example learning an adaptive, sensor-driven interaction between two coupled agents [58]. Instead of learning/placing basis functions for the forcing term, Pervez et al. presented a DMP-based method that accommodates spatial and temporal variations in demonstrations, different initial and final conditions, and partial executions by directly encoding the phase variable and forcing term value in a Gaussian mixture model (GMM) and synthesizing the forcing term at test time using Gaussian mixture regression (GMR) [92].

Another promising learning framework is using a GMM and GMR directly on the demonstrated trajectories, where multiple channels of information (e.g., position and velocity of the robot gripper) are encoded jointly by a GMM. The conditional probability density function of the outputs on the inputs can be calculated and used in GMR for a wide range of applications such as trajectory retrieval [4]. The task-parameterized GMM (TP-GMM) framework utilizes task parameters to annotate demonstrations, and it allows generalization to undemonstrated situations by manipulating the demonstrated data with respect to the undemonstrated task parameters [21]. Rozo et al. used the TP-GMM framework for human-robot collaborative tasks, additionally modeling the physical interactions with an impedance model that has unit mass and constant damping while stiffness is learned from data [98]. In a similar spirit, Pervez and Lee developed task-parameterized DMP (TP-DMP) to include such task parameters in a mixture of GMMs [93], extending the GMM that previously encoded only the phase variable and the forcing term value in [92].

In addition to utilizing various learning frameworks, LfD approaches create the opportunity for the robot to determine when new demonstrations are needed, thus avoiding poor or even dangerous actions. Uncertainty in generalized trajectories or cost functions has been used as a trigger for requesting demonstrations, where uncertainty can be calculated from, e.g., Query by Bagging [104] or a Gaussian Process (GP) [73]. Chernova and Veloso used confidence in execution to detect unfamiliar or ambiguous states that require new demonstrations [24]. In another approach, a GMM gating model that is based on observed human motions determines whether the test task is likely to be contained in the Interaction ProMPs that the robot has already learned, or whether new demonstrations are necessary [55]. In these approaches, the new test is compared to what the robot has experienced, and new demonstrations are requested when the robot deems it necessary. On the other hand, Abi-Farraj et al. considered generalized trajectories for refining the learned distribution via an information gain threshold, so that the robot does not need to request additional demonstrations [2].

Many approaches aim to build skill libraries from demonstrations. For example, Muelling et al. learn a library of DMPs from demonstrations for table tennis, and at test time the output control policy is a weighted average of the DMP skills generated from a kernel function on input stimuli (hitting position and velocity) and weight parameters that prioritize certain skills obtained using reinforcement learning [83]. Since a weighted average was used as a means for generalization, predicting task performance for tests was difficult, i.e., the combination of a set of good demonstrations may not necessarily result in good behavior, and thus reinforcement learning was necessary to prune or prioritize certain skills in the DMP library [83]. As a result, when a new motion primitive is added or an existing one is deleted, the library may need to be re-trained for it to converge again. In contrast, online learning is achieved in [55] by incrementally building the GMM gating model and the Interaction ProMPs, while the gating model selects only one most likely model for generalization. Other examples that incrementally build skill libraries from new demonstrations include [57, 64, 22]. Existing methods that incrementally train GMMs have also been adapted for TP-GMMs [37].

4.2 Overview of the Task-Parameterized Gaussian Mixture Algorithm

We use TP-GMM [21] as the basis of our framework. The general approach encodes demonstration trajectories into a model with help of task parameters, and at test time, new trajectories can be retrieved efficiently even if the robot has never seen that particular task instance before. TP-GMM has been used to enable a robot to learn collaborative beam lifting [30] as well as object transportation and chair assembly [98]; its typical implementation, which we call vanilla TP-GMM (VT), has shown good generalization capabilities in these applications.

In the following sections, we use task situations to denote particular instances of a task: for example, in the object-moving task, moving from point A to point B and moving from point A to point C are two different task situations. In the context of TP-GMM, the task parameters fully define a task situation.

4.2.1 The TP-GMM Algorithm

The n-th demonstration ($n = 1, 2, ..., N$) contains L_n data points ($\{\boldsymbol{\xi}_{n,l}\}_{l=1}^{L_n}$), and each data point may have dimensions of time, position, velocity, etc., at a given time step. The task parameters ($\mathbf{p}$) are defined as P affine transformations ($\{\mathbf{A}_p, \mathbf{b}_p\}_{p=1}^{P}$) that include information about the task situation (e.g., poses of the start and goal

Algorithm 1 Given demonstrations $\{\{\boldsymbol{\xi}_{n,l}\}_{l=1}^{L_n}\}_{n=1}^{N}$ for N situations $\{\mathbf{p}_{\text{train,n}}\}_{n=1}^{N}$ and test situation $\mathbf{p}_{\text{test}}$, find a trajectory $\boldsymbol{\xi}^{\mathcal{O}}$ for the test situation using vanilla TP-GMM.

```
procedure VT-Train($\{\{\boldsymbol{\xi}_{n,l}\}_{l=1}^{L_n}\}_{n=1}^{N}, \{\mathbf{p}_{\text{train,n}}\}_{n=1}^{N}$)
    $\mathbf{\Pi} = \textsc{Encode}(\{\{\boldsymbol{\xi}_{n,l}\}_{l=1}^{L_n}\}_{n=1}^{N}, \{\mathbf{p}_{\text{train,n}}\}_{n=1}^{N})$
    return $\mathbf{\Pi}$
procedure VT-Test($\mathbf{\Pi}, \mathbf{p}_{\text{test}}$)
    $\boldsymbol{\pi} = \textsc{Decode}(\Pi, \mathbf{p}_{\text{test}})$
    $\boldsymbol{\xi}^{\mathcal{O}} = \text{GMR}(\boldsymbol{\pi}, \boldsymbol{\xi}^{\mathcal{I}})$
    return $\boldsymbol{\xi}^{\mathcal{O}}$
```

frames). In addition, task parameters need to be compatible with the data; for example, if each data point contains the instantaneous 3D Cartesian position and velocity of the robot gripper, i.e., $\boldsymbol{\xi} = \begin{bmatrix}\mathbf{x}^\top & \mathbf{v}^\top\end{bmatrix}^\top$, one can define $\mathbf{A}_p = \begin{bmatrix}\mathbf{R}_p & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{R}_p\end{bmatrix}$ and $\mathbf{b}_p = \begin{bmatrix}\mathbf{r}_p^\top & \mathbf{0}_{1\times3}\end{bmatrix}^\top$, where $\mathbf{R}_p$ and $\mathbf{r}_p$ represent the orientation and position of the p-th relevant reference frame, so that matrix operations such as $\mathbf{A}_p^{-1}(\boldsymbol{\xi} - \mathbf{b}_p)$ are valid and physically meaningful.

Alg. 1 includes a brief overview of the TP-GMM algorithm, with involved functions defined in Alg. 2. The training step contains only the Encode function, which transforms data points into each task frame and then fits a Gaussian mixture model $\mathbf{\Pi} = \{\pi_m, \{\boldsymbol{\mu}_m^{(p)}, \mathbf{\Sigma}_m^{(p)}\}_{p=1}^{P}\}_{m=1}^{M}$, where M is the number of Gaussian clusters, π_m is the mixture coefficient, and $\{\boldsymbol{\mu}_m^{(p)}, \mathbf{\Sigma}_m^{(p)}\}$ are the Gaussian mean and covariance matrices of the m-th cluster in the p-th task frame. The FitGMM function is close to the standard procedure in fitting a GMM with expectation maximization. The testing step includes two functions: Decode transforms each Gaussian cluster in $\mathbf{\Pi}$ according to the test task parameters and generates a GMM $\boldsymbol{\pi}$ in the global frame, and GMR computes a trajectory for the test situation, which can be used as controller commands. For example, $\boldsymbol{\xi}^{\mathcal{O}}$ could be the velocity and $\boldsymbol{\xi}^{\mathcal{I}}$ the position of a robot gripper, i.e., during automatic execution, the robot could derive the desired velocity given its current position.

Algorithm 2 Involved functions used in TP-GMM.

```
procedure ENCODE($\{\{\boldsymbol{\xi}_{n,l}\}_{l=1}^{L_n}\}_{n=1}^{N}, \{\mathbf{p}_{\text{train,n}}\}_{n=1}^{N}$)
    for $n = 1$ to $N$ do
        for $l = 1$ to $L_n$ do
            for $p = 1$ to $P$ do
                $\mathbf{X}_{n,l}^{(p)} = \mathbf{A}_{\text{train},n,p}^{-1}(\boldsymbol{\xi}_{n,l} - \mathbf{b}_{\text{train},n,p})$
            $\mathbf{X}_{n,l} = \begin{bmatrix} \mathbf{X}_{n,l}^{(1)\top} & \mathbf{X}_{n,l}^{(2)\top} & ... & \mathbf{X}_{n,l}^{(P)\top} \end{bmatrix}^\top$
    $\boldsymbol{\Pi} = \{\pi_m, \{\boldsymbol{\mu}_m^{(p)}, \boldsymbol{\Sigma}_m^{(p)}\}_{p=1}^{P}\}_{m=1}^{M}$
      $= \text{FITGMM}(\{\{\mathbf{X}_{n,l}\}_{l=1}^{L_n}\}_{n=1}^{N})$
    return $\boldsymbol{\Pi}$
procedure DECODE($\boldsymbol{\Pi}, \mathbf{p}_{\text{test}}$)
    for $m = 1$ to $M$ do
        for $p = 1$ to $P$ do
            $\hat{\boldsymbol{\mu}}_m^{(p)} = \mathbf{A}_{\text{test},p}\boldsymbol{\mu}_m^{(p)} + \mathbf{b}_{\text{test},p}$
            $\hat{\boldsymbol{\Sigma}}_m^{(p)} = \mathbf{A}_{\text{test},p}\boldsymbol{\Sigma}_m^{(p)}\mathbf{A}_{\text{test},p}^\top$
        $\boldsymbol{\Sigma}_m = \left(\sum_{p=1}^{P} \hat{\boldsymbol{\Sigma}}_m^{(p)^{-1}}\right)^{-1}$
        $\boldsymbol{\mu}_m = \boldsymbol{\Sigma}_m \sum_{p=1}^{P} \hat{\boldsymbol{\Sigma}}_m^{(p)^{-1}} \hat{\boldsymbol{\mu}}_m^{(p)}$
    $\boldsymbol{\pi} = \{\pi_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}_{m=1}^{M}$
    return $\boldsymbol{\pi}$
procedure GMR($\boldsymbol{\pi}, \boldsymbol{\xi}^{\mathcal{I}}$)
    for $m = 1$ to $M$ do
        $\boldsymbol{\mu}_m = \begin{bmatrix} \boldsymbol{\mu}_m^{\mathcal{I}\top} & \boldsymbol{\mu}_m^{\mathcal{O}\top} \end{bmatrix}^\top$
        $\boldsymbol{\Sigma}_m = \begin{bmatrix} \boldsymbol{\Sigma}_m^{\mathcal{I}} & \boldsymbol{\Sigma}_m^{\mathcal{I}\mathcal{O}} \\ \boldsymbol{\Sigma}_m^{\mathcal{O}\mathcal{I}} & \boldsymbol{\Sigma}_m^{\mathcal{O}} \end{bmatrix}$
        $\hat{\boldsymbol{\mu}}_m = \boldsymbol{\mu}_m^{\mathcal{O}} + \boldsymbol{\Sigma}_m^{\mathcal{O}\mathcal{I}}\boldsymbol{\Sigma}_m^{\mathcal{I}^{-1}}(\boldsymbol{\xi}^{\mathcal{I}} - \boldsymbol{\mu}_m^{\mathcal{I}})$
        $h_m = \frac{\pi_m \mathcal{N}(\boldsymbol{\xi}^{\mathcal{I}}|\boldsymbol{\mu}_m^{\mathcal{I}}, \boldsymbol{\Sigma}_m^{\mathcal{I}})}{\sum_{m'=1}^{M} \pi_{m'} \mathcal{N}(\boldsymbol{\xi}^{\mathcal{I}}|\boldsymbol{\mu}_{m'}^{\mathcal{I}}, \boldsymbol{\Sigma}_{m'}^{\mathcal{I}})}$
    $\boldsymbol{\xi}^{\mathcal{O}} = \sum_{m=1}^{M} h_m \hat{\boldsymbol{\mu}}_m$
return $\boldsymbol{\xi}^{\mathcal{O}}$
```

TP-GMM exploits locally consistent features among demonstrations in each task frame (ENCODE) and transforms them according to new situations (DECODE), generally yielding reliable performance for both interpolation and extrapolation in many applications [30, 98]. However, TP-GMM doesn't have an explicit estimate of how well the local information would perform with respect to the new task situation in GMR. For example, it may have poor generalization when the new task situation is too different from what has been demonstrated. In addition, GMR may have trouble even when the test situation is exactly the same as a demonstrated one, depending on the quality and consistency of the demonstrated data. These potential issues are illustrated next.

4.2.2 TP-GMM with a Simplistic Task

Suppose the task of interest is moving from a known start position to a known goal position on a one-dimensional line, and the strategy that generates demonstrations uniformly connects the start and goal points in 100 time steps, as shown in Fig. 4.1. In this example, data points $\boldsymbol{\xi} = \begin{bmatrix} t & x \end{bmatrix}^\top$, where $t = 1, 2, ..., 100$ is the time step and x is the coordinate. Consequently, we include the start and the goal task frames, and for each we have $\mathbf{A} = \mathbf{I}_2$ and $\mathbf{b} = \begin{bmatrix} 0 & r \end{bmatrix}^\top$, where r represents the location of the frame. Without loss of generality, the start positions of all task situations are at $x = 0$.

Using vanilla TP-GMM with $M = 3$ in Alg. 1, we can generalize trajectories for different test situations ($\boldsymbol{\xi}^{\mathcal{I}} = t$ and $\boldsymbol{\xi}^{\mathcal{O}} = x$), as shown in Fig. 4.2. We use 3 clusters in this simulation because they generally cover the trajectories well and require little time to train, but other numbers work as well.

It can be seen that the generalized trajectories cover the distance between the start and goal positions well (Fig. 4.2a), but they no longer contain uniform step lengths (Fig. 4.2b). If the start and goal are close to each other (e.g., demonstration 1 in Fig. 4.2), the generalized trajectory may even reverse the direction of motion at time values around 30 and 70, where the dominant Gaussian cluster changes. On the other hand, if the start and goal are far from each other (e.g., test 3 in Fig. 4.2), the generalized trajectory may cause very large velocities at the same time points. Inconsistencies such as reversed motion direction and high magnitude velocity may cause instabilities on real hardware and might seem like faulty behaviors to a naive user. Another inconsistency is that the generalized first and last trajectory points don't always align with the prescribed start and goal positions, as shown in Fig. 4.2b, which may cause jumps at the beginning and end of autonomous execution. Importantly, although the types and locations of the inconsistencies can

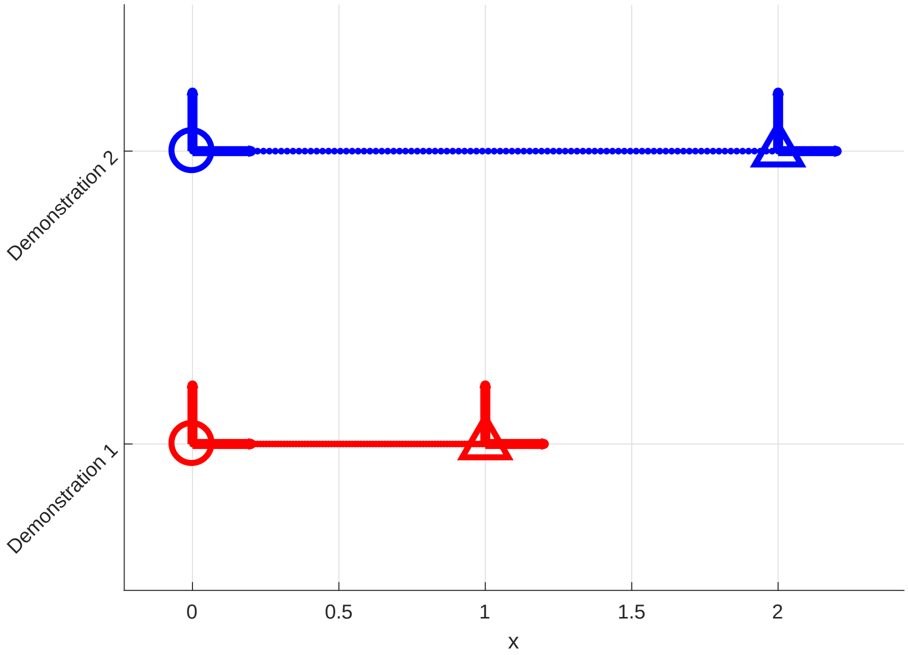
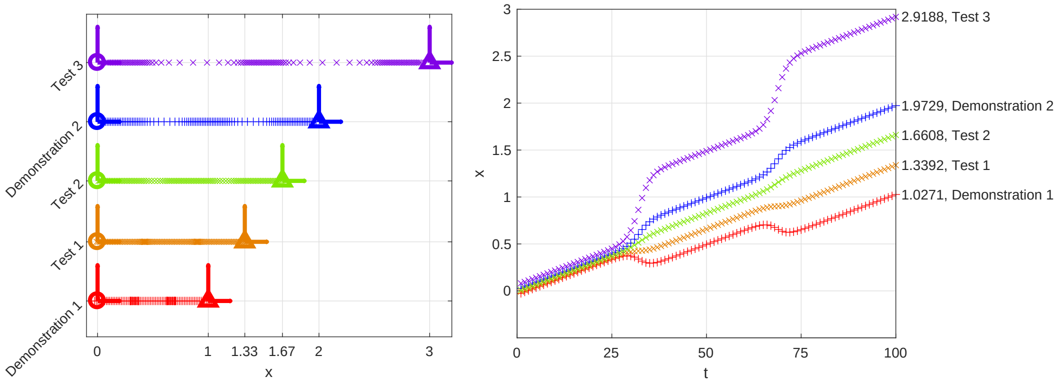


Figure 4.1: Demonstrated trajectories for a simplistic movement task. Circles represent the start positions and triangles represent the goal positions of the two demonstrated task situations.

vary in different implementations, their existence is not specific to the task or the number of clusters used in simulation.

What causes this difficulty with generalization? TP-GMM can be viewed as being similar to a regression algorithm: task parameters are the independent variables, and trajectories are generalized from local information in each task frame. Consequently, the exact information from each individual training point may be lost (e.g., demonstration 1 in Fig. 4.2). In addition, because the function that maps task parameters to trajectories is highly nonlinear, the regression model does not have enough information to accurately generalize for a test point outside of the trained region (e.g., test 3 in Fig. 4.2). Vanilla TP-GMM can have good generalization properties (test 2 in Fig. 4.2), but it doesn't offer an estimate of generalized performance and thus cannot differentiate between tests 1, 2, and 3. Instead, it has to rely on the robot controller to handle the potential peculiarities of the generated trajectory. Additionally, because information from demonstrations is stored locally with respect to each task frame, the original global strategy (in this case, connecting start and goal positions with uniform step lengths) is largely lost after modeling in



(a) Generalized trajectories.

(b) Generalized trajectories over time.

Figure 4.2: Generalized trajectories for the two demonstrated situations and three new test situations using vanilla TP-GMM.

vanilla TP-GMM.

4.3 Hierarchical Task-Parameterized Learning from Demonstration

To preserve the powerful generalization capability of TP-GMM and overcome the previously discussed shortcomings, we developed a hierarchical structure (HS) that explicitly reasons about task parameters using three utility functions in the following steps.

First, we define a distance function that operates on a pair of task situations and outputs a scalar value, representing how similar the two situations are to each other. We argue that this scalar value can serve to estimate test performance, which can then be a trigger for requesting demonstrations. Second, sets of demonstrations associated with the same task parameters are each encoded as their own TP-GMM. Third, given a test situation, we use the distance function to the TP-GMM from only the most similar situation. In addition, with only one situation per TP-GMM, we can manipulate the Gaussian clusters to enhance generalization with two morphing functions, because the exact information and strategy from that individual training data set are preserved. Finally, the hierarchical structure makes it straightforward to encode a large variety of task situations in the same framework.

This section details the steps in the hierarchical structure, empirically validates

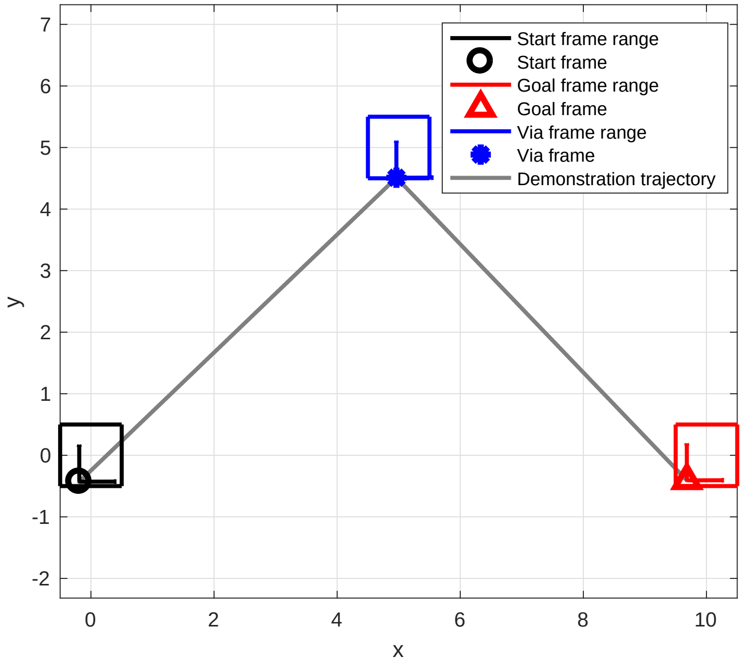


Figure 4.3: An example task situation and its trajectory. The frame ranges represent the regions from where each task frame is randomly sampled.

the distance function as a test performance estimator, and shows the improvements of our approach compared to vanilla TP-GMM for a simulated movement task.

4.3.1 Simulated Task Definition

We use a movement task with three task frames, representing the start $(\mathbf{R}_1, \mathbf{r}_1)$, goal $(\mathbf{R}_2, \mathbf{r}_2)$, and via $(\mathbf{R}_3, \mathbf{r}_3)$ points on a two-dimensional plane. Task parameters for these frames include rotation matrices with x-axes parallel to a vector pointing from the start to the goal, z-axes pointing out of the page, and y-axes following the right-hand rule. The demonstration strategy uniformly connects the start and via points using a straight line with 100 time steps and then uniformly connects the via and goal points using a straight line with another 100 time steps ($\boldsymbol{\xi} = \begin{bmatrix} t & x & y \end{bmatrix}^\top$, where $t = 1, 2, ..., 200$), as shown in Fig. 4.3. We call a unique specification of the frame ranges a *task configuration*.

We use this example task to instantiate the utility functions and the evaluation procedure in the following subsections. Nevertheless, the hierarchical structure can also be used in other tasks and/or with different data dimensions (such as in

Section 4.4).

4.3.2 Distance Function

For the example task, we define the distance function in Alg. 3 for two task situations $(\mathbf{p}_A, \mathbf{p}_B)$: the task frames of the compared situations are transformed into their start frames, and the distance function value is calculated as the sum of squares of the distances between the corresponding goal and via points. Note that there may be many possible definitions for the distance function; for example, one could also choose to include a norm on the rotation matrices as well. This particular definition comes from understanding that TP-GMM aligns clusters in each task frame, and thus the task frames' positions with respect to each other are more important than their absolute positions in the world frame.

Algorithm 3 Distance function.

```
procedure DISTANCE($\mathbf{p}_A, \mathbf{p}_B$)
    $d = 0$
    for $j = 2$ to $3$ do
        $d = d + \left|\left|\mathbf{R}_{1,A}^{\top}(\mathbf{r}_{j,A} - \mathbf{r}_{1,A}) - \mathbf{R}_{1,B}^{\top}(\mathbf{r}_{j,B} - \mathbf{r}_{1,B})\right|\right|_2^2$
    return $d$
```

To understand the distance function, consider the two-frame task in Section 4.2.2, where Alg. 3 would simply calculate the square of the distances between the goals of each situation. In particular, $\textsc{Distance}(\mathbf{p}_{\text{test},3}, \mathbf{p}_{\text{train},1}) = (3-1)^2 = 4$, while $\textsc{Distance}(\mathbf{p}_{\text{test},3}, \mathbf{p}_{\text{train},2}) = (3-2)^2 = 1$. Therefore, test 3 would be considered more similar to demonstration 2 than it is to demonstration 1.

4.3.3 Situation and GMM Morphing

Similar to using DMP on generalized trajectories from a GP to ensure that the prescribed goals are reached [73], we introduce a generalization-enhancing strategy that is specific to the movement task: if the start and goal points become farther or closer to each other, then the trajectory can be proportionately stretched or

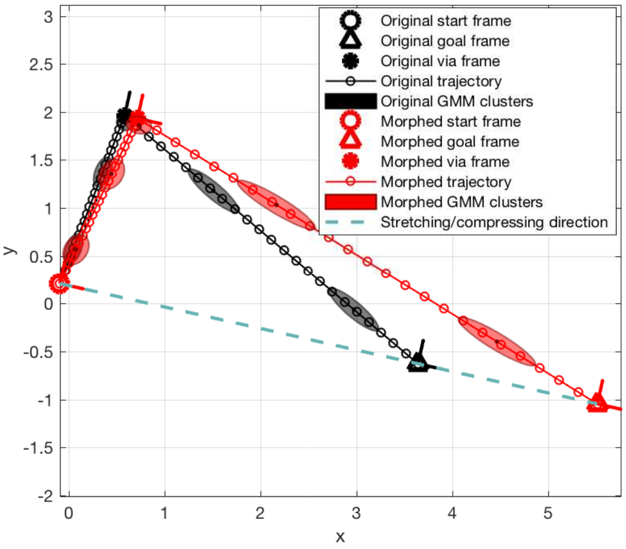


Figure 4.4: Illustration of eq. (4.3.1) and the morphing functions in Alg. 4. The original task frames are randomly sampled, and the original trajectory and GMM clusters are generated from the demonstration strategy and trained accordingly. The morphed frames, trajectory, and GMM clusters are calculated from the original ones with $\alpha = 1.5$. Note that the original and morphed start frames coincide.

compressed in the start-goal direction to accommodate the change:

$$\begin{aligned}
\mathbf{T} &= \frac{(\mathbf{r}_2 - \mathbf{r}_1)(\mathbf{r}_2 - \mathbf{r}_1)^\top}{(\mathbf{r}_2 - \mathbf{r}_1)^\top(\mathbf{r}_2 - \mathbf{r}_1)}, \\
\mathbf{r} &= \begin{bmatrix} x & y \end{bmatrix}^\top, \\
\mathbf{r}' &= \mathbf{r}_1 + \alpha\mathbf{T}(\mathbf{r} - \mathbf{r}_1) + (\mathbf{I} - \mathbf{T})(\mathbf{r} - \mathbf{r}_1) \\
&= (\mathbf{I} + \alpha\mathbf{T} - \mathbf{T})\mathbf{r} - (\alpha - 1)\mathbf{T}\mathbf{r}_1,
\end{aligned} \tag{4.3.1}$$

where $\mathbf{T}$ is the projection operator along a unit vector from start ($\mathbf{r}_1$) to goal ($\mathbf{r}_2$), $\mathbf{r}$ is an arbitrary point of the original trajectory, α is the scalar value representing the extent of stretching or compression, and $\mathbf{r}'$ is the proportionately changed new trajectory point, as shown in Fig. 4.4.

Consequently, we can generate task parameters and Gaussian clusters for the new situation in the same manner. Alg. 4 details how this process can be carried out in accordance with the current definition of $\boldsymbol{\xi} = \begin{bmatrix} t & x & y \end{bmatrix}^\top$ (e.g., see definitions of $\mathbf{T}'$ and $\mathbf{T}''$), where $\mathbf{T}$ is defined as in eq. (4.3.1). The morphed task situation and GMM clusters are plotted in Fig. 4.4.

Algorithm 4 Utility functions that morph task situations and Gaussian clusters in GMMs.

```
procedure MorphTask($\mathbf{p}, \alpha$)
    for $i = 1$ to $3$ do
        $\mathbf{R}'_i = \mathbf{R}_i$
        $\mathbf{r}'_i = (\mathbf{I} + \alpha\mathbf{T} - \mathbf{T})\mathbf{r}_i - (\alpha - 1)\mathbf{T}\mathbf{r}_1$
        $\mathbf{A}'_i = \begin{bmatrix} 1 & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{2\times 1} & \mathbf{R}'_i \end{bmatrix}$
        $\mathbf{b}'_i = \begin{bmatrix} 0 & \mathbf{r}'^{\top}_i \end{bmatrix}^{\top}$
    return $\mathbf{p}' = \{\mathbf{A}'_i, \mathbf{b}'_i\}_{i=1}^{3}$
procedure MorphGMM($\pi, \alpha, \mathbf{p}$)
    $\mathbf{T}' = \begin{bmatrix} 1 & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{2\times 1} & \mathbf{I} + \alpha\mathbf{T} - \mathbf{T} \end{bmatrix}$
    $\mathbf{T}'' = \begin{bmatrix} 1 & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{2\times 1} & (\alpha - 1)\mathbf{T} \end{bmatrix}$
    for $m = 1$ to $M$ do
        $\boldsymbol{\mu}'_m = \mathbf{T}'\boldsymbol{\mu}_m - \mathbf{T}''\mathbf{b}_1$
        $\boldsymbol{\Sigma}'_m = \mathbf{T}'\boldsymbol{\Sigma}_m\mathbf{T}'^{\top}$
    $\boldsymbol{\pi}' = \{\pi_m, \boldsymbol{\mu}'_m, \boldsymbol{\Sigma}'_m\}_{m=1}^{M}$
    return $\boldsymbol{\pi}'$
```

4.3.4 Hierarchical Structure

With the utility functions defined in Algs. 3 and 4, our hierarchical structure for TP-GMM is shown in Alg. 5. Demonstrations from each situation are encoded in separate TP-GMMs ($\boldsymbol{\Pi}_n$) in the training step. The test situation is maximally matched with each demonstrated situation (the $\arg\min$ step in the **for** loop), and we select the overall best match $n^{\star}$ for generalization in the Decode function in the test step if the matched result $d_{n^{\star}}$ is below a prescribed threshold. If the threshold is exceeded, new demonstrations should be requested. This process serves as a gating function, similar to the one in [73]. Finally, the Gaussian clusters in the generated world-frame GMM $\boldsymbol{\pi}^{\star}$ are inversely morphed with $\alpha_{n^{\star}}^{-1}$ to ensure that the final GMM

$\boldsymbol{\pi}$ is compatible with and applicable to the actual, desired test situation $\mathbf{p}_{\text{test}}$. Note that the minimization step in the **for** loop in Alg. 5 can be solved analytically, because of the linear operations on the task parameters and the L2-norms.

Algorithm 5 Given demonstrations $\{\{\boldsymbol{\xi}_{n,l}\}_{l=1}^{L_n}\}_{n=1}^{N}$ for N situations $\{\mathbf{p}_{\text{train,n}}\}_{n=1}^{N}$ and test situation $\mathbf{p}_{\text{test}}$, find a trajectory $\boldsymbol{\xi}^{\mathcal{O}}$ for the test situation using hierarchical TP-GMM.

```
procedure HS-TRAIN($\{\{\boldsymbol{\xi}_{n,l}\}_{l=1}^{L_n}\}_{n=1}^{N}, \{\mathbf{p}_{\text{train,n}}\}_{n=1}^{N}$)
    for $n = 1$ to $N$ do
        $\boldsymbol{\Pi}_n = \text{ENCODE}(\{\boldsymbol{\xi}_{n,l}\}_{l=1}^{L_n}, \mathbf{p}_{\text{train,n}})$
    return $\{\boldsymbol{\Pi}_n\}_{n=1}^{N}$
procedure HS-TEST($\{\boldsymbol{\Pi}_n\}_{n=1}^{N}, \{\mathbf{p}_{\text{train,n}}\}_{n=1}^{N}, \mathbf{p}_{\text{test}}, d_{\text{threshold}}$)
    for $n = 1$ to $N$ do
        $\alpha_n = \arg\min_{\alpha'} \text{DISTANCE}(\text{MORPHTASK}(\mathbf{p}_{\text{test}}, \alpha'), \mathbf{p}_{\text{train},n})$
        $\mathbf{p}_n = \text{MORPHTASK}(\mathbf{p}_{\text{test}}, \alpha_n)$
        $d_n = \text{DISTANCE}(\mathbf{p}_n, \mathbf{p}_{\text{train},n})$
    $n^\star = \arg\min_n d_n$
    if $d_{n^\star} < d_{\text{threshold}}$ then
        $\boldsymbol{\pi}^\star = \text{DECODE}(\boldsymbol{\Pi}_{n^\star}, \mathbf{p}_{n^\star})$
        $\boldsymbol{\pi} = \text{MORPHGMM}(\boldsymbol{\pi}^\star, \alpha_{n^\star}^{-1}, \mathbf{p}_{\text{test}})$
        $\boldsymbol{\xi}^{\mathcal{O}} = \text{GMR}(\boldsymbol{\pi}, \boldsymbol{\xi}^{\mathcal{I}})$
        return $\boldsymbol{\xi}^{\mathcal{O}}$
    else
        Request demonstration
        return
```

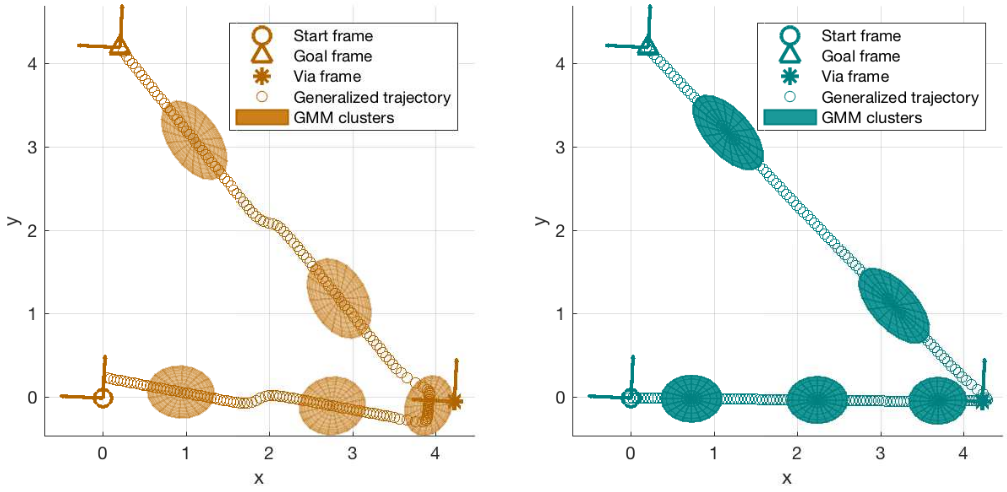
4.3.5 Validation of HS in Simulation

We conducted simulations to empirically validate the hierarchical structure and compare its performance with vanilla TP-GMM (VT). The simulation procedure is detailed in Alg. 6. We used $N_{\text{train}} = 2, 3, ..., 10$ and $M_{\text{test}} = 100$ to explore how test performance changes with an increasing number of demonstrated situations. We repeated SIM 20 times for each value of N_{train}, so that we could extensively sample both training and test situations. We used $d_{\text{threshold}} = \infty$ with HS to disable new demonstration requests because VT cannot preemptively stop execution. Finally, we used three different task configurations of start, goal, and via frame sampling ranges to verify the hierarchical framework's performance.

Algorithm 6 Simulation procedure.

```
procedure Sim($N_{\text{train}}$, $M_{\text{test}}$, start_range, goal_range, via_range)
    for $n = 1$ to $N_{\text{train}}$ do                                              ▷ Sample training data
        Sample $\mathbf{p}_{\text{train},n}$ from sampling ranges
        Generate a demonstration trajectory $\{\boldsymbol{\xi}_{n,l}\}_{l=1}^{200}$
    $\boldsymbol{\Pi}_{\text{VT}}$ = VT-Train($\{\{\boldsymbol{\xi}_{n,l}\}_{l=1}^{200}\}_{n=1}^{N_{\text{train}}}, \{\mathbf{p}_{\text{train,n}}\}_{n=1}^{N_{\text{train}}}$)
    $\{\boldsymbol{\Pi}_n\}_{n=1}^{N_{\text{train}}}$ = HS-Train($\{\{\boldsymbol{\xi}_{n,l}\}_{l=1}^{200}\}_{n=1}^{N_{\text{train}}}, \{\mathbf{p}_{\text{train,n}}\}_{n=1}^{N_{\text{train}}}$)
    for $m = 1$ to $M_{\text{test}}$ do                                                ▷ Sample test data
        Sample $\mathbf{p}_{\text{test},m}$ from sampling ranges
        for $n = 1$ to $N_{\text{train}}$ do
            $d_{m,n}$ =Distance($\mathbf{p}_{\text{test},m}$, $\mathbf{p}_{\text{train},n}$)
        $d_m = \min_n d_{m,n}$                                                   ▷ Distance function value
        $\boldsymbol{\xi}_{\text{VT}}^{\mathcal{O}}$ = VT-Test($\boldsymbol{\Pi}_{\text{VT}}, \mathbf{p}_{\text{test},m}$)
        $\boldsymbol{\xi}_{\text{HS}}^{\mathcal{O}}$ = HS-Test($\{\boldsymbol{\Pi}_n\}_{n=1}^{N_{\text{train}}}, \{\mathbf{p}_{\text{train,n}}\}_{n=1}^{N_{\text{train}}}, \mathbf{p}_{\text{test},m}, \infty$)
        $e_{\text{VT},m}$ = Eval($\mathbf{p}_{\text{test},m}$, $\boldsymbol{\xi}_{\text{VT}}^{\mathcal{O}}$)                          ▷ Evaluation of VT
        $e_{\text{HS},m}$ = Eval($\mathbf{p}_{\text{test},m}$, $\boldsymbol{\xi}_{\text{HS}}^{\mathcal{O}}$)                          ▷ Evaluation of HS
    return $\{d_m\}_{m=1}^{M_{\text{test}}}$, $\{e_{\text{VT},m}\}_{m=1}^{M_{\text{test}}}$, $\{e_{\text{HS},m}\}_{m=1}^{M_{\text{test}}}$
procedure Eval($\mathbf{p}$, $\boldsymbol{\xi}$)
    $e_1 = ||\xi_1 - \mathbf{r}_1||_2$                                      ▷ First trajectory point and start
    $e_2 = ||\xi_{200} - \mathbf{r}_2||_2$                                   ▷ Last trajectory point and goal
    $e_3 = ||\xi_{100} - \mathbf{r}_3||_2$                                 ▷ 100-th trajectory point and via
    $e = \{e_1, e_2, e_3\}$
    return $e$
```

Fig. 4.5 shows sample generalized trajectories for a test with three demonstrated task situations. We used Eval in Alg. 6 to calculate *generalization errors* as the squares of the distances between the first generalized trajectory point and the start frame, the last trajectory point and the goal frame, and the 100-th trajectory point and the via frame. These three pairs of points should coincide for a trajectory that was perfectly generated according to the original demonstration strategy, yielding zero error.



(a) With vanilla TP-GMM (VT). **(b)** With hierarchical structure (HS).

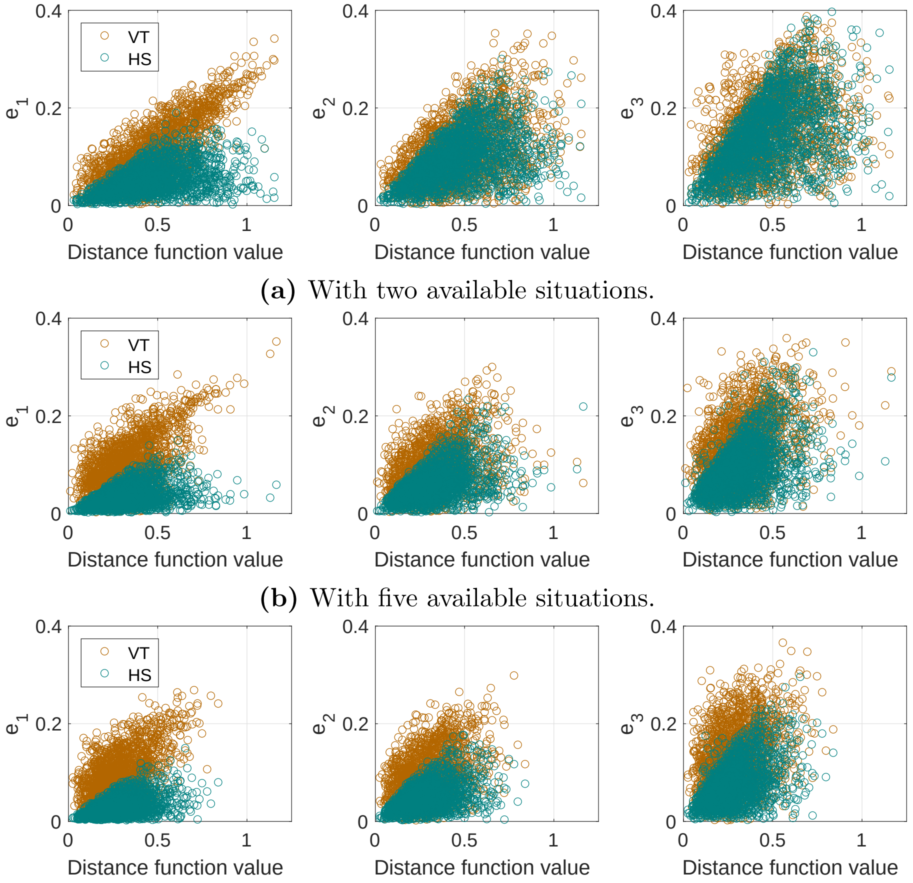
Figure 4.5: Generalized trajectories for a test situation when three demonstrations are available.

Distance function as a performance estimator

For the task configuration shown in Fig. 4.3, Fig. 4.6 shows the accumulated generalization errors against the corresponding distance function values. For brevity, we show results only when two, five, or eight situations were demonstrated. The general trend that the upper bounds of the generalization errors are roughly linearly proportional to the distance function value is true for all numbers of demonstrations. In other words, the distance value can serve as a cautious estimator of generalization errors, so that if the user needs to maintain a minimum level of performance (maximum allowed generalization error), he or she can prevent task execution if the distance value is above a certain threshold and can instead provide more demonstrations to the robot.

Generalization performance

Fig. 4.6 shows that HS achieved better performance (lower evaluation metric values) than VT. The accumulated generalization errors are shown for a few different task configurations in Fig. 4.7 to fully verify that observation. It can be seen that as the number of demonstrated situations increased from 2 to 9 (we explain the results with 10 situations in the next subsubsection), the generalization errors became



(a) With two available situations.

(b) With five available situations.

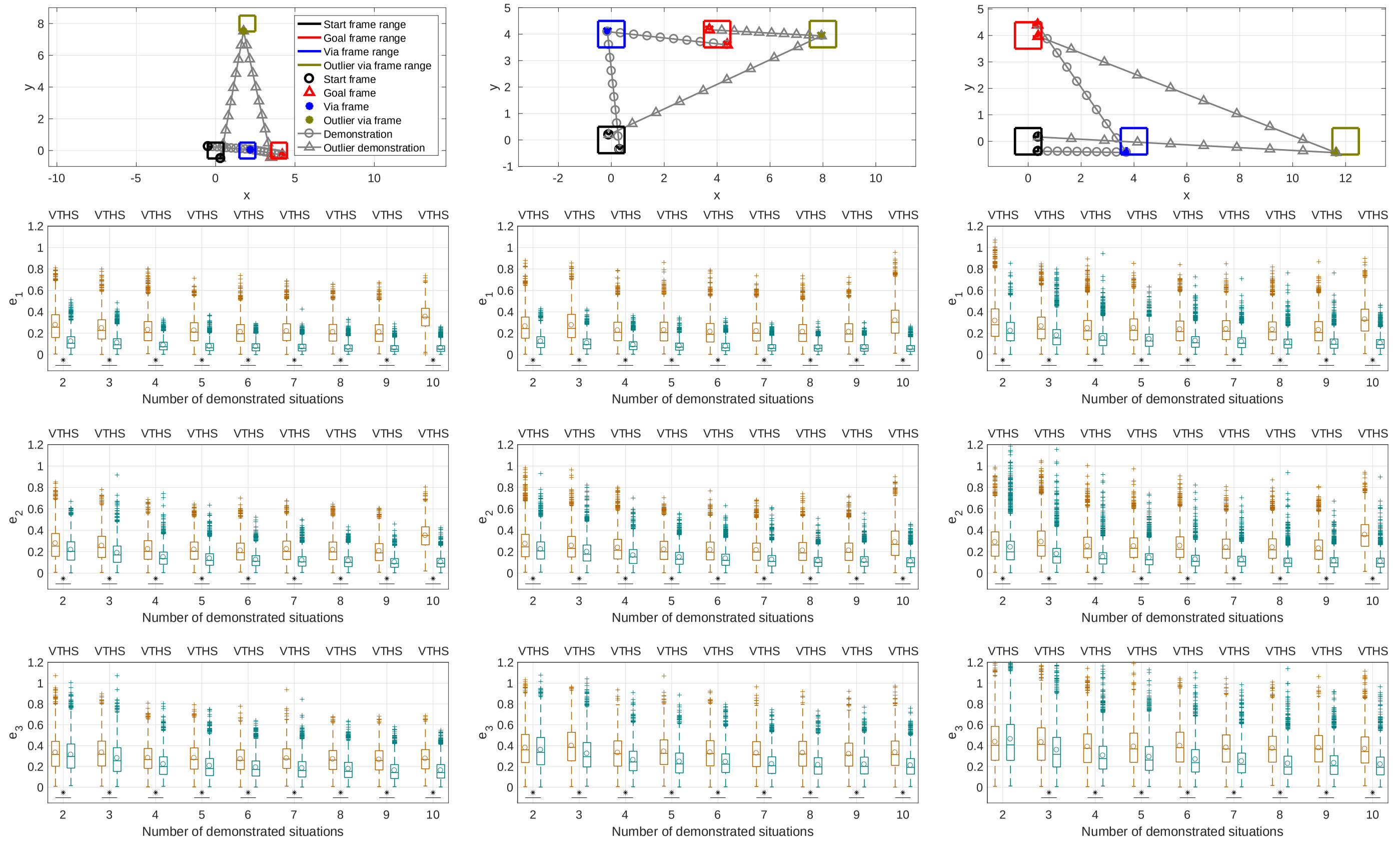
(c) With eight available situations.

Figure 4.6: Generalization errors against distance metric when different numbers of demonstrations were available during training.

lower for both VT and HS. Nevertheless, HS almost always had significantly better performance than VT.

Effect of an outlier demonstration

We mentioned that TP-GMM can be seen as a regression algorithm, with the task parameters as independent variables and the trajectories as dependent variables. Therefore, we explored the effect of having outlier training data in TP-GMM when there were ten demonstrated situations in the simulation procedure: the tenth situation had a different sampling range for the via frame, as shown in the top panels of Fig. 4.7. It can be seen that VT suffered from this single outlier: compared to results with nine and sometimes even two demonstrated situations (see e_1 in Fig. 4.7a



(a) Configuration 1.

(b) Configuration 2.

(c) Configuration 3.

Figure 4.7: Comparison of generalization errors for the vanilla TP-GMM (VT) and the hierarchical structure (HS) when different numbers of demonstrations were available. Top panels show task configurations and samples of task situations, where each task frame was sampled from the specified range. When ten situations were demonstrated, the via frame of the tenth situation was drawn from an outlier range. Panels in the bottom three rows show boxplots of error metrics with different numbers of demonstrations in different task configurations. The center box lines represent the medians, and the box edges are the 25th and 75th percentiles. Circles show mean values. A star and a horizontal line below boxplots show data from HS is significant lower than data from VT with $p < 0.001$.

and e_2 in Fig. 4.7b): average generalization errors increased when demonstrations from 10 situations were available, even though more training data is often assumed to improve test performance. On the other hand, HS was not affected by the outlier and maintained the same level of performance, because task parameters were used to first filter the training data and only the most similar situation was used in generalization.

Note that although the outlier situations differ from the majority of the training data from the model's point of view, they may still be of interest to users and hence should be learned by the robot. Therefore, it may be more sensible for vanilla TP-GMM to encode the different situations in a separate model, so that the robot can handle the outlier situations without affecting performance for the regular situations. However, to the best of our knowledge, allowing a robot to automatically determine when to create a new model is still an open problem regarding TP-GMM, precisely because the strengths of TP-GMM include handling data from varied task situations.

Training time

Another advantage that HS offers is reducing computational load when demonstrations become available gradually. The VT-Train function always encodes all demonstrations together, which means previously encoded demonstrations have to be stored for reuse later. On the other hand, because HS-Train encodes each situation in a separate TP-GMM, only new data needs processing when it becomes available. Fig. 4.8 shows the time spent only in training TP-GMMs as the number of available demonstrations increase in the simulation procedure: HS consistently took little time to encode new data, while VT had to spend more time encoding everything.

Note that here we stored all demonstrations and computed a new TP-GMM every time for VT, instead of using one of the incremental TP-GMM methods in [37] because of the following reasons. First, the generative technique in [37] doesn't save computation time compared to VT because it samples trajectories using the existing model to represent previously encoded trajectories, which are then encoded with new trajectories to form a completely new model. Furthermore, performance may suffer because sampled trajectories are used in the new model

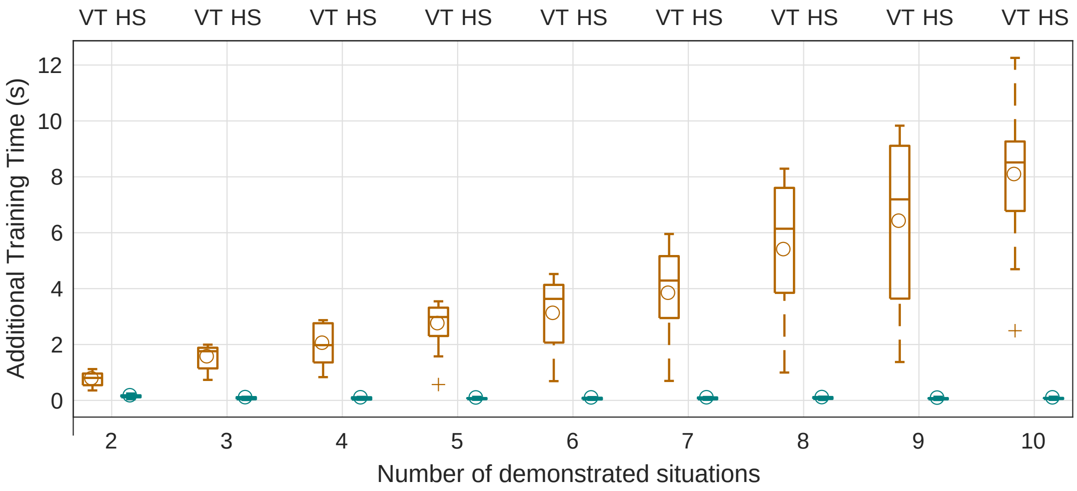


Figure 4.8: Additional time spent to train TP-GMMs increased as more demonstrations became available for VT but remained low for HS.

instead of actual demonstrations. Second, the model addition technique in [37] will take strictly more time than HS because it encodes new demonstrations in a new model like HS and then has to concatenate and optimize the previous and the new model together. Third, the direct update technique [37] assumes that the old demonstrations and the new ones are drawn from the same distribution, which is problematic because we sample from a relatively large number of task situations or even an outlier situation.

The advantage of reduced computational load is also true when removing demonstrations. In this example, HS can identify the outlier in the 10 demonstrations, because it was never selected for generalization, and thus the corresponding TP-GMM could be deleted from the robot's database without affecting performance. In contrary, there is no inherent method in VT to identify the outlier, and if a human operator identifies the outlier demonstration to be deleted, a new TP-GMM has to be retrained to recover the performance of the nine remaining demonstrations. Incremental methods such as [37, 73] don't seem to consider removing demonstrations.

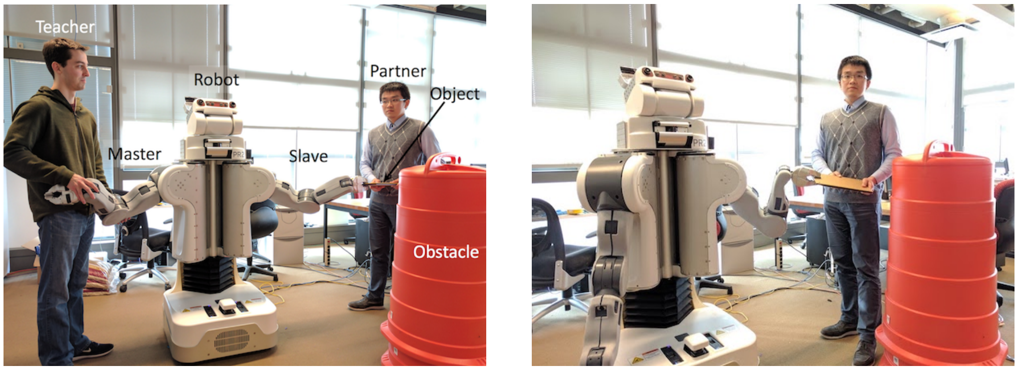
4.3.6 Comparison Summary between HS and VT

Compared to vanilla TP-GMM, the hierarchical structure has higher complexity because it encodes a separate TP-GMM for each demonstrated task situation and has several utility functions to compare and morph test situations against demonstrated ones. However, this structured approach enables a robot to differentiate between the demonstrated situations and select the most similar experience for trajectory generalization. As a result, the robot can halt autonomous execution if its expected performance is lower than a predefined threshold. In contrast, VT does not offer these abilities. Because HS selectively uses demonstrations during generalization, test performance can also be improved via situation-specific manipulation of trajectories using the morphing function. Moreover, it is possible to include a wide variety of task situations in the same HS framework without decreasing performance, while VT may lose situation-specific information because it effectively averages all demonstrations. Lastly, new demonstrations can easily be added to the skill library by creating a new TP-GMM, and poor or no-longer-wanted demonstrations that are already encoded can easily be removed with the hierarchical structure.

Special care needs to be considered when defining the utility functions in HS. For tasks such as object movement, Cartesian distances between corresponding task frames can readily be used in the distance function because they utilize the same information that TP-GMM considers. As for the morphing functions, our example validates the effectiveness of the hierarchical structure even with their simple and intuitive definitions. Other approaches, such as using reinforcement learning to optimize task parameters [41], may be used as well at the expense of additional design effort and computational load.

4.4 Experimental Validation on Real Hardware

We tested our learning structure using a Willow Garage PR2 in the real collaborative object-manipulating task shown in Fig. 4.9. The PR2 has two mirrored arms, each with four revolute arm joints and three revolute wrist joints. The PR2 has Robot Operation System (ROS) installed on its onboard computers, which handled all sensor and actuator communications. We collected demonstrations for three different task situations and conducted a human-subject study to validate the generalization performance of the hierarchical structure in this task. Fifteen adults participated in the study, each completing the collaborative object-manipulation task under various task situations, experiencing the demonstration process, and filling out questionnaires to evaluate their interactions with the robot. The Penn



(a) The teacher guides the robot to manipulate an object with the partner.

(b) The robot collaborates with the partner during evaluation.

Figure 4.9: Training and testing scenarios for collaborative manipulation with a Willow Garage PR2.

IRB approved all experimental procedures under protocol 829536. Subjects gave informed consent and received no compensation for participating.

4.4.1 Task Definition

The left arm of the PR2 robot holds an object (a rigid rectangular plate) together with the human partner; the robot and the participant collaborate to move the object from a start position to a goal position while avoiding an obstacle. The plate has a mass of 0.77 kg and a size of 0.30 m by 0.20 m by 0.01 m, and the obstacle is a slightly tapered plastic cylinder with a top radius of 0.23 m, as shown in Fig. 4.9. The minimum and maximum distances between the robot end-effector at robot-shoulder height and the shoulder joint are about 0.4 m and 0.82 m, so the size of the obstacle is significant when compared to the robot's workspace. Thus the robot's trajectories during collaborative movement need to make sense for the human partner for the task to be successful.

4.4.2 Demonstrating Procedure

When collecting demonstrations, we used the PR2's right arm as the master and its left arm as the slave in bilateral teleoperation [87]. The teacher guided the master to help the human partner accomplish the desired task with the slave arm, therefore directly feeling the motions and limits of the robot arm, similar to how demonstrations are done in kinesthetic teaching. Demonstration recordings included the Cartesian position of the slave's wrist center calculated from forward kinematics. Force feedback between the master and the slave was achieved through a joint-level proportional-derivative (PD) torque controller, and therefore no force or torque sensor was required.

For the master-side wrist joints, an additional virtual fixture [1] was applied to help the teacher control the robot's hand orientation. The virtual fixture torques were calculated using a PD controller with zero desired velocity:

$$\tau_{i,\mathrm{vf}} = K_{p,\mathrm{vf}}(q_{i,\mathrm{vf}} - q_{i,m}) - K_{d,\mathrm{vf}}\,\dot{q}_{i,m}, \tag{4.4.1}$$

where $q_{i,\mathrm{vf}}$ are the desired wrist joint angles for the virtual fixture. In the current work, we used the virtual fixture to constrain one degree of freedom of the gripper orientation: the desired center axis of the gripper frame was constrained to be horizontal in the world frame, and $q_{i,\mathrm{vf}}$ were found using inverse kinematics. The virtual fixture could also be used to satisfy task-specific requirements by choosing a different desired gripper orientation, e.g., when the carried object needs to be tilted to go through a doorway.

4.4.3 Training Procedure and Robot Controller

The task frames were defined to include positions $\mathbf{r}_p$ and orientations $\mathbf{R}_p$ of the start ($p = 1$), the goal ($p = 2$), and the obstacle ($p = 3$). When collecting demonstrations, we used forward kinematics to determine the start and the goal poses, and we calculated the obstacle's pose by making the robot's end-effector touch the edge of the cylindrical obstacle along its radial direction and adding an offset of the cylinder's radius. Because the start, goal, and obstacle frames may have different orientations in each task situation, we expanded the previously listed distance function (Alg. 3) to iteratively align all task frames, as shown in Alg. 7.

We assumed the existence of a desired trajectory corresponding to each task situation. Given the robot's wrist trajectories ($\mathbf{x}$) for a situation of interest, we resampled all trajectories to L data points based on trajectory length and used standard GMM/GMR to generate an *average trajectory* ($\mathbf{x}_{\mathrm{avg}}$) in the world frame for that task situation. We then derived a *desired trajectory* for the task situation:

Algorithm 7 Distance function used in the user study.

```
procedure DistanceFunction($\mathbf{p}_A, \mathbf{p}_B$)
    $d = 0$
    for $i = 1$ to $3$ do
        for $j = 1$ to $3$ do
            $d = d + \left|\left|\mathbf{R}_{i,A}^{\top}\left(\mathbf{r}_{j,A} - \mathbf{r}_{i,A}\right) - \mathbf{R}_{i,B}^{\top}\left(\mathbf{r}_{j,B} - \mathbf{r}_{i,B}\right)\right|\right|_2^2$
    return $d$
```

$\mathbf{x}_{\text{des}} = \left[\{\mathbf{x}_{\text{avg},l}\}_{l=l_d+1}^{L} \quad \mathbf{x}_{\text{interp}}\right]$, where l_d serves as a look-ahead variable to make the robot appear more active during execution, and $\mathbf{x}_{\text{interp}}$ is a linearly interpolated trajectory between the last average trajectory point ($\mathbf{x}_{\text{avg},L}$) and the goal of the task situation ($\mathbf{r}_2$) in l_d steps. In the current work, $L = 500$ and $l_d = 50$. In the experiment, $\mathbf{x}_{\text{avg},L}$ was typically close to the goal point, and thus $\mathbf{x}_{\text{interp}}$ generally was a short line-segment connecting $\mathbf{x}_{\text{avg},L}$ to the exact goal location.

We chose to learn TP-GMMs that use the robot's 3D wrist center position as the input and the desired 3D trajectory point calculated from above as the output: $\boldsymbol{\xi}^{\mathcal{I}} = \mathbf{x}$, and $\boldsymbol{\xi}^{\mathcal{O}} = \mathbf{x}_{\text{des}}$. Consequently, the task parameters were defined as $\mathbf{A}_p = \begin{bmatrix} \mathbf{R}_p & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{R}_p \end{bmatrix}$ and $\mathbf{b}_p = \begin{bmatrix} \mathbf{r}_p^{\top} & \mathbf{r}_p^{\top} \end{bmatrix}^{\top}$, $p = 1, 2, 3$. We chose to use position rather than time for parameterization to increase robustness [95] and eliminate the need for phase estimation (e.g., [72]) or dynamic time warping. In addition, we updated definitions of $\mathbf{T}'$ and $\mathbf{T}''$ in Alg. 4 to accommodate 3D trajectories and the dimensions of our TP-GMM:

$$\begin{aligned} \mathbf{T}' &= \begin{bmatrix} \mathbf{I} + \alpha\mathbf{T} - \mathbf{T} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I} + \alpha\mathbf{T} - \mathbf{T} \end{bmatrix}, \\ \mathbf{T}'' &= \begin{bmatrix} (\alpha - 1)\mathbf{T} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & (\alpha - 1)\mathbf{T} \end{bmatrix}. \end{aligned} \tag{4.4.2}$$

We selected three different task situations to collect demonstrations, as shown in Fig. 4.11a. We collected five demonstrations for each situation to ensure that the variability of trajectories was captured. The collected demonstrations and their task parameters were encoded in VT-Train and HS-Train, and the TP-GMMs ($\boldsymbol{\Pi}_{\text{VT}}, \{\boldsymbol{\Pi}_n\}_{n=1}^{3}$) were tested and evaluated in the user study. We used the Bayesian Information Criterion (BIC) [103] to determine the number of Gaussian clusters M

for each TP-GMM (15 clusters for $\mathbf{\Pi}_{\mathrm{VT}}$ and 4 or 5 clusters for each $\mathbf{\Pi}_n$).

At test time, the PR2 robot could calculate a desired wrist position from its current wrist position at each time step using VT-Test or HS-Test, and a generic PD controller was used to generate the motor commands:

$$\boldsymbol{\tau} = \mathbf{J}^\top(\mathbf{K}(\mathbf{x}_{\mathrm{des}} - \mathbf{x}) - \mathbf{B}\dot{\mathbf{x}}), \tag{4.4.3}$$

where $\mathbf{J}$ is the Jacobian matrix of the position dimensions, and we chose $\mathbf{K}$ and $\mathbf{B}$ as diagonal matrices with $120\,\mathrm{N\,m^{-1}}$ and $10\,\mathrm{N\,s\,m^{-1}}$. We also included a passive mode where the robot provided only gravity compensation for the half of the object's weight, with

$$\boldsymbol{\tau} = -\frac{1}{2}\mathbf{J}^\top m\mathbf{g}, \tag{4.4.4}$$

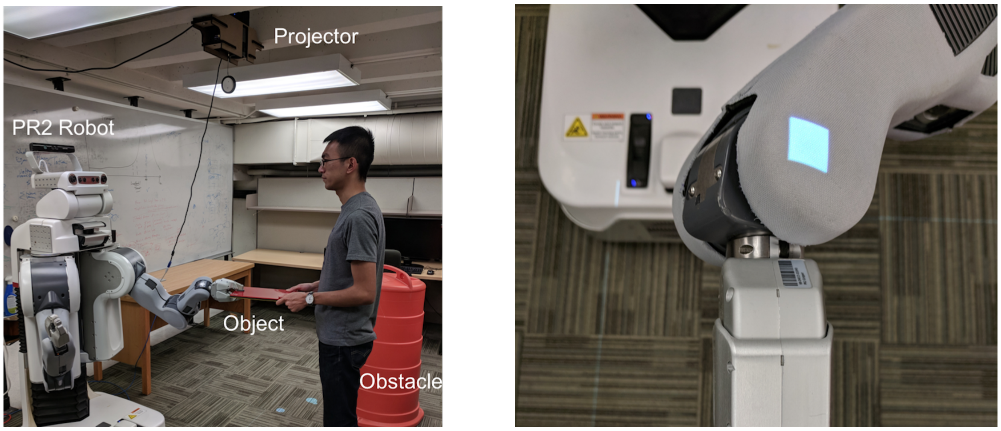
where m is the mass of the held object and $\mathbf{g}$ is the gravity vector, assuming that the robot and the human share the object's weight equally. This control mode was included so we could directly test whether the complexity of TP-GMM confers any benefits.

4.4.4 User Study

We conducted a human-subject study to evaluate how the three described control algorithms affect task performance and how users perceive the robot behaviors in the collaborative movement task. Since the task situations were defined in the robot frame, we used a projector mounted on the ceiling to help participants identify and find the desired start, goal, and obstacle positions, as shown in Fig. 4.10.

Participants

Our participant pool consisted of 15 University of Pennsylvania affiliates, including undergraduate and graduate students, postdoctoral associates, and visiting researchers. Of the 15 participants, three were female and 12 were male, with ages ranging from 22 to 35 years ($\mu = 26.3, \sigma = 3.42$).



(a) A projector mounted on the ceiling helps subjects find the start, goal, and obstacle positions.

(b) A square is projected on the PR2's wrist when the Start position is reached.

Figure 4.10: The experimental setup for the human-subject study.

Dependent measures

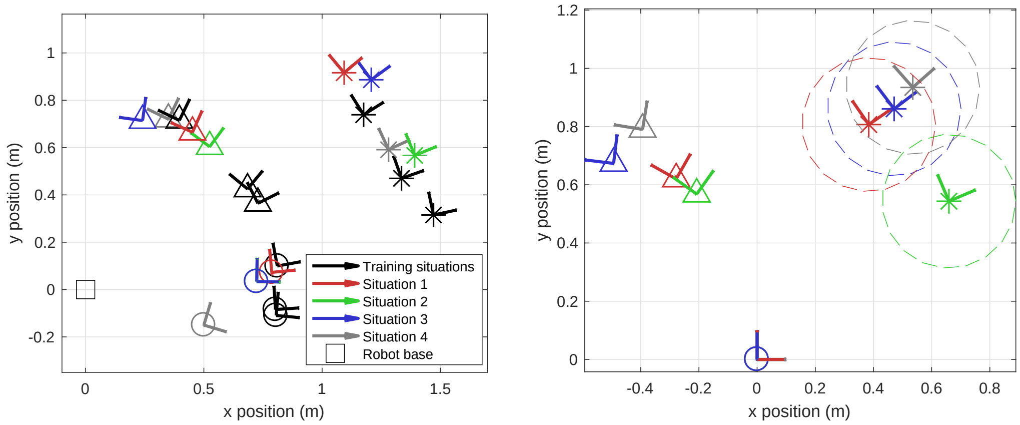
We recorded robot motions during each trial, and we used two quantitative measures to evaluate task performance: average trajectory length and average task completion time.

In addition, we used three questionnaires to evaluate the quality of the human-robot interactions during the task. First, the subject completed a Unified Theory of Acceptance and Use of Technology (UTAUT) survey [110] at the beginning and the end of the study. Results from the two surveys were compared to determine how interacting with the robot affected the general perception of subjects toward using the robot in everyday tasks, and UTAUT questions include:

•

Second, we adapted the questionnaire used in [61] and asked participants to answer the following questions on a 100-point scale from strongly disagree to strongly agree after each collaboration trial:

- Q1. The robot *moved too fast*.
- Q2. The robot *moved too slowly*.
- Q3. The robot *had problems* doing the task.



(a) Training and the four common situations in world frame.

(b) The four common situations aligned by their start frames.

Figure 4.11: Task situations involved in the experiment. Circles represent the start frames, triangles the goal frames, and stars the obstacle frames. Dashed circles represent the boundary of the obstacle. Note that the start frames of situations 2 and 3 coincide. With respect to the three demonstrated situations, The $d_{n^\star}$ values for these four situations in Alg. 5 with the distance function definition in Alg. 7 are $0.071\,\mathrm{m}^2$, $0.151\,\mathrm{m}^2$, $0.295\,\mathrm{m}^2$, and $0.398\,\mathrm{m}^2$.

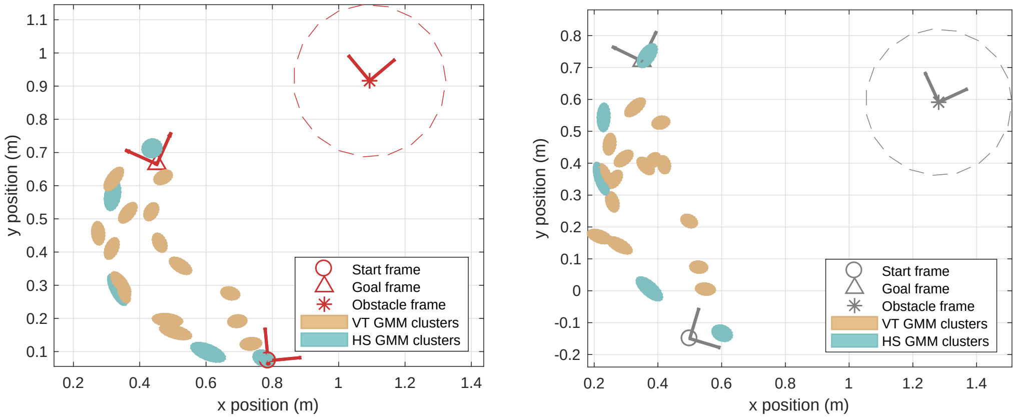
- Q4. I *felt safe* when working with the robot.
- Q5. I *trusted the robot* to do the right thing at the right time.
- Q6. The robot and I *worked well* together.

These questions sought to evaluate how subjects perceived the robot's behaviors and performance.

Third, a NASA-TLX survey [34] was administered after the participants experienced the process of providing demonstrations via teleoperation to gauge the workload of this interaction.

Procedure

The human-subject study consists of two main components: collaborating with the robot and providing demonstrations. We chose to put the collaborating component first because it allows participants to become more familiar with the robot before demonstrating new movements.



(a) Situation 1.

(b) Situation 4.

Figure 4.12: Generalized GMM clusters ($\boldsymbol{\xi}^{\mathcal{O}}$ dimensions) for two test situations from Fig. 4.11.

The first component, collaborating with the robot, took place after the opening UTAUT survey. Participants were asked to collaborate with the robot to move the object from the start to the goal in five different situations. Four common situations, shown in Fig. 4.11, were pre-determined by the research team and were the same for all subjects, with situation 1 having the lowest and situation 4 having the highest $d_{n^\star}$ value. These common situations were chosen to be somewhat close to the training situations so that both VT and HS should intuitively work well, and they span a relatively large portion of the robot's workspace (Fig. 4.11a) and have varied configurations (Fig. 4.11b). Recall that we use TP-GMMs to encode data with $\boldsymbol{\xi}^{\mathcal{I}} = \mathbf{x}$ and $\boldsymbol{\xi}^{\mathcal{O}} = \mathbf{x}_{\text{des}}$, and Fig. 4.12 shows the $\boldsymbol{\xi}^{\mathcal{O}}$ dimensions of generalized VT and HS GMM clusters for two example test situations. Situation 5 was chosen by each subject so that a large number of situations were sampled in the study.

Participants compared three different robot behavior modes: (P) passive with gravity compensation for the object, (VT) active with vanilla TP-GMM, and (HS) active with hierarchical structure and TP-GMM. Therefore, each participant evaluated the collaboration in 15 different combinations of task situation and robot behavior. For each situation, we asked participants to experience all three robot behaviors sequentially, completing two collaborations under each robot behavior, and then re-experience and rate each robot behavior. After the subject evaluated all three robot behaviors, a new situation was presented. This process repeated until all five situations had been shown. For each subject, the five situations were presented in a randomized order, and within each situation, the presentation order of the three robot behaviors was also randomized.

Then participants were presented with the second component of the study: providing demonstrations. The experimenter acted as the partner and moved the object with the robot, and the participant manipulated the robot's right arm to teleoperate the slave arm. Each participant experienced the demonstrating process for three to five minutes and then filled out the NASA-TLX survey. Finally, the participant filled out the closing UTAUT survey.

Hypotheses

Based on the dependent measures, our main hypotheses for this experiment were as follows:

- The hierarchical structure (HS) will lead to better task performances, including shorter trajectory lengths and lower task completion times, compared to using vanilla TP-GMM (VT) and passive gravity compensation (P).

- Participants will be more satisfied with the robot when it uses the hierarchical structure (HS) to generate motion controls, compared to VT and P.

- Given feedback after task execution in a wide variety of situations, the robot will be able to learn decision boundary for when to ask for new demonstrations.

4.4.5 Results

Fig. 4.13 shows results from the NASA-TLX survey that subjects completed after providing demonstrations. The median value for 'How mentally demanding was the task?' was 53 out of 100, and the median values for the other five questions were all below 50, indicating that the subjects perceived teaching by teleoperation as a low-to-moderate-effort task.

Objective performance at the collaborative movement task was determined by calculating the two quantitative measures of average trajectory length and average task completion time for the four common task situations. Results are shown in Fig. 4.14. One-way repeated measures analysis of variance (ANOVA) was used

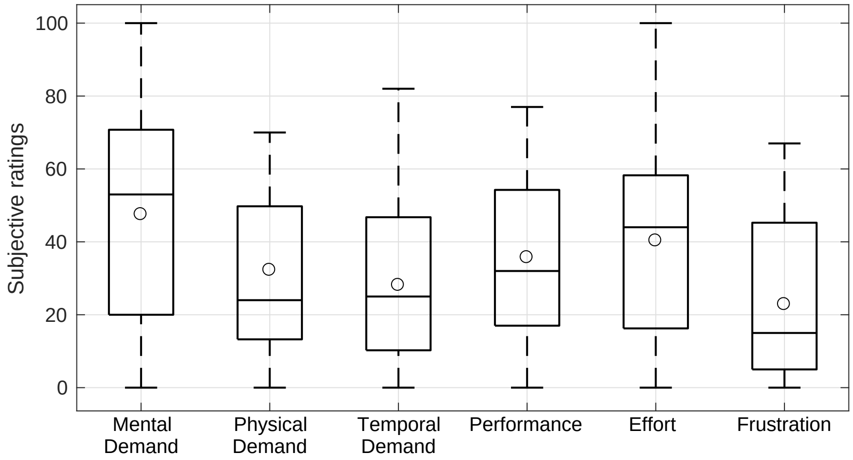


Figure 4.13: NASA-TLX ratings of the teaching procedure (lower is better). For the Performance question, 0 indicates perfect and 100 indicates failure.

to determine whether the differences in these measures between the three behavior modes under the same task situation were significant. If there was, a Tukey-Kramer post-hoc multiple comparison test was conducted to determine which robot behaviors produced significantly different ratings. It can be seen that with the hierarchical structure, traversed trajectories were significantly shorter and took significantly less time for almost all situations, while VT reduced average completion time compared to passive gravity compensation only in test situation 1, which is the most similar to the demonstrated situations.

Results from the subjective ratings under the four common task situations are shown in Fig. 4.15. Plotted ratings of Q1, Q2, and Q3 were subtracted from 100 so that a higher rating is better for all questions. The same procedure used for the quantitative measures was used to determine significant differences. It can be seen that with passive gravity compensation, the robot almost always appeared significantly slower (Q1, Q2) and safer (Q4). No significant differences in perceived pace and safety were found between the two active modes. Compared to vanilla TP-GMM, the hierarchical structure appeared to have significantly fewer problems doing the task (Q3) in all situations. In situations 2 and 3, participants had more trust in the robot doing the right thing at the right time (Q5) with the hierarchical structure than vanilla TP-GMM. Finally, participants felt they worked better with the robot (Q6) with the hierarchical structure than the other two modes in situation 1. Fig. 4.16 shows the sums of these ratings. Vanilla TP-GMM generally had

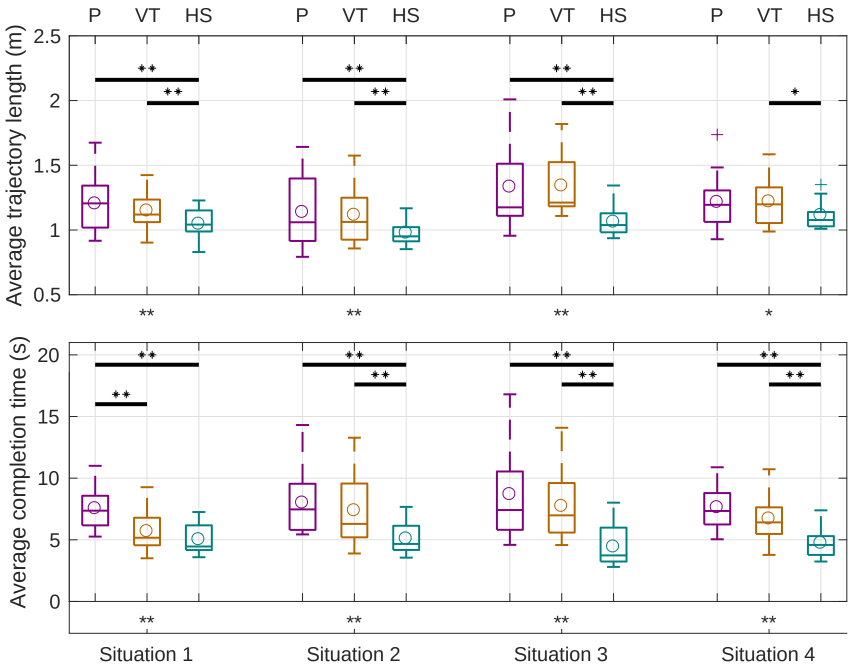


Figure 4.14: Comparison of quantitative measures for the four common task situations (lower is better). Stars under the x-axes show significant differences in each group, and stars and horizontal lines above boxplots show pairwise significant differences. *: $p < 0.1$. **: $p < 0.05$.

the lowest rating sums. Significant differences were found between the hierarchical structure and vanilla TP-GMM in situations 2 and 4 as well as between passive gravity compensation and vanilla TP-GMM in situation 3.

Results from the subjective ratings under the participant-selected situations were used to validate the distance function as a performance estimator in the collaborative task. To acquire labels of successful/failed execution, we defined the following criterion: an HS success has a subjective rating sum that is greater than 80% of the largest sum of ratings for any control mode under any task situation from the corresponding subject. Fig. 4.17a shows the classification results using the above criterion, where we additionally include a second feature dimension $(\max(\alpha_{n^\star}, 1/\alpha_{n^\star}))$, which is calculated from the distance function in HS-Test and represents the degree of stretching or compression in the morphing functions. We also plot the common situations in Fig. 4.17a for completeness; these are manually labeled as successes with the hierarchical structure, because there were significant advantages in the quantitative measures and some significant advantages in the subjective rat-

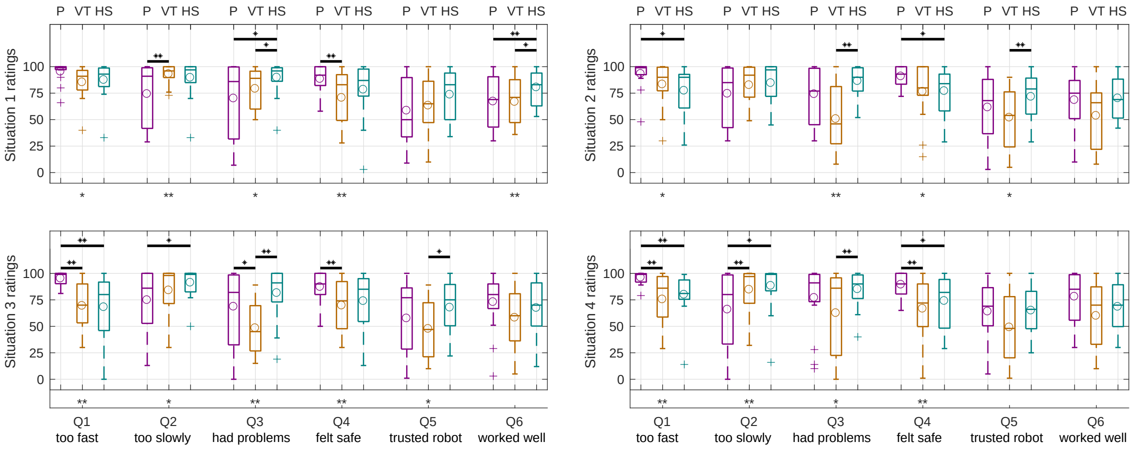


Figure 4.15: Comparison of subjective ratings (higher is better). Questions Q1–Q6 are detailed in Section 4.4.4. *: $p < 0.1$. **: $p < 0.05$.

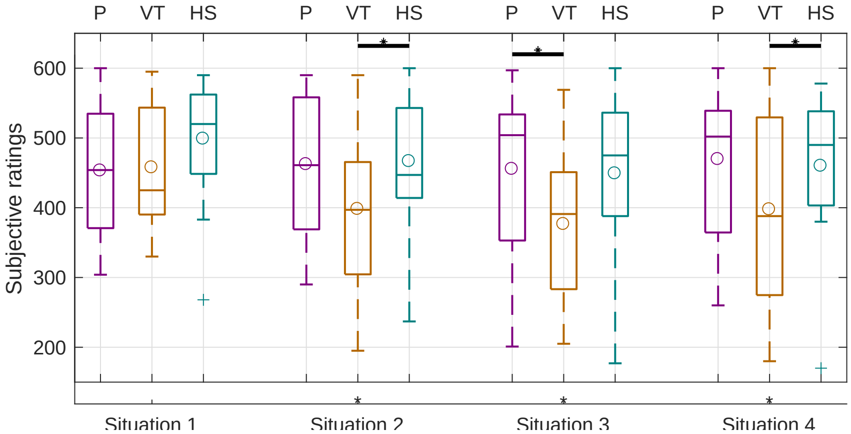
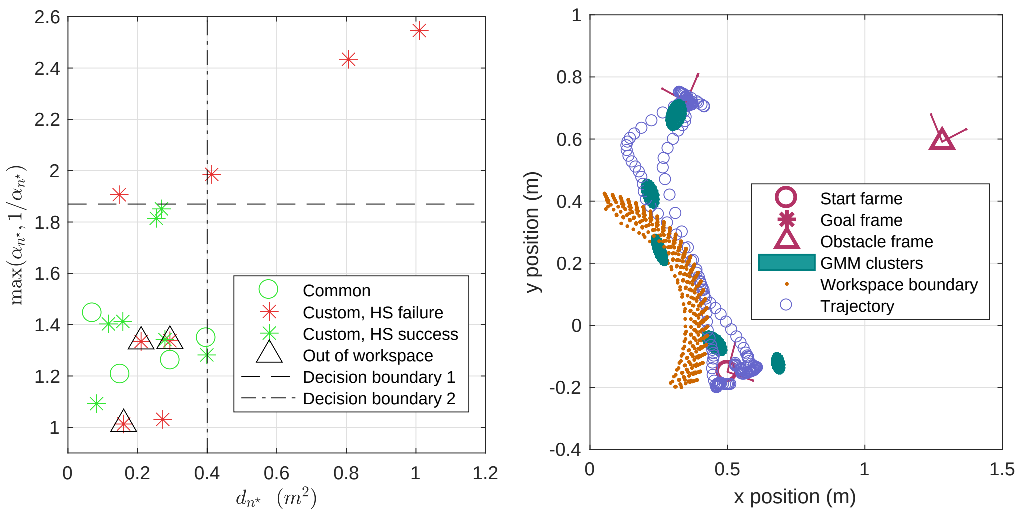


Figure 4.16: Comparison of sums of subjective ratings (higher is better). *: $p < 0.1$.

ings. Three of the custom situations for which the hierarchical structure was rated as a failure are manually marked as being out of workspace, because the obstacle was placed so close to the robot that some GMM clusters of the hierarchical structure policy were outside of the robot's workspace. As a result, the robot arm would become stuck when it first reached the workspace boundary following the control policy, and the robot arm would then slide along the workspace boundary and appear less smooth to participants. This behavior at least partially caused these three poor ratings; an example is shown in Fig. 4.17b.

Note that each participant tested only one custom situation and hence contributed one data point to Fig. 4.17a. The distribution might look different if a



(a) Rated success with the hierarchical structure.

(b) An example with some GMM clusters outside of the workspace.

Figure 4.17: Learning when to request new demonstrations.

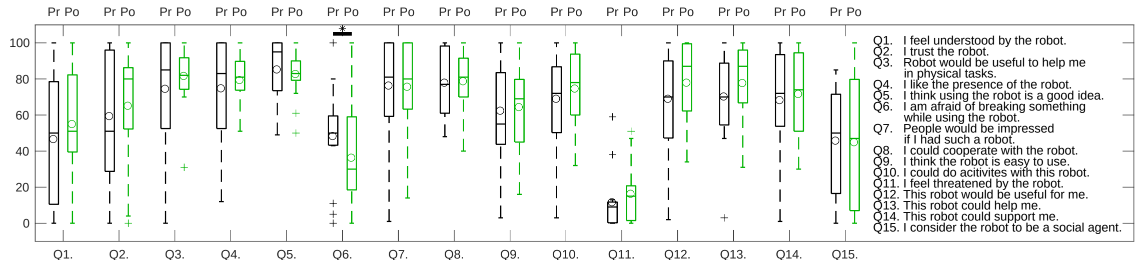


Figure 4.18: UTAUT ratings from pre-study (Pr) and post-study (Po) surveys. *: $p < 0.1$.

single participant did all of the tested situations, and it might change for different participants. Nevertheless, not counting the out-of-workspace situations, one could simply place a decision boundary for when to ask for new demonstrations at $\max(\alpha_{n^\star}, 1/\alpha_{n^\star}) = 1.87$ with one mis-classified task situation, or at $d_{n^\star} = 0.4\,\mathrm{m}^2$ with two mis-classified data points.

Fig. 4.18 shows the ratings from the UTAUT questions. Paired-sample t-tests were used to determine whether the differences between the means of these ratings were significant. A significant difference was found for the question ‘I am afraid of breaking something while using the robot’, indicating subjects were less afraid of breaking something after the study. No significant differences were found for other questions.

4.4.6 Discussion

The results from the user study provide strong support for our first hypothesis: the significant differences in the quantitative measures in favor of HS indicate that explicitly reasoning about task situations and generalizing over selected demonstrations could lead to significantly better trajectories and hence better task performance. In comparison, although VT also uses task parameters to annotate and process the demonstrations in Encode and Decode, it doesn't differentiate between demonstrated situations and thus cannot apply situation-specific trajectory and model morphing to better accommodate the test situation. As a result, VT falls short of HS in task performance. Passive gravity compensation offers little help to users and led to the longest trajectory lengths and task completion times.

Hypothesis 2 centers on user perception of the three tested robot control modes. In the subjective ratings, HS was rated to be significantly less problematic (Q3) than VT in all situations, indicating that users were able to differentiate the two modes and preferred HS. Furthermore, HS achieved significantly higher ratings than VT for trust (Q5) and working well together (Q6) in particular situations. The sum of the subjective ratings generally favored the hierarchical structure and passive gravity compensation, with the former being perceived as fast and effective and the latter as slow and safe. Thus, we conclude that the subjective ratings support our second hypothesis.

Results from the custom situations provide some support for our third hypothesis, which stated that the robot would be able to learn a decision boundary for when to ask for new demonstrations in the object movement task. In the particular instance of Fig. 4.17a, only one situation would have been mis-classified by the robot with decision boundary 1. Note that instead of using human ratings as labels and learning a decision boundary (as a classification problem), the robot could use other features, such as time taken and distance traveled for each task, as outputs in a regression problem so that the robot could predict objective performance as well. This approach would enable the robot to learn when to request new demonstrations automatically, but the robot might not be able to adapt to different users' preferences given the lack of human feedback.

We think predicting the performance of generalized behaviors is a critical component of LfD, since demonstrations are typically available only for a small subset of task situations and robot designers often cannot test every possible one. In addition to uncertainty- or confidence-based methods, TP-GMM offers unique opportunities in this effort because it utilizes the task parameters that offer additional information about the demonstrations as well as enhance generalization capabilities.

The NASA-TLX survey and the pre- and post-study UTAUT surveys were not used to evaluate our hypotheses, but they provide some insights on our approach. When subjects provided new demonstrations using our kinesthetic teleoperation method, they indicated the heaviest workload was mental, most likely because the robot's motion was a mirror image of the demonstrated motion. In the pre- and post-study UTAUT surveys, subjects became less afraid of breaking things when using the robot, suggesting the human-robot interactions during the study had a slight positive effect on their opinions about the robot. Importantly, the participants experienced multiple robot behavior modes and multiple task situations where the robot might have worked well or poorly, which may explain the lack of significant changes in other questions.

4.5 Summary

In this chapter, we developed a hierarchical structure for learning from demonstration, instantiated in a collaborative object-movement task. Both simulation and a user study show that by adding complexity in the learning framework, HS offered better performance and led to more favorable human perception toward the robot. In HS, the distance function helps to differentiate between demonstrations, and the morphing functions allow finer generalization as if more training data were present. Other benefits of HS include the ability to add or remove demonstrations without having to retrain the entire model, thus enabling online learning, and the ability to accommodate a wide range of demonstrations in the same framework. The latter could be essential. Imagine a robot walking forward sees an obstacle in front of it; some demonstrations may tell the robot to go left and some right, but the robot had better choose one or the other instead of averaging the two equally good options.

A generic indirect force regulation approach was used, because we focused on designing a demonstration procedure that's easy to use and on a learning framework that could learn efficiently from data so that users could easily customize the robot's behavior. Moreover, it is important for the robot to generalize and adapt to new task situations from limited amount of training data, and if the adaptation is too difficult (e.g., if the distance metric value is higher than a pre-determined threshold), the robot should stop execution and ask for more demonstrations. After all, following the framework into failure is exactly the type of robot behavior that LfD approaches aim to address.

This chapter thus lays the ground work for the next chapter, where we try to address the question at the end of Chapter 3: can a robot learn customized behavior and thus be able to regulate physical interactions with a person during therapy, so that robot designers get to simplify the design process and don't have to worry (too much) about parameter tuning?

Chapter 5

Learning Therapeutic Activities from Demonstration

Conventional approaches to robotic therapy typically ask the user to achieve a nominal metric: reach a target in a certain amount of time, follow a trajectory without deviating too much, etc. Thus makers of these therapy tools need to tune the design carefully to make the robot effective, potentially via extensive human-subject studies. Alternatively, they can leave the burden of tuning to the therapist, who then adjusts the device on the fly, but the therapist will have to understand how the robot works to adjust the parameters for their desired performance. The GPT in Chapter 3, for example, requires this parameter-tuning effort.

Thus in this chapter, we explore options to have the robot *learn* physical exercises with people. Following the training procedure in Chapter 4, a therapist holds the PR2's right arm to control its left arm, which holds a small object together with a stroke survivor to perform various exercise motions for the upper limb. Because the two robot arms are identical and provide force feedback, actions from the therapist can be recorded, and thus direct force regulation can be learned for certain types of exercises. On the other hand, an 'average' trajectory can also be obtained from the same demonstrations, so the robot can be used in the conventional approach to help the user track a trajectory over time. Chapter 5 describes a user study where an occupational therapist and six stroke survivors with upper-limb impairment provided demonstrations and tested the PR2 in these two learning approaches. Results show the new approach we introduced could be viable alternative in therapy.

After discussing related work in Section 5.1, we detail the methods of the user study with three types of exercises and two learning approaches in Section 5.2. Results are given in Section 5.3 and discussed in Section 5.4. Section 5.5 summarizes this chapter.

5.1 Related Work

Robot-assisted therapy could provide intensive and repeatable training, which can increase rehabilitation performance [122, 46]. Furthermore, robots can help patients train semi-autonomously [122], reduce the load on rehabilitation professionals, and augment their ability to provide care to patients [19]. In addition, robots can provide objective assessment of a patient's progress via onboard sensors, which is important and necessary for therapy [122, 68, 48]. Finally, robotics, among other technologies, makes therapy possible outside of clinics, such as collaborative tele-rehabilitation [65].

Various robotic therapy devices for upper-limb impairment have been developed, including MIT-MANUS [56], Gentle/s [66], Gentle/g [67], MIME [20, 69], and TheraDrive [49]. Rehabilitation robots have been shown to work well in these studies, but some argue that their results may be inconsistent and don't carryover to real life because they do not allow real tasks in the therapy environment [115]. In contrast, task-related therapy can lead to better results [116, 106]. First, studies using the ADLER device have found that human trajectories may be significantly different from the linear, minimum-jerk trajectories commonly used in point-to-point reaching when an object is present [50, 114]. This mismatch may partially explain the performance difference between typical robotic and task-related therapies. Second, many therapy HRI approaches are based on intuitive notions and ad-hoc controllers such as desired trajectories that are enforced [75]. But such enforced trajectories could lead to less effective motor training because humans learn better when movement variability is greater [101]. In addition, the traditional position error metric, which correlates negatively with performance, may not be optimal in cases where the robot could perfectly track the desired trajectories and the patient simply follows along [90]. Surface electromyography (sEMG) can thus be added to actively monitor participation level [90] so that degree of assistance from the robot can be adjusted. These results suggest that further research is required to determine the most effective robotic training scheme. It may be wise to look at human-human interactions (HHI), particularly therapist-patient interactions, to draw inspiration and guidelines for designing robot hardware and controllers. Perhaps rehabilitation robots should seek to emulate how human therapists behave in related tasks.

Jerasse et al. proposed a categorization of HHI: competition, collaboration, and cooperation, while cooperation can be further divided into assistance and education [45]. In this categorization, education is suitable to model therapy, but most existing HHI work is for collaboration (as in Chapter 4). Furthermore, general principles that could be translated to the design and use of rehabilitation robots have not yet been identified [101]. An early attempt to establish models defining physical therapist-patient interactions found that the inexpensive Kinect sensor is not accurate enough for studying human-human therapy kinematics [47]. In a follow-up study, kinematics (wrist positions and velocities of a therapist and a patient) obtained from a motion-capture system was used to calculate custom-designed interaction metrics, which indicate possible close interaction if below certain thresholds [81]. This approach is effective in determining when therapist-patient interaction (contact) happens, and thus it may be useful to collect nominal therapy behaviors as training data, but it does not address how a robotic therapist should behave during such interactions.

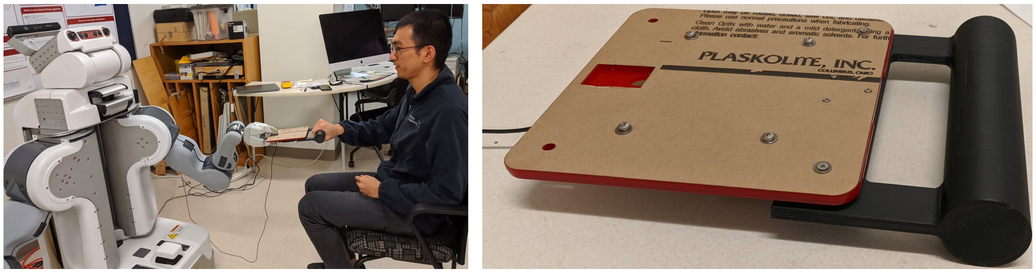
Apart from kinematics, haptic information between the therapist and the patient could offer insights in designing a robotic therapist, but literature examining the force interaction between individuals in cooperative sensorimotor assistance and education is lacking [101], which may be due to the challenge of finding a tactile sensor that does not interfere with interaction [81]. These facts suggest that although HHI is an active field of research, we still need a framework and more experimental data to understand how human therapists physically work with patients during rehabilitation [101] in order to guide the design of robotic therapists.

Thus instead of relying on mathematical models and principles of HHI, learning from demonstration (LfD) [9] may be a useful technique to guide robot controller design with how human therapists behave. For example, one approach uses dynamic motion primitives (DMPs) to model trajectories with an end-effector-type robot arm to assist with activities of daily living [63]. For exoskeletons, DMP has been extended with a neural network to learn and generalize anthropomorphic trajectories in joint space [62]. Limitations of DMP-based methods include that they learn motions in each degree-of-freedom (DOF) separately, require second-order derivatives to compute the forcing term, and are mostly limited to trajectory-retrieval tasks.

Other approaches exist that primarily utilize a Gaussian mixture model (GMM), which statistically encodes various types of data from demonstrations, and Gaussian mixture regression (GMR), which generalizes robot commands. Examples including learning Cartesian trajectories to track with an impedance controller [85, 84]. For this type of trajectory-learning task, Martinez and Tavakoli applied the SEDS extension [54] to learn stable dynamical systems of robot position [77], but only episodic motions/tasks were tested. To learn periodic motions, one approach obtains Fourier coefficients from a single trajectory, which are encoded with trajectories in polar coordinates in a GMM [76]. But this approach may require a large amount of training data to avoid overfitting because the number of data dimensions in the GMM may be high due to the number of Fourier coefficients.

The approaches mentioned above utilize GMM/GMR to model a variety of tasks, but they all explicitly or implicitly depend on time to parameterize trajectories. In contrast, Fong and Tavakoli used the difference in patient foot position and foot velocity as input to regress the robot's end-effector motion [31], thus accommodating different walking speeds and possible start and stop during training. In addition, the human-robot interaction force is typically implicitly regulated by an impedance controller in the above approaches, which requires impedance parameters that need to be experimentally tuned or learned from data. Explicit interaction force learning has been presented in [71], where a healthy person (therapist) provided demonstrations, and the time, position, and interaction force were encoded in a GMM. At autonomous execution, interaction force could be regressed based on time and position, and a PID controller tracked the force output using feedback from a force sensor. The GMM with SEDS extension in [77] could also be used for interaction-force learning, by encoding recorded force and its time derivative in the GMM. In either of these approaches, the system's dependence on time makes periodic tasks difficult to learn.

Following Chapter 4, we adapt our LfD framework to learn direct force output from the therapist, and we address periodic tasks by using the system state instead of time as the input to regression, as detailed in the next section.



(a) A user sits in front of the robot.

(b) The object used in this study. The black handle is for the human user, and the red rectangular depression on the other end is for the robot gripper

Figure 5.1: Example scenario of the user study.

5.2 Methods

We conducted a human-subject study to test whether robots utilizing learning-from-demonstration techniques could be a viable tool for occupational therapy for stroke, especially for periodic motions and learning therapist forces directly. In this study, the user sits in front of the robot for the exercises, as shown in Fig. 5.1a. We assume that the robot and the user have both grasped an object, shown in Fig. 5.1b, at the two ends before the exercise starts and that they can hold the object securely during the exercise. The object is approximately 380 mm by 180 mm by 30 mm in size and has a mass of 510 g.

The Penn IRB approved all experimental procedures under protocol 834211. All subjects gave informed consent. Stroke survivors each received 30 USD compensation, and the occupational therapist received 50 USD for participating in each session. Participants demographics are shown in Table 5.1.

5.2.1 Study Procedure

We conducted six study sessions, each lasting between 60 and 90 minutes depending on how quickly the subject finished the tasks (detailed below). Each stroke survivor listed in Table 5.1 participated in one study session, while the occupational therapist participated in all six sessions. At the beginning of each session, the stroke survivor participant did a Box and Blocks Test [79] and a Montreal Cognitive Assessment [86] to evaluate his or her upper extremity and cognitive impairment,

Table 5.1: Participant demographics in the user study. The BBT column lists scores of the Box and Blocks Test [79], where L and R represent data from the left and right arms, respectively. The MoCA column lists scores on the Montreal Cognitive Assessment [86].

	Age	Gender	BBT	MoCA	Other details
Stroke survivor 1	60	Male	L:56 R: 14	24	Had stroke in 2011.
Stroke survivor 2	52	Male	L:56 R: 45	18	Had stroke in 2016.
Stroke survivor 3	63	Female	L:50 R: 28	28	Had stroke in 2010.
Stroke survivor 4	73	Male	L:37 R: 32	29	Had stroke in 2008.
Stroke survivor 5	64	Male	L:43 R: 31	20	Had stroke in 2006.
Stroke survivor 6	56	Female	L:0 R: 42	28	Had stroke in 2003.
Therapist	26	Female	-	-	Has three months of professional experience.

respectively. Then in each session, both subjects filled in two UTAUT surveys to report their impressions of the robot at the beginning and the end of the session. They also each filled in a demographics survey at the end of the session.

For the main portion of each session, first the subjects provided demonstrations to the robot in three different physical exercises (the training phase), and the robot's joint-space position, velocity, and effort data were recorded using ROS. The exercises are detailed in Section 5.2.2. Then the demonstration data were processed and encoded by two learning approaches (see Section 5.2.4), which enable the robot to be autonomous when supporting people in the exercises. Next, the stroke survivor tested the autonomous robot in the three exercises and with the two learning approaches, while the occupational therapist monitored the interaction and noted how the stroke survivor responded to the exercises; we will refer to these activities as the first testing phase. Thus, the first testing phase of each session included six trials, covering the three exercises and two learning approaches for each exercise. Then the occupational therapist tested the autonomous robot while the stroke survivor rested, which we will refer to as the second testing phase. Similarly, there were six trials in the second testing phase.

Finally, the three above phases (training, first testing, and second testing) were repeated for one exercise, whose type was determined by the therapist. This additional portion of the study allows the therapist and the stroke survivor to re-experience the study activities after becoming familiar with the exercises and how the robot functions; we call these phases retraining, first retesting, and second retesting. Thus, a comparison between the first interactions and the retraining ones helps us evaluate how familiarity could affect the efficiency of training with the robot.

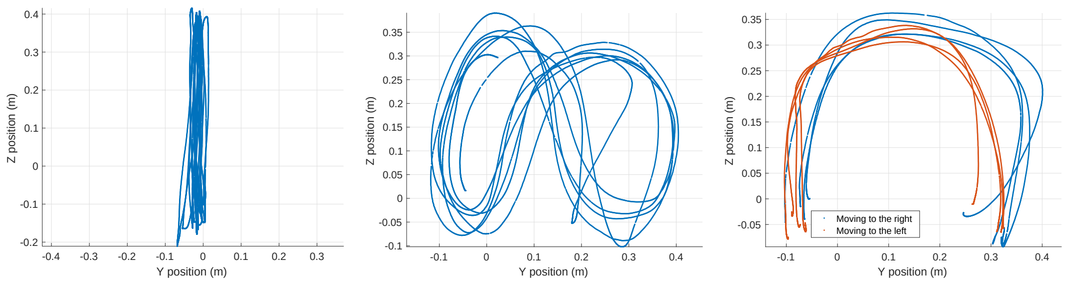
For the six sessions, the presentation order of the three motions was randomized and balanced, and the same order was used in the training and testing phases in each session. For each motion, the presentation order of the two learning approaches was randomized and unknown to the participants. In other words, during autonomous robot execution, the participants knew which exercise they should do but not what learning approach the robot was using.

After each human-robot interaction in the training and testing phases, participants filled in questionnaires evaluating the interactions, which are detailed in Section 5.2.6.

5.2.2 Exercises

We considered three types of exercises for their practicality in occupational therapy, so that the study would simulate a simplified but realistic clinical scenario. In addition, these types of exercises were chosen for their simplicity, so that subjects who were not familiar with the robot could quickly understand how to interact with it. Subjects could then focus on the exercises instead of how the robot works. Finally, we chose to include both periodic and episodic motion, as detailed below.

For the first type of exercise, the participant needed to move the object up and down vertically (1D), as shown in Fig. 5.2a. For the second type of exercise, the participant needed to hold the object and draw an infinity (∞) shape (2D) in the frontal plane, as shown in Fig. 5.2b. For the third type, the participant needed to raise the object up for a short distance, move it to the left or the right for a short distance, and then lower it, thus mimicking picking and placing (PP) an object,



(a) 1D exercise. **(b)** 2D exercise. **(c)** PP exercise.

Figure 5.2: Example demonstration trajectories for the three types of exercises, expressed in the robot shoulder frame. Only y (horizontal) and z (vertical) dimensions are shown for clarity. Movements in the x direction (depth) were of much smaller magnitude throughout the study.

which is an activity of daily living, as shown in Fig. 5.2c. Thus the 1D and 2D exercises are periodic motions, and the PP exercise is episodic with start and goal positions in each repetition. These exercises require the participant to use their arm and shoulder to move the object and their fingers to grasp it, thus activating their impaired upper limb.

In all three exercises, there was no visual feedback regarding where or when the object should be moved; instead, only the general shape was mentioned. The instruction to participants was to do the exercises as well as they could.

5.2.3 Demonstrating Procedure

The procedure in Chapter 4.4.2 was used to collect training data for the robot for each of the three exercises considered in this study. The therapist acted as the teacher and held the robot's right arm, and the stroke survivor acted as the partner and held the object with the robot's left arm. Force feedback between the two arms enabled the therapist to adjust robot motion given how well the stroke survivor was doing during the exercises.

In the study, each demonstration generally lasted until both participants understood the motion well and they could finish the exercise smoothly with the robot, with the exceptions of the second stroke survivor who couldn't understand the motion in the 2D exercise and the sixth stroke survivor who couldn't complete the 2D exercise (see Section 5.3 for details). The time it took the therapist and the stroke survivor to provide each demonstration ranged from 15.1 s to 57.3 s ($\mu = 30.9\,\text{s}$, $\sigma = 9.0\,\text{s}$).

5.2.4 Learning Approaches

Conventional robotic therapy tools typically have a desired trajectory or target that the client should track or reach, and the robot's output is a function of the tracking error, e.g., how far the user deviates from the trajectory or how fast the user moves [56, 69, 66, 50]. Thus, the first modeling approach we used in the study aimed to find an average trajectory from the demonstrations that the robot should follow during autonomous execution with fixed gains. This approach will be referred to as 'time-based' in the remainder of this thesis, because the robot tries to follow time-parameterized trajectories.

However, with the above approach it is possible for the client to simply follow the robot motion and do little to no work, so that the exercises are finished (by the robot alone), but the rehabilitation would not be effective. Therefore, in some other approaches, the robot provides assistance as needed; for example, the robot could adjust the tracking gain if the person fails to execute a trajectory after some predefined time period [70]. However, the gain values and the desired trajectory still need to be defined before the exercise starts. Thus, to overcome this issue, the second approach we used in the study aimed to learn reactions directly from the therapist during the demonstration procedure, instead of assuming a conventional model (e.g., a second-order impedance model or a PID model). This approach will be described as 'state-based' because the robot regresses the therapist's reaction based on the current robot state, including position and possibly velocity, at each time step.

Given that the 1D and 2D exercises are periodic motions that could last an indefinite period of time and that the PP exercise is episodic and has meaningful start and goal positions, we designed different procedures for the two approaches based on the exercise type, as explained below.

1D and 2D Exercises

The trajectories for these two types of exercises are periodic; therefore, we use sinusoidal waves to fit the trajectories in each dimension for the time-based approach. For example, given a trajectory x_t, the following optimization problem can be solved:

$$\arg\min_{\Theta} \sum_{t} ||o + A\sin(\omega t + \phi) - x_t||^2 \tag{5.2.1}$$

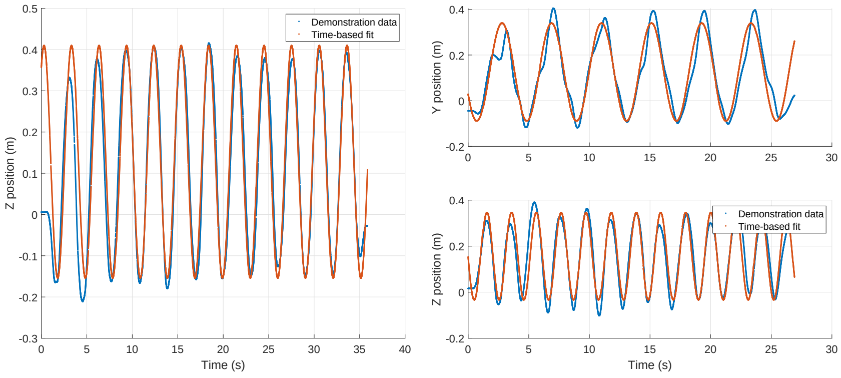
where $\Theta = \{o, A, \omega, \phi\}$ are the offset, amplitude, frequency, and phase parameters of the sinusoidal trajectory. Fig. 5.3 shows results from example demonstrations of 1D and 2D exercises.

For the state-based approach, we use the GMM/GMR approach in Chapter 4 for each relevant dimension (y or z directions in the robot-frame). The input is chosen to include the robot end-effector's position and velocity, which represent the state of the human-object-robot interaction because we assume a rigid grasp of the object from the robot and from the user. The output is chosen to be the robot's effort, which is controlled by the therapist during demonstrations using joint-level feedback, to ensure that the robot could exert similar work during autonomous execution. Fig. 5.4 shows examples of fitted Gaussian clusters; it can be seen that the position and velocity dimensions resemble a typical harmonic oscillator's phase plot, and the effort dimension depends on both the position and velocity dimensions.

PP Exercise

For PP exercise, we use a similar procedure for the time-based approach as the one in Section 4.4.3 that derives $\mathbf{x}_{\text{avg}}$: demonstration trajectories are used to train a GMM with time as input. Note that to avoid placing too many data points where the object is moved slowly, as in Chapter 4, we resample time and the trajectories based on trajectory length. Fig. 5.5 shows results from example demonstrations of PP exercise.

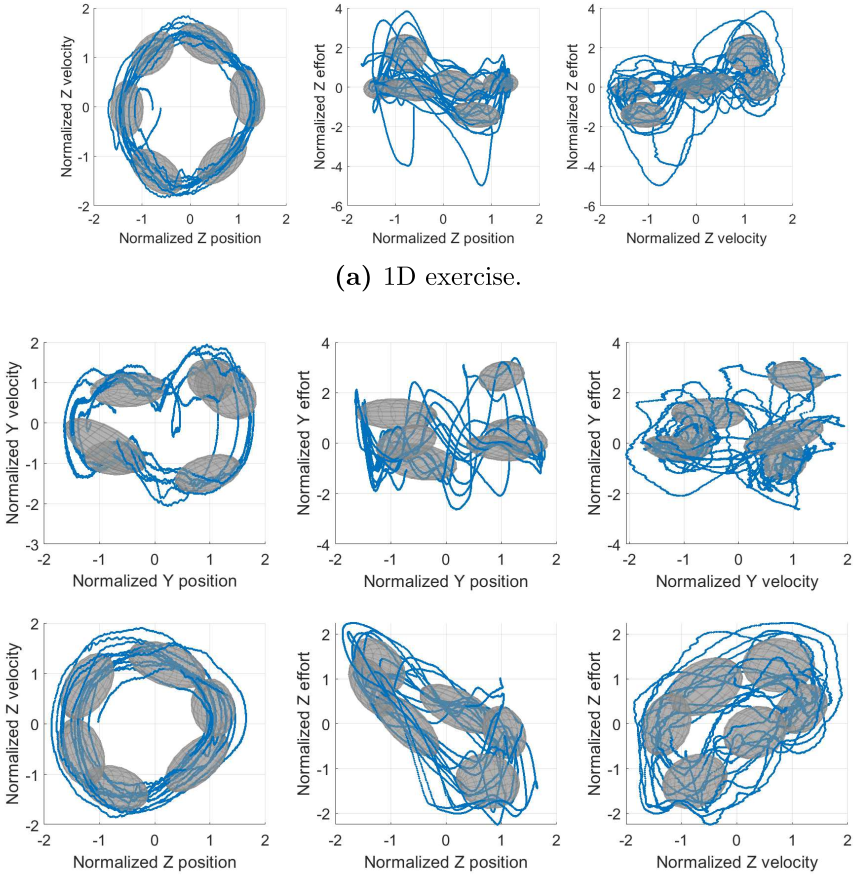
For the state-based model, we use the procedure in Chapter 4.4.3 that models $\mathbf{x}_{\text{des}}$: the output is $\mathbf{x}_{\text{des}}$, which is $\mathbf{x}_{\text{avg}}$ shifted forward in time to make the robot appear more active, and the input is the robot's current position $\mathbf{x}$, as shown in Fig. 5.6. Here the robot velocity is dropped from the robot's state for the PP exercise because the motions are largely uni-directional. Reducing the number of dimensions also helps to avoid overfitting. However, note that in other types of exercises that mimic ADL, velocity may be added to represent the state more accurately.



(a) 1D exercise.

(b) 2D exercise.

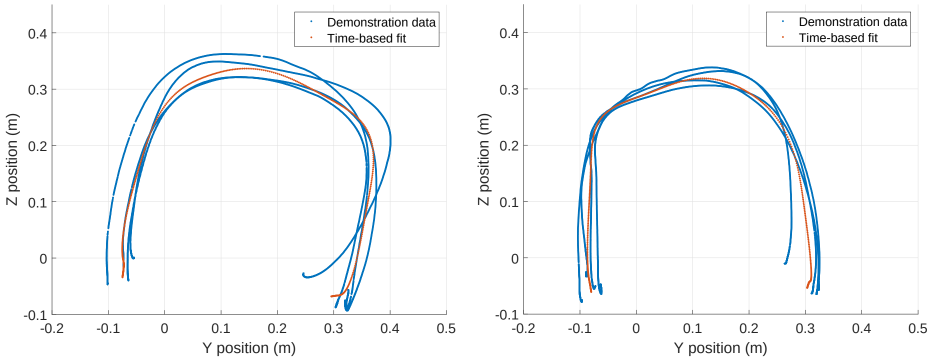
Figure 5.3: Example time-based (sinusoidal) fit of demonstration data for the 1D and 2D exercises.



(a) 1D exercise.

(b) 2D exercise.

Figure 5.4: Example state-based fit of demonstration data for the 1D and 2D exercises. The ellipses represent Gaussian clusters. Training data is normalized by standard deviation after subtracting the mean.



(a) Moving to the right.

(b) Moving to the left.

Figure 5.5: Example time-based fit of demonstration data for the PP exercise.

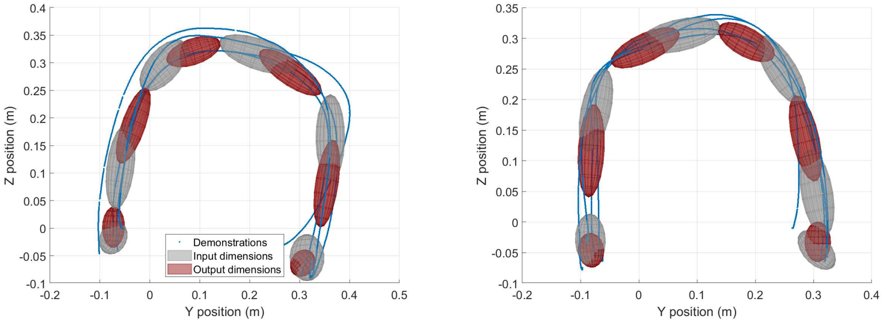
5.2.5 Robot Controller during Autonomous Training

At test time, the PD joint torque controller from eq. (4.4.3) is used in the state-based approach for PP exercises and in the time-based approach for all exercises, because the robot tracks a desired position. $\mathbf{K}$ is chosen to be slightly lower at $100\,\mathrm{N\,m^{-1}}$ to make the robot softer, given that the majority of subjects have some degree of motor impairment, and $\mathbf{B}$ is kept as the same value of $10\,\mathrm{N\,s\,m^{-1}}$ to slightly dampen the motion. In the state-based approach for 1D and 2D exercises, since the output is robot effort in the Cartesian frame, joint torque can be directly calculated using the Jacobian, assuming quasi-static conditions.

Crucially, because the time-based approach tracks a trajectory and because the object's position can vary at the beginning of each execution, it is necessary to adjust for this discrepancy so that the robot does not exert too much power at the beginning. Thus, for 1D and 2D exercises, we discard the phase value obtained from the training data and calculate it based on the robot's x position at the beginning of each execution:

$$\phi = \begin{cases} \frac{\pi}{2}, & x - o > A \\ -\frac{\pi}{2}, & x - o < -A \\ \arcsin(\frac{x-o}{A}), & \text{otherwise} \end{cases} \tag{5.2.2}$$

where o is the offset and A is the amplitude obtained from training, and ϕ is the new phase variable.



(a) Moving to the right.

(b) Moving to the left.

Figure 5.6: Example state-based fit of demonstration data for the PP exercise. For visualization, each Gaussian cluster is broken down, and the input dimensions and the output dimensions are plotted separately, while the cross-terms in the covariance are not plotted. It can been seen that the output dimensions are generally ahead of the input dimensions with respect to the movement directions; therefore the robot will appear active and lead the movement.

For the PP exercise, with the time-based approach, we pre-compute the trajectory to track (see Fig. 5.5) based on a set of linearly sampled time steps. Then before autonomous execution starts, the point on the pre-computed trajectory that is the closest to the current robot position is found, and the time stamp associated with the closest point is used as an offset during execution, so that the robot tracks the part of the desired trajectory starting from that point.

5.2.6 Dependent Measures

We recorded robot motions during each trial, and quantitative measures were obtained in post-processing to evaluate how the learning approach affected the exercise interactions:

- Acceleration: the ratio between the mean acceleration and the peak acceleration, which can be used as a measurement of motion smoothness [80].
- Excursion: the range of motion exercised. Excursion is measured as the difference in height between the highest and the lowest points reached in each trial for 1D exercise, the area enclosed by the ∞-pattern for 2D exercise, and the average trajectory length for PP exercise.

- Power: the mean of the dot product of robot effort and velocity. This value measures how much power output is generated from the robot; a very high power output means that the robot does most of the work in the exercises, and a negative power output means that the robot works against the person and hinders motion.

- Speed: the mean speed at which the object was moved during each exercise.

- Frequency: how many repetitions are done on average, i.e., how much therapy 'dosage' can be achieved, per second. Because motions during the exercises were not exactly uniform, we used the highest point reached in the 1D exercise and the most-left point in the 2D exercise to determine repetitions. For the PP exercise, each interaction (raising, moving, then lowering the object) counted as one repetition.

Participants also filled in questionnaires during each phase of the study session. A UTAUT survey was completed at the beginning and the end of each session to evaluate whether and how participants' perceptions of the robot changed after having interacted with it.

After each demonstration in the training phase, the therapist filled a NASA-TLX survey to report the workload. In the meantime, the stroke survivor rated how much they agree with each of the following statements (on a continuous scale from strongly disagree to strongly agree):

- Q1. The robot appeared to do the *right* thing at the right time.

- Q2. The robot *worked well* in practicing the exercise with me.

- Q3. I *felt safe* when working with the robot.

- Q4. The robot moved *too fast.*

- Q5. The robot moved *too slowly.*

In addition, the stroke survivor also rated their feelings on the following scales after practicing the exercise with the robot, based on the Self-Assessment Manikin [18]:

- F1. Unhappy/sad – *Happy*/elated

- F2. Calm/bored – *Stimulated*/involved
- F3. Not in control – *In control*/dominant
- F4. *No pain* – Most severe pain imaginable
- F5. *No exhaustion* – Complete exhaustion

In the first testing phase, one statement was added to the first part of the above survey:

- Q6. The robot behavior appeared similar to when it was controlled by the therapist.

The stroke survivor filled in the survey questions (Q1-Q6, F1-F5) after each trial. The therapist answered the same questions at the same time, but the wording was slightly changed to indicate that she was asked to rate the stroke survivor's interactions with the robot based on visual observations. At the end of the first testing phase, the therapist was additionally asked to rate which interaction was the best and which was the worst of the six. Note that since the therapist did not know the details of the learning approaches, the answers would refer to the overall ordering, e.g., 'the first interaction was the best, and the fourth one was the worst'. In the second testing phase, the therapist answered Q1-Q6 and F1-F5 after practicing the exercise with the robot in each trial.

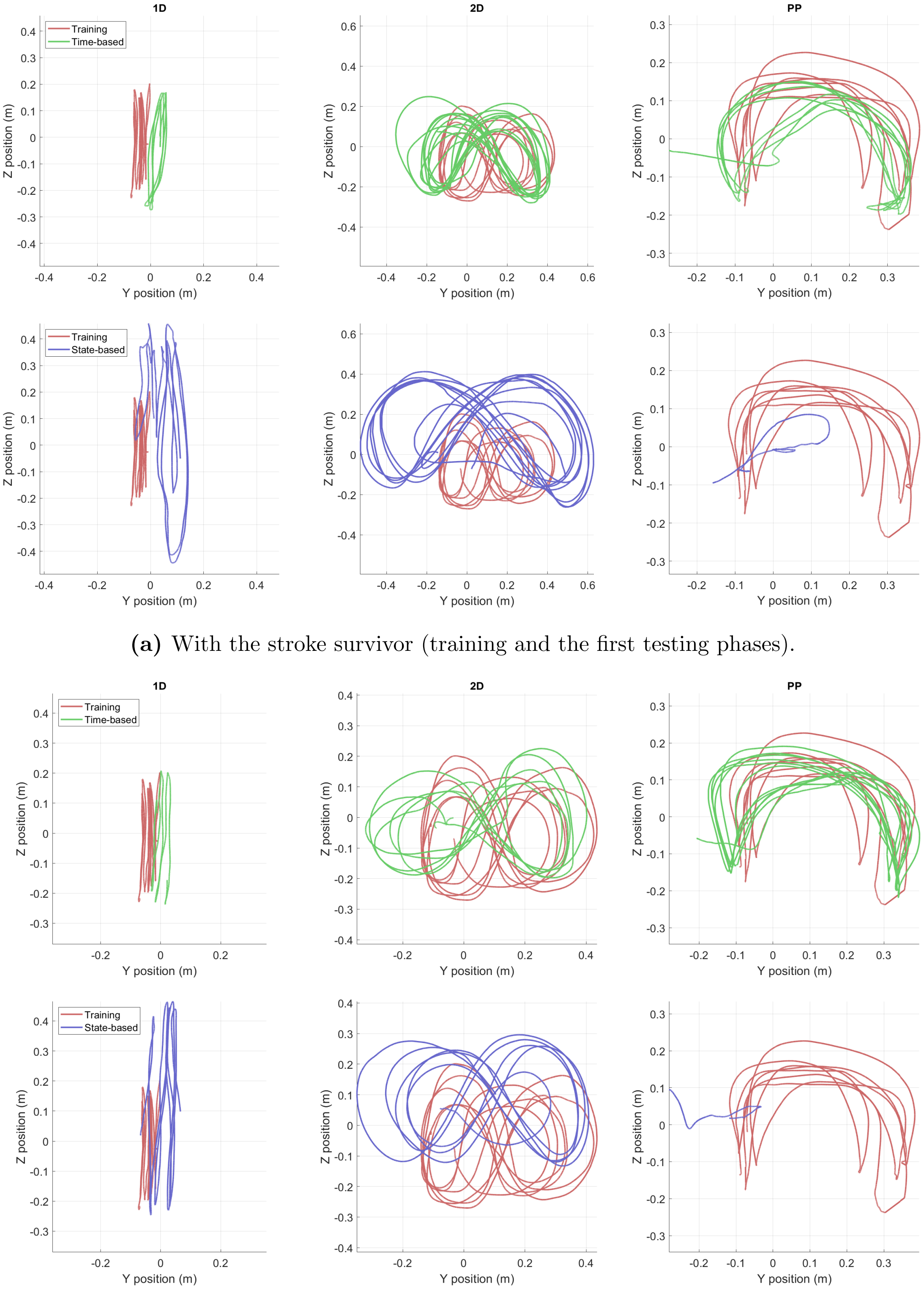
After the second testing phase, the therapist decided which motion to retrain and answered the following questions:

- R1. Which movement would you like to retrain?
- R2. Why do you choose to retrain this movement?

The same questionnaires from the training and testing phases were used in the corresponding retraining and retesting phases.

5.3 Results

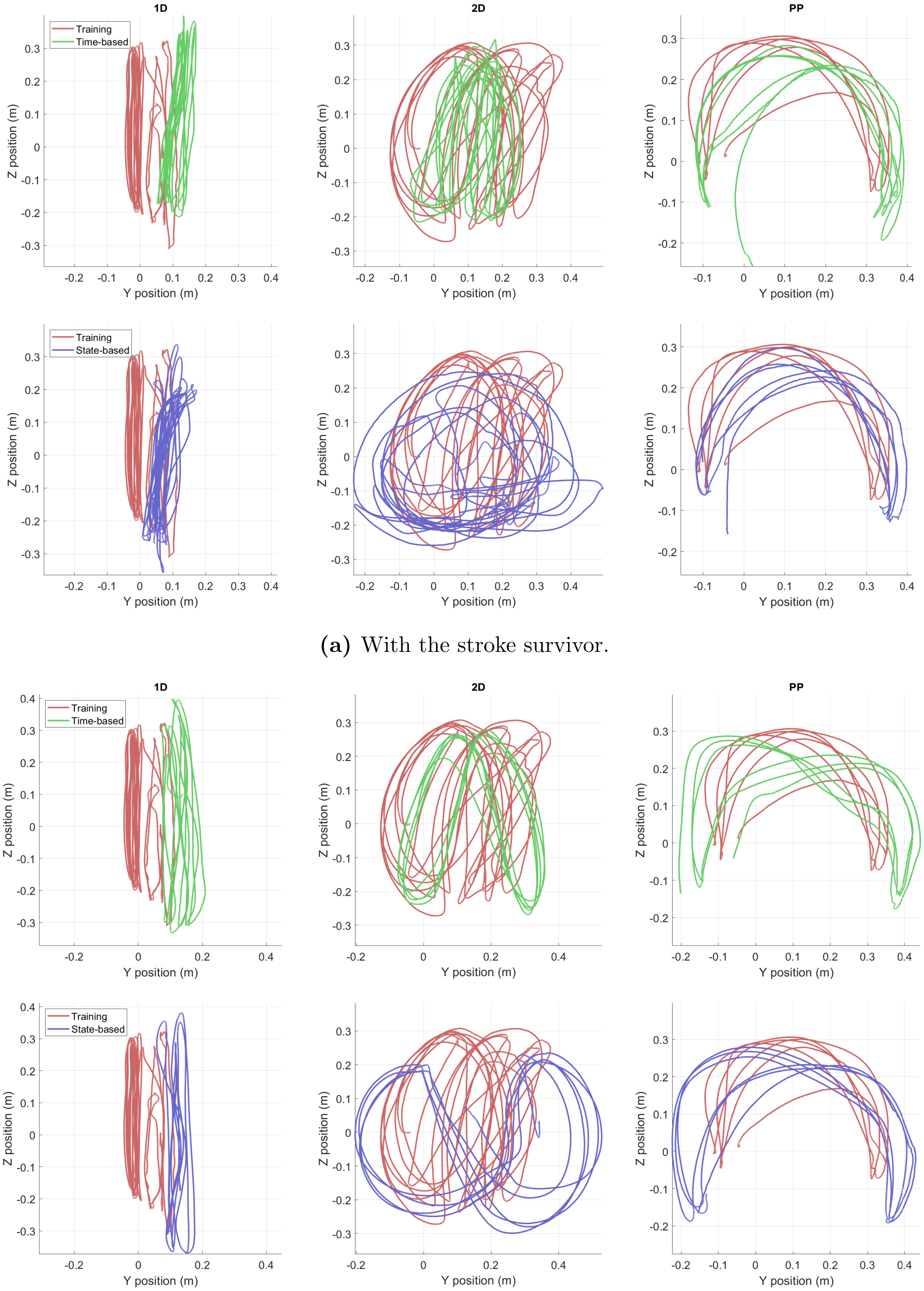
First we calculated the normalized Box and Blocks Test (nBBT) scores using the average performance of adults in the same age group [79] for each stroke survivor



(a) With the stroke survivor (training and the first testing phases).

(b) With the therapist(training and the second testing phases).

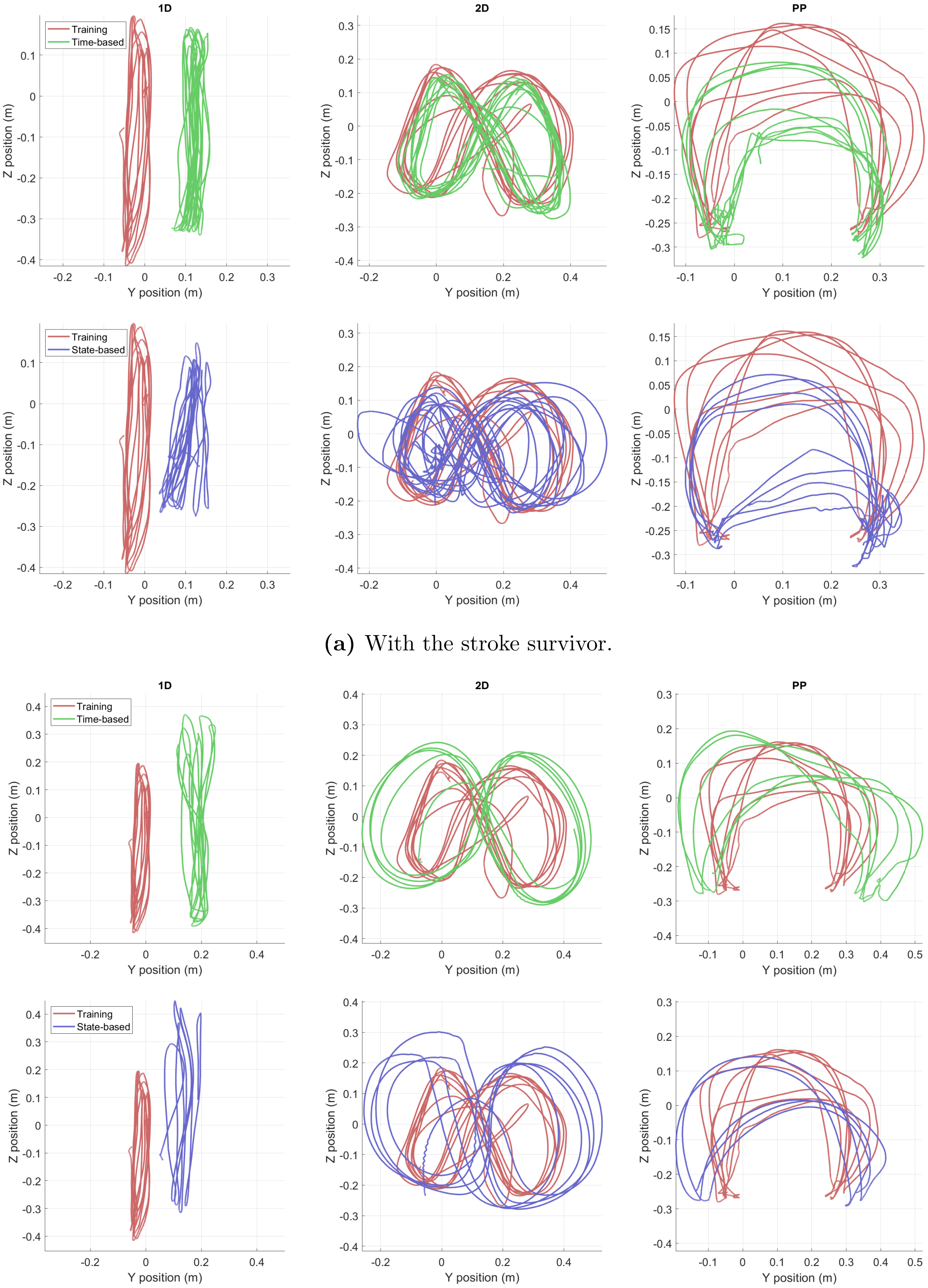
Figure 5.7: Trajectories in the first study session.



(a) With the stroke survivor.

(b) With the therapist.

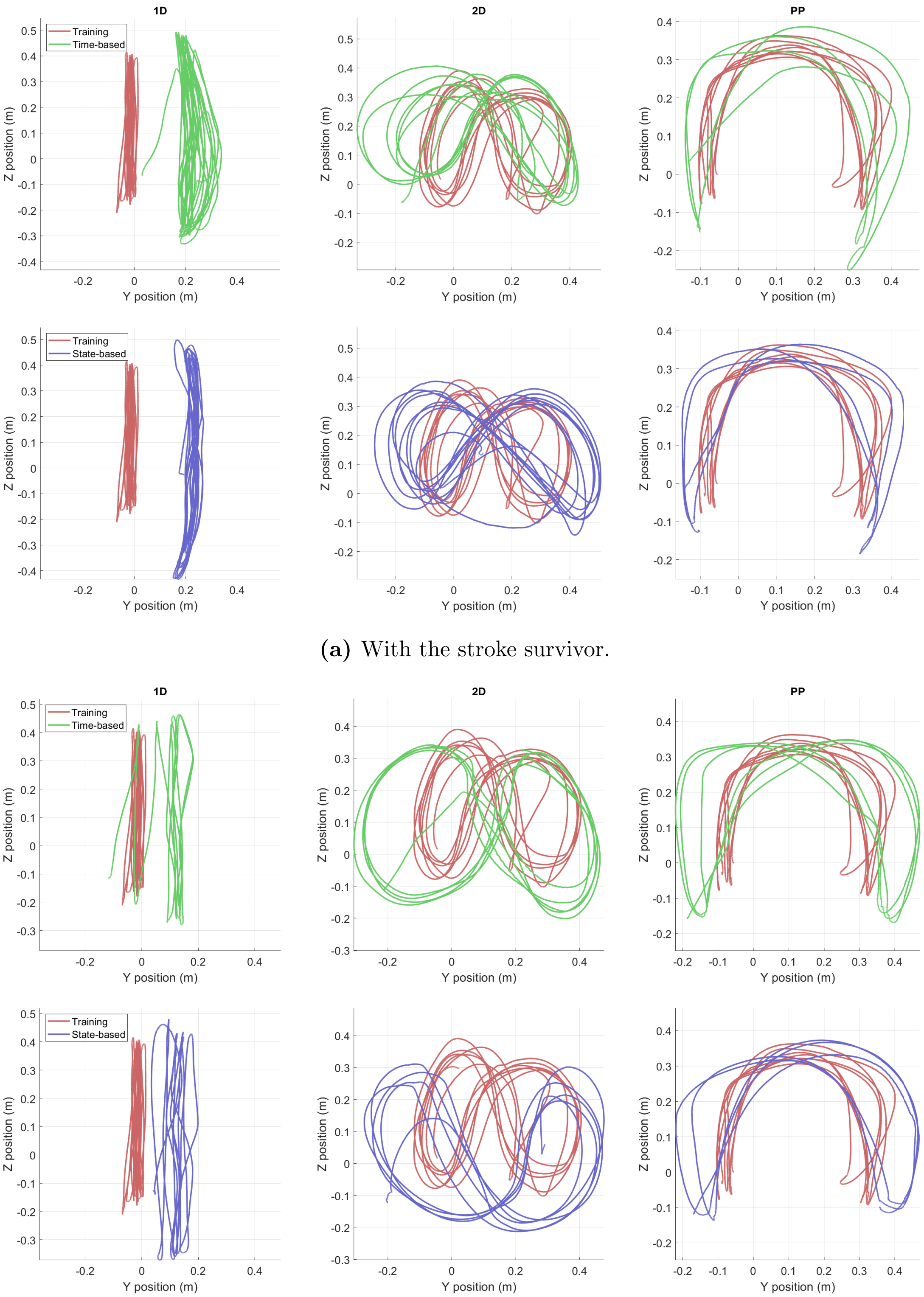
Figure 5.8: Trajectories in the second study session.



(a) With the stroke survivor.

(b) With the therapist.

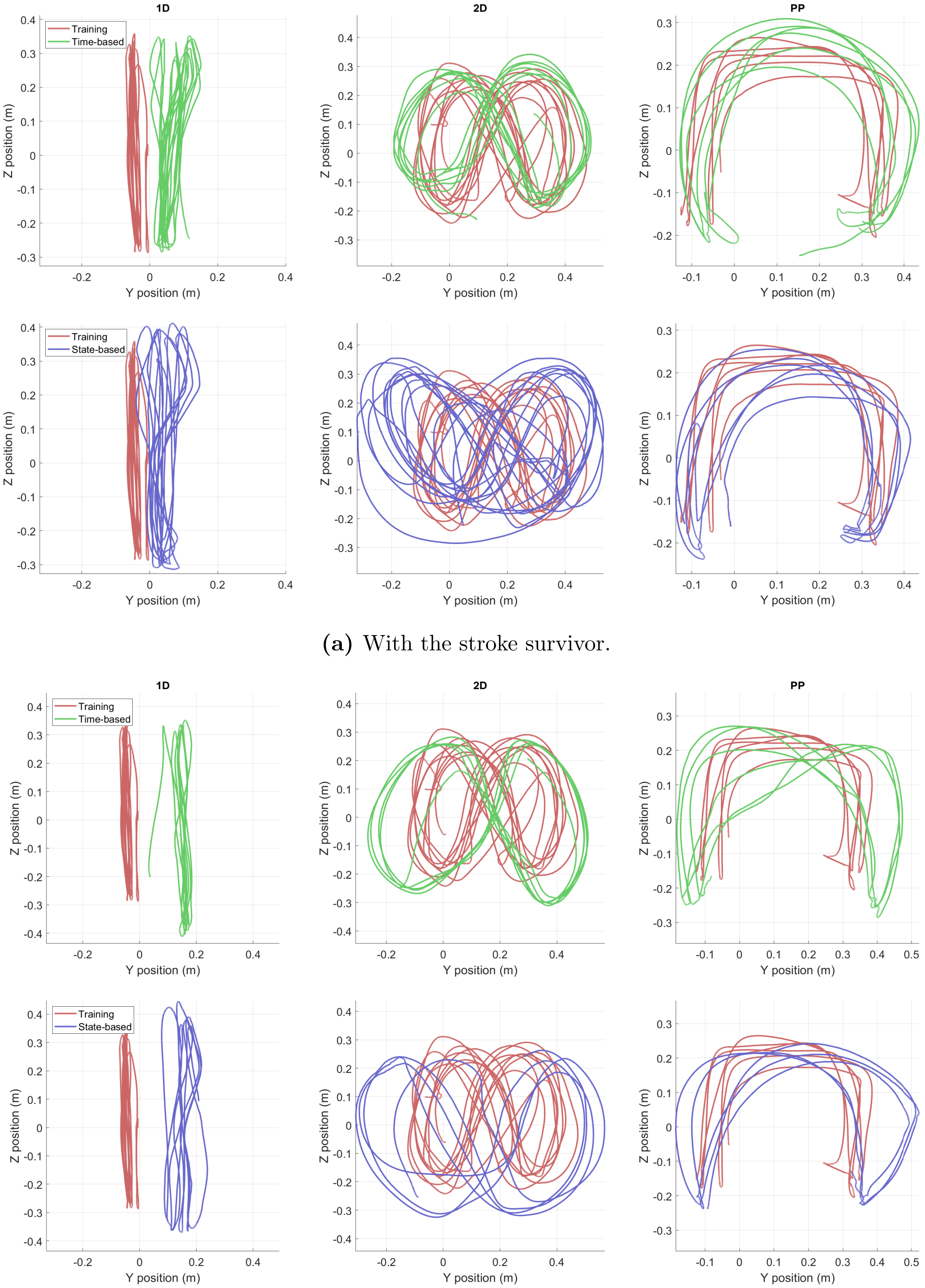
Figure 5.9: Trajectories in the third study session.



(a) With the stroke survivor.

(b) With the therapist.

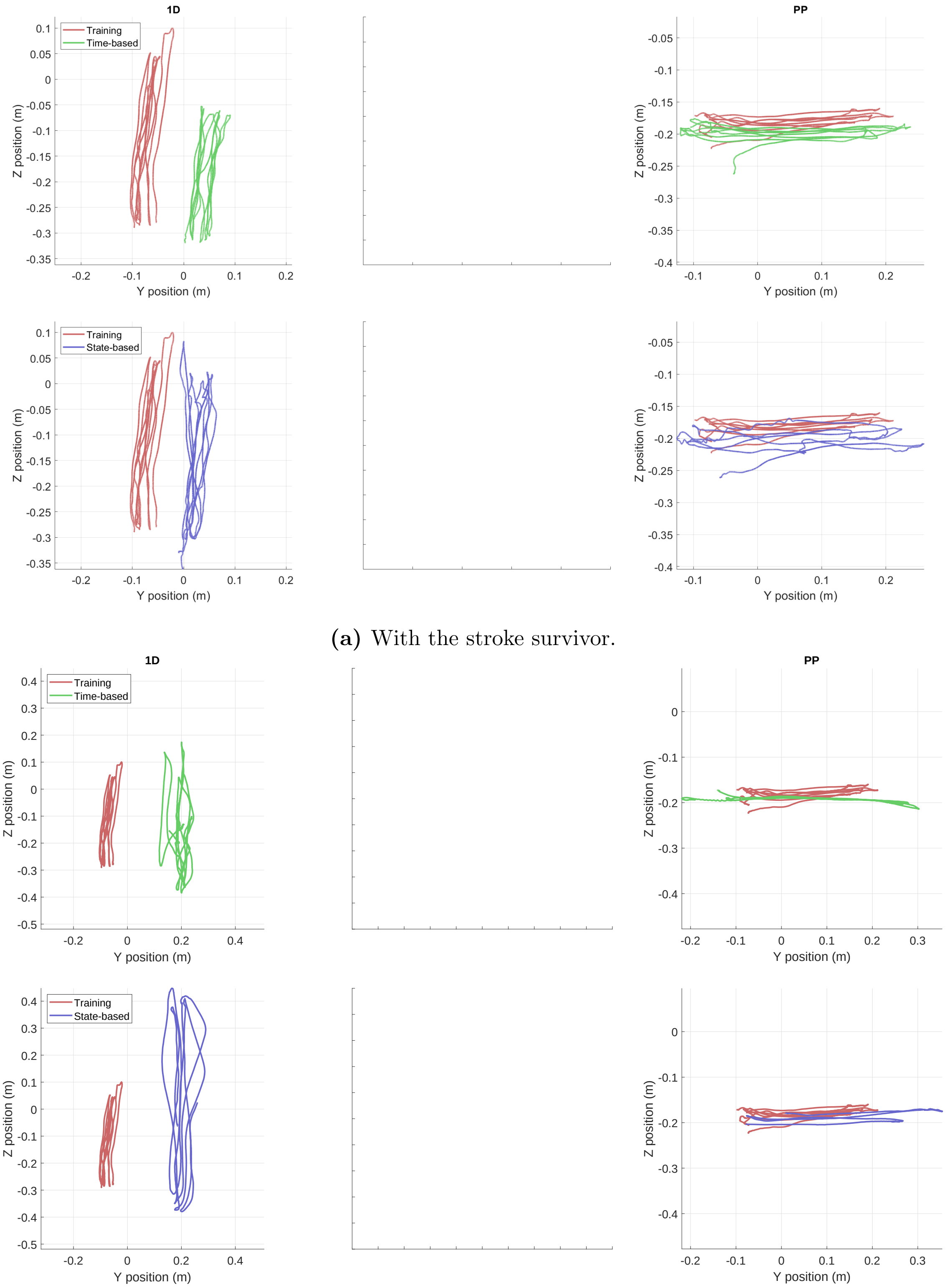
Figure 5.10: Trajectories in the fourth study session.



(a) With the stroke survivor.

(b) With the therapist.

Figure 5.11: Trajectories in the fifth study session.



(a) With the stroke survivor.

(b) With the therapist.

Figure 5.12: Trajectories in the sixth study session.

Table 5.2: nBBT scores of the participants.

Participant	nBBT score
Stroke Survivor 1	0.20
Stroke Survivor 2	0.58
Stroke Survivor 3	0.37
Stroke Survivor 4	0.49
Stroke Survivor 5	0.44
Stroke Survivor 6	0
Therapist	1

subject as a measure of their impairment level. Because the therapist has no upper-limb impairment, we assign 1 as her nBBT score. The results are shown in Table 5.2.

Figs. 5.7 to 5.12 show the trajectories of the robot's end-effector during training, first testing, and second testing phases for each of the six sessions. The participants were able to complete most exercises in both training and testing phases. However, in the first session, the demonstration data was not consistent enough during the PP exercise, possibly due to the unfamiliarity of the robot to the therapist at the time. The model learned for moving to the right is shown in Fig. 5.13. Compared to the typical and functional models shown in Fig. 5.6, the clusters in Fig. 5.13 are not aligned approximately evenly along the trajectories, which caused some clusters of the output dimensions to overlap and consequently the robot to appear stuck during autonomous testing. Therefore the state-based learning approach could not generate a functioning controller for PP in the first study session, and the corresponding testing phase was skipped (see lower-right panels in Fig. 5.7a and 5.7b). In addition, the sixth stroke survivor had relatively severe motion impairment and could not complete the 2D exercise in either training or testing phases; the center panels in Fig. 5.12 are consequently blank. Thus, data of the respective exercises from these two sessions were not included in the results (e.g., Fig. 5.14a lack a data point for the nBBT score of 0 in the 2D exercise and a data point for the nBBT score of about 0.2 in the PP exercise). The PP exercise for the sixth stroke survivor was reduced to largely horizontal movements due to the motion impairment, but the quantitative metrics were still used in the analysis because the participant could

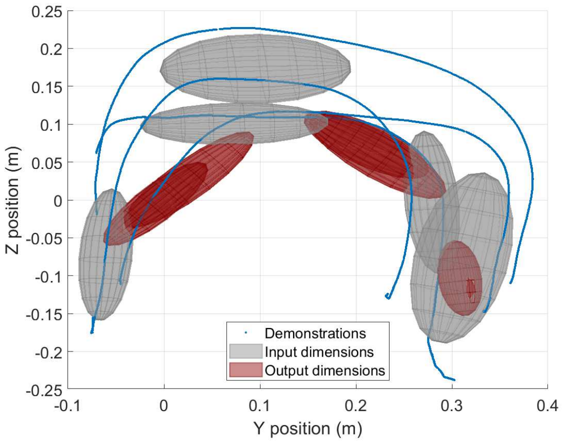


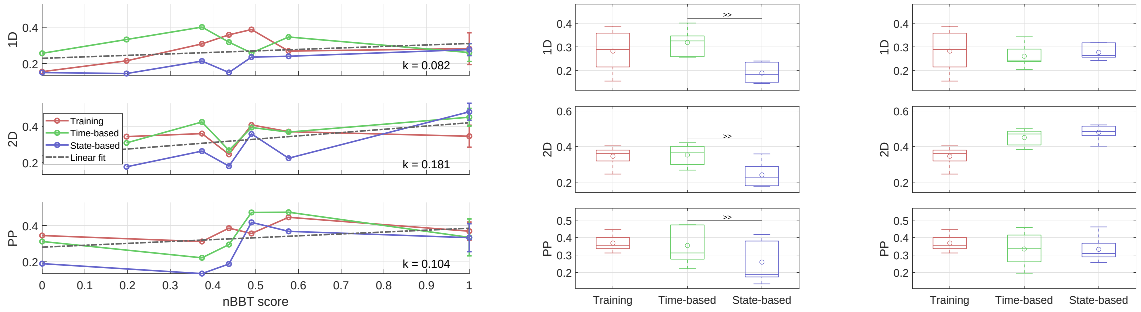
Figure 5.13: State-based model for PP (moving to the right) from the first session, where the clusters were not distributed evenly.

finish the exercise.

Figs. 5.14 to 5.18 show the quantitative metrics for each of the study sessions. Since the therapist attended all six sessions, the means across the sessions were used for the line plot, and standard deviations were added as error bars when plotting each metric as a function of the nBBT score in Figs. 5.14 to 5.18. A linear fit is added for each metric, and the slope of the line, k, represents the general trend of how the metric changes as nBBT score increases (the user is more capable). In the box plots, data from all of the sessions were grouped according to the study condition (training, time-based model, and state-based model), and Fisher-Pittman permutation tests were run for each metric to compare the two learning approaches.

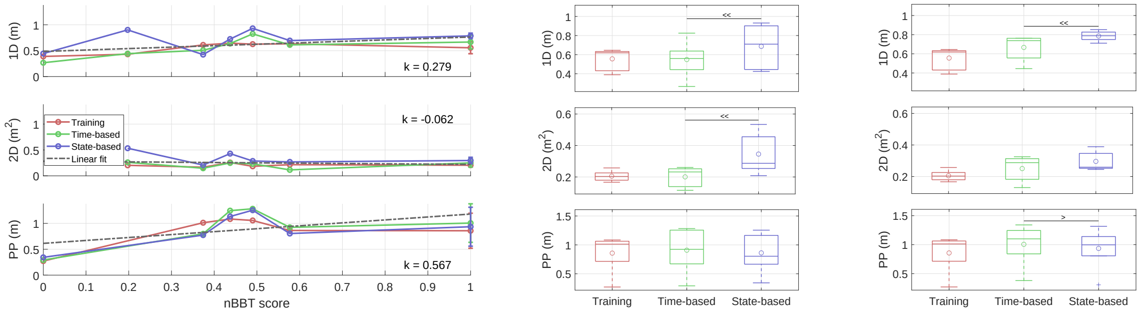
Questionnaire answers from stroke survivors in the training and first testing phases of the study are shown in Fig. 5.19; answers from Q4, Q5, F4 and F5 were reversed so that higher rating is better for all questions in the figure. Fisher-Pittman permutation tests were run to compare the two learning approaches (T1 and S1 groups). Note that Q6 was asked only in the testing phase; therefore Fig. 5.19 has no data for Tr in the Q6 group.

The therapist filled in the NASA-TLX survey in the training phase of each session, and the results are shown in Fig. 5.20. The overall ratings are consistent across the types of exercises: physical demand and effort are the highest, mental demand is medium, and others are low (note that low ratings for performance mean



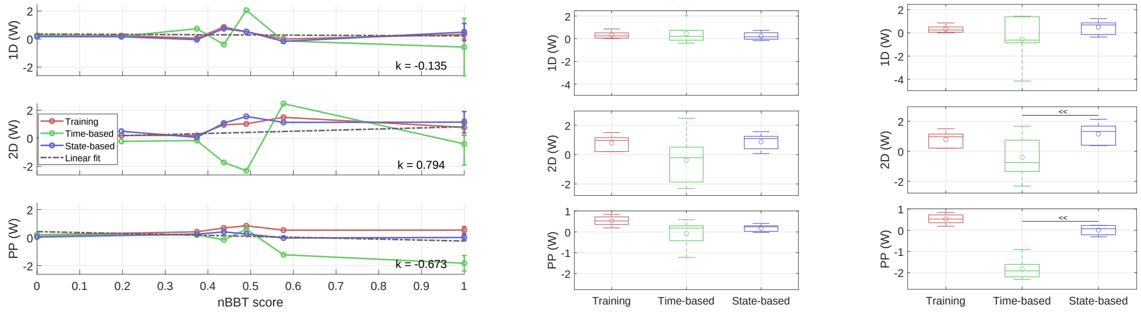
(a) As a function of the nBBT score. **(b)** Stroke survivor data. **(c)** Therapist data.

Figure 5.14: Acceleration metric (higher is better). In Fig. 5.14a, the slopes of linear fit (k) are recorded in each panel. In Figs. 5.14b and 5.14c, if a significant difference between the time-based and state-based groups was found, a horizontal bar and either '>' or '<' symbols were added on the graph to indicate the direction of the difference: one symbol represents the significant difference was found with $p < 0.1$, while two symbols represent $p < 0.05$.



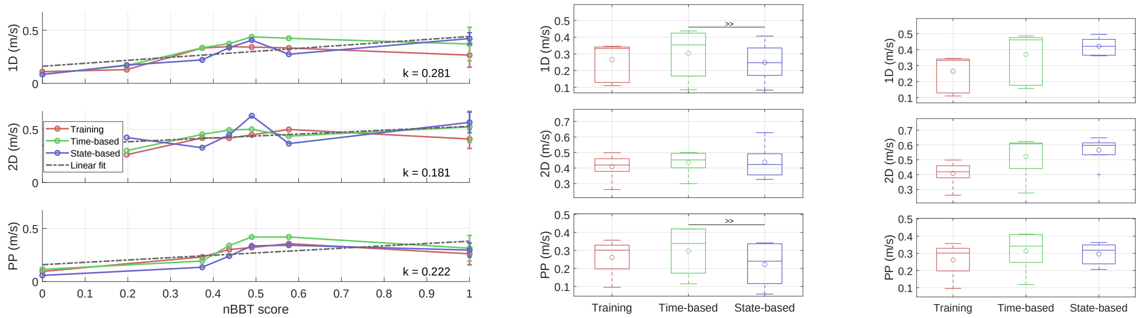
(a) As a function of the nBBT score. **(b)** Stroke survivor data. **(c)** Therapist data.

Figure 5.15: Excursion metric (higher is better).



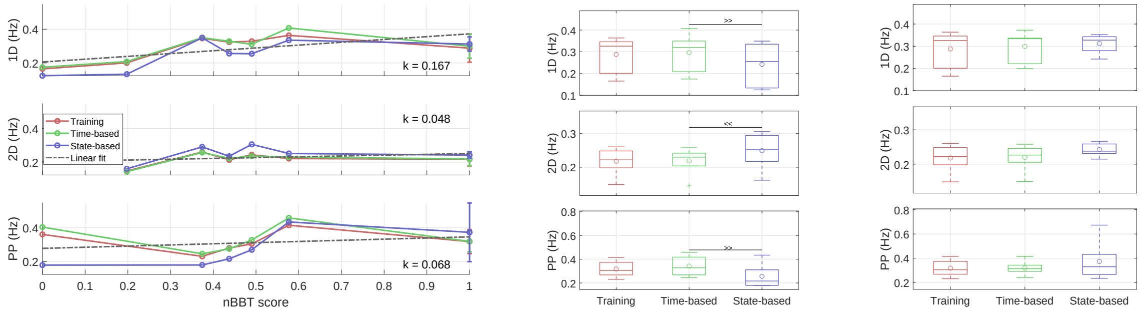
(a) As a function of the nBBT score. **(b)** Stroke survivor data. **(c)** Therapist data.

Figure 5.16: Power metric (negative value means robot resists motion).



(a) As a function of the nBBT score. **(b)** Stroke survivor data. **(c)** Therapist data.

Figure 5.17: Speed metric (higher is better).



(a) As a function of the nBBT score. **(b)** Stroke survivor data. **(c)** Therapist data.

Figure 5.18: Frequency metric (higher is better).

good performances). In particular, frustration ratings from the 2D exercise appear higher, largely because the second stroke survivor had difficulty understanding the infinity (∞) motion pattern.

Ratings from the first and the second testing phases are shown in Fig. 5.21, where the therapist observed the exercises by stroke survivors and the robot (T1 and S1 in Fig. 5.21) or interacted with the robot herself (T2 and S2 in Fig. 5.21). Answers from Q4, Q5, F4 and F5 were similarly reversed so that a higher rating is better for all questions in the figure.

In addition, subjective rating sums for each exercise were calculated as an evaluation of the overall sentiment toward the exercises with the robot. The maximum value was 1100 (11 questions each with a maximum of 100). Fig. 5.22 shows the sums after the stroke survivors practiced the exercises with the robot, and Fig. 5.23 shows the rating sums given by the therapist. Table 5.3 shows the selections given by the therapist regarding which interaction was the best and which was the worst

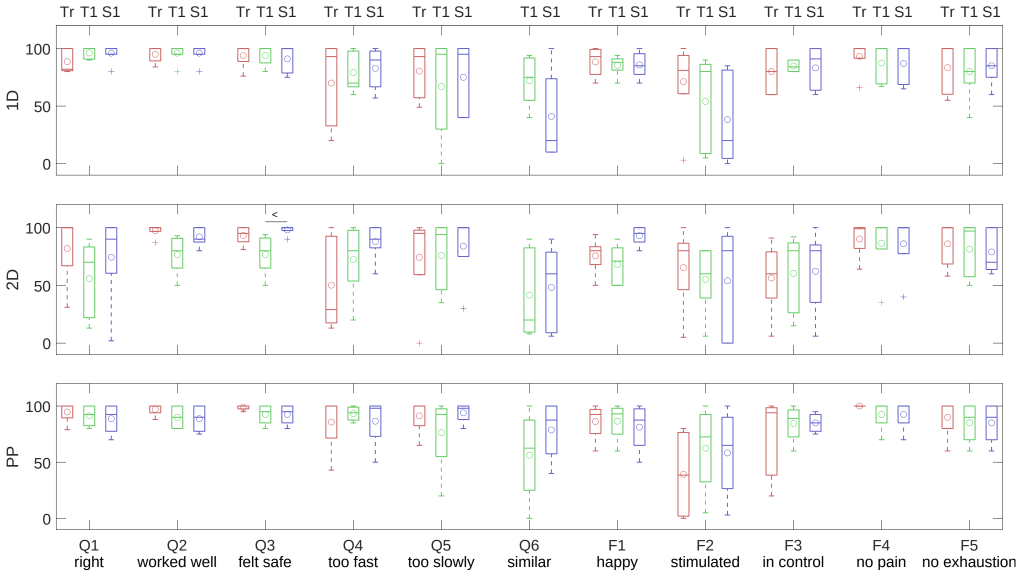


Figure 5.19: Questionnaire ratings by stroke survivors (higher is better). ‘Tr’ represents data from the training phase, and ‘T1’ and ‘S1’ represent when the robot was using the **T**ime-based or **S**tate-based approach, and the number 1 indicates that these answers were given in the first testing phase.

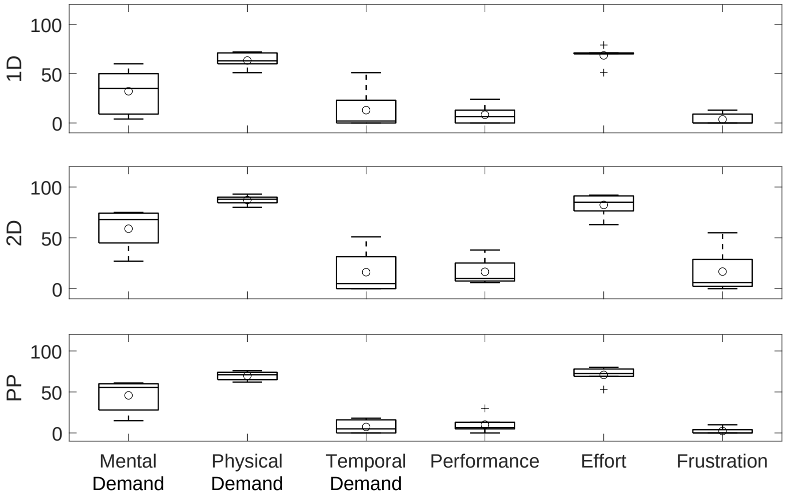


Figure 5.20: NASA-TLX ratings by the therapist for providing demonstrations (lower is better).

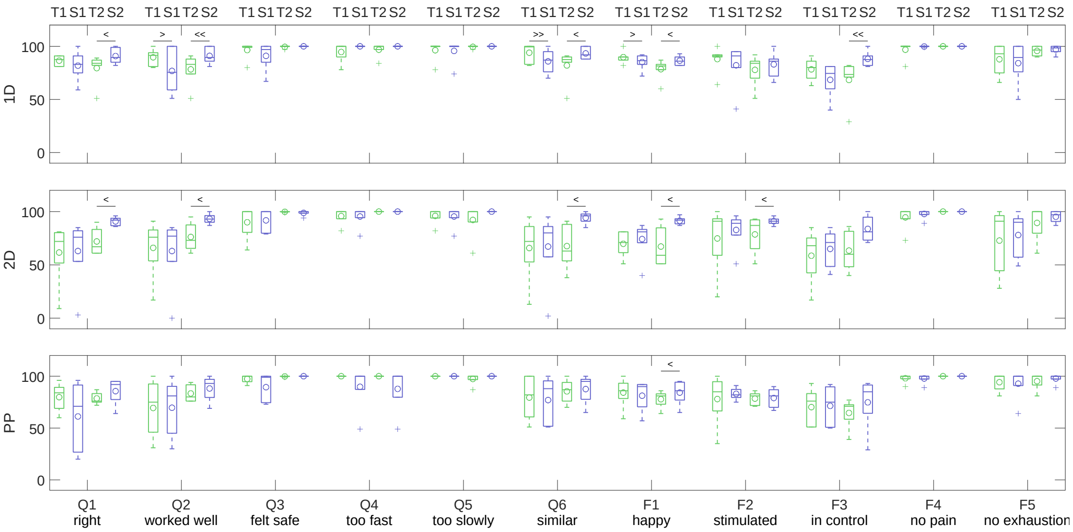


Figure 5.21: Questionnaire ratings by the therapist (higher is better). 'T' and 'S' are ratings for when the robot was using the **T**ime-based or **S**tate-based approach, and the numbers after the letters indicate the first (when the therapist observed robot-patient exercises) or the second (when the therapist practiced the exercises with the robot) testing phase.

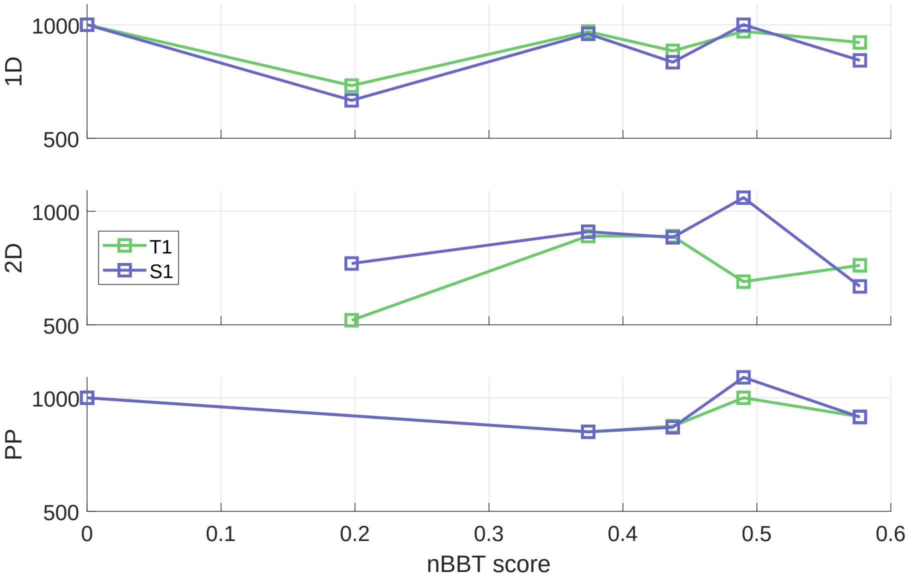
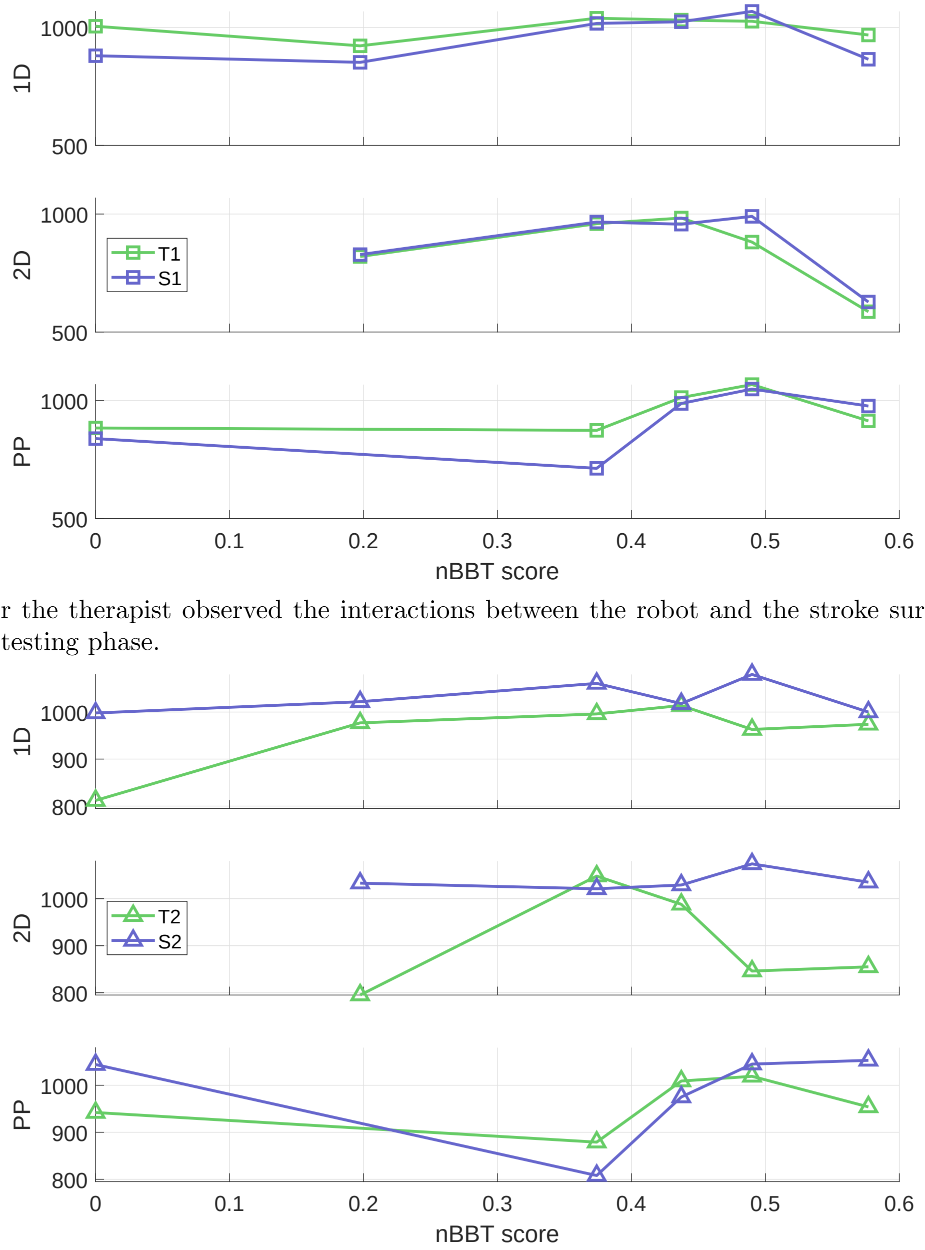


Figure 5.22: Sum of ratings by the stroke survivors after they interacted with the robot in the first testing phase. Higher is better.



(a) After the therapist observed the interactions between the robot and the stroke survivors in the first testing phase.

(b) After the therapist practiced the exercises with the robot in the second testing phase.

Figure 5.23: Sum of ratings by the therapist. Higher is better.

Table 5.3: Number of selections for best and worst interactions by the therapist. Note that because the stroke survivor with nBBT score of 0 had very limited range of motion, the therapist had difficulty choosing the worst interactions, and thus this table reports only five worst but six best interactions in each testing phase

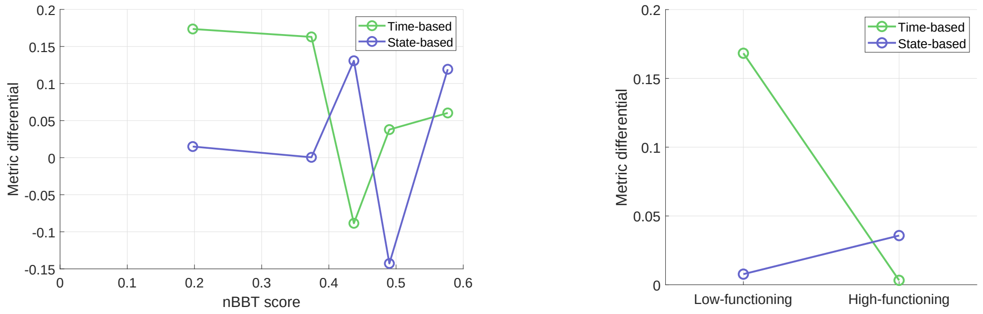
	First testing phase Selection from observation		Second testing phase Selection from interaction	
	Time-based T1	State-based S1	Time-based T2	State-based S2
1D	Best: 1	Best: 2 Worst: 1		Best: 1
2D	Best: 2 Worst: 2	Best: 1	Worst: 2	Best: 3
PP		Worst: 2		Best: 2 Worst: 3
Sum	Best: 3 Worst: 2	Best: 3 Worst: 3	Best: 0 Worst: 2	Best: 6 Worst: 3

out of the six overall after each of the testing phases.

As for the retraining portion of the study, only five sessions completed the related exercises because the retraining phase was skipped in the sixth study session due to the very limited motor function of the stroke survivor. Of the five sessions that completed the retraining portion, the therapist chose to retrain the 2D exercise in two sessions and the PP exercise in the other three. Typical reasons for her choices included that PP is used more frequently for self-care and is thus worthy of spending more time and that the robot didn't perform well enough in the first and the second testing phases for the chosen exercise.

Figs. 5.24 to 5.28 show the differences in quantitative metrics when the stroke survivors practiced with the robot in the first testing phase and retraining phase. If there is a positive difference, the metric increased during retraining. Furthermore, we divide the five stroke survivors into two groups: a low-functioning group with nBBT score below 0.45 (3 people) and a high-functioning group above 0.45 (2 people). We show the mean changes from these two groups.

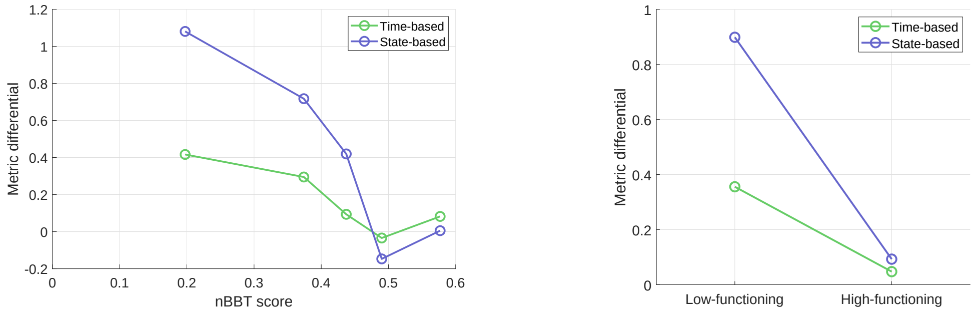
Figs. 5.29 and 5.30 show the differences in the sum of subjective ratings by the stroke survivors and the therapist before and after retraining, and we also organize the data into low- and high-functioning groups and show the mean changes.



(a) As a function of nBBT score.

(b) Mean metric differential for the two groups.

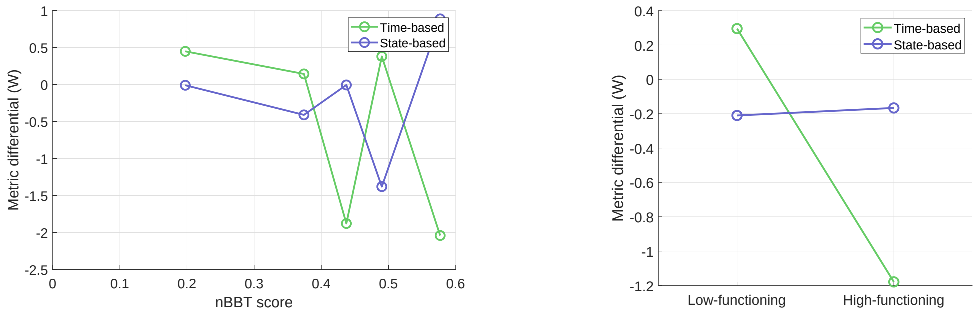
Figure 5.24: Acceleration metric differential by the stroke survivors from first exposure to second exposure (retraining). Positive values indicate that the metric increased.



(a) As a function of nBBT score.

(b) Mean metric differential for the two groups.

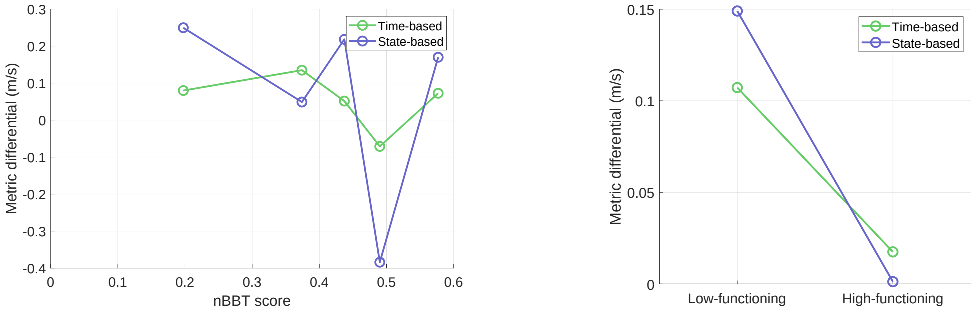
Figure 5.25: Excursion metric differential by the stroke survivors. Note that because the excursion metric has a unit of m^2 in the 2D exercise and m in the PP exercise, we omit the units and directly report numerical values in this figure.



(a) As a function of nBBT score.

(b) Mean metric differential for the two groups.

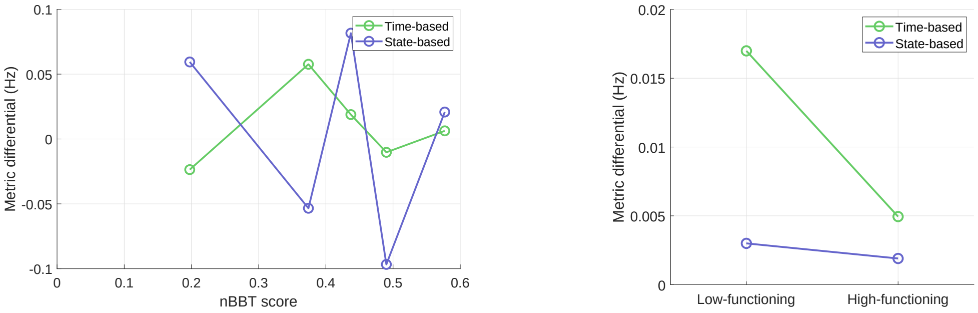
Figure 5.26: Power metric differential by the stroke survivors.



(a) As a function of nBBT score.

(b) Mean metric differential for the two groups.

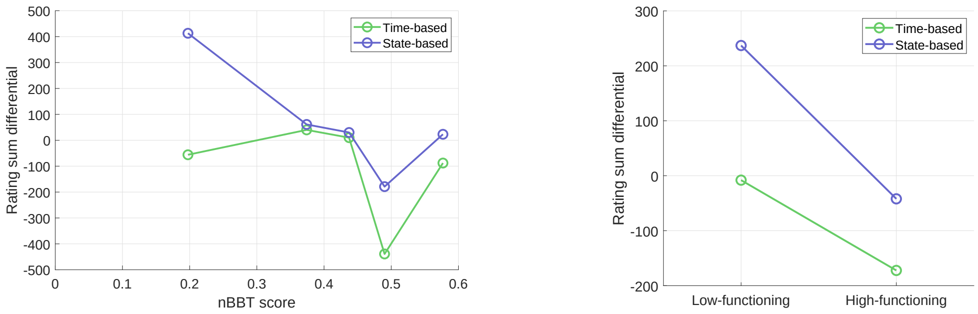
Figure 5.27: Speed metric differential by the stroke survivors.



(a) As a function of nBBT score.

(b) Mean metric differential for the two groups.

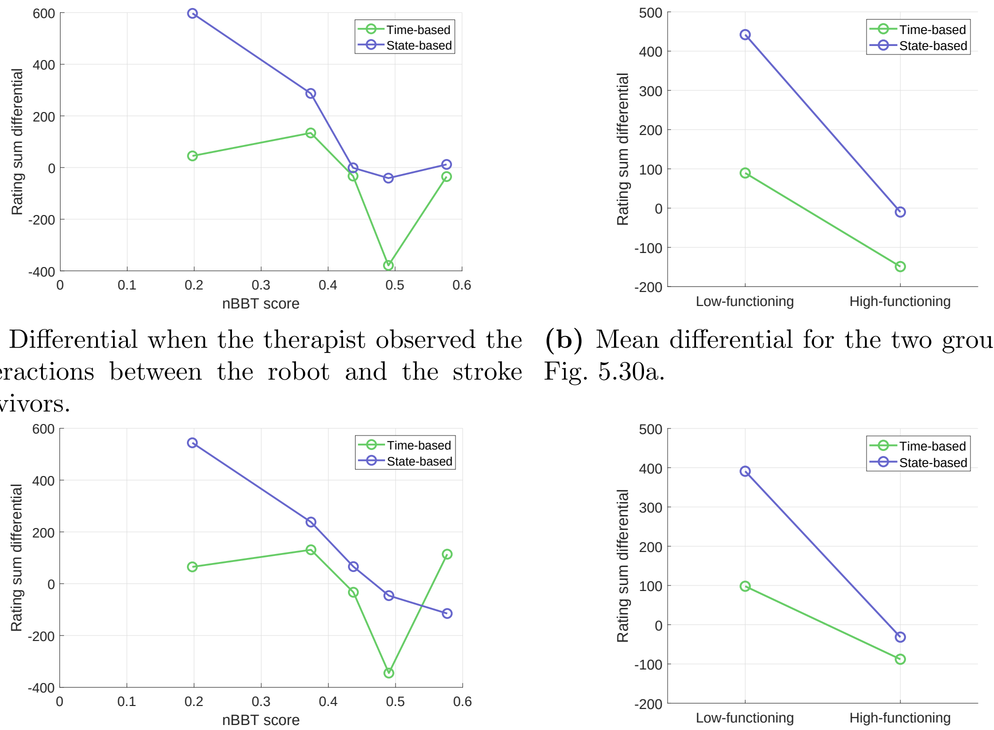
Figure 5.28: Frequency metric differential by the stroke survivors.



(a) As a function of nBBT score.

(b) Mean differential for the two groups.

Figure 5.29: Rating sum differential by the stroke survivors.



(a) Differential when the therapist observed the interactions between the robot and the stroke survivors.

(b) Mean differential for the two groups in Fig. 5.30a.

(c) Differential when the therapist practiced with the robot.

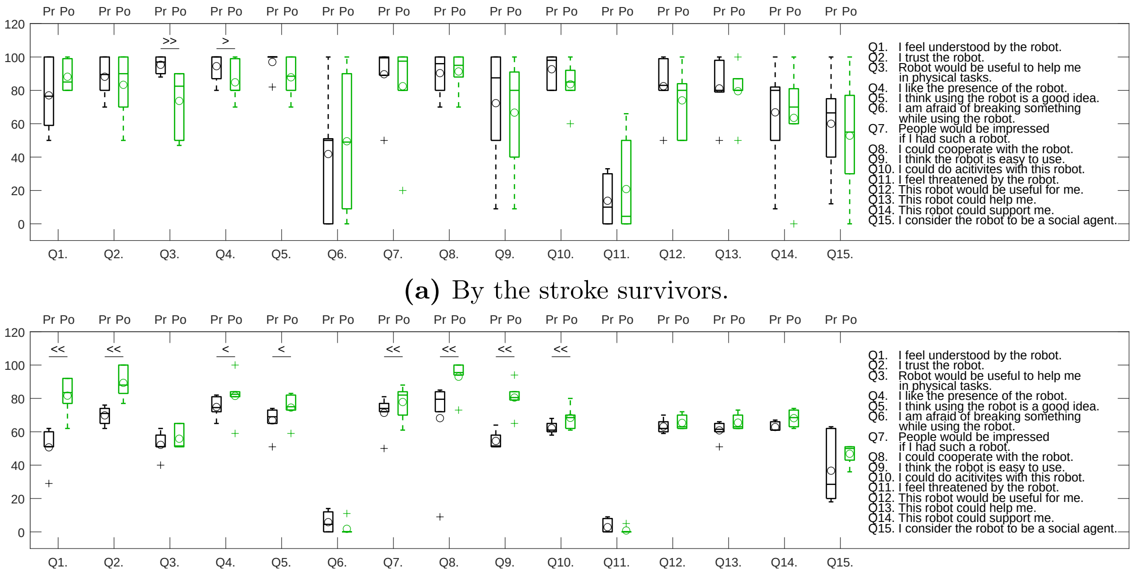
(d) Mean differential for the two groups in Fig. 5.30c.

Figure 5.30: Rating sum differential by the therapist.

Ratings from the UTAUT surveys at the beginning and the end of each session are shown in Fig. 5.31.

5.4 Discussion

The quantitative metrics illustrate how the two modeling approaches may affect the interactions during autonomous exercises, even if trained from the same demonstrations. For example, the acceleration metric, which is a representation for motion smoothness (higher means smoother) and is shown in Fig. 5.14, shows that generally the metric increases as the nBBT score increases (Fig. 5.14a) for both modeling approaches. This trend agrees with our expectation that people whose upper limbs are more able should have smoother motions. However, when comparing the two modeling approaches, it can be seen that the time-based approach enabled smoother movements for stroke survivors (Fig. 5.14b), while there was no significant differ-



(a) By the stroke survivors.

(b) By the therapist.

Figure 5.31: UTAUT ratings from the pre-study (Pr) and post-study (Po) surveys.

ence for the therapist in the two modeling approaches (Fig. 5.14c). One possible reason is that the conventional time-based approach tracks a trajectory, and stroke survivors, at least those who have more impairment (e.g., nBBT ¡ 0.45), could simply hold on to the object and follow the motions created by the robot. On the other hand, in the state-based approach, the robot regresses the effort exerted by the therapist during training, thus requiring more input from the stroke survivor to continue the exercises. For a healthy person like the therapist, she could hold on to the object and either follow the robot or give enough input to finish the exercises, thus displaying no significant difference in the acceleration metric between the two approaches.

The excursion metric, shown in Fig. 5.15, also shows that participants with higher nBBT scores had larger excursion (see k values in Fig. 5.15a) in the 1D and PP exercises, because they tended to have better control of the object and larger range of motion. The excursion metric remained constant across nBBT scores for the 2D exercise. However, when comparing the learning models, it can be seen that in many exercises, the state-based approach enabled larger excursions for both the stroke survivors and the therapist (Figs. 5.15b and 5.15c), as opposed to the acceleration metric where the time-based approach led to higher results. This result can also be expected from the fact that the time-based approach tracks a trajectory while the state-based approach allows more movement from the person. Note that the time-based approach led to significantly higher excursion ($p < 0.1$) in the PP exercise from therapist data (Fig. 5.15c), while there was no significant difference in the stroke survivor data; a possible explanation is that the episodic nature of the PP exercise made it easier for subjects to memorize the overall motion, which reduced the difference or even reversed it slightly.

Interestingly, the power metric from the training phase and the state-based approach remain similar (Fig. 5.16a), while the robot tended to have negative power outputs with the time-based approach especially for those with high nBBT scores. This result suggests that the robot correctly learned the reactions from demonstrations and provided similar levels of support in the state-based approach, while the robot was sometimes slowing down the motion because of the fixed pace that was learned from demonstrations in the time-based approach. On the other hand, the negative power output showed that the participants, when able to, were exerting effort to lead the robot to complete the exercises. However, we note that when negative power took place, it was not regarded as a form of resistance training. Instead, some participants verbally commented that the robot wouldn't give them control. A possible reason is the premise of the study: participants were told that the robot would support them in the exercises, so they did not expect the robot to be 'stubborn' and oppose their motion. More importantly, we also note that an extension of this study could be to evaluate the two learning approaches in resistance training, i.e., when the therapist controls the robot to resist the patient's motion while providing demonstration data.

Figs. 5.17 and 5.18 show that generally people with higher nBBT scores could move the object and finish repetitions faster. Similar to the acceleration metric, the time-based approach led to higher metrics than the state-based approach for the 1D and PP exercises, but higher frequency metric values were found with the state-based approach in the 2D exercise. For the therapist, no significant difference was found in these two metrics. Therefore, overall, the speed and frequency metrics gave similar observations that the state-based approach requires more input from the participant instead of tracking a time-parameterized trajectory. Note that the high frequency metric in training and the time-based approach from the sixth participant who has an nBBT score of 0 in Fig. 5.18a was largely because of the low excursion (see Fig. 5.15a), and it can be seen from Fig. 5.17a that the movement speed was indeed low for that participant.

From subjective ratings by the stroke survivors shown in Fig. 5.19, only one significant difference was found by the Fisher–Pitman permutation test: the robot appeared safer to the subjects with the state-based approach in the 2D exercise. The scarcity of significant difference showed that stroke survivors either could not subjectively differentiate or did not have a consistent preference between the two learning approaches. Nevertheless, Fig. 5.22 shows that some participants rated the state-based approach more favorably than the time-based approach, suggesting that it may be advantageous to use different learning models for different approaches depending on each person's level of motor function, but more data is needed to validate this hypothesis. In addition, the sixth participant who has an nBBT score of 0 rated everything as 'perfect' (with the exception of F2 where she chose completely calm/bored every time), thus leading to scores of 1000 in every scenario; during the session, this participant was motivated and tried very hard to finish every exercise. She then said everything was great because it was novel and exciting, and she thought the use of robot may help more people eventually. Thus it is also possible that the novelty factor of the PR2 overshadowed the subject's perception of the learning approaches.

Subjective ratings from the therapist revealed some interesting findings. First of all, Fig. 5.20 shows that physical demand was higher than mental demand, while the participants of the user study in Chapter 4 rated these items in the opposite way (see Fig. 4.13). One possible reason is that in the current demonstration procedure, the therapist had to bend her back when manipulating the robot's right arm because the robot was generally in a low position to accommodate the stroke survivor, who sat in a chair while holding the object with the robot's left arm. The bending posture may have caused the high physical demand and effort. Having the stroke survivor sitting in a higher chair or adding an adjustable height difference between the two arms may help with this issue.

In addition, Fig. 5.21 shows that when the therapist practiced with the robot, she rated the experience differently from when she observed the interactions between the stroke survivors and the robot. Three significant differences were found where the time-based approach in the first testing phase (T1) was rated more favorably than the state-based approach (S1), while eleven were found where S2 were rated more favorably than T2. Because the time-based approach led to smoother and faster motions, as illustrated by the acceleration, speed, and frequency metrics, it may have been regarded as the better approach from observations in the first testing phase. However, after hands-on interactions, the state-based approach appeared to be superior to the therapist and more similar to what she had intended to have the robot do (Q6 in Fig. 5.21). This finding suggests that physical interactions may be required, in addition to visual observation, for people to get a good sense of what the robot does, and that the therapist could differentiate the two learning approaches and may be able to recommend a learning model based on the stroke survivor's motor impairment level. Note that the lower rating sums in the 2D exercise in Fig. 5.23 for the second participant, who has a MoCA score of 18, were because the participant had trouble understanding the task and couldn't finish the required movement, as shown in the center panels of Fig. 5.8a.

Selections in Table 5.3 echo the therapist's ratings. It can be seen that in the first testing phase, the best and worst interactions were distributed almost evenly between the time- and state-based approaches, but all best interactions in the second testing phase were from the state-based approach. Thus, it can be confirmed that the therapist could differentiate between the two learning approaches, at least after hands-on experience in the exercises.

Of the five sessions with retraining, the therapist never chose to retrain with the 1D exercises; she chose the other two because the robot did not appear to do a good job in learning them, potentially because those exercises are more complex and needed more or more consistent training data than what was available during the initial training phase. If the robot performed well in all exercises, she chose PP for retraining because it is the most important type of motion for everyday life. Thus an important extension of this study would be to include a wider variety of exercises for activities of daily living.

From the quantitative metrics, the differences between before and after retraining were rather noisy, and there was no consistent change regarding the high- and low-functioning groups (see Figs. 5.24 to 5.28). However, the sums of subjective ratings exhibited consistent trends during retraining, as shown in Figs. 5.29 and 5.30. For the low-functioning group, the state-based approach appeared much more favorable after retraining, while the time-based approach led to similar or slightly higher subjective ratings. This result suggests that after getting familiar with the robot and how it moves, the subjects became more open and receptive to training with the robot, especially with the state-based approach. On the other hand, the high-functioning group rated the time-based approach less favorably after retraining, while the state-based approach remained virtually the same. Thus for high-functioning people, the time-based approach may appear as non-compliant and resistant, and even more so after they spend more time working with the robot.

The UTAUT surveys saw opposite trends in answers from the stroke survivors and the therapist, as seen in Fig. 5.31. Stroke survivors found the robot less useful for physical tasks and liked the presence of the robot less after the sessions, and their comments show they feared to break the machine and wanted to have more control over the robot. The therapist generally liked the robot better in the post-study UTAUT survey, and commented that adding more variety to the exercises that the robot could do, such as a diagonal pattern, would be useful. These comments suggest there is much room to improve for the robot and the learning models to become truly useful for robotic rehabilitation.

Finally, a common approach which was not tested in our study is to fine-tune the controller gains, $\mathbf{K}$ and $\mathbf{B}$, for each participant, in the time-based approach, or adapt the gains during training based on exercise performance. The gain values we chose were soft but active enough with slight damping, so that users could feel the motion of the robot but not be overwhelmed by it. Fine-tuning and adapting the gains could at least in theory make the time-based approach better and more suitable for each person, but doing so for each person may prolong the duration of the study sessions and does not change how the exercises are represented. We believe the user study in this chapter shows the differences between the state- and time-based approaches, even with the constant gain values throughout the six sessions. Nevertheless, it would be an interesting topic to explore how the state-based approach compares with the time-based one with optimal gains.

5.5 Summary

This chapter adapts the learning structure, demonstration procedure, and study design of Chapter 4 for use in a clinical environment, where stroke survivors with upper-limb impairments could teach the robot with a therapist and then practice various physical exercises with the autonomous robot. With about 30 s of training data, our learning approaches obtained working models that could autonomously support people for the three types of exercises we chose in this study. We evaluated five quantitative metrics and found the more conventional approach, which uses time to parameterize trajectories to track, led to smoother and quicker motions. The state-based approach, where the robot regressed the effort that it should output based on the current system state, gave more control to people and allowed larger excursion and generated similar power outputs to the demonstrations. This observation was also reflected in the ratings given by the therapist after visually observing the interactions between the robot and the stroke survivors. However, the therapist preferred the state-based approach over the time-based one after she interacted with the robot (especially in her answers of Q6 in Fig. 5.21), suggesting that the state-based behavior is more agreeable and what she intended to teach the robot.

The retraining phases of the user study offered some interesting insights. The state-based approach became much more favorable for people who are more impaired, suggesting that getting used to how the robot works was an important factor in their perception of the robot. On the other hand, the time-based approach was rated slightly worse after retraining for the higher-functioning group, because the robot did not give enough control to the user and appeared slow and resisting.

Thus, our findings support that the state-based approach can offer a unique alternative for training, especially as people become more familiar and comfortable with how the robot acts. In addition, for more complex exercises, the state-based approach may be more suitable because it doesn't rely on time and thus the training data does not need to be aligned, although it may require more work for model design, e.g., what dimensions to include as state, how to adjust model complexity (number of clusters in GMM approaches), etc.

Finally, although the hierarchical TP-GMM framework in Chapter 4 was not directly used in this chapter, it could be readily adapted when there is a sufficient amount of training data. With HS, it would be possible to generalize exercise motions between stroke survivors. For example, for 1D exercise, the highest and lowest positions during training could be used to define the task parameters, and a library of models could then be learned from data. For a new person, the highest and lowest points that he or she could comfortably reach could be easily measured at the beginning of therapy as the new task parameters, which could then be used to generalize robot controllers with the TP-GMM hierarchical framework. In fact, this measurement-generalization procedure can be used in every session to help the robot controller adapt to the patient's training progress, i.e., if over time, the patient could reach higher and lower positions, the robot should then be able to accommodate a larger range of motion. Furthermore, it may be a viable extension to have the robot measure how the patient performs to obtain new task parameters *during* training, so that the robot controller can be adapted online to afford even more natural and smoother reactions. However, although patient performance can be readily measured from motion data, more research is needed to validate the controller adaptation schedule (e.g., how often should the robot generalize controllers based on new performance measurements) and ensure controller stability.

Chapter 6

Conclusion

We envision a suite of robotic devices that assist people in their daily, physical tasks. Toward that goal, robot designers need to create a harmony of hardware and software so that users can enjoy safe, sophisticated, and customizable interactions. Many existing devices and robotic platforms strive to offer these, yet most robots today are used in factories and warehouses where they occupy different spaces from humans and perform different tasks. The most successful robots in the home environment so far are perhaps the cleaning robots, which quietly roam the rooms and spend sensing and computation efforts to get out of their masters' way, let alone actively interacting with them. A robot that could give massage and help carry the desk may still be far from existence.

This thesis thus began with a hybrid force display that combines active and passive actuators, which is shown to offer good simulation performance for various haptic elements. This hybrid force display was then adapted for gait rehabilitation training for people with hemiparesis, thanks to its powerful and safe output. Here we notice that the design parameters for these assistive robots require careful choice, which incur a heavy cost on designers and may be a major reason why robots have not been as ubiquitous as other devices like the smartphone. Therefore, the remainder of this thesis focused on a learning from demonstration framework of software, with which users can easily teach a humanoid robot how to perform tasks and the robot can learn efficiently with limited data and adapt to new task situations that it has never seen. This framework is then used for an upper-limb occupational therapy setting, where stroke survivors and a therapist taught and tested the robot. This thesis thus considered both hardware and software design for robotic platforms, and we explored both direct and indirect force regulation for human assistive technologies. Just like the smartphones of the world, with which even toddlers seem to intuitively know how to interact and enjoy, our envisioned suite of assistive robotic devices should be intuitive and empowering.

Below, we discuss our contributions toward this vision in Section 6.1, and then we outline the next steps that could continue to improve the existing technologies in Section 6.2.

6.1 Summary of Contributions

The primary contributions of this thesis are the following:

- A one-DOF hybrid force display combining a magnetic particle brake and a DC motor. By using the brake as the source of output, forces as high as 100 N can be easily generated, and the motor can reduce the inherent friction to as low as 2 N. Stiff virtual springs can be simulated well using this hybrid device, and virtual dampers may be simulated with higher noise because of hysteresis of the brake.

- An over-ground gait propulsion trainer (GPT) for stroke rehabilitation. Based on the one-DOF hybrid force display, a lightweight, wireless gait phase detection system is added and a new controller is implemented. The GPT enables unilateral stance phase resistance training targeted to increase propulsion forces from the impaired leg for people with hemiparesis from stroke. Various design parameters are considered and adjustable, offering flexibility for different training schemes.

- A hierarchical learning from demonstration framework. Compared to the state-of-the-art, generalization performance is estimated and improved, and computation load is reduced by differentiating between the training data using task parameters. This framework ensures that only relevant information is used for generalization, thus improving performance, and it enables estimation of performance, which is particularly important because the robot can then preemptively halt execution and ask for more demonstrations if generalization is deemed to fail.

- A complete procedure for using LfD with a humanoid robot, including data collection via kinesthetic teleoperation, modeling, and autonomous execution. Users with limited knowledge about robotics can easily adapt and use it for various tasks.

- An approach to learn direct force responses from demonstration for periodic physical exercises in upper-limb therapy. Using state (position and velocity) as input, periodic motions can be learned without relying on time, which simplifies the model. In addition, directly learning force responses reduces the effort on determining the optimal level of assistance during rehabilitation; instead, therapists only need to provide a few short demonstrations and a custom model could be obtained for each patient.

6.2 Conclusions and Future Directions

This dissertation discusses various approaches for modulating physical interactions between human users and robots, via both hardware design and software integration. After the initial development and testing, more research will be needed to make these designs and frameworks applicable in more settings and for more users, and further investigations will continue to build upon the current and future findings.

The first natural extension from this thesis is to extend the design of the hybrid force display and the gait propulsion trainer to include multiple degrees of freedom, i.e., the GPT could be able to resist multiple joints/links of the body and potentially during swing phase as well as stance phase, similar to how exoskeleton-type devices can guide the exact movements of the entire limb. Second, the cable transmission allows a larger workspace, but as a result, the brake can only generate resistance forces when the user is moving away from the base. Thus after each walking trial with the GPT, the user needs to return to the device and the cable has to be reeled. To make the training more efficient, one could include additional mechanical components, such as using the spool as a capstan and also route the other end of the cable to the user via a pulley, to be able to fully constrain the user's position and apply resistance forces when the user walks in either direction. The compromise between workspace and output directions will then depend on the application.

An extension for the GPT that has been mentioned by medical experts who have seen the device is to use it with a treadmill, so that body weight support and fall prevention could be incorporated easily. However, the GPT was designed for over-ground walking, and the resistance force from the brake can only be applied with increasing relative distance between the user and the GPT. In contrast, when one walks on a treadmill, the position of the pelvis remains relatively constant, so that the brake resistance cannot be applied. We note that using a motor to generate the resistance force could solve this issue, but it may introduce other safety concerns such as loss of backdrivability due to the gearbox. Nevertheless, body weight support and fall prevention are desirable functions, and thus it may be beneficial to change the GPT design to work with a treadmill. For example, one could have the GPT base move away from the treadmill at the person's walking speed (relative to the belt) when the resistance force is desired.

An exciting work would be to further develop the hierarchical learning from demonstration framework to include longer tasks and to automatically obtain the task parameters. In the current setup, the relevant task frames determine the complexity of the training model, and tasks that take multiple steps will make the model train slower and require more demonstrations, in addition to the fact that each demonstration will be longer. A possible solution for this problem could be to use LfD for each individual step, learn a separate planner for these steps, and then chain the learned steps by the planner, but then it would be cumbersome to provide demonstrations and divide them. Thus if task parameters could be automatically obtained from demonstrations, instead of from manual human labels (e.g., telling robot the current position is the first task frame), it may be possible to divide the long demonstrations according to the task parameters.

Finally, continuing the efforts in our therapy approaches could help determine what types of hardware and actuation are most suitable for therapy and what types of behavior from the robot would lead to the most acceptance from users and offer best results. For the GPT, although the study in this thesis reported increased propulsion force from the resisted leg and it is hypothesized that a person with hemiparesis will have reduced propulsion asymmetry after repeated and intensive training, more data is needed to show the actual effects and to determine the optimal training scheme. The latter may be summarized or even learned from data. Our collaborators at Stony Brook University are actively testing with people with stroke and continuing to evaluate and improve the GPT. For LfD with PR2, although a major part of our motivation was to avoid much of the design burden and parameter tuning by having the therapist directly teach the robot how it should behave, it is necessary to get more involvement from the clinical side to guide the design of the exercises. For example, the therapist in the user study in Chapter 5 recommended training with a diagonal pattern, which was not included in the study. It would also be beneficial to include more types of activities of daily living. Toward this end, a more general learning framework and a corresponding robot controller would become essential, as it may even be possible to teach the robot how to run a study in addition to how to perform in each step of the study.

Bibliography

[1] Jake J Abbott and Allison M Okamura, *Virtual fixture architectures for telemanipulation*, Proc. of the IEEE International Conference on Robotics and Automation, vol. 2, 2003, pp. 2798–2805.

[2] Firas Abi-Farraj, Takayuki Osa, Nicoló Pedemonte Jan Peters, Gerhard Neumann, and Paolo Robuffo Giordano, *A learning-based shared control architecture for interactive task execution*, Proc. IEEE Int. Conf. on Robotics and Automation, 2017, pp. 329–335.

[3] Don Joven Agravante, Andrea Cherubini, Antoine Bussy, Pierre Gergondet, and Abderrahmane Kheddar, *Collaborative human-humanoid carrying using vision and haptic sensing*, IEEE International Conference on Robotics and Automation, IEEE, 2014, pp. 607–612.

[4] Baris Akgun, Maya Cakmak, Jae Wook Yoo, and Andrea Lockerd Thomaz, *Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective*, Proc. ACM/IEEE Int. Conf. on Human-Robot Interaction, 2012, pp. 391–398.

[5] Jessica L Allen, Steven A Kautz, and Richard R Neptune, *Forward propulsion asymmetry is indicative of changes in plantarflexor coordination during walking in individuals with post-stroke hemiparesis*, Clinical Biomechanics **29** (2014), no. 7, 780–786.

[6] Fady S Alnajjar, Abdulrahman Majed Renawi, Massimiliano Cappuccio, and Omar Mubain, *A low-cost autonomous attention assessment system for robot intervention with autistic children*, IEEE Global Engineering Education Conference, IEEE, 2019, pp. 787–792.

[7] Jinung An and Dong-Soo Kwon, *Virtual friction display of hybrid force feedback interface with actuators comprising dc motor and magnetorheological brake*, IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on, IEEE, 2006, pp. 3997–4002.

[8] ______, *Five-bar linkage haptic device with dc motors and mr brakes*, Journal of Intelligent Material Systems and Structures **20** (2009), no. 1, 97–107.

[9] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning, *A survey of robot learning from demonstration*, Robotics and Autonomous Systems **57** (2009), no. 5, 469–483.

[10] Louis N Awad, Darcy S Reisman, Trisha M Kesar, and Stuart A Binder-Macleod, *Targeting paretic propulsion to improve poststroke walking function: a preliminary study*, Archives of Physical Medicine and Rehabilitation **95** (2014), no. 5, 840–848.

[11] Richard Balliet, Brenda Levy, and K Blood, *Upper extremity sensory feedback therapy in chronic cerebrovascular accident patients with impaired expressive aphasia and auditory comprehension*, Archives of Physical Medicine and Rehabilitation **67** (1986), no. 5, 304–310.

[12] Sai K Banala, Suni K Agrawal, and John P Scholz, *Active leg exoskeleton (ALEX) for gait rehabilitation of motor-impaired patients*, Proc. IEEE International Conference on Rehabilitation Robotics, 2007, pp. 401–407.

[13] Julien Baumeyer, Pierre Vieyres, Sylvain Miossec, Cyril Novales, Gerard Poisson, and Samuel Pinault, *Robotic co-manipulation with 6 DoF admittance control: Application to patient positioning in proton-therapy*, IEEE International Workshop on Advanced Robotics and its Social Impacts, IEEE, 2015, pp. 1–6.

[14] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal, *Robot programming by demonstration*, Springer handbook of robotics (2008), 1371–1394.

[15] Jonathan Blake and Hakan B Gurocak, *Haptic glove with MR brakes for virtual reality*, IEEE/ASME Trans. on Mechatronics **14** (2009), no. 5, 606–615.

[16] Mark G Bowden, Chitralakshmi K Balasubramanian, Richard R Neptune, and Steven A Kautz, *Anterior-posterior ground reaction forces as a measure of paretic leg contribution in hemiparetic walking*, Stroke **37** (2006), no. 3, 872–876.

[17] Mark G Bowden, Aaron E Embry, and Chris M Gregory, *Physical therapy adjuvants to promote optimization of walking recovery after stroke*, Stroke Research and Treatment **2011** (2011).

[18] Margaret M Bradley and Peter J Lang, *Measuring emotion: the self-assessment manikin and the semantic differential*, Journal of Behavior Therapy and Experimental Psychiatry **25** (1994), no. 1, 49–59.

[19] Kevin D Bui and Michelle J Johnson, *Designing robot-assisted neurorehabilitation strategies for people with both HIV and stroke*, Journal of NeuroEngineering and Rehabilitation **15** (2018), no. 1, 75.

[20] Charles G Burgar, Peter S Lum, Peggy C Shor, and HF Machiel Van der Loos, *Development of robots for rehabilitation therapy: The Palo Alto VA/Stanford experience*, Journal of Rehabilitation Research and Development **37** (2000), no. 6, 663–674.

[21] Sylvain Calinon, *A tutorial on task-parameterized movement learning and retrieval*, Intelligent Service Robotics **9** (2016), no. 1, 1–29.

[22] Sylvain Calinon and Aude Billard, *Incremental learning of gestures by imitation in a humanoid robot*, Proc. ACM/IEEE Int. Conf. on Human-Robot Interaction, 2007, pp. 255–262.

[23] Sylvain Calinon, Paul Evrard, Elena Gribovskaya, Aude Billard, and Abderrahmane Kheddar, *Learning collaborative manipulation tasks by demonstration using a haptic interface*, Proc. IEEE Int. Conf. on Advanced Robotics, 2009, pp. 1–6.

[24] Sonia Chernova and Manuela Veloso, *Interactive policy learning through confidence-based autonomy*, Journal of Artificial Intelligence Research **34** (2009), 1–25.

[25] François Conti and Oussama Khatib, *A new actuation approach for haptic interface design*, The International Journal of Robotics Research **28** (2009), no. 6, 834–848.

[26] John J Craig, Ping Hsu, and S Shankar Sastry, *Adaptive control of mechanical manipulators*, The International Journal of Robotics Research **6** (1987), no. 2, 16–28.

[27] Tyler A Davis, Yung C Shin, and Bin Yao, *Observer-based adaptive robust control of friction stir welding axial force*, IEEE/ASME Transactions on Mechatronics **16** (2010), no. 6, 1032–1039.

[28] Ruth Dickstein, *Rehabilitation of gait speed after stroke: A critical review of intervention approaches*, Neurorehabilitation and Neural Repair **22** (2008), no. 6, 649–660.

[29] Janice J Eng and Pei Fang Tang, *Gait training strategies to optimize walking ability in people with stroke: A synthesis of the evidence*, Expert Review of Neurotherapeutics **7** (2007), no. 10, 1417.

[30] Paul Evrard, Elena Gribovskaya, Sylvain Calinon, Aude Billard, and Abderrahmane Kheddar, *Teaching physical collaborative tasks: Object-lifting case study with a humanoid*, Proc. IEEE-RAS Int. Conf. on Humanoid Robots, 2009, pp. 399–404.

[31] Jason Fong, Hossein Rouhani, and Mahdi Tavakoli, *A therapist-taught robotic system for assistance during gait therapy targeting foot drop*, IEEE Robotics and Automation Letters **4** (2019), no. 2, 407–413.

[32] Berk Gonenc and Hakan Gurocak, *Virtual needle insertion with haptic feedback using a hybrid actuator with DC servomotor and MR-brake with Hall-effect sensor*, Mechatronics **22** (2012), no. 8, 1161–1176.

[33] Chris M Gregory and C Scott Bickel, *Recruitment patterns in human skeletal muscle during electrical stimulation*, Physical Therapy **85** (2005), no. 4, 358–364.

[34] Sandra G Hart and Lowell E Staveland, *Development of NASA-TLX (task load index): Results of empirical and theoretical research*, Advances in Psychology, vol. 52, 1988, pp. 139–183.

[35] Elwood Henneman, George Somjen, and David O Carpenter, *Functional significance of cell size in spinal motoneurons*, Journal of Neurophysiology **28** (1965), no. 3, 560–580.

[36] N Hogan, *Impedance control: an approach to manipulation, Part 1 - Theory*, ASME Journal of Dynamic Systems, Measurement, and Control **107** (1985), no. 1, 1–7.

[37] Jose Hoyos, Flavio Prieto, Guillem Alenyà, and Carme Torras, *Incremental learning of skills in a task-parameterized gaussian mixture model*, Journal of Intelligent & Robotic Systems **82** (2016), no. 1, 81–99.

[38] HaoYuan Hsiao, Louis N Awad, Jacqueline A Palmer, Jill S Higginson, and Stuart A Binder-Macleod, *Contribution of paretic and nonparetic limb peak propulsive forces to changes in walking speed in individuals poststroke*, Neurorehabilitation and Neural Repair **30** (2016), no. 8, 743–752.

[39] Siyao Hu and Katherine J Kuchenbecker, *High magnitude unidirectional haptic force display using a motor/brake pair and a cable*, Proc. of the IEEE World Haptics Conference, IEEE, 2017, pp. 394–399.

[40] ———, *Hierarchical task-parameterized learning from demonstration for collaborative object movement*, Applied Bionics and Biomechanics **2019** (2019).

[41] Yanlong Huang, João Silvério, Leonel Rozo, and Darwin G Caldwell, *Generalized task-parameterized skill learning*, Proc. IEEE Int. Conf. on Robotics and Automation, 2018, pp. 1–5.

[42] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal, *Trajectory formation for imitation with nonlinear dynamical systems*, Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, vol. 2, 2001, pp. 752–757.

[43] Jeffrey F Israel, Donielle D Campbell, Jennifer H Kahn, and T George Hornby, *Metabolic costs and muscle activity patterns during robotic-and therapist-assisted treadmill walking in individuals with incomplete spinal cord injury*, Physical Therapy **86** (2006), no. 11, 1466–1478.

[44] FM Ivey, RF Macko, AS Ryan, and CE Hafer-Macko, *Cardiovascular health and fitness after stroke*, Topics in Stroke Rehabilitation **12** (2005), no. 1, 1–16.

[45] Nathanaël Jarrassé, Themistoklis Charalambous, and Etienne Burdet, *A framework to describe, analyze and generate interactive motor behaviors*, PloS one **7** (2012), no. 11, e49945.

[46] Michelle J Johnson, *Recent trends in robot-assisted therapy environments to improve real-life functional performance after stroke*, Journal of NeuroEngineering and Rehabilitation **3** (2006), no. 1, 29.

[47] Michelle J Johnson, Seethu M Christopher, Mayumi Mohan, and Rochelle Mendonca, *Assessing human-human therapy kinematics for retargeting to human-robot therapy*, Proc. of the IEEE International Conference on Rehabilitation Robotics, 2015, pp. 870–875.

[48] Michelle Jillian Johnson, Roshan Rai, Sarath Barathi, Rochelle Mendonca, and Karla Bustamante-Valles, *Affordable stroke therapy in high-, low-and middle-income countries: From theradrive to rehab cares, a compact robot gym*, Journal of Rehabilitation and Assistive Technologies Engineering **4** (2017), 2055668317708732.

[49] MJ Johnson, M Trickey, E Brauer, and X Feng, *TheraDrive: a new stroke therapy concept for home-based, computer-assisted motivating rehabilitation*, Proc. of the International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 2, 2004, pp. 4844–4847.

[50] MJ Johnson, KJ Wisneski, J Anderson, D Nathan, and RO Smith, *Development of ADLER: The activities of daily living exercise robot*, Proc. of the IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006, pp. 881–886.

[51] Johannes Kaiser, *An exact and a Monte Carlo proposal to the Fisher–Pitman permutation tests for paired replicates and for independent samples*, The Stata Journal **7** (2007), no. 3, 402–412.

[52] Diomidis I Katzourakis, David A Abbink, Riender Happee, and Edward Holweg, *Steering force feedback for human–machine-interface automotive experiments*, IEEE Transactions on Instrumentation and Measurement **60** (2010), no. 1, 32–43.

[53] Margaret Kelly-Hayes, Alexa Beiser, Carlos S Kase, Amy Scaramucci, Ralph B D'Agostino, and Philip A Wolf, *The influence of gender and age on disability following ischemic stroke: the Framingham study*, Journal of Stroke and Cerebrovascular Diseases **12** (2003), no. 3, 119–126.

[54] S Mohammad Khansari-Zadeh and Aude Billard, *Learning stable nonlinear dynamical systems with gaussian mixture models*, IEEE Transactions on Robotics **27** (2011), no. 5, 943–957.

[55] Dorothea Koert, Susanne Trick, Marco Ewerton, Michael Lutter, and Jan Peters, *Online learning of an open-ended skill library for collaborative tasks*, Proc. IEEE-RAS Int. Conf. on Humanoid Robots, 2018, pp. 1–9.

[56] Hermano I Krebs, Bruce T Volpe, Mark Ferraro, S Fasoli, J Palazzolo, B Rohrer, L Edelstein, and N Hogan, *Robot-aided neurorehabilitation: from evidence-based to science-based rehabilitation*, Topics in Stroke Rehabilitation **8** (2002), no. 4, 54–70.

[57] Dana Kulic, Wataru Takano, and Yoshihiko Nakamura, *Combining automated on-line segmentation and incremental clustering for whole body motions*, Proc. IEEE Int. Conf. on Robotics and Automation, 2008, pp. 2591–2598.

[58] Tomas Kulvicius, Martin Biehl, Mohamad Javad Aein, Minija Tamosiunaite, and Florentin Wörgötter, *Interaction learning for dynamic movement primitives used in cooperative robotic tasks*, Robotics and Autonomous Systems **61** (2013), no. 12, 1450–1459.

[59] Tae-Bum Kwon and Jae-Bok Song, *Force display using a hybrid haptic device composed of motors and brakes*, Mechatronics **16** (2006), no. 5, 249–257.

[60] Tania Lam, Markus Wirz, Lars Lünenburger, and Volker Dietz, *Swing phase resistance enhances flexor muscle activity during treadmill locomotion in incomplete spinal cord injury*, Neurorehabilitation and Neural Repair **22** (2008), no. 5, 438–446.

[61] Przemyslaw A Lasota and Julie A Shah, *Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration*, Human Factors **57** (2015), no. 1, 21–33.

[62] Clemente Lauretti, Francesca Cordella, Anna Lisa Ciancio, Emilio Trigili, Jose Maria Catalan, Francisco Javier Badesa, Simona Crea, Silvio Marcello Pagliara, Silvia Sterzi, Nicola Vitiello, et al., *Learning by demonstration for motion planning of upper-limb exoskeletons*, Frontiers in Neurorobotics **12** (2018), 5.

[63] Clemente Lauretti, Francesca Cordella, Eugenio Guglielmelli, and Loredana Zollo, *Learning by demonstration for planning activities of daily living in rehabilitation and assistive robotics*, IEEE Robotics and Automation Letters **2** (2017), no. 3, 1375–1382.

[64] Dongheui Lee and Christian Ott, *Incremental kinesthetic teaching of motion primitives using the motion refinement tube*, Autonomous Robots **31** (2011), no. 2-3, 115–131.

[65] RCV Loureiro, MJ Johnson, and WS Harwin, *Collaborative tele-rehabilitation: a strategy for increasing engagement*, Proc. of the International Conference on Biomedical Robotics and Biomechatronics, 2006, pp. 859–864.

[66] Rui Loureiro, Farshid Amirabdollahian, and William Harwin, *A Gentle/S approach to robot assisted neuro-rehabilitation*, Advances in Rehabilitation Robotics, Springer, 2004, pp. 347–363.

[67] Rui CV Loureiro and William S Harwin, *Reach & grasp therapy: design and control of a 9-dof robotic neuro-rehabilitation system*, Proc. of the International Conference on Rehabilitation Robotics, 2007, pp. 757–763.

[68] Rui CV Loureiro, William S Harwin, Kiyoshi Nagai, and Michelle Johnson, *Advances in upper limb stroke rehabilitation: a technology push*, Medical & Biological Engineering & Computing **49** (2011), no. 10, 1103.

[69] Peter Lum, David Reinkensmeyer, Richard Mahoney, William Z Rymer, and Charles Burgar, *Robotic devices for movement therapy after stroke: current status and challenges to clinical acceptance*, Topics in Stroke Rehabilitation **8** (2002), no. 4, 40–53.

[70] Mahdi Maaref, Alireza Rezazadeh, Kimia Shamaei, Renz Ocampo, and Tavakoli Mahdi, *A bicycle cranking model for assist-as-needed robotic rehabilitation therapy using learning from demonstration*, IEEE Robotics and Automation Letters **1** (2016), no. 2, 653–660.

[71] Mahdi Maaref, Alireza Rezazadeh, Kimia Shamaei, and Mahdi Tavakoli, *A Gaussian mixture framework for co-operative rehabilitation therapy in assistive impedance-based tasks*, IEEE Journal of Selected Topics in Signal Processing **10** (2016), no. 5, 904–913.

[72] Guilherme Maeda, Marco Ewerton, Gerhard Neumann, Rudolf Lioutikov, and Jan Peters, *Phase estimation for fast action recognition and trajectory generation in human–robot collaboration*, The International Journal of Robotics Research **36** (2017), no. 13-14, 1579–1594.

[73] Guilherme Maeda, Marco Ewerton, Takayuki Osa, Baptiste Busch, and Jan Peters, *Active incremental learning of robot movement primitives*, Conf. on Robot Learning, 2017, pp. 37–46.

[74] Guilherme J Maeda, Gerhard Neumann, Marco Ewerton, Rudolf Lioutikov, Oliver Kroemer, and Jan Peters, *Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks*, Autonomous Robots **41** (2017), no. 3, 593–612.

[75] Laura Marchal-Crespo and David J Reinkensmeyer, *Review of control strategies for robotic movement training after neurologic injury*, Journal of NeuroEngineering and Rehabilitation **6** (2009), no. 1, 20.

[76] Carlos Martinez and Mahdi Tavakoli, *Learning and reproduction of therapist's semi-periodic motions during robotic rehabilitation*, Robotica, 1–13.

[77] Carlos Martínez and Mahdi Tavakoli, *Learning and robotic imitation of therapist's motion and force for post-disability rehabilitation*, Proc. of the IEEE International Conference on Systems, Man, and Cybernetics, 2017, pp. 2225–2230.

[78] Thomas H Massie and J Kenneth Salisbury, *The PHANToM haptic interface: a device for probing virtual objects*, Proc. ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, vol. 55, 1994, pp. 295–300.

[79] Virgil Mathiowetz, Gloria Volland, Nancy Kashman, and Karen Weber, *Adult norms for the Box and Block Test of manual dexterity*, American Journal of Occupational Therapy **39** (1985), no. 6, 386–391.

[80] Stefano Mazzoleni, Patrizio Sale, Micol Tiboni, Marco Franceschini, F Posteraro, and MC Carrozza, *Upper limb robot-assisted therapy in chronic and subacute stroke patients: a kinematic analysis*, Converging Clinical and Engineering Research on Neurorehabilitation, Springer, 2013, pp. 129–133.

[81] Mayumi Mohan, Rochelle Mendonca, and Michelle J Johnson, *Towards quantifying dynamic human-human physical interactions for robot assisted stroke therapy*, Proc. of the International Conference on Rehabilitation Robotics, 2017, pp. 913–918.

[82] Susan L Morris, Karen J Dodd, and Meg E Morris, *Outcomes of progressive resistance strength training following stroke: a systematic review*, Clinical Rehabilitation **18** (2004), no. 1, 27–39.

[83] Katharina Muelling, Jens Kober, and Jan Peters, *Learning table tennis with a mixture of motor primitives*, Proc. IEEE-RAS Int. Conf. on Humanoid Robots, 2010, pp. 411–416.

[84] Mohammad Najafi, Kim Adams, and Mahdi Tavakoli, *Robotic learning from demonstration of therapist's time-varying assistance to a patient in trajectory-*

following tasks, Proc. of the International Conference on Rehabilitation Robotics, 2017, pp. 888–894.

[85] Mohammad Najafi, Mojtaba Sharifi, Kim Adams, and Mahdi Tavakoli, *Robotic assistance for children with cerebral palsy based on learning from tele-cooperative demonstration*, International Journal of Intelligent Robotics and Applications **1** (2017), no. 1, 43–54.

[86] Ziad S Nasreddine, Natalie A Phillips, Valérie Bédirian, Simon Charbonneau, Victor Whitehead, Isabelle Collin, Jeffrey L Cummings, and Howard Chertkow, *The Montreal Cognitive Assessment, MoCA: a brief screening tool for mild cognitive impairment*, Journal of the American Geriatrics Society **53** (2005), no. 4, 695–699.

[87] Günter Niemeyer, Carsten Preusche, Stefano Stramigioli, and Dongjun Lee, *Telerobotics*, Springer handbook of robotics, Springer, 2016, pp. 1085–1108.

[88] Michelle M Ouellette, Nathan K LeBrasseur, Jonathan F Bean, Edward Phillips, Joel Stein, Walter R Frontera, and Roger A Fielding, *High-intensity resistance training improves muscle strength, self-reported function, and disability in long-term stroke survivors*, Stroke **35** (2004), no. 6, 1404–1409.

[89] Ion PI Pappas, Milos R Popovic, Thierry Keller, Volker Dietz, and Manfred Morari, *A reliable gait phase detection system*, IEEE Trans. on Neural Systems and Rehabilitation Engineering **9** (2001), no. 2, 113–125.

[90] Shrey Pareek, Hemanth Manjunath, Ehsan T Esfahani, and Thenkurussi Kesavadas, *Myotrack: tracking subject participation in robotic rehabilitation using sEMG and IMU*, Proc. of the International Symposium on Medical Robotics, 2019, pp. 1–7.

[91] Jacquelin Perry and J. M. Burnfield, *Gait analysis: Normal and pathological function*, Slack Inc., New York, NY, 2010.

[92] Affan Pervez, Arslan Ali, Jee-Hwan Ryu, and Dongheui Lee, *Novel learning from demonstration approach for repetitive teleoperation tasks*, Proc. IEEE World Haptics Conf., 2017, pp. 60–65.

[93] Affan Pervez and Dongheui Lee, *Learning task-parameterized dynamic movement primitives using mixture of GMMs*, Intelligent Service Robotics **11** (2018), no. 1, 61–78.

[94] Edwin JG Pitman, *Significance tests which may be applied to samples from any populations*, Supplement to the Journal of the Royal Statistical Society **4** (1937), no. 1, 119–130.

[95] Joel Rey, Klas Kronander, Farbod Farshidian, Jonas Buchli, and Aude Billard, *Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies*, Autonomous Robots **42** (2018), no. 1, 45–64.

[96] Sarah A Roelker, Mark G Bowden, Steven A Kautz, and Richard R Neptune, *Paretic propulsion as a measure of walking performance and functional motor recovery post-stroke: a review*, Gait & posture **68** (2019), 6–14.

[97] Carlos Rossa, José Lozada, and Alain Micaelli, *Design and control of a dual unidirectional brake hybrid actuation system for haptic devices*, IEEE Trans. on Haptics **7** (2014), no. 4, 442–453.

[98] Leonel Rozo, Sylvain Calinon, Darwin G Caldwell, Pablo Jimenez, and Carme Torras, *Learning physical collaborative robot behaviors from human demonstrations*, IEEE Trans. on Robotics **32** (2016), no. 3, 513–527.

[99] Leonel Rozo, João Silvério, Sylvain Calinon, and Darwin G Caldwell, *Learning controllers for reactive and proactive behaviors in human-robot collaboration*, Frontiers in Robotics and AI **3** (2016), 30.

[100] Jee-Hwan Ryu and Moon-Young Yoon, *Memory-based passivation approach for stable haptic interaction*, IEEE/ASME Trans. on Mechatronics **19** (2014), no. 4, 1424–1435.

[101] Andrew Sawers and Lena H Ting, *Perspectives on human-human sensorimotor interactions for the design of rehabilitation robots*, Journal of NeuroEngineering and Rehabilitation **11** (2014), no. 1, 142.

[102] Peter H Schönemann, *A generalized solution of the orthogonal Procrustes problem*, Psychometrika **31** (1966), no. 1, 1–10.

[103] Gideon Schwarz, *Estimating the dimension of a model*, The Annals of Statistics **6** (1978), no. 2, 461–464.

[104] David Silver, J Andrew Bagnell, and Anthony Stentz, *Active learning from demonstration for robust autonomous navigation*, Proc. IEEE Int. Conf. on Robotics and Automation, 2012, pp. 200–207.

[105] Ajay Kumar Tanwani and Sylvain Calinon, *Learning robot manipulation tasks with task-parameterized semitied hidden semi-Markov model*, IEEE Robotics and Automation Letters **1** (2016), no. 1, 235–242.

[106] Gergory T Thielman, Catherine M Dean, and AM Gentile, *Rehabilitation of reaching after stroke: task-related training versus progressive resistive exercise*, Archives of Physical Medicine and Rehabilitation **85** (2004), no. 10, 1613–1618.

[107] Aleš Ude, Andrej Gams, Tamim Asfour, and Jun Morimoto, *Task-specific generalization of discrete and periodic dynamic movement primitives*, IEEE Transactions on Robotics **26** (2010), no. 5, 800–815.

[108] Heike Vallery, Edwin HF Van Asseldonk, Martin Buss, and Herman Van Der Kooij, *Reference trajectory generation for rehabilitation robots: complementary limb motion estimation*, IEEE Transactions on Neural Systems and Rehabilitation Engineering **17** (2008), no. 1, 23–30.

[109] Jan F Veneman, Rik Kruidhof, Edsko EG Hekman, Ralf Ekkelenkamp, Edwin HF Van Asseldonk, and Herman Van Der Kooij, *Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation*, IEEE Transactions on Neural Systems and Rehabilitation Engineering **15** (2007), no. 3, 379–386.

[110] Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis, *User acceptance of information technology: Toward a unified view*, MIS Quarterly (2003), 425–478.

[111] Joachim von Zitzewitz, Georg Rauter, Reto Steiner, Andreas Brunschweiler, and Robert Riener, *A versatile wire robot concept as a haptic interface for*

sport simulation, Proc. IEEE Int. Conf. on Robotics and Automation, 2009, pp. 313–318.

[112] Joachim von Zitzewitz, Peter Wolf, Vladimir Novaković, Mathias Wellner, Georg Rauter, Andreas Brunschweiler, and Robert Riener, *Real-time rowing simulator with multimodal feedback*, Sports Technology **1** (2008), no. 6, 257–266.

[113] Angela Weiss, Toshimi Suzuki, Jonathan Bean, and Roger A Fielding, *High intensity strength training improves strength and functional performance after stroke*, American Journal of Physical Medicine & Rehabilitation **79** (2000), no. 4, 369–376.

[114] Kimberly J Wisneski and Michelle J Johnson, *Insights into modeling functional trajectories for robot-mediated daily living exercise environments*, Proc. of the IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006, pp. 99–104.

[115] ______, *Quantifying kinematics of purposeful movements to real, imagined, or absent functional objects: implications for modelling trajectories for robot-assisted adl tasks*, Journal of NeuroEngineering and Rehabilitation **4** (2007), no. 1, 7.

[116] George F Wittenberg, Robert Chen, Kenji Ishii, Khalafalla O Bushara, Edward Taub, Lynn H Gerber, Mark Hallett, and Leonardo G Cohen, *Constraint-induced therapy in stroke: magnetic-stimulation motor maps and cerebral activation*, Neurorehabilitation and Neural Repair **17** (2003), no. 1, 48–57.

[117] Elizabeth C Wonsetler and Mark G Bowden, *A systematic review of mechanisms of gait speed change post-stroke. Part 2: exercise capacity, muscle activation, kinetics, and kinematics*, Topics in Stroke Rehabilitation **24** (2017), no. 5, 394–403.

[118] Ming Wu, Jill M Landry, Janis Kim, Brian D Schmit, Sheng-Che Yen, and Jillian MacDonald, *Robotic resistance/assistance training improves locomotor function in individuals poststroke: a randomized controlled study*, Archives of Physical Medicine and Rehabilitation **95** (2014), no. 5, 799–806.

[119] Sheng-Che Yen, Brian D Schmit, and Ming Wu, *Using swing resistance and assistance to improve gait symmetry in individuals post-stroke*, Human Movement Science **42** (2015), 212–224.

[120] Shun-Zheng Yu, *Hidden semi-Markov models*, Artificial Intelligence **174** (2010), no. 2, 215–243.

[121] JA Zeni Jr, JG Richards, and JS Higginson, *Two simple methods for determining gait events during treadmill and overground walking using kinematic data*, Gait & Posture **27** (2008), no. 4, 710–714.

[122] Jaka Ziherl, Domen Novak, Andrej Olenšek, Matjaž Mihelj, and Marko Munih, *Evaluation of upper extremity robot-assistances in subacute and chronic stroke subjects*, Journal of NeuroEngineering and Rehabilitation **7** (2010), no. 1, 52.