Design and Development of a Dual-arm Robotic Clothing Assistance System using Imitation Learning

著者	Prakash Joshi Ravi
year	2020-09-14
その他のタイトル	模倣学習を用いた両腕ロボット着衣介助システムのデザインと開発
学位授与年度	令和2年度
学位授与番号	17104甲生工第384号
URL	http://hdl.handle.net/10228/00007950

KYUTECH-LSSE-17899033

Doctoral Dissertation

Design and Development of a Dual-arm Robotic Clothing Assistance System using Imitation Learning

模倣学習を用いた両腕ロボット着衣介助システムのデザインと開発

JOSHI RAVI PRAKASH

September 14, 2020

Department of Life Science and Systems Engineering
Graduate School of Life Science and Systems Engineering
Kyushu Institute of Technology

A Doctoral Dissertation
submitted to Graduate School of Life Science and Systems Engineering,
Kyushu Institute of Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

JOSHI RAVI PRAKASH

Thesis Committee:

Professor Tomohiro Shibata	(Supervisor)
Professor Kazuo Ishii	(Co-supervisor)
Associate Professor Hiroaki Wagatsuma	(Co-supervisor)
Associate Professor Kazuto Takashima	(Co-supervisor)

Design and Development of a Dual-arm Robotic Clothing Assistance System using Imitation Learning[*]

JOSHI RAVI PRAKASH

Abstract

The recent demographic trend across developed nations shows a dramatic increase in the aging population and fallen fertility rates. With the aging population, the number of elderly who need support for their Activities of Daily Living (ADL) such as dressing, is growing. The use of caregivers is universal for the dressing task due to the unavailability of any effective assistive technology. Unfortunately, across the globe, many nations are suffering from a severe shortage of caregivers. Hence, the demand for service robots to assist with the dressing task is increasing rapidly.

Robotic Clothing Assistance is a challenging task. The robot has to deal with the following two complex tasks simultaneously, (a) non-rigid and highly flexible cloth manipulation, and (b) safe human-robot interaction while assisting a human whose posture may vary during the task. On the other hand, humans can deal with these tasks rather easily.

In this thesis, a framework for Robotic Clothing Assistance by imitation learning from a human demonstration to a compliant dual-arm robot is proposed. In this framework, the dressing task is divided into the following three phases, (a) reaching phase, (b) arm dressing phase, and (c) body dressing phase. The arm dressing phase is treated as a global trajectory modification and implemented by applying the Dynamic Movement Primitives (DMP). The body dressing phase is represented as a local trajectory modification and executed by employing the

[*] Doctoral Dissertation, Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, KYUTECH-LSSE-17899033, September 14, 2020.

Bayesian Gaussian Process Latent Variable Model (BGPLVM). It is demonstrated that the proposed framework developed towards assisting the elderly is generalizable to various people and successfully performs a sleeveless T-shirt dressing task.

Furthermore, in this thesis, various limitations and improvements to the framework are discussed. These improvements include the followings (a) evaluation of Robotic Clothing Assistance, (b) automated wheelchair movement, and (c) incremental learning to perform Robotic Clothing Assistance. Evaluation is necessary for our framework. To make it accessible in care facilities, systematic assessment of the performance, and the devices' effects on the care receivers and caregivers is required. Therefore, a robotic simulator that mimicks human postures is used as a subject to evaluate the dressing task. The proposed framework involves a wheeled chair's manually coordinated movement, which is difficult to perform for the elderly as it requires pushing the chair by himself. To this end, using an electric wheelchair, an approach for wheelchair and robot collaboration is presented. Finally, to incorporate different human body dimensions, Robotic Clothing Assistance is formulated as an incremental imitation learning problem. The proposed formulation enables learning and adjusting the behavior incrementally whenever a new demonstration is performed. When found inappropriate, the planned trajectory is modified through physical Human-Robot Interaction (HRI) during the execution.

This research work is exhibited to the public at various events such as the International Robot Exhibition (iREX) 2017 at Tokyo (Japan), the West Japan General Exhibition Center Annex 2018 at Kokura (Japan), and iREX 2019 at Tokyo (Japan).

Keywords:

Assistive Robotics, Robotic Clothing Assistance, Dynamic Movement Primitives, Bayesian Gaussian Process Latent Variable Model, Human-Robot Interaction, Learning from Demonstration, Imitation Learning

Contents

List of Figures

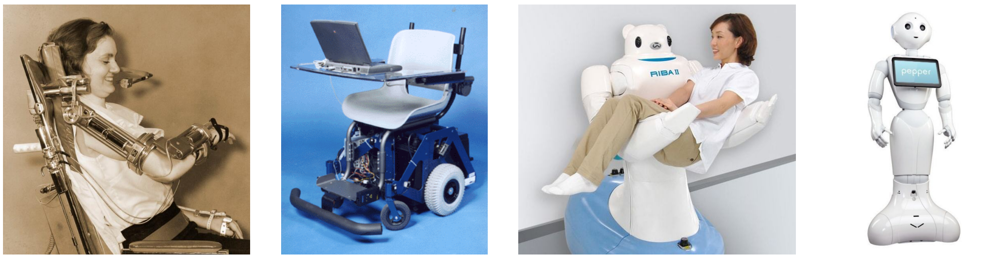
List of Tables

1 Introduction

Assistive robotics is playing a key role in empowering and endorsing human life. These types of robots build a new category because of the following reason. They need to share their workspace, communicate with humans, and assist individuals with the aid of sensors. Various types of service robots are shown in Fig. 1.1.

Assistive robots are designed to compensate for any lost function of a person. These robots enable motor therapy procedures for recovering motor control and motor capabilities for independent living for the disabled and elderly. For example, Rancho arm was the first electric orthosis amenable to external control. Wheelesley is another assistive device used for indoor navigation. It is a robotic wheelchair system that comprises a user interface.

Assistive robots come in many forms. For example, the Robot for Interactive Body Assistance (RIBA) can pick and lift patients off the floor. On the other hand, Pepper is a humanoid robot which focuses on assisting users through social and physical interaction. It can provide an adaptive and engaging conversation with humans for companionship.



(a) Rancho [1] (b) Wheelesley [2] (c) RIBA [3] (d) Pepper [4]

Figure 1.1: Various types of service robots

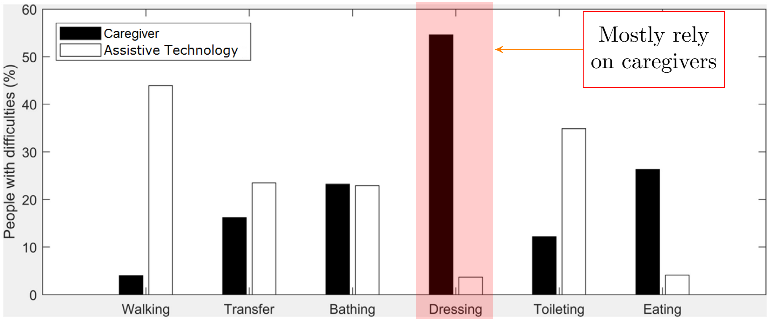


Figure 1.2: People using caregiver/assistive technology for ADL [10]

1.1 Motivation of the Research

The world's population is rapidly ageing. The number of people aged 60 years or older is expected to rise from 12% to 22% of the total global population between 2015 and 2050 [5]. Most of the developed countries across the globe have been ageing for decades. With the ageing population, life expectancy is rising beyond 80 years in countries such as Japan [6]. The number of elderly who need support for Activities of Daily Living (ADL) [7] in developing countries is forecast to quadruple by 2050 [5, 8]. This dramatic increase in ageing population combined with fallen fertility rates are reaching up to an alarming situation. It brings a rising need for long-term care, including home nursing. Many initiatives are being taken care of to address these issues for sustainable growth. European Commission (EU) and the member states have invested more than €1 billion to empower research and development for the welfare of the elderly [9]. According to a survey focusing on difficulties in performing various ADLs, the use of caregivers was seen as more common for tasks such as dressing. Only 3.7% of people were found using assistive technology, whereas other rely on personal caregivers [10] as shown in Fig. 1.2.

As per the Japanese ministry's estimate, the nation will need 2.53 million caregivers in fiscal 2025, but the number will fall short of demand by 377,000 [11] as shown in Fig. 1.3.

Therefore, with the rapidly ageing population and the shortage of caregivers, a robotic solution for assisting in dressing can significantly improve the quality of life of the elderly and disabled.

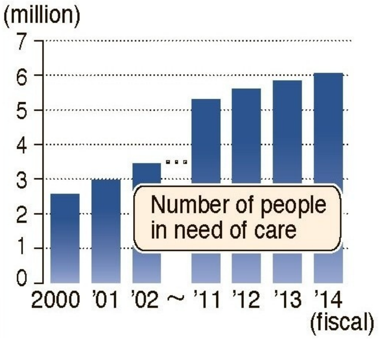


Figure 1.3: Demand of caregivers in Japan [11]

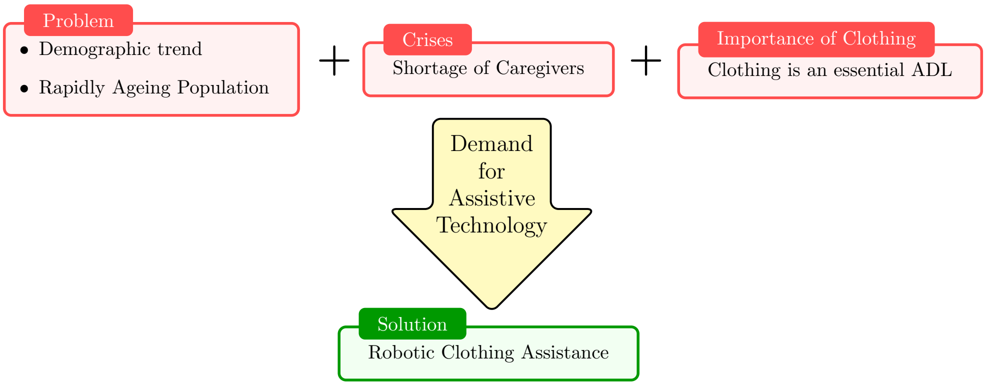

Problem
• Demographic trend
• Rapidly Ageing Population
+
Crises
Shortage of Caregivers
+
Importance of Clothing
Clothing is an essential ADL
Demand for Assistive Technology
Solution
Robotic Clothing Assistance

Figure 1.4: A pictorial representation of Robotic Clothing Assistance

1.2 Robotic Clothing Assistance

The recent demographic trends across the developed nations combined with a shortage of caregivers has created a demand for Robotic Clothing Assistance. A pictorial representation is shown in Fig. 1.4. Robotic Clothing Assistance has the immense potential to improve the Quality of Life (QoL) of the elderly while reducing the burden on caregivers considerably. It can cope with the shortage of caregivers in the care-house.

Robotic Clothing Assistance is hard to accomplish because of the following two major difficulties-

- Cloth manipulation
- Safe human-robot interaction

Clothes are non-rigid and highly flexible objects, which make it difficult to manipulate. Unlike rigid object manipulation, which heavily relies on precise robot control, cloth manipulation requires adaptive control. The tight coupling between the human and the clothing article is difficult to model, wherein the clothing article undergoes severe deformations. Concerning safety, the robot needs to take care of the human whose posture may vary while assisting.

Robotic Clothing Assistance is designed for hospital environments such as elderly care house. A dual-arm robot can be installed in a care house facility to provide dressing assistance for all elderly people. Assuming that person is sitting in a wheelchair (electric), dressing will be performed by robot wheelchair coordination. According to CDC HRQOL-14 [12] and EQ-5D-5L [13], dressing is one of the essential factors for measuring the QoL. Robotic Clothing Assistance can impart freedom to the elderly as they do not need to depend on caregivers for dressing. On the other hand, Robotic Clothing Assistance will reduce the burden of work on caregivers. In this way, we confirm that Robotic Clothing Assistance will improve the QoL of the elderly.

1.3 Thesis Overview

In this thesis, we propose a framework for Robotic Clothing Assistance by imitation learning from a human demonstration to a compliant dual-arm robot, since clothing assistance is generally not difficult for humans. We divide the dressing task into three phases, i.e., reaching phase, arm dressing phase, and body dressing phase. We apply various learning from demonstration approaches on each phase. We enumerate the limitations of the framework and propose improvements. These improvements include (a) evaluation of Robotic Clothing Assistance, (b) automated wheelchair movement, and (c) incremental learning to perform Robotic Clothing Assistance.

The outline of the thesis is as follows. The thesis is devided into 9 chapters. Chapter 2 presents an overview of the related works. Chapter 3 summarizes various imitation learning methods used for the modeling of the problems in this thesis. Chapter 4 describes the setup of our experiment. In Chapter 5, we explain the proposed framework and provide results for each dressing phase. We enumerate the limitations of the framework and propose improvements. Therefore, in Chapter 6 we use a robotic subject to evaluate the proposed framework. Chapter 7 mentions futher improvements by providing an approach for collaboration of the electric wheelchair and the robot. In Chapter 8, we formulate the problem of Robotic Clothing Assistance as an incremental imitation learning. Finally, Chapter 9 concludes the thesis with directions for future research and open questions for real-world implementation of Robotic Clothing Assistance.

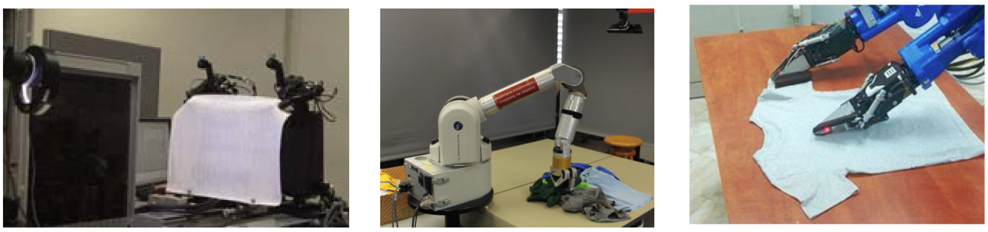
2 Related Work

The state of the art research related to cloth manipulation is mainly implemented for cloth folding tasks. In order to understand the challenges involved in Robotic Clothing Assistance, first, we need to look at the sophisticated research related to the cloth manipulation. In the subsequent section, we will summarize the related works in the field of cloth manipulation followed by the actual Robotic Clothing Assistance task.

2.1 Cloth Manipulation

In order to deal with tedious and complex cloth modeling, Monsó *et al.* [14] proposed a probabilistic planner based on Partially Observable Markov Decision Process (POMDP). The planner was applied to perform clothes sorting task from a pile of laundry. Yamazaki *et al.* [15] exploited optical flow on image sequences for cloth state estimation in trouser dressing tasks. They propose a method to recognize the cloth state by matching the optical flow with the training dataset. Yamakawa *et al.* [16] proposed visual feedback control for dynamic manipulation of sheet-like flexible objects by a high-speed robotic system that learns necessary motor-skills from demonstration performed by a human subject. Kita *et al.* [17] applied a model-driven strategy to recognize the shape of the clothing article based on observations during manipulation. They used a humanoid robot and provided a three-dimensional view to pick up an arbitrarily placed clothing article on a desk and spread it open by holding the cloth at specific locations.

Willimon *et al.* [19] performed classification of clothing article lying in a pile of laundry by comparing against a dataset of known items. A robot equipped with an overhead stereo and side-facing cameras was used to interact with the pile to isolate each item one at a time. Graph-based segmentation algorithm and



(a) Yamakawa *et al.* [16] (b) Monsó *et al.* [14] (c) Doumanoglou *et al.* [18]

Figure 2.1: Related studies showing cloth manipulation

nearest neighbor algorithm (K-NN) on a dataset labeled in a supervised manner were used. Li *et al.* [20] worked on cloth folding by finding an optimal trajectory to move robotic arm given start and end folding position. A quadratic objective function defined using material properties of the garment and frictional force was trained offline in a simulation environment, which was later executed on a real robotic arm. Towner *et al.* [21] tackled the problem of identifying clothing articles from an unknown configuration and bringing it to the desired configuration. They used Hidden Markov model (HMM) for estimating the identity of the clothing article. Miller *et al.* [22] performed a shape-based classification by examining the best-parametrized model fit for recognizing the configuration of clothing articles spread in a pile of laundry. Ono *et al.* [23] developed a two finger hand for cloth handling which can separate a clothing article from the pile of clothes. The thickness of clothing articles was found proportional to the output of the strain gauge of the hand. Doumanoglou *et al.* [18] presented a pipeline for folding a pile of clothes using a dual-arm robot. Cloth spreading was done by detecting deformations of the cloth contours, and grasping points were detected using Active Random Forests.

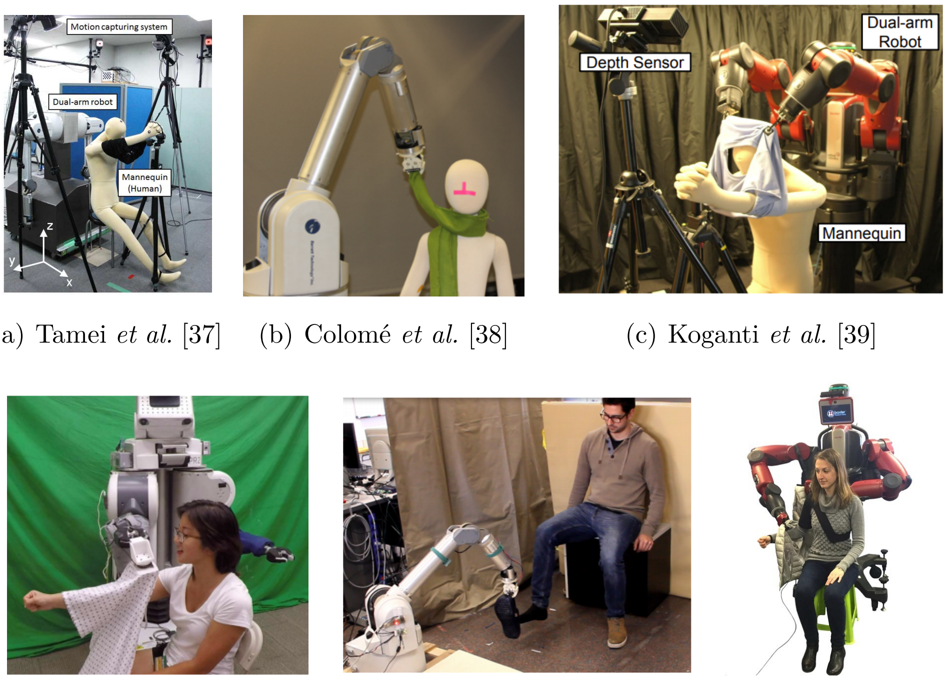
Studies such as Kita *et al.* [17], Willimon *et al.* [19], Li *et al.* [20] and Towner *et al.* [21] heavily rely on offline simulation of the environment wherein no human intervention is considered. The clothing articles are spread out on a flat surface in Towner *et al.* [21] and Miller *et al.* [22], and polygon mesh models are used to approximate clothing articles. These assumptions together with slow manipulation of clothing articles such as Doumanoglou *et al.* [18] can produce inaccurate results during the dressing process since clothing article goes through severe deformations while dressing on a human body.

2.2 Clothing Assistance

Our method is categorized into imitation learning in which the desired behavior is learned from the demonstrations provided by the expert [24]. Learning from Demonstration (LfD) is applied extensively on DMP by [25, 26]. DMP represents a movement with a set of differential equations and captures a deterministic behavior. A method for incrementally learning parameters of DMP is proposed in [27]. They focused on developing an algorithm to address when a robot should learn new motor skills. They used GPs to provide a measure of confidence which is then used to request a new demonstration. DMPs are learned hereafter on the output of GPs. DMP captures the mean behavior from demonstrations, whereas a probabilistic distribution is learned on ProMP [28]. This probabilistic formulation enables the encoding of the mean and variance across the demonstrations. Context-dependent motor skills are taught to a robot by employing ProMP in [29]. They proposed another incremental learning by modeling a joint probability distribution over refined trajectories and corresponding context variables. The policy for generating the trajectory parameter is incrementally learned in the context of shared control for teleoperation [30]. The human feedback through physical interactions is also studied in [31, 32].

Reinforcement Learning (RL) is closely related to imitation learning. In RL, the agent needs to perform numerous interactions with the environment. The high dimensionality of state space in RL can be taken care by Deep Q-learning [33, 34] which uses Deep Neural Networks (DNN). Furthermore, GAIL [35] uses an adversarial approach to learn a policy directly from expert behavior. In these methods, the agent needs to perform countless interactions with a test environment. Therefore, Deep Q-learning and GAIL are very hard to employ on a real robot. Previous studies [35] are performed on simulation environment OpenAI Gym [36]. When a teacher/demonstrator is available, imitation learning is a more efficient way to learn a policy [24].

RL has been applied to tackle the problem of Robotic Clothing Assistance by [37]. They utilized topological coordinates to represent the human-cloth relation-



(a) Tamei *et al.* [37] (b) Colomé *et al.* [38] (c) Koganti *et al.* [39]

(d) Erickson *et al.* [40] (e) Jevtić *et al.* [41] (f) Zhang *et al.* [42]

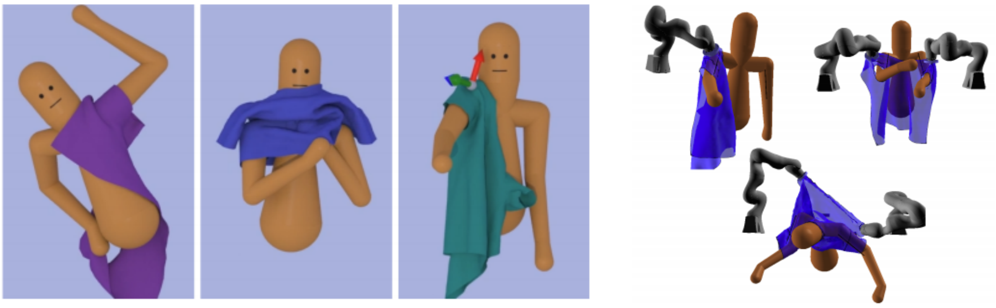
Figure 2.2: Related studies showing Robotic Clothing Assistance

ship and used finite difference policy gradient on a subset of policy parameters. Although, their method can improve the performance through trial and error, but the generalization capabilities of their work is limited. To deal with the complexity of clothing assistance, [43] applied the Bayesian Gaussian Process Latent Variable Model (BGPLVM) to obtain an efficient representation of motor-skills in a latent space. They used a RL method on the latent space generated by the BGPLVM. They assumed that the cloth is already in the arms of a static mannequin. Colomé et al. [38] proposed a framework for scarf dressing by incorporating an analytical model of friction in the robot joints using RL. In Robotic Clothing Assistance, the robot needs to take care of the posture of the person, which may change during the task. Therefore, [40] tackled the problem of human pose tracking during the clothing assistance task by using proximity sensing. Zhang *et al.* [42] performed tracking of the user's movements to perform Robotic Clothing Assistance. Jevtić *et al.* [41] developed a personalized robot with speech and gesture recognition to enhance HRI.

The studies presented above related to Robotic Clothing Assistance have critical limitations such as limited generalizability, inability to learn incrementally from new demonstrations. We overcome these limitations by proposing an incremental learning process that learns whenever a new demonstration is performed and adjusts itself according to the new demonstration. Our incremental learning process allows correcting the robot behavior by performing physical interaction with the robot.

2.3 Deep Learing

In literature, Deep Neural Networks (DNN) is well known for classification and regression problems. Clegg *et al.* [44] used Deep Reinforcement Learning (DRL) for hospital gown dressing by human-robot collaboration. They used a simulation environment to learn control policies for human and robot simultaneously. They employed a co-optimization approach to train two dense DNN and optimized it to maximize expected long-term rewards. DNN is very promising in classification and regression problems demands a massive amount of data to learn. Collecting a large dataset in robotics, especially in Robotic Clothing Assistance, is highly



(a) Clegg *et al.* [45]

(b) Clegg *et al.* [44]

Figure 2.3: Related studies using Deep Learning

challenging and time-consuming and hence discouraged by the research community in robotics. On the other hand, in DRL based methods, the agent needs to perform numerous interactions with a test environment [45]. The test/simulation environment contains limited knowledge about the environment. It can be extremely unfeasible to model a simulation environment for a real and complex environment. In our research, we employed Manifold Relevance Determination (MRD) [46], because it adopts Bayesian treatment to learn from a little amount of data. It constructs a low dimensional latent space by determining a common representation among underlying high dimensional data.

3 Imitation Learning

3.1 Dynamic Movement Primitives

DMP aims at designing a controller for learning and generalization of motor skills by learning from demonstration [25, 47]. The controller is based on a nonlinear dynamical system and uses Locally Weighted Regression (LWR) to learn complex, discrete or rhythmic movements demonstrated by a human subject [48]. The basic idea behind DMP formulation is to use an analytically well-understood dynamical system and add a nonlinear term so that it produces the desired behavior [49]. Originally, for a one-dimensional system, DMP is defined by a linear spring model combined with an external force as follows

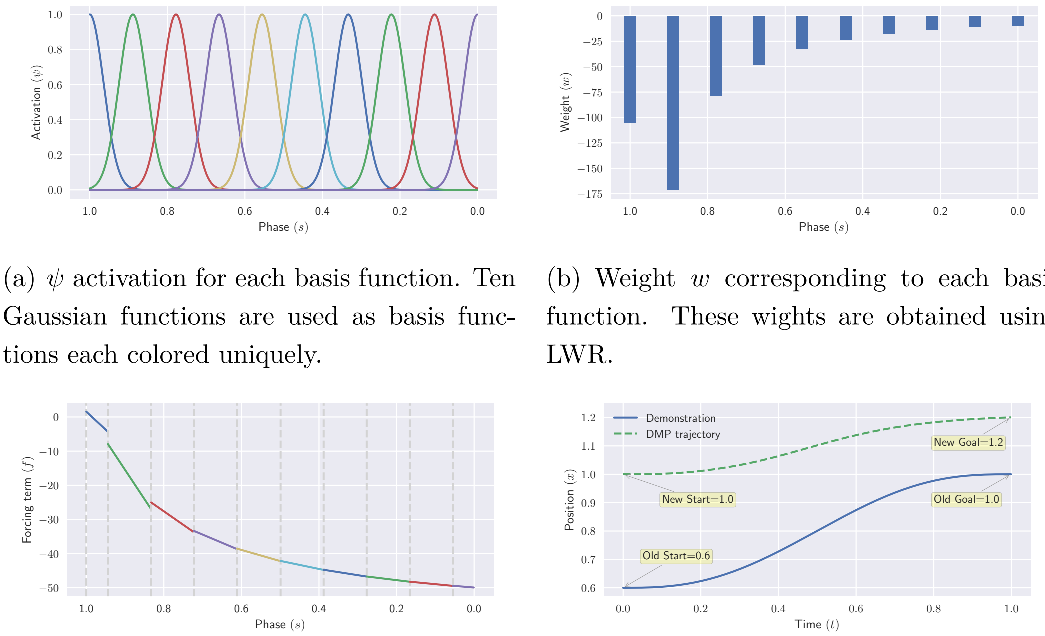
$$\tau \dot{v} = K(g - x) - Dv + (g - x_0)f \tag{3.1}$$

where

$$\tau \dot{x} = v$$

The term x and v are position and velocity of the system respectively, x_0 and g are start and goal position respectively. τ is the temporal scaling term, K acts like spring constant and D is damping factor chosen in a way such that system is critically damped. However, the above formulation of DMP suffers from stability issues such as high accelerations for special cases. Hence, a new formulation was proposed by Pastor *et al.* [50] in which Eq. (3.1) was redefined as follows

$$\tau \dot{v} = K(g - x) - Dv - K(g - x_0)s + Kf(s) \tag{3.2}$$



(a) ψ activation for each basis function. Ten Gaussian functions are used as basis functions each colored uniquely.

(b) Weight w corresponding to each basis function. These wights are obtained using LWR.

(c) Forcing term f corresponding to each basis function and colored uniquely. For each Gaussian, function f gets activated for a certain time period as shown by the gray colored dashed vertical lines.

(d) Learning a trajectory using DMP. Start and goal points of a demonstrated trajectory are modified to generate a DMP trajectory.

Figure 3.1: Brief overview of a one-dimensional discrete DMP system.

where s is called phase variable. Phase variable s starts from 1 and monotonically decreases to 0, defined by following equation $\tau\dot{s} = -\alpha s$, where α is a positive gain term. In Eq. (3.2), notice the term $K(g-x_0)s$ which is necessary for avoiding a sudden jump at the beginning of a movement. The nonlinear function f, which is also called the forcing term is a non-linear function to be learned to allow complex movements. The forcing function f is chosen as

$$f(s) = \frac{\Sigma_i w_i \psi_i(s)}{\Sigma_i \psi_i(s)} s \tag{3.3}$$

where ψ_i is defined as Gaussian basis function as

$$\psi_i = \exp\left(-h_i\left(s - c_i\right)^2\right) \tag{3.4}$$

where h_i and c_i are constants that determine, respectively, width and centers of basis functions. w_i represents weight defined for each Gaussian. Forcing function f depends on phase variable s. Our goal is to design a forcing function that can learn from demonstration and allows us to scale the movement defined by start and goal state, i.e., x_0 and g respectively. So that the system can follow a specified path. The forcing term can be redefined as

$$f_{target}(s) = \frac{Dv + \tau\dot{v}}{K} - (g - x) + (g - x_0)s \tag{3.5}$$

where desired acceleration $\dot{v}(t)$ can be calculated by taking the second derivative of the positional data recorded from demonstration as

$$\dot{v}(t) = \frac{\partial v}{\partial t} = \frac{\partial^2 x}{\partial t^2} \tag{3.6}$$

The forcing function in Eq. (3.3) is comprised of the weighted summation of Gaussians that are going to be activated as system converges to goal as shown in Fig. 3.1. We want that forcing function matches the desired trajectory, i.e., f_{target} should be as close as possible to f. Mathematically, we can formulate it as an optimization problem such as

$$J = \sum_{s} \left(f_{target}(s) - f(s)\right)^2 \tag{3.7}$$

This ends by calculating weight parameters across Gaussians. Optimization methods such as LWR [51] can be used, so that forcing function matches the desired trajectory. In this way, DMP can be made to imitate the desired path [50].

3.2 Bayesian Gaussian Process Latent Variable Model

BGPLVM is an extension of GPLVM [52] in which inputs are unobserved, treated as latent variables and outputs are observed using multiple output Gaussian Process (GP) regression model as shown in Fig. 3.2.

Let $\boldsymbol{Y} \in \mathbb{R}^{N \times D}$ be the observed data where N is the number of observations, and D is the dimensionality of each observation. In Latent Variable Model (LVM) methodology, we assume that these observations are generated from an unobserved space also known as latent space $\boldsymbol{X} \in \mathbb{R}^{N \times Q}$ such that $Q \ll D$. We start by defining a generative mapping from latent space to observation space as

$$\boldsymbol{y}_n = f(\boldsymbol{x}_n) + \epsilon_n \; \forall \; \epsilon_n \sim \mathcal{N}(\boldsymbol{0}, \beta^{-1}\boldsymbol{I}) \tag{3.8}$$

where $\boldsymbol{y}_n$, $\boldsymbol{x}_n$ represents n^{th} row of $\boldsymbol{Y}, \boldsymbol{X}$ respectively. The random noise variable ϵ is derived from a Gaussian distribution with mean zero and covariance $\beta^{-1}\boldsymbol{I}$. It should be noted that the above mapping is governed by GP[53] wherein GPs are taken to be independent across the features. Hence conditional likelihood is written as

$$\begin{aligned} p(\boldsymbol{Y}|\boldsymbol{X}) &= \prod_{d=1}^{D} p(\boldsymbol{y}_d|\boldsymbol{X}) \\ &= \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{y}_d|\boldsymbol{0}, K_{NN} + \beta^{-1}I_N) \end{aligned} \tag{3.9}$$

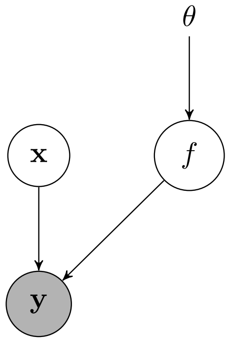


Figure 3.2: Graphical representation of BGPLVM. Gray circle represents the observed variable whereas the white circle shows the latent variable.

In the above equation, Gaussian distribution has zero mean and covariance $K_{NN} + \beta^{-1} I_N$. Term K_{NN} is $N \times N$ covariance matrix composed by kernel function $k(\boldsymbol{x}, \boldsymbol{x}\prime)$, which is defined by Automatic Relevance Detection (ARD) kernel [54, 53]. ARD kernel solves model selection problem by determining the dimensionality of latent space and is written as

$$k(\boldsymbol{x}, \boldsymbol{x}\prime) = \sigma^2_{ARD} \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} \alpha_q (x_q - x\prime_q)^2\right) \tag{3.10}$$

ARD kernel consists of weight parameters $\{\alpha_q\}_{q=1}^{Q}$ which describe the relevance of each dimension. Term σ_{ARD} describes the scale of GP mapping function [39].

The model consists of hyperparameters $\theta = (\beta, \sigma^2_{ARD}, \{\alpha_q\}_{q=1}^{Q})$. The objective is to infer latent variable $\boldsymbol{X}$. We can assign a Gaussian prior into it as follows

$$p(\boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n | \mathbf{0}, I_Q) \tag{3.11}$$

The joint probability of the model can be defined as

$$p(\boldsymbol{Y}, \boldsymbol{X}) = p(\boldsymbol{Y}|\boldsymbol{X}) p(\boldsymbol{X}) \tag{3.12}$$

Here, we use a variational Bayesian approach for the marginalization of the latent variable $\boldsymbol{X}$, which allows optimizing resulting lower bound on marginal likelihood w.r.t. the hyperparameters θ [55]. The marginal likelihood of the data is written as

$$p(\boldsymbol{Y}) = \int p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X} \tag{3.13}$$

To compute the above equation, we need to use $p(\boldsymbol{Y}|\boldsymbol{X})$ from Eq. (3.9). Eq. (3.9) contains $\boldsymbol{X}$ nonlinearly inside $K_{NN} + \beta^{-1}I_N$ as shown in Eq. (3.10), which makes it intractable. Here, we make use of variational inference to approximate posterior distribution $p(\boldsymbol{X}|\boldsymbol{Y})$ by introducing a variational distribution $q(\boldsymbol{X})$ written as

$$q(\boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_n, S_n) \tag{3.14}$$

where $\{\boldsymbol{\mu}_n, S_n\}_{n=1}^{N}$ are variational parameters. The idea behind variational inference is to use Kullback-Leibler (KL) divergence as a measure of distance between $p(\boldsymbol{X}|\boldsymbol{Y})$ and $q(\boldsymbol{X})$. It allows computing Jensen's lower bound on marginal likelihood as follows

$$\begin{aligned}\log p(\boldsymbol{Y}) &\geq F(q) \\ &= \int q(\boldsymbol{X}) \log p(\boldsymbol{Y}|\boldsymbol{X})d\boldsymbol{X} - \int q(\boldsymbol{X}) \log \frac{q(\boldsymbol{X})}{p(\boldsymbol{X})} d\boldsymbol{X} \\ &= \tilde{F}(q) - \mathrm{KL}(q\|p)\end{aligned} \tag{3.15}$$

The hyperparameters θ are dropped for notation clarity. Term $\tilde{F}(q)$ can be computed by using Eq. (3.9) and by breaking it down to separate computation at each dimension of observation as follows

$$\begin{aligned}\tilde{F}(q) &= \sum_{d=1}^{D} \int q(\boldsymbol{X}) \log p(\boldsymbol{y}_d|\boldsymbol{X})d\boldsymbol{X} \\ &= \sum_{d=1}^{D} \tilde{F}_d(q)\end{aligned} \tag{3.16}$$

The term $\tilde{F}_d(q)$ contains an intractable integration because conditional likelihood term $p(\boldsymbol{y}_d|\boldsymbol{X})$ contains $\boldsymbol{X}$ nonlinearly inside the inverse of covariance matrix, i.e., $K_{NN} + \beta^{-1}I_N$. In order to derive a closed-from lower bound for $\tilde{F}_d(q)$, variational sparse GP regression [56] was used in which auxiliary variables were introduced in an augmented probability model [55]. The augmented model defines $\boldsymbol{f}_d \in \mathbb{R}^N$ associated with $\boldsymbol{y}_d$ as follows

$$\begin{aligned} p(\boldsymbol{y}_d|\boldsymbol{f}_d) &= \mathcal{N}(\boldsymbol{y}_d|\boldsymbol{f}_d, \beta^{-1}I_N) \\ p(\boldsymbol{f}_d|\boldsymbol{X}) &= \mathcal{N}(\boldsymbol{f}_d|\boldsymbol{0}, K_{NN}) \end{aligned} \tag{3.17}$$

M auxiliary variables also known as inducing variables, i.e., $\boldsymbol{u}_d \in \mathbb{R}^M$ are defined in pseudo input locations $\hat{\boldsymbol{X}} \in \mathbb{R}^{M\times Q}$. The augmented joint probability of the model is given by following

$$p(\boldsymbol{y}_d, \boldsymbol{f}_d, \boldsymbol{u}_d|\boldsymbol{X}, \hat{\boldsymbol{X}}) = p(\boldsymbol{y}_d|\boldsymbol{f}_d)p(\boldsymbol{f}_d|\boldsymbol{u}_d, \boldsymbol{X}, \hat{\boldsymbol{X}})p(\boldsymbol{u}_d|\hat{\boldsymbol{X}}) \tag{3.18}$$

GP prior evaluated at input $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ factorizes as

$$p(\boldsymbol{f}_d, \boldsymbol{u}_d|\boldsymbol{X}, \hat{\boldsymbol{X}}) = p(\boldsymbol{f}_d|\boldsymbol{u}_d, \boldsymbol{X}, \hat{\boldsymbol{X}})p(\boldsymbol{u}_d|\hat{\boldsymbol{X}})$$

gives following

$$p(\boldsymbol{f}_d, \boldsymbol{u}_d|\boldsymbol{X}, \hat{\boldsymbol{X}}) = \mathcal{N}(\boldsymbol{f}_d|\alpha_d, K_{NN} - K_{NM}^{-1}K_{MN}) \tag{3.19}$$

where $\alpha_d = K_{NM}K_{MM}^{-1}\boldsymbol{u}_d$ and marginal GP prior over inducing variables is given by $p(\boldsymbol{u}_d|\hat{\boldsymbol{X}}) = \mathcal{N}(\boldsymbol{u}_d|\boldsymbol{0}, K_{MM})$.

Above calculations leads to an interpretation which says that unlike $\boldsymbol{X}$, inducing inputs $\hat{\boldsymbol{X}}$ are neither random variables nor model hyperparameters. They are treated as variational parameters [57]. Variational distribution over inducing variables was found independent of $\boldsymbol{X}$. The model leads to tractable Jensen's lower bound $\tilde{F}_d(q)$ by using mean field approach which forces independent distribution from the input variable $\boldsymbol{X}$ thereby making the approximation tractable. Detailed derivation of the model is further explained in [55].

In order to make predictions $p(y_*|\boldsymbol{Y})$ in unseen data $y_* \in \mathbb{R}^D$, latent variables $\boldsymbol{X}$ and new test latent variables x_* are introduced, and the ratio of two marginal likelihoods is calculated as follows

$$\begin{aligned} p(y_*|\boldsymbol{Y}) &= \frac{p(y_*, \boldsymbol{Y})}{p(\boldsymbol{Y})} \\ &= \frac{\int\int p(y_*, \boldsymbol{Y}|\boldsymbol{X}, x_*)p(\boldsymbol{X}, x_*)dXdx_*}{\int p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X}} \end{aligned} \tag{3.20}$$

The term $\int p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X}$ gives variational distribution $q(\boldsymbol{X})$ and is fixed during test time. $\int\int p(y_*, \boldsymbol{Y}|\boldsymbol{X}, x_*)p(\boldsymbol{X}, x_*)dXdx_*$ is approximated by the ratio of lower bounds as follows

$$\begin{aligned} p(y_*|\boldsymbol{Y}) &\approx q(y_*|\boldsymbol{Y}) \\ &= \exp\left(F(q(\boldsymbol{X}, x_*) - F(q(\boldsymbol{X}))\right) \end{aligned} \tag{3.21}$$

Detailed steps of the prediction process are further explained in [55].

3.3 Manifold Relevance Determination

MRD is a nonlinear dimensionality reduction technique proposed by Damianou *et al.* [46]. It is used to learn a shared latent space among multiple observation spaces. It involves the use of Bayesian inference as it was proposed as an extension to the BGPLVM proposed by Titsias *et al.* [55].

MRD aims to relate two observation spaces $\boldsymbol{Y} \in \mathbb{R}^{N \times D_Y}$ and $\boldsymbol{Z} \in \mathbb{R}^{N \times D_Z}$ within a single model. Here, N represents the number of observations. D_Y and D_Z represent the dimensionality of each observation, i.e., $\boldsymbol{Y}$ and $\boldsymbol{Z}$ respectively. The two observations spaces are assumed to be generated from a low-dimensional latent space $\boldsymbol{X} \in \mathbb{R}^{N \times L}$ such that $L \ll D$ (to account for the dimensionality reduction) and corrupted by Gaussian noise:

$$\begin{aligned} \boldsymbol{y}_n &= f^Y(\boldsymbol{x}_n) + \epsilon_n^Y, \ \epsilon_n^Y \in \mathcal{N}(\boldsymbol{0}, \beta_Y^{-1}\boldsymbol{I}), \\ \boldsymbol{z}_n &= f^Z(\boldsymbol{x}_n) + \epsilon_n^Z, \ \epsilon_n^Z \in \mathcal{N}(\boldsymbol{0}, \beta_Z^{-1}\boldsymbol{I}) \end{aligned} \tag{3.22}$$

where β_Y, β_Z denote the inverse variance parameters for the noise random variables ϵ_n^Y, ϵ_n^Z. A Gaussian Process (GP) prior is placed on the mapping function f, $f(\boldsymbol{x}) \sim \mathcal{GP}(\mathbf{0}, k(\boldsymbol{x}, \boldsymbol{x}\prime))$, where $k(\boldsymbol{x}, \boldsymbol{x}\prime)$ is the covariance function. $k(\boldsymbol{x}, \boldsymbol{x}\prime)$ is defined using the automatic relevance determination (ARD) kernel (Eq. (3.23)).

The likelihood under the model is denoted by, $\mathsf{P}(\boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta})$ where $\boldsymbol{\theta} = \{\boldsymbol{\theta}^Y, \boldsymbol{\theta}^Z\}$ collectively denotes the parameters of the mapping functions and the noise variances β_Y, β_Z.

The selection of the latent space dimensionality is performed automatically using the ARD kernel,

$$k_Y(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_Y^2 \ \exp\left(-\frac{1}{2}\sum_{l=1}^{L} \alpha_l^Y (x_{i,l} - x_{j,l})^2\right) \tag{3.23}$$

and similarly for the $\boldsymbol{Z}$ observation space. The relevance of each latent dimension is determined by its ARD weight α_l, and the scale of the GP mapping function is determined by σ.

The ARD weights α_l^Y, α_l^Z also help in partitioning the latent space into shared ($\boldsymbol{X}_S$) and private spaces ($\boldsymbol{X}_Y, \boldsymbol{X}_Z$). This is done by using a threshold δ which is set heuristically on the normalized ARD weights to determine the relevance of a latent dimension in reconstructing each observation space. The shared and private spaces are defined as follows:

$$\begin{aligned}
\boldsymbol{X}_S &= \{\boldsymbol{x}_l\}_{l=1}^L : \boldsymbol{x}_l \in \boldsymbol{X}, \ \alpha^Y > \delta, \ \alpha^Z > \delta \\
\boldsymbol{X}_Y &= \{\boldsymbol{x}_l\}_{l=1}^L : \boldsymbol{x}_l \in \boldsymbol{X}, \ \alpha^Y > \delta, \ \alpha^Z < \delta \\
\boldsymbol{X}_Z &= \{\boldsymbol{x}_l\}_{l=1}^L : \boldsymbol{x}_l \in \boldsymbol{X}, \ \alpha^Y < \delta, \ \alpha^Z > \delta
\end{aligned} \tag{3.24}$$

3.3.1 Inference Procedure

In this section, we briefly explain the inference procedure in MRD. Given a set of observed test points $\boldsymbol{Y}^\star \in \mathbb{R}^{N^\star \times D_Y}$, we aim to generate a new set of outputs $\boldsymbol{Z}^\star \in \mathbb{R}^{N^\star \times D_Z}$. This is done in the following three steps:

1. We predict the set of latent points ($\boldsymbol{X}_Y^\star$, $\boldsymbol{X}_S^\star$), which may have generated $\boldsymbol{Y}^\star$. To do this, we compute an approximation of the posterior $\mathsf{P}(\boldsymbol{X}^\star|\boldsymbol{Y}^\star, \boldsymbol{Y})$.

It is given by a variational distribution as in BGPLVM. This variational distribution is found by optimizing a variational lower bound on the marginal likelihood $\mathsf{P}(\boldsymbol{Y}, \boldsymbol{Y}^{\star})$, as given in Titsias *et al.* [55].

2. The shared latent space $\boldsymbol{X}_S^{\star}$ is then used to find the nearest neighbors among the latent points corresponding to the training data and obtain the information on the private dimension of $\boldsymbol{Z}$, $\boldsymbol{X}_Z^{NN}$.

3. We use the full latent state $\boldsymbol{X}_S^{\star}$, $\boldsymbol{X}_Z^{NN}$ to infer the outputs $\boldsymbol{Z}^{\star}$.

Detailed explanation of MRD is given in [46].

4 Experimental System

4.1 Baxter Research Robot

Baxter is a research robot manufactured by Rethink Robotics. It is a 3 foot tall 2 armed humanoid robot with an LCD display on the head as shown in Fig. 4.1.

Each arm of the robot has 7 independent joint and they can move much like human hand. Seven DOF arms are desirable as they provide a kinematic redundancy greatly improving manipulability and safety. Each joint has inbuilt torque sensors, which provides real time joint torque values. The specification of each joint is given in Table 4.1. The robot is easily controlled by using ROS [58]. It provides python API for interacting with the Baxter Research Robot [59].

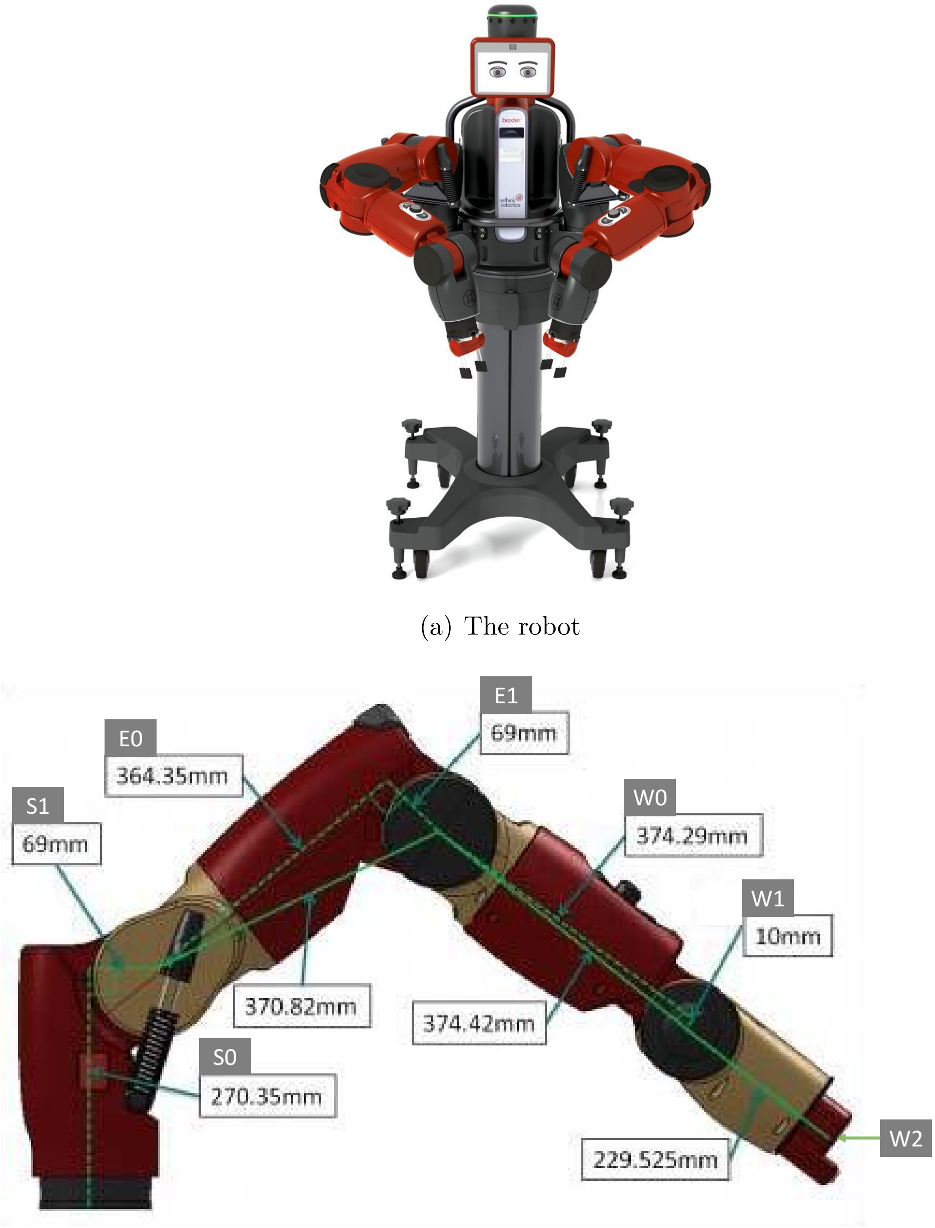
The robot is designed to work effectively directly alongside human in a work place. Series Elastic Actuators [60, 61] are the actuation technique, or mechanism responsible for moving the robot links. This inherently provide flexibility and safety at each joint.

4.1.1 Baxter Safety Information

Unlike typical industrial robots that operate behind wire enclosures, Baxter is a collaborative robot and does not require such safety guards. The robot is designed while concerning the following features-

- Physical interaction between a human and the robot
- Avoiding accidental contact
- Minimizing forces and slowing/stopping during human contact

The robot meets the following safety standards [62, 63, 64]-


(a) The robot

(b) All joints on an arm

Figure 4.1: Baxter research robot and all joints on an arm

Table 4.1: Joints specifications of Baxter arm

Joint	Min Limit (deg)	Max Limit (deg)	Range (deg)	Max Joint Speed (deg/sec)
S0	-97	+97	194	114
S1	-123	+60	183	114
E0	-174	+174	349	114
E1	-3	+150	153	114
W0	-175	+175	350	228
W1	-90	+120	210	228
W2	-175	+175	350	228

- **ISO 10218-1:2006** : Requirement of the power and force-limited control
- **ISO 10218-2** : Risk assessment of the user application to determine the needed safety performance and safeguarding
- **ANSI RIA R15.06-2012** : U.S. adoption of ISO 10218-1 & 2
- **UL 60950-1** : Fire and electrical safety

Following are the safety and compliance features provided due to the advance mechanical design of the Baxter-

- The robot is covered with smooth and impact-absorbing shells. It has padding in key areas, such as the elbows and wrists, and is designed to have reduced sharp points/corners.
- The robot claims injury-free operation, or at worse, a slight injury - "S1" injury or below as per ISO 13849-1 standard.
- The robot is consists of Series Elastic Actuators (SEAs) [60, 61]. It has springs at all joints that provide passive compliance to minimize any contact or impact force.
- All the joints are fully back-drivable and can easily be rotated by hand, even when the robot is powered off.

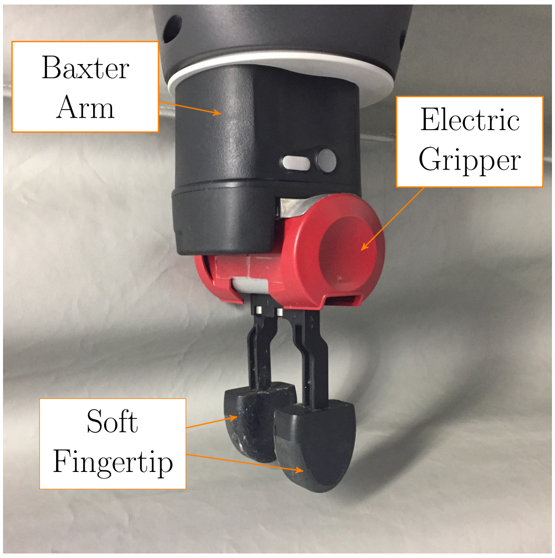


Figure 4.2: Soft fingertips mounted on fingers of the electric gripper of Baxter robot

- The robot operates at moderate velocity (human-equivalent), which makes it easier for people to avoid any unintended contact with the robot.
- The robot has force sensing at each joint which enables the followings-
  - Provides detection and immediate response to contact and impact
  - Prevents Baxter from applying continuous or excessive pressure
  - Keeps impact static forces under design limits

4.2 Soft Fingertip

We have used two finger electric gripper provided by Baxter. We designed soft fingertips which were plugged into these fingers tightly as shown in Fig. 4.2. These soft fingertips 3D printed using soft-material, are necessary for firm gripping of flexible clothing article hence provides better cloth manipulation. The clothing article is held by these fingertips, and it is put in the arms of Baxter robot manually by a human assistant.

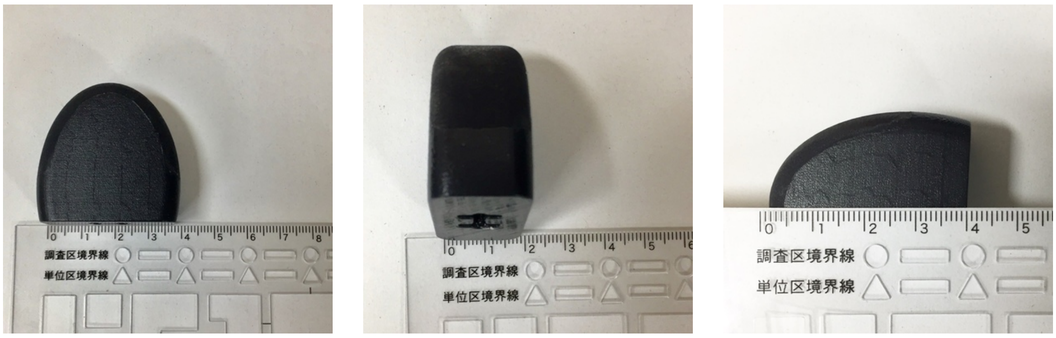


Figure 4.3: Fingertip Dimensions

Figure 4.4: Microsoft Kinect version 2

4.2.1 Fingertip Dimensions and Material Information

To 3D print the fingertips, the Stratasys printer was utilized. A rubber-Like material Agilus30 was used. Agilus30 is a polyjet photopolymer having tear-resistance, capable of withstanding repeated flexing and bending [65]. During the printing, the hardness parameter, i.e., shore, was set to 50.

Dimensions of the fingertips are shown in Fig. 4.3.

4.3 Microsoft Kinect Sensor

Kinect [66] is a line of motion sensing input devices by Microsoft for Xbox 360 and Xbox One video game consoles and Microsoft Windows PCs. Kinect sensor contains an RGB camera that stores three channel data in a 1280 x 960 pixels

Figure 4.5: Intel RealSense sensor model D435

resolution which makes capturing a colour image possible, an infrared (IR) emitter and an IR depth sensor, a multi-array microphone and a 3-axis accelerometer configured for a 2G range. The video and depth sensor cameras have a 640 × 480 pixels resolution and run at 30 FPS. The sensor is shown in Fig. 4.4.

4.4 Intel RealSense Sensor

We are using Intel RealSense Sensor [67, 68] model number D435. It is a stereo camera and provides depth information similar to the Mircosoft Kinect sensor. While the Kinect V2 is designed for a distance of 0.5m - 4.5m, the RealSense is intended for a 0.2m-1.2m operating range. This camera has a very compact form factor, and it is very lightweight too. The RGB and depth resolution is 1920 × 1080 pixels and 1280 × 720 pixels, respectively. A detailed comparison with Kinect sensor can be found here [69].

4.5 Mannequin

Instead of directly experimenting on human, a soft full size male mannequin is used for the experiment. The mannequin is covered with soft form as shown in Fig. 4.6. It has flexible joints and it can be configured to any pose by gently pushing its body.

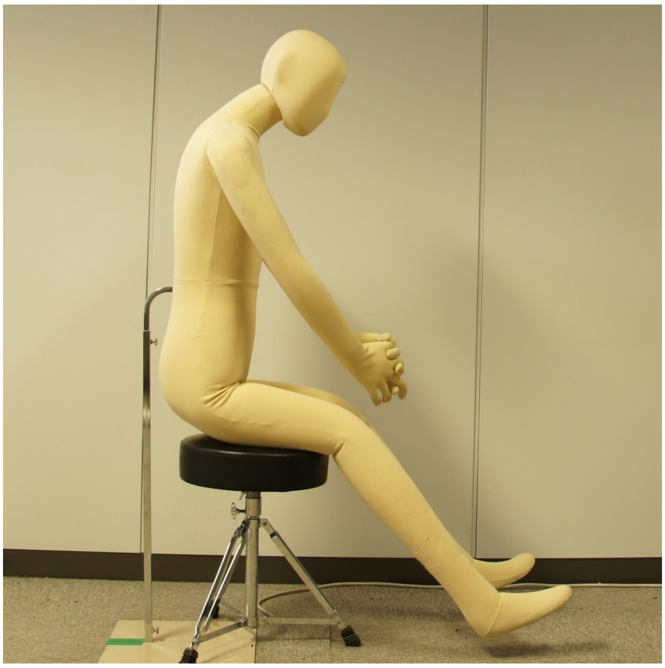


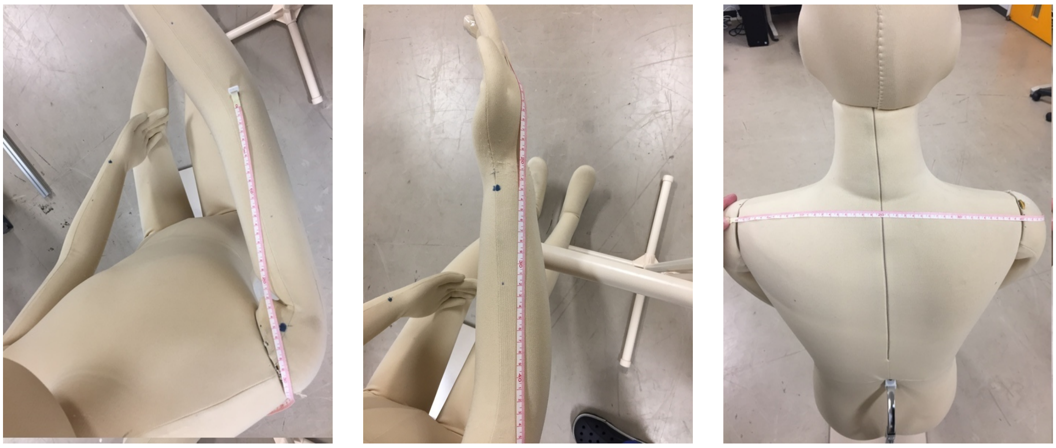
Figure 4.6: Soft-mannequin

Dimensions of the soft-mannequin are shown in Fig. 4.7.

4.6 Clothing Articles

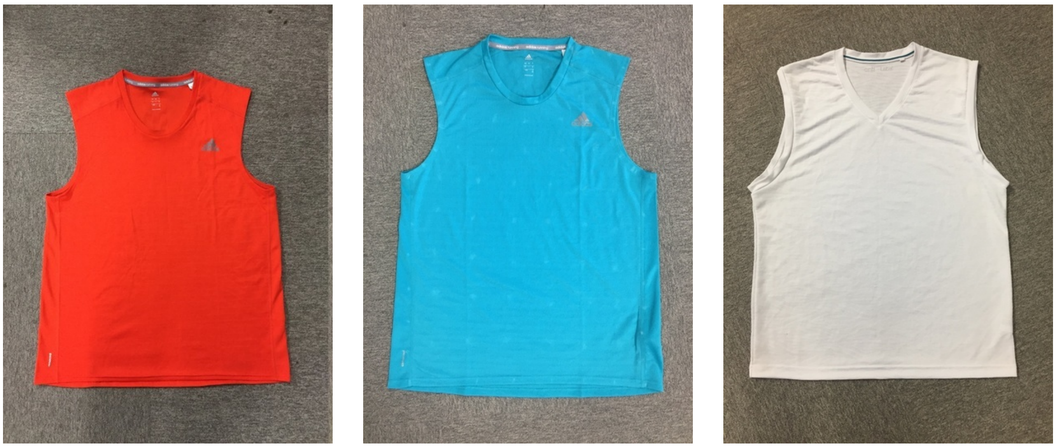
We used 3 polyester T-shirts for performing the clothing task. The clothing articles are sleeveless T-shirts, as shown in Fig. 4.8. The orange and blue color T-shirts are from Adidas, and the white color is from Avail. All of them are 100% polyester sleeveless T-shirts. The size of all T-shirts is L.

Grasping points are determined empirically by observing the picking points of a human demonstrator while he was performing the dressing task. Fig. 4.9 shows grasping points situated at the bottom of each shirt.



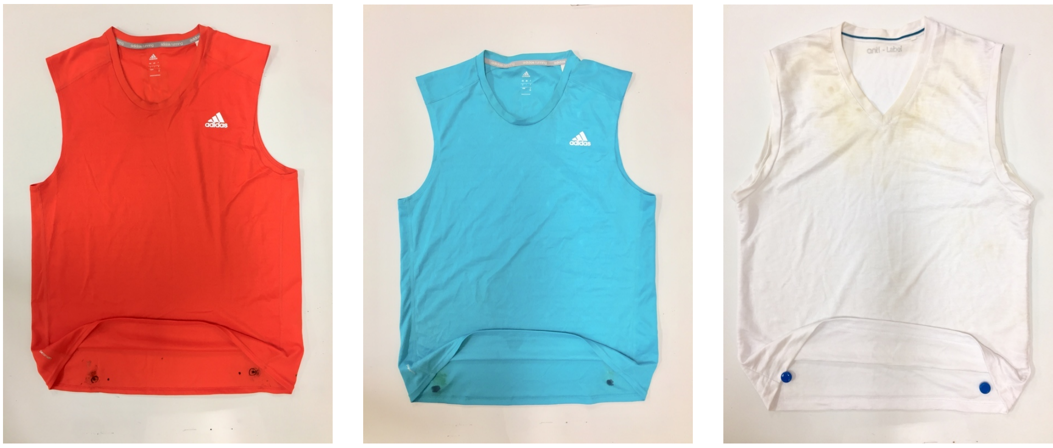
(a) 30 cm (b) 40 cm (c) 40 cm

Figure 4.7: Dimension of the soft-mannequin



(a) T-shirt A (b) T-shirt B (c) T-shirt C

Figure 4.8: Clothing articles (Sleeveless T-shirts)



(a) T-shirt A

(b) T-shirt B

(c) T-shirt C

Figure 4.9: Grasping Points in T-shirts

5 Proposed Framework

Robotic Clothing Assistance is hard to accomplish because of two significant difficulties, (a) cloth manipulation, and (b) safe human-robot interaction. Clothes are non-rigid and highly flexible objects which make them difficult to manipulate. Unlike rigid object manipulation, which heavily relies on precise robot control, cloth manipulation requires complex adaptive control. The tight coupling between the human and the clothing article is difficult to model wherein the clothing article undergoes severe deformations. Concerning safety, the robot needs to take care of the human whose posture may vary while assisting.

In this chapter, we propose a framework for Robotic Clothing Assistance by imitation learning from a human demonstration to a compliant dual-arm robot, since clothing assistance is generally not difficult for humans. In this framework, we divide the dressing task into three phases, i.e., reaching phase, arm dressing phase, and body dressing phase. The reaching phases can be achieved through point-to-point motion planning. We apply Dynamic Movement Primitives (DMP) for the arm dressing phase and Bayesian Gaussian Process Latent Variable Model (BGPLVM) for the body dressing phase. The arm dressing phase is considerably a trajectory planning problem, and it is taken care of by DMP. In the body dressing phase, the robot operates close to the subject. We employ BGPLVM in the body dressing phase. DMP by formulation provides adaptive control of the robot. DMP parameterizes the robot trajectory acquired from the kinesthetic demonstration by a human. By changing the start and goal parameters of DMP, the generated trajectory can be modified globally. Hence, we can say that DMP is goal-directed since a change in the goal affects the entire trajectory. On the other hand, the latent space generated by BGPLVM provides local modification. We assume that performing a dressing task requires a consistent set of motor skills. These motor-skills can be acquired from the robot trajectory by constraining it to a lower dimensional latent space using BGPLVM. The body dressing phase requires a complicated trajectory as compared to the arm dressing phase. Hence, we apply BGPLVM to encode these complicated motor-skills. We have shown that generated latent space by BGPLVM provides safe human-robot interaction during the body dressing phase wherein a tight coupling between the human and the cloth happens.

Clothing assistance is difficult to tackle using reinforcement learning as it takes a long time which is undesirable in dressing tasks, to find the optimal policy. This difficulty can be resolved by using imitation learning as a form of prior knowledge. The prior knowledge is incorporated by using Learning from Demonstration (LfD) frameworks to avoid the complexity and uncertainty associated with the modeling of the clothing assistance environment which consists of the human, robot and clothing article. We evaluated the proposed framework on human subjects by dressing a sleeveless T-shirt using the Baxter robot. Our approach is focused on assisting the elderly and disabled having limited upper arm movement. Hence while assisting, the robot cooperates with the subject and expects minimal upper arm movement to provide a relaxed experience. The robot used in this research meets the international ANSI RIA R15.06-2012 and ISO 10218-1:2006 safety standards [70], thus suitable for our research.

5.1 Setup of the System

The experimental setup contains a compliant dual-arm humanoid robot Baxter. Each arm of the Baxter robot has 7 degrees of freedom (DOF). The setup of our system is shown in Fig. 5.1. A Kinect v2 [66] depth sensor is mounted below the LCD on the chest of Baxter by a custom designed mount. We have used two finger electric gripper provided by Baxter. We designed soft fingertips which were plugged into these fingers tightly as shown in Fig. 4.2. These soft fingertips 3D printed using soft-material, are necessary for firm gripping of flexible clothing article hence provides better cloth manipulation. The clothing article is held by these fingertips, and it is put in the arms of Baxter robot manually by a human assistant. A chair is provided to the subject to sit on during the dressing task and face the Baxter. The robot puts the clothing article on the human subject.

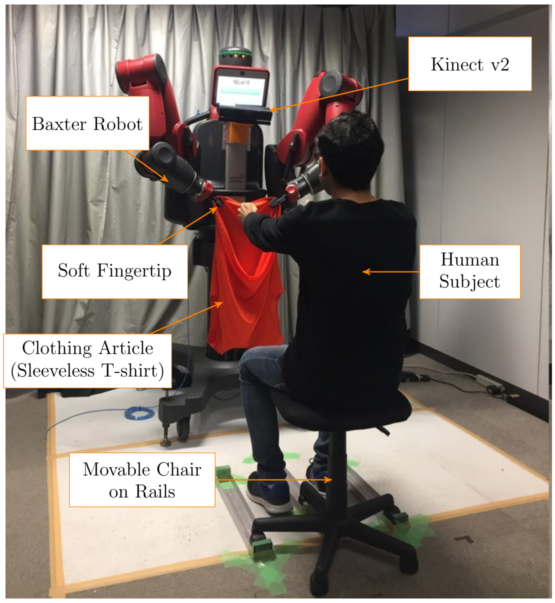


Figure 5.1: Setup of Robotic Clothing Assistance task showing various components of the system.

During the process clothing article goes over the arms of the human subject. Hence it is essential for the subject to keep his arms straight and to face towards the robot. It should be noted that Baxter's arms have limited workspace and it cannot reach the torso of the subject while arms of the subject are extended. Therefore we propose to use a portable chair wherein the movement of the chair is restricted by keeping it on rails. This arrangement of the chair provides sufficient movement required for performing a dressing assistance task.

We use Robot Operating System (ROS) [58] to implement our framework in Ubuntu OS. Baxter robot is connected to this computer using an Ethernet cable. For human pose tracking, official Kinect APIs were used in a separate Windows OS which is also connected to Ubuntu OS. The skeleton tracking data is transferred from Windows OS to Ubuntu OS in real-time using ZeroMQ [71], a high performance distributed asynchronous messaging library. We used Ubuntu 14.04 LTS 64 Bit OS having 8GB RAM on Intel Core i7, 3.40 GHz x 8 CPU for training and testing our framework. The clothing articles used in this study are Adidas and Avail 100% polyester (size L) sleeveless T-shirt as shown in Fig. 4.8.

5.2 Dressing Task

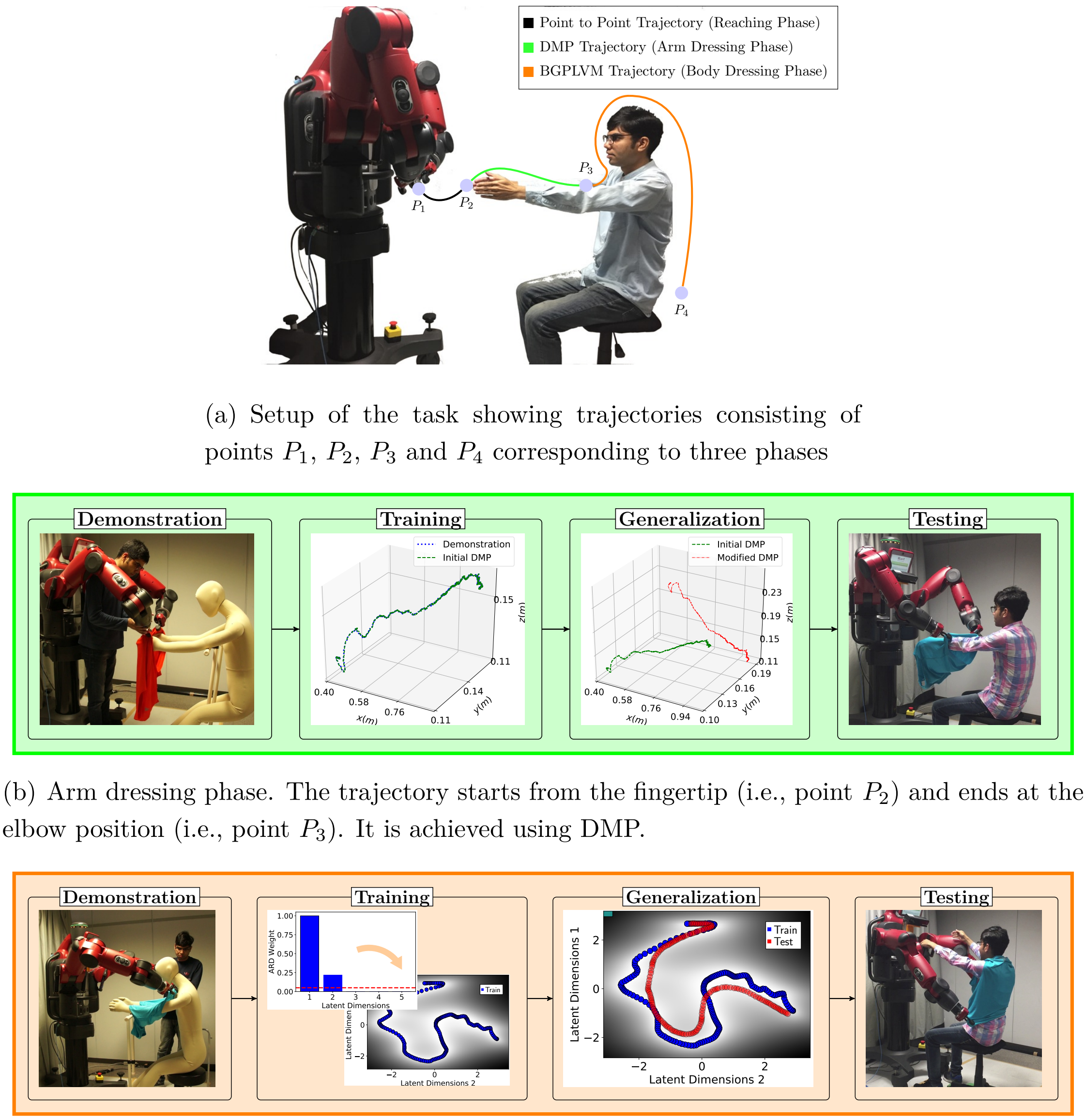
Our framework is shown in Fig. 5.2. We have divided the trajectory into the following three phases

1. Reaching phase, which refers to the trajectory that starts from the home position of the robot, i.e., P_1 and ends at the fingers of the subject, i.e., P_2.
2. Arm dressing phase, which refers to the trajectory that starts from fingers of the subject, i.e., P_2 and reaches up to the elbow, i.e., P_3.
3. Body dressing phase, which refers to the trajectory that starts from the elbow, i.e., P_3 of the subject goes over the head and reaches up to the torso of the subject, i.e., P_4.

The reaching phase is a point-to-point trajectory, performed using a simple position based controller. However, for the rest of the phases, we need an efficient controller which can learn required motor skills from the demonstration and can generalize to various people. The arm dressing phase is reasonably a trajectory planning problem, and DMP takes it to care. DMP can be generalized to various postures of the subject's arm by changing the start, and goal parameters thus enable adaptive control of the robot. More information is provided in Section 5.2.1. In body dressing phase, the robot operates close to the subject. Hence to address safe human-robot interaction, we propose to use BGPLVM wherein tight coupling between the human and the clothing item occurs. We assume that performing a dressing task requires a consistent set of motor skills. The robot trajectory can be constrained to a lower dimensional latent space using BGPLVM. Both DMP and BGPLVM being data-efficient can learn even from a single demonstration thus suitable for our task. In the subsequent sections, we are explaining two phases of dressing task and the mathematical formulation of them.

5.2.1 Arm-Dressing Phase

In this section, we incorporate DMP for putting the clothing article on the arms of a subject. As per the formulation described in Section 3.1, DMP can learn from the demonstration. Therefore we start by performing a kinesthetic demonstration



(a) Setup of the task showing trajectories consisting of points P_1, P_2, P_3 and P_4 corresponding to three phases

(b) Arm dressing phase. The trajectory starts from the fingertip (i.e., point P_2) and ends at the elbow position (i.e., point P_3). It is achieved using DMP.

(c) Body dressing phase. The trajectory starts from the elbow position (i.e., point P_3) and finishes at the torso position (i.e., point P_4). It is achieved using BGPLVM.

Figure 5.2: Overview of the framework. The complete task is divided into three phases. The first phase, i.e., reaching phase, is a point-to-point trajectory. The remaining two phases are arm dressing and body dressing phase for which four stages such as demonstration, training, generalization, and testing are defined.

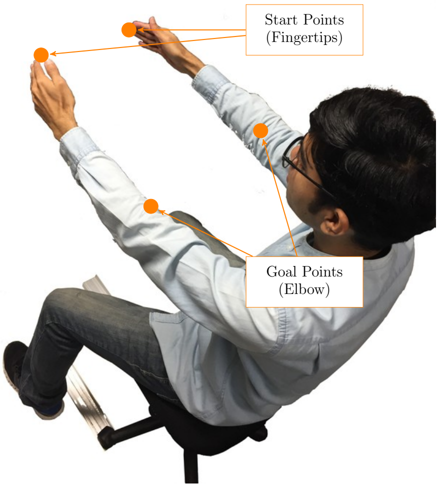


Figure 5.3: Control points for arm dressing phase showing start and goal points of DMP system colored as orange. Fingertips and elbow positions are chosen as start points and goal points respectively.

with the robot controlled in gravity compensation mode as shown in Fig. 5.2. This is referred to as "Demonstration Stage" since, in this stage, an expert provides a demonstration of the dressing task while the robot is under gravity compensation. During the demonstration, pose trajectory of end-effector is recorded using Baxter API and stored in a file. The term pose collectively refers to as position in Cartesian space $p = (p_x, p_y, p_z) \in \mathbb{R}^3$ and orientation. The orientation is defined in terms of quaternion $q = (q_x, q_y, q_z, q_w) \in \mathbb{R}^4$. Once the demonstration is finished, DMP is parameterized using the recorded trajectory file. This is termed as "Training Stage". The parameterized DMP can represent all the characteristics of the original trajectory. Here, three DMP systems, one for each coordinate axis, i.e., x, y, and z are initialized for one arm. In this way, we have totally six DMP systems, which can control both the arms of the Baxter robot. The orientation of the end-effector is not considered as a part of the DMP system and kept the same as it was at the time of "Demonstration Stage". It should be noted that the expert demonstrations were performed on a mannequin as it can be time-consuming for a human subject to sit while recording demonstrations. After training the DMP system, it can generalize to various arm postures. Now, we need to set the control points which are start and goal parameters of DMP as fingertip and elbow positions of the subject respectively shown in Fig. 5.3. The control points of DMP are retrieved by using a Kinect v2 sensor. More information is given in Section 5.1. In this way, we have modified DMP system, which can adapt modified posture referred to as “Generalization Stage”. The adaptation is verified by executing the generated trajectory during “Testing Stage”.

5.2.2 Body-Dressing Phase

In this section, we provide the details for generating the latent space using BGPLVM described in Section 3.2. BGPLVM is used to dress the body part of the subject. Similar to DMP; we start by performing a kinesthetic demonstration with the robot controlled in gravity compensation mode as shown in Fig. 5.2. We record joint angle trajectories of Baxter’s arms using Baxter API and use them for modeling BGPLVM.

5.3 Robot-Camera Calibration

Our experimental setup of Robotic Clothing Assistance contains a Kinect v2 depth sensor that is used for human pose tracking. We need to calibrate Kinect camera w.r.t. Baxter robot so that observations can be transformed into Baxter reference frame. To perform this calibration, we need to hold a marker in one arm of Baxter as shown in Fig. 5.4. In our experiment, we are using a green colored ball of radius 5 *cm*. The ball is firmly held by the end-effector of the robot through a stick. The idea behind the calibration technique is to record the position of the marker as observed from both the frames, i.e., Baxter and Kinect. We start the calibration process by recording marker position w.r.t. both frames. It should be noted that an additional transformation on Baxter’s end-effector was incorporated while recording marker position w.r.t. Baxter. Now, we moved the end-effector and recorded the position again. We recorded the data for several (≈ 50) positions and created a dataset of points. The calibration process is automatic and finished in a few (≈ 5) minutes. Next, we estimated a rigid transformation between two point sets by computing Singular-Value Decomposition (SVD) proposed by Umeyama [72].

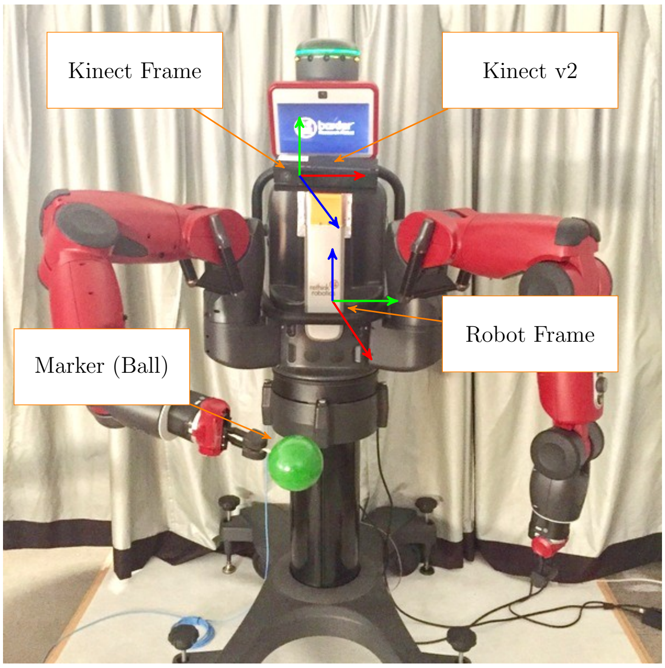


Figure 5.4: Setup for the robot-camera calibration

5.3.1 Rigid Transformation Estimation

The core of our calibration process depends on the estimation of the rigid transformation. We are using SVD based procedure proposed by Umeyama [72]. The procedure is described in Algorithm 1. It computes the rigid transformation in the form of rotation and translation. It takes two point sets x and y, which are in correspondence hence have the same dimensionality. The algorithm starts by computing the mean of point sets, which is then used to center the data. In this way, both point sets have zero mean. Next, SVD is performed over the covariance matrix defined by two point sets. The rotation $\boldsymbol{R}$ and translation $\Delta\boldsymbol{p}$ is computed accordingly as shown in the algorithm.

In order to apply rigid transformation estimation Algorithm 1, we need to define two point sets x and y in correspondence. The point set x is ball center location w.r.t. Kinect. Formally, $x \equiv c$ w.r.t. Kinect frame and $y \equiv c$ w.r.t. robot frame where $c = (c_x, c_y, c_z) \in \mathbb{R}^3$ is ball center position. In the next two sections, we are explaining the procedure to define point sets x and y.

Algorithm 1 Rigid Transformation Estimation

```
Input Two point sets $x$, $y$ in correspondence
Compute the mean
    $x_{mean} \leftarrow mean(x)$
    $y_{mean} \leftarrow mean(y)$
Center the points
    $x \leftarrow x - x_{mean}$
    $y \leftarrow y - y_{mean}$
Compute the covariance $\sigma \leftarrow y * x^{\top}$
Compute singular-value decomposition
    $\boldsymbol{U}, \boldsymbol{D}, \boldsymbol{V} \leftarrow svd(\sigma)$
$\boldsymbol{S} \leftarrow diag([1\ 1\ sign(|\boldsymbol{U}| * |\boldsymbol{V}|)])$    ▷ $diag$: diagonal matrix, $sign$: signum function
Rotation $\boldsymbol{R} \leftarrow \boldsymbol{U} * \boldsymbol{S} * \boldsymbol{V}^{\top}$
Translation $\Delta\boldsymbol{p} \leftarrow y_{mean} - \boldsymbol{R} * x_{mean}$
Output Transformation parameters $\boldsymbol{R}$, $\Delta\boldsymbol{p}$
```

Define Point Set x

We employ point cloud, acquired from Kinect sensor and perform sphere fitting on it using Random Sample Consensus (RANSAC) algorithm [73]. The depth range of a Kinect sensor is large (up to 5 *m*). Hence, it is difficult to find a sphere of radius 5 *cm* on a point cloud captured from a single view since only a portion of the ball is visible. Therefore, we propose to define a region of interest (ROI) for the sphere as shown in Fig. 5.5. We compose an RGB image from the point cloud and use HSV color space [74] for image segmentation. We obtain 2D points from the segmented region, convert them into 3D points by utilizing the captured point cloud and define an ROI in 3D space. The point cloud is then cropped as per defined ROI, and we apply sphere fitting using RANSAC. In order to make the entire procedure robust, as a validation step, we compare color values of the segmented sphere and accept only if it is within range. We have used PCL [75] for point cloud processing and Kinect drivers [76, 77] for capturing point cloud data.

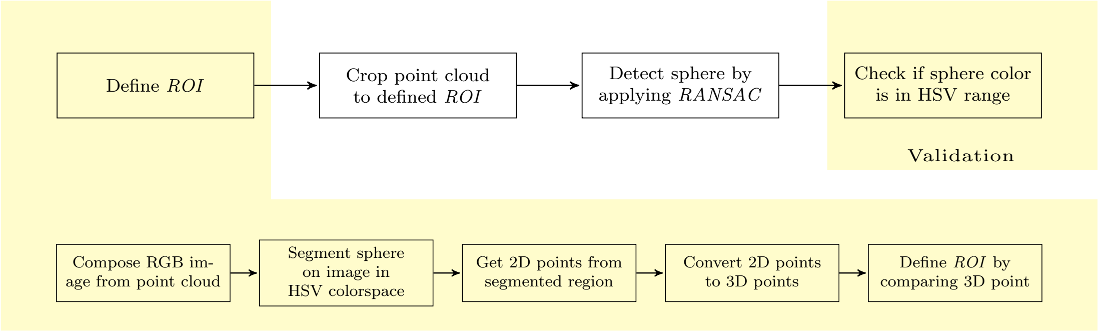

Define *ROI*
Crop point cloud to defined *ROI*
Detect sphere by applying *RANSAC*
Check if sphere color is in HSV range
Validation
Compose RGB image from point cloud
Segment sphere on image in HSV colorspace
Get 2D points from segmented region
Convert 2D points to 3D points
Define *ROI* by comparing 3D point

Figure 5.5: Procedure to define point set x

Define Point Set y

In order to define point set y, we rely on the robot end-effector pose data. Since, the ball is attached to the end-effector through a stick, the ball center position is not the same as the end-effector position. The location of the ball frame and end-effector frame is depicted in Fig. 5.6.

We are interested to know the ball center location w.r.t. robot base frame, which can easily be calculated by multiplying the transformation of end-effector w.r.t. robot base with the transformation of ball center w.r.t. end-effector. Formally, it can be written as follows-

$$\boldsymbol{T}_c^b = \boldsymbol{T}_e^b * \boldsymbol{T}_c^e \tag{5.1}$$

where $\boldsymbol{T}_c^b$ is the transformation of ball center w.r.t. robot base, $\boldsymbol{T}_e^b$ is the transformation of end-effector w.r.t. robot base and $\boldsymbol{T}_c^e$ is the transformation of ball center w.r.t. end-effector. The term $\boldsymbol{T}_e^b$ is composed by acquiring end-effector pose from the robot and here, we completely rely on the forward-kinematics of the robot. The term $\boldsymbol{T}_c^e$ can be written as follows-

$$\boldsymbol{T}_c^e = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

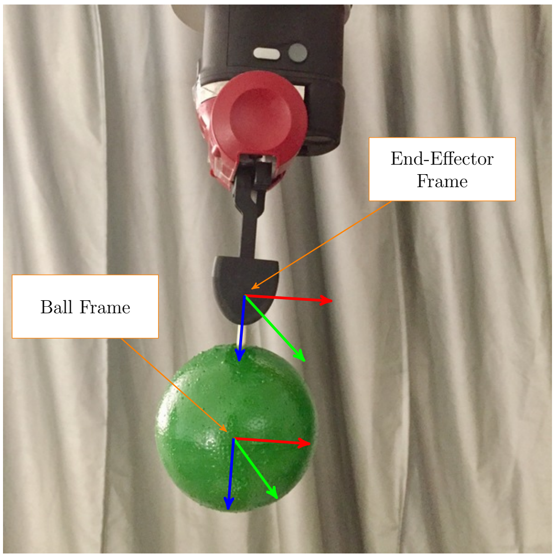


Figure 5.6: Location of the end-effector frame and ball frame. The origin of the ball frame is situated at the center of the ball.

where d is the distance of ball center from end-effector frame. Note that $\boldsymbol{T}_c^e$ does not change during the experiment hence can be pre-computed. In our experiment the value of d is 8.43 cm.

5.3.2 Calibration Procedure

The flowchart of Baxter-Kinect calibration is shown in Fig. 5.7. The process starts by attaching a marker to the end-effector of the robot. We are using a green color ball of radius 5 cm. We then record the marker position w.r.t. both the frames, i.e., Baxter and Kinect frame. While recording the marker position, we move the end-effector until enough data is collected. Once we have obtained point sets x and y, a rigid transformation is estimated based on Algorithm 1. The transformation consists of a rotation matrix $\boldsymbol{R}$ and translation vector t, which is serialized to a file for later use. In our experiment, we are using ROS [58] for robot and camera control. The serialized data is published via ROS publisher.

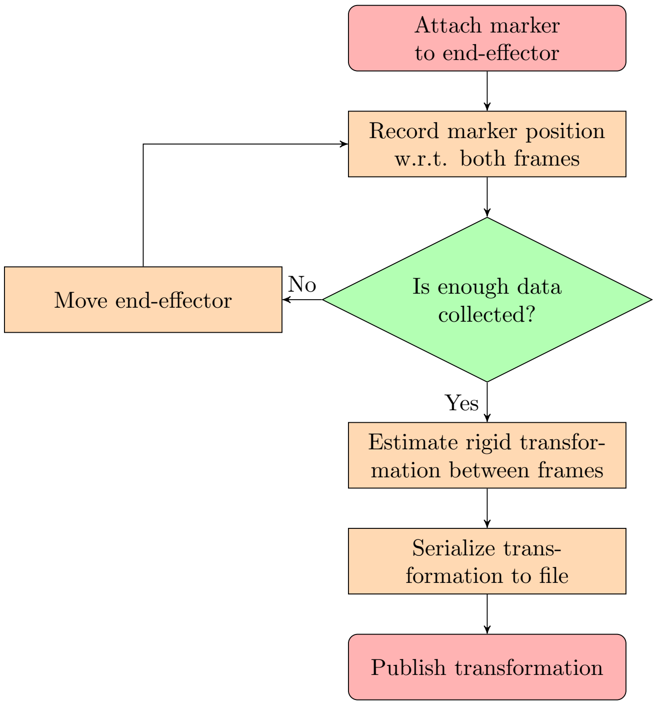

Attach marker to end-effector
Record marker position w.r.t. both frames
Is enough data collected?
No
Move end-effector
Yes
Estimate rigid transformation between frames
Serialize transformation to file
Publish transformation

Figure 5.7: Flowchart of the calibration. The last two steps are related to ROS framework.

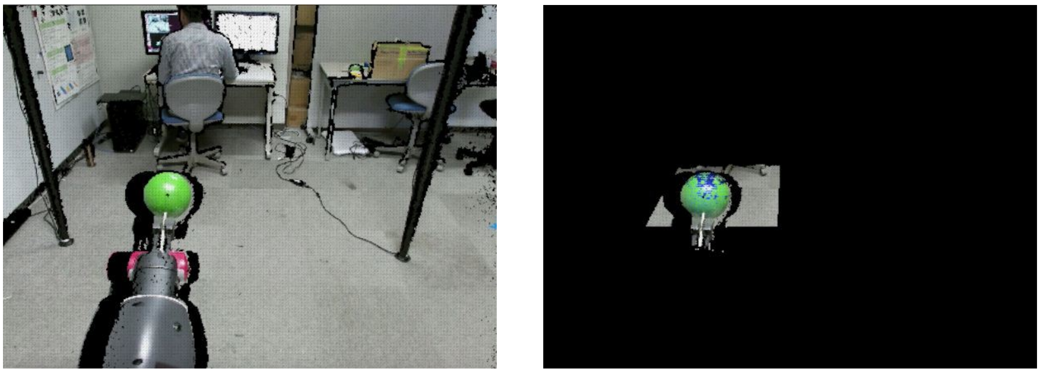


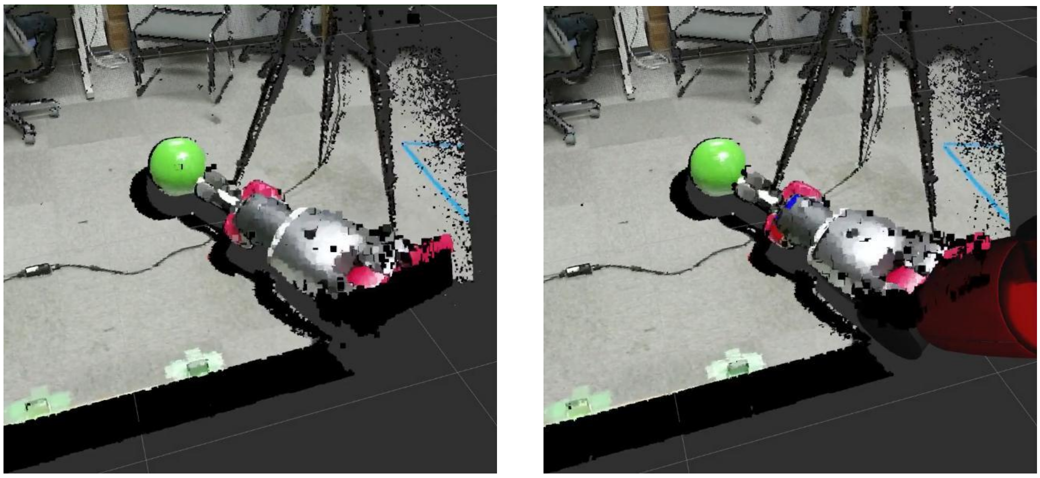
Figure 5.8: Raw (left) and segmented (right) point cloud. The detected sphere is colored as blue. (The blue region is appearing smaller than the ball due to the visualization in 3D.)

5.3.3 Calibration Results

We performed single Kinect Baxter calibration. The raw point cloud is acquired, and sphere fitting is applied over a defined ROI. The detected sphere is highlighted using blue color as shown in Fig. 5.8. Note that the blue region is appearing smaller than the ball due to the visualization in 3D.

During the calibration process, we moved the end-effector and recorded many points. More precisely, we randomly defined ten positions in 3D space and at each position at max five points were recorded. Hence, in total, we have approximately fifty points. This process took five minutes approximately since robot needs to move to each point slowly while reducing the vibrations to avoid shaking of the attached Kinect camera and the marker, i.e., ball. After computing the calibration, as described in Section 5.3.2, we transformed Kinect frame into Baxter frame. We visualized both frames after calibration inside RViz. RViz is a 3D visualizer for the ROS framework [78]. The real-time data acquired from Kinect data was superimposed on the Baxter model as shown in Fig. 5.9.

The result of our calibration process is shown in Fig. 5.10. As an outcome, we can see that Kinect points (shown in red color) are now close to Baxter points (shown in blue color). Quantitatively, to define the quality of our calibration, we further investigated and computed the calibration error at each point. The mean calibration error is also computed and found 4.54 *mm* as shown in Fig. 5.11.



(a) Kinect frame

(b) Kinect and Baxter frames

Figure 5.9: Visualization of both frames after calibration inside RViz. The left side image shows real-time data acquired from Kinect and right side image shows Kinect data superimposed on the Baxter model.

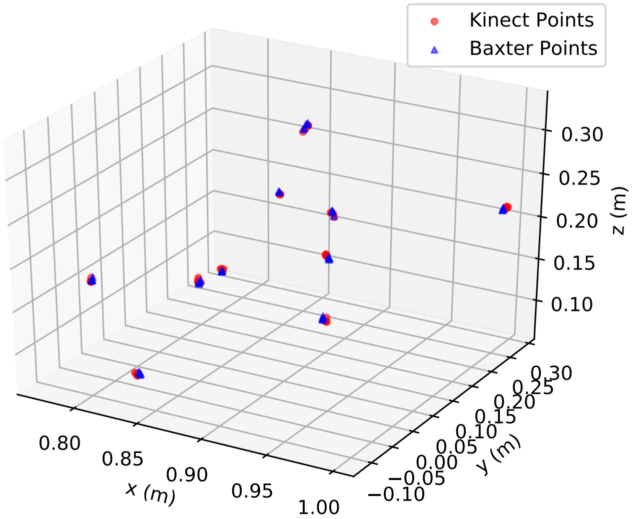


Figure 5.10: Result of the calibration procedure. Kinect frame is transformed into Baxter frame.

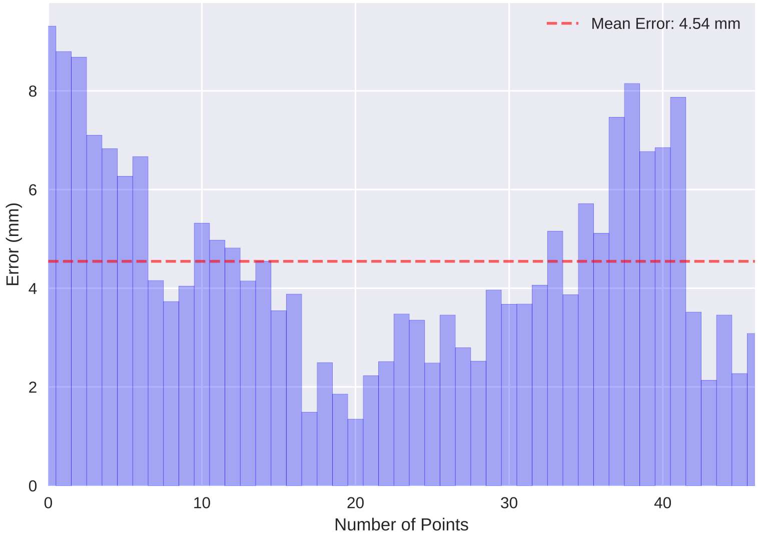


Figure 5.11: Calibration error. Showing error for each point and highlighted mean error using the dotted red line.

5.3.4 Sample Efficiency of Calibration Procedure

In this section, we measure the efficiency of our calibration method w.r.t. the number of samples. This analysis is helpful to investigate the number of samples required for robust calibration. The sample efficiency is shown in Fig. 5.12. We have taken five trials for the mean (μ) and standard deviation (σ) calculation. As the number of samples is growing, the calibration error is converging; however, the variance is reducing intuitively which makes the calibration robust and more suitable.

5.3.5 Discussion

Our calibration process uses point cloud data and applies sphere fitting on it. The maximum depth value for Kinect v2 sensor is 5 m, and the radius of the sphere is only 5 cm. Hence we defined an ROI since it is difficult to segment sphere from a single view, i.e., front view, capturing only a portion of the ball. The error in calibration can be explained further. Following are the major source of errors-

- We rely on the internal forward-kinematics of Baxter robot to compose

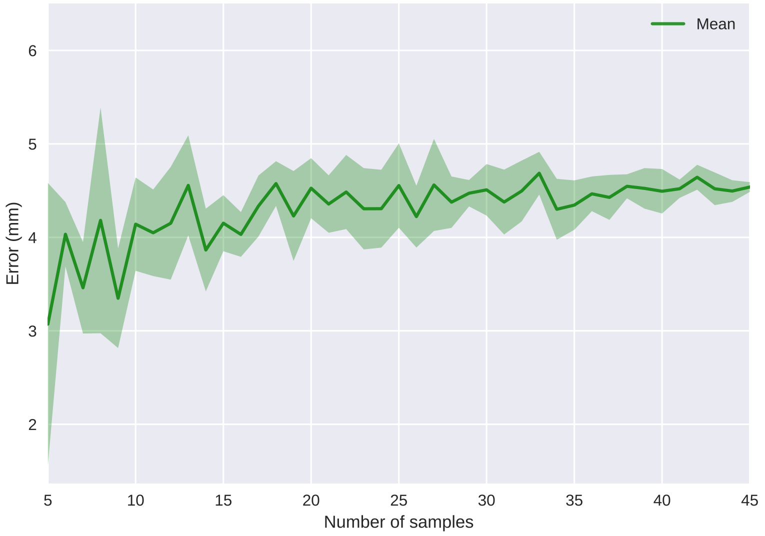


Figure 5.12: Sample efficiency of the calibration method. The dark green line is representing the mean error of calibration. The region $\mu \pm \sigma$ along mean error is filled with light green color. The mean calibration error grows and converges. At the same time, the variance is reduced hence makes the calibration robust and more suitable.

the term $\boldsymbol{T}_e^b$. Originally, Baxter is a compliant robot, and it suffers from poor accuracy and repeatability due to its design. The reported accuracy and repeatability is as high as +/- 5 *mm* and 3.3 *mm* respectively[79, 80]. Indirectly, it generates an error in $\boldsymbol{T}_e^b$. It is advised to perform Baxter arm calibration beforehand [81] to reduce this error up to some extent.

- The term $\boldsymbol{T}_c^e$ assumes only translation without any rotation. However, it may be possible to have a minor error in measuring translation since it is measured manually. Furthermore, a small amount of rotation may exist since the ball is mounted with a stick to the end-effector manually.

- The point cloud data generated by a Kinect sensor is noisy in nature. The noise is significant in case of a reflective surface. In our experiment, we noticed that the reflective marker is very difficult to recognize due to the noise in depth values converting spherical surface to look like a mountain. In Fig. 5.13, we have shown how a reflective surface can corrupt the point cloud. This can be seen easily in an isometric view by rotating the view 90 degrees.

5.4 Hand Location Estimation

The Robotic Clothing Assistance is comprised of several subtasks as mentioned above. This section focuses on the initialization task in which is perfromed by employing DMP. For adaptive control, we set the start and goal parameters of DMP automatically by real-time estimation of the 3D hand location by a template matching algorithm.

Our solution to estimate DMP parameters leverage on three dimensional (3D) data from the depth sensor for extracting the hand of the mannequin at different poses. For the 3D visual information, a Kinect depth sensor is mounted on the top of the Baxter Robot as shown in Fig. 5.14. The mannequin is placed beforehand in front of the sensor, so that the sensor can clearly see its hands. Point cloud data (PCD) is used and analyzed for detecting the hand location. Point clouds contain the xyz coordinate data of each point in reference to the viewpoint of the sensor. In template matching technique, we are matching previously recorded

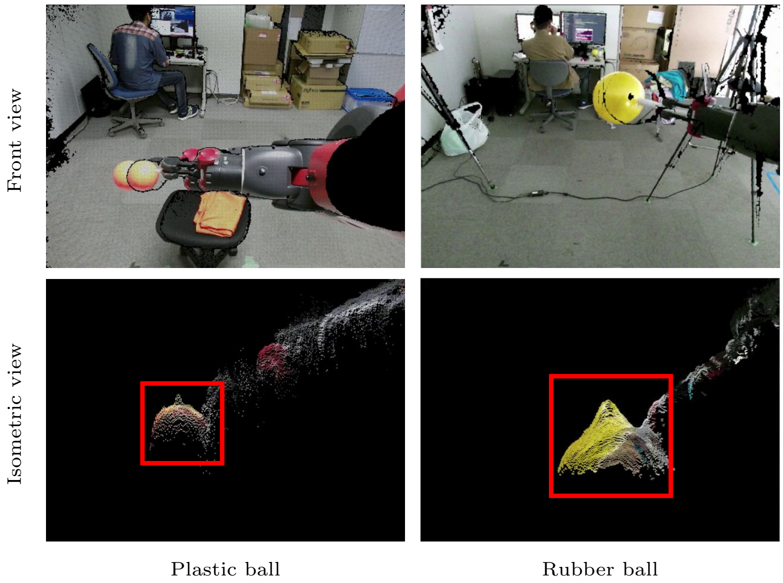


Figure 5.13: Noise on point cloud due to the reflective surface. The front view looks perfect however the noise can be seen in an isometric view. The spherical surface is converted due to the noise and looking like a mountain.

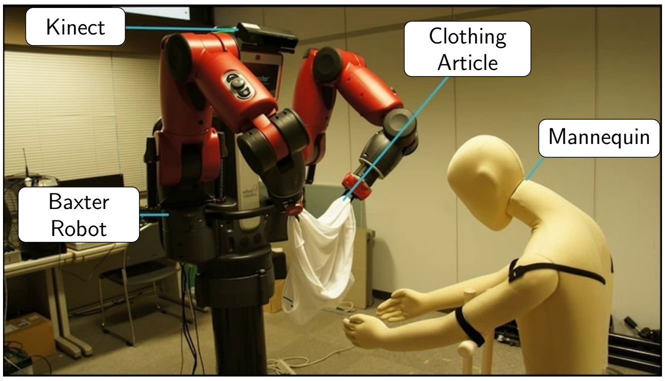


Figure 5.14: Setup of the task for estimating hand location in 3D

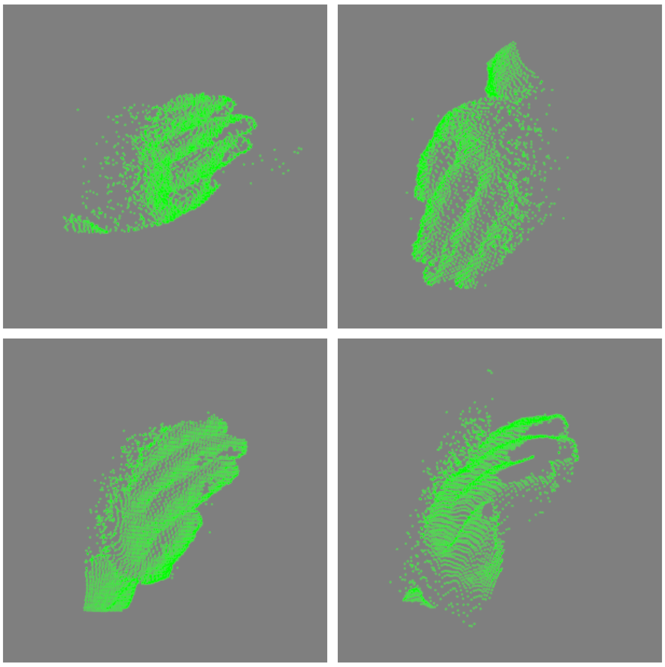


Figure 5.15: Input templates (PCD) of wrist

object templates with the new data and find the position of the object in new data [82].

By using the information of point cloud, we recorded dataset of the mannequin and get point clouds of the hand at several orientations manually. These PCD on each hand are saved and set as input templates to the extraction algorithm. For our experiment we used 5 different hand templates for each left and right arm which created from five different pose of the hand. Template samples can be seen in Fig. 5.15. The technique takes these templates one by one and patches each iteration to the target point cloud, search through the whole data and find where the templates match. Then, we stored the matching results so that later on will be used to locate the coordinate position of the hand within the matched region.

The technique starts by setting a given template as the source cloud of Sample Consensus Initial Alignment (SAC-IA) algorithm, and then aligning these input templates to the target [82]. SAC-IA is an implementation of matching various overlapping 3D point cloud data views into a complete model, also known as 3D registration [83, 82]. During the alignment of each input template, fitness scores

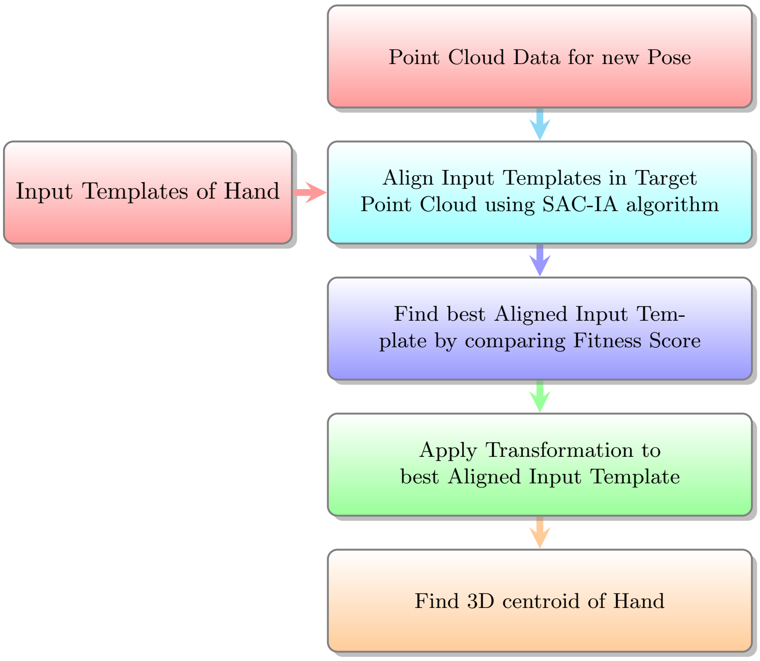

Point Cloud Data for new Pose
Input Templates of Hand
Align Input Templates in Target Point Cloud using SAC-IA algorithm
Find best Aligned Input Template by comparing Fitness Score
Apply Transformation to best Aligned Input Template
Find 3D centroid of Hand

Figure 5.16: Flowchart of the task

are produced on each template and the best one is selected. After finding the best matched template, the method also calculates rotational, translational matrices, including the region's centroid. The complete framework is shown in Fig. 5.16.

5.4.1 Results

The posture of mannequin was modified by changing the shoulder elevation. The template matching algorithm was applied for the new 3D point cloud. We experimented this method for six different hand positions by changing the angle of inclination of the hand w.r.t to horizontal line in 2D space. The template matching method was able to locate the hand smoothly in every posture and able to represent its position with extracted 3D centroid coordinates as shown in Fig. 5.17.

The template matching algorithm highly depends on the quality of input template. The templates were designed from the point cloud data captured from Kinect sensor. The position of Kinect was fixed hence the generated template

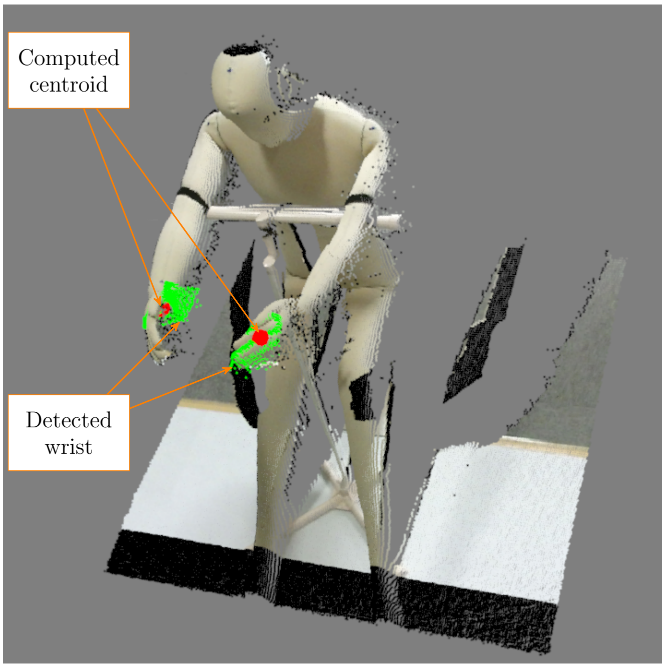


Figure 5.17: Result of the hand location estimation

files don't have complete 360 degree view of the wrist. The point cloud data recorded from Kinect also contains noise. Not to mention that there were also failure scenario because of occlusions in target point cloud file. In such cases the detection was failed as shown in Fig. 5.18.

5.5 Evaluation

5.5.1 Arm-Dressing Phase

The arm dressing phase was accomplished by the DMP system. Initial DMP was modified to accommodate a new posture by changing start and goal parameters. The generated trajectory from the modified DMP system was then run on Baxter robot as shown in Fig. 5.19. The initial DMP trajectory (shown in green color) is parametrized and able to represent all characteristics of the demonstration trajectory (shown in blue color). The modified DMP trajectory (shown in red color) was found well suited as it is capable of performing the task. Since the

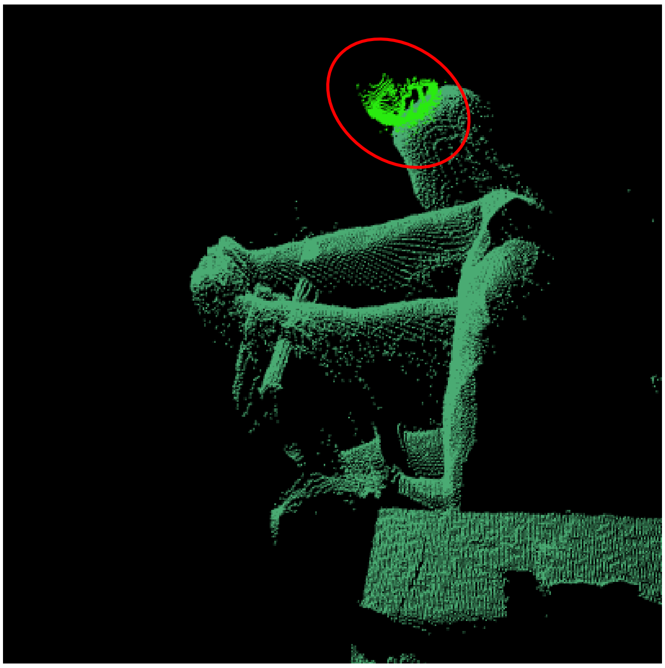


Figure 5.18: Failure scenario in wrist detection

DMP trained on a mannequin and tested on human subjects, we can notice the difference in control points of DMPs.

To further investigate our DMP system, we performed arm dressing task on ten subjects. For each subject, we repeated the experiment ten times. During the experiment, the end-effector trajectory generated by the DMP system is recorded. We have plotted the mean and along with the standard deviation for all subjects as shown in Fig. 5.20. Three sub-plots are showing x, y and z coordinates of end-effector w.r.t. time. The time is normalized to [0, 1] range. The largest range of coordinates is noticed across x-axis which reflect the length of the human's arm starting from the fingertip up to elbow. The y- and z-axis corresponds to the bending and height of the human's arm respectively. The different height of the human's arm is induced from the fact that many (ten to be precise) subjects having different body dimensions were involved in the evaluation of the experiment. The trajectory starts from the fingertip of the subject and reaches up to the elbow of the subject. The standard deviation is higher at the beginning, and it decreases as time passes. The beginning and the ending represent the fingertip

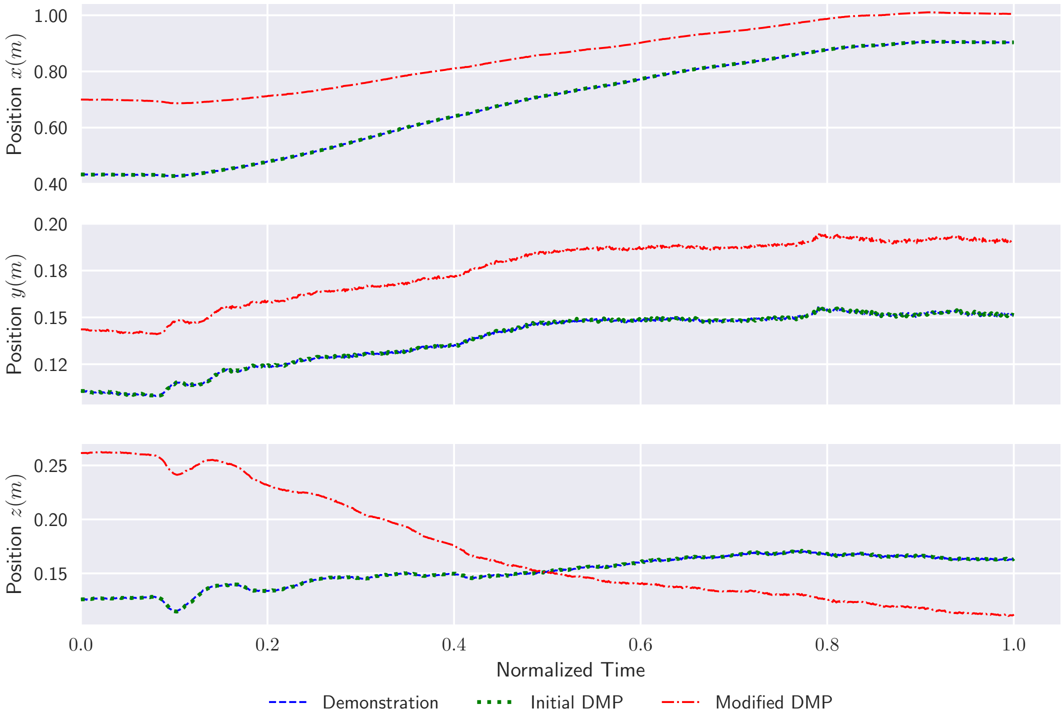


Figure 5.19: DMP trajectory corresponding to the left-arm of Baxter while performing arm dressing on a subject. The modified DMP is acquired by changing control points of initial DMP, which is modeled by parameterizing demonstration trajectory using DMP. The time is normalized to [0, 1] range.

and the elbow position of the subject respectively. For every subject at each trial, the fingertip position changes substantially higher compared to the elbow position. Hence the standard deviation is high at the beginning.

5.5.2 Body-Dressing Phase

Due to the strong human-cloth coupling during the body dressing phase, BGPLVM was utilized. It was implemented using GPy python library [84]. The input variable $\boldsymbol{X}$ was initialized using PCA from demonstration data as the first step for GPy library. ARD kernel and 100 inducing input points were supplied to BGPLVM model. The training of the model was done in two steps. Firstly, the signal-to-noise ratio (SNR) which was fixed to 10, was used to constrain the variance of model parameters. In this configuration, the model was optimized for 20 iterations. Secondly, we trained the model without any constraints and optimized for 500 iterations. Generated latent space with the relevance of each latent dimension is shown in Fig. 5.21(a). The model can find two latent dimensions which are having the maximum contribution in defining the data.

We further exploit BGPLVM model by sampling trajectory from latent space. This trajectory was executed on Baxter robot 100 times, i.e., ten times on ten subjects. During the experiment, forces acting on the end-effector were recorded then plotted as shown in Fig. 5.22. The time is normalized to [0, 1] range. For each subject, the mean force is plotted with a dark color and the region $\mu \pm \sigma$ along mean force is filled with a light color. End-effector forces are noisy; however, they all are following a similar trend. The clothing article is already in the arms of the subject in this phase, i.e., body dressing phase. During this phase, the clothing article is dressed on the body, and it reaches up to the torso of the subject. In between, the clothing article undergoes severe deformations; especially it is stretched out to expand while passing by the head and neck of the subject, which increases (in -ve direction) end-effector forces considerably on the y-axis of the base (robot) frame during the time $t \approx 0.2$. The increase (in -ve direction) in force is highlighted on F_y in Fig. 5.22. Once the clothing article reaches near to the chest of the subject, the robot exerts forces so that the clothing article reaches up to the torso of the subject, which can be noticed on the vertical axis. i.e., z-axis of the base (robot) frame during the time $t \approx 1$. It signifies that BGPLVM model is

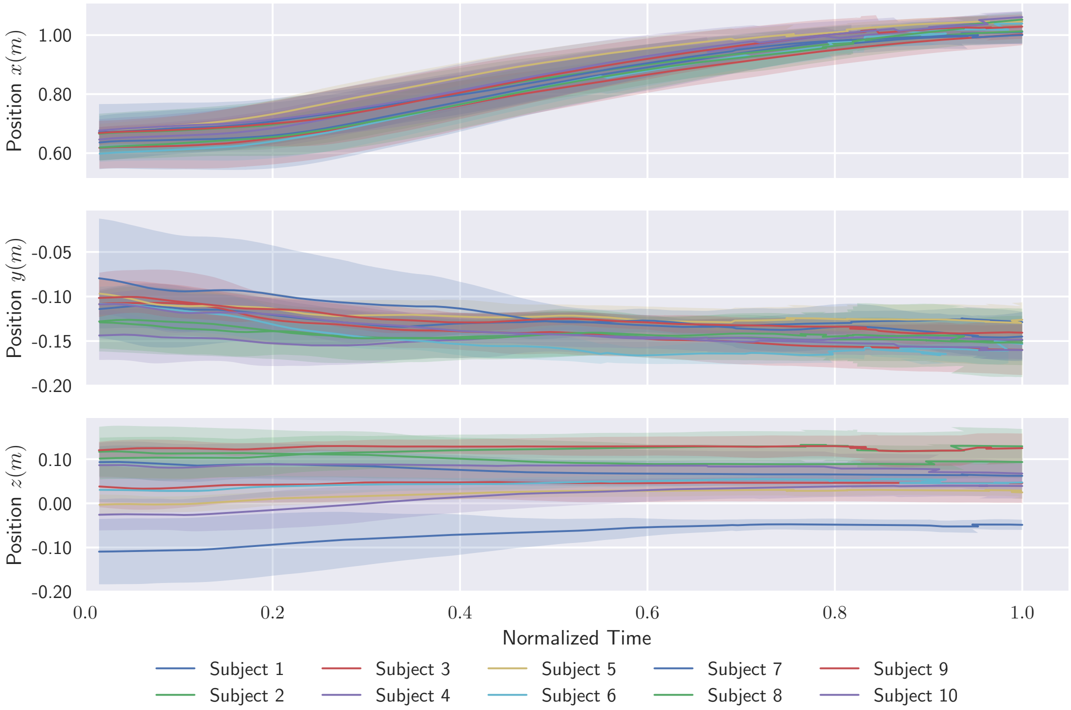
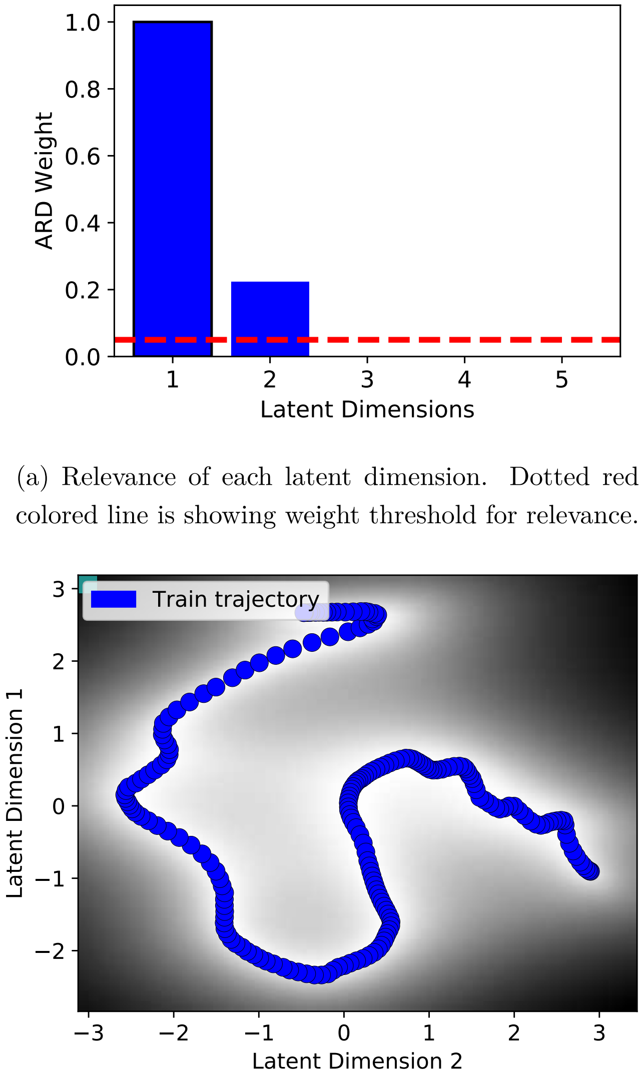


Figure 5.20: Right-arm trajectory of Baxter while performing arm dressing on multiple subjects. The time is normalized to [0, 1] range. For each subject, the mean position is plotted with a dark color and the region $\mu \pm \sigma$ along mean position is filled with a light color. The x, y and z-axis corresponds to the length (from fingertip up to elbow), bending and height of the human's arm respectively.



(a) Relevance of each latent dimension. Dotted red colored line is showing weight threshold for relevance.

(b) Two-dimensional latent space generated by BGPLVM

Figure 5.21: BGPLVM Model

capable of performing the task efficiently. The motivation behind presenting end-effector forces is the following. End-effector forces are a measure of the amount of discomfort felt by the subject. Lower the force applied better for the subject. Additionally, these forces are intuitively representing various stages during the body dressing phase. For example, F_y and F_z show expending the cloth and pulling down the cloth respectively as mentioned above already.

To show what the latent space encoded, we explored the latent space by choosing four points on latent space. At each point, we sampled and inferred the configuration of the robot as shown in Fig. 5.23. Blue colored trajectory shows the training trajectory whereas sampled test trajectory is shown in red color. During the exploration of latent space, the robot along with its end-effector trajectories was visualized. The end-effector trajectories are shown in yellow color. The exploration confirms that the generated two-dimensional latent space is restricted to perform dressing tasks only and hence provides safe human-robot interaction, which is also shown in Fig. 5.22. This figure plots forces acting on the end-effector recorded during the body dressing phase. End-effector forces are noisy; however, they all are following a similar trend, which indicates that our controller with the BGPLVM model achieved safe human-robot interaction.

5.5.3 Complete Robotic Clothing Assistance

We performed the dressing experiment on ten healthy people and observed the task. If the robot is able to dress sleeveless T-shirt (Fig. 4.8(b)) such that the T-shirt reaches up to the torso, we labeled it as a successful dressing; otherwise, it is treated as failure dressing. Sometimes failure occurred due to the clothing article get slipped from the fingers while pulling it. The overall results are shown in Table 5.1. Overall, 93% of the trials of the dressing task were found successful. The complete dressing task took 45 seconds approximately. The dressing task at various timestamps is shown in Fig. 5.24. After finishing the task, the robot arm is moved to its home position. This movement time is also included in dressing time shown in Table 5.1. The dressing time varies due to the different body dimensions of each subject. Intuitively, for thin subjects such as subject no. 3, the robot exerts less force while pulling the sleeveless T-shirt and hence dressing is quicker than others.

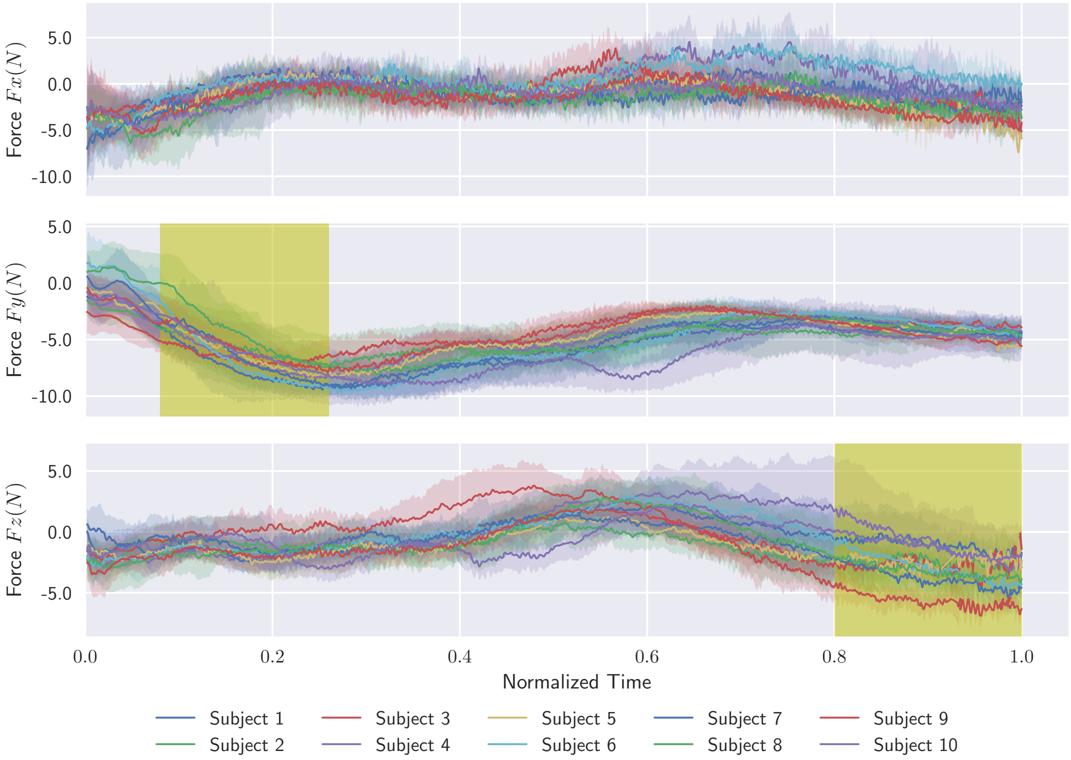
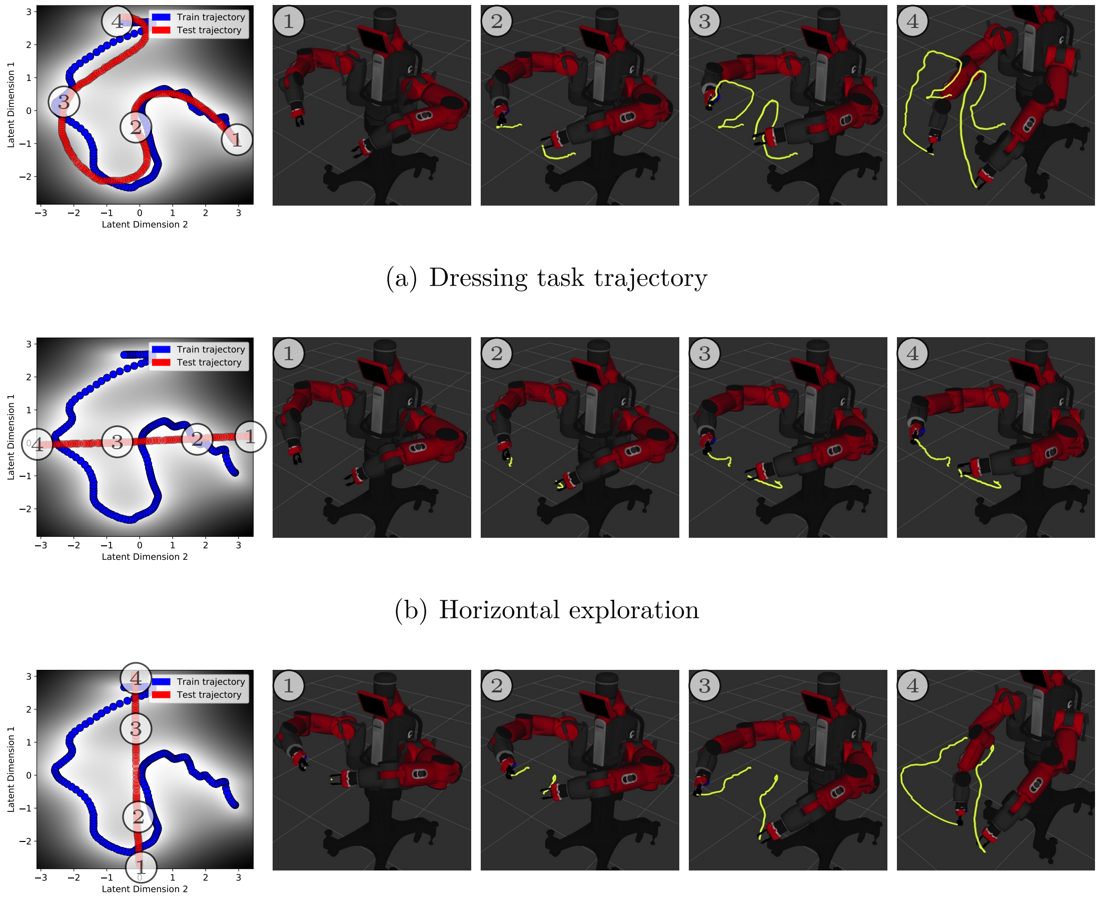


Figure 5.22: Forces acting on the right arm of Baxter while performing body dressing on multiple subjects. The time is normalized to [0, 1] range. For each subject, the mean force is plotted with a dark color and the region $\mu \pm \sigma$ along mean force is filled with a light color. The highlighted region in F_y shows increasing (in -ve direction) force while stretching out the clothing article to expand so that it can pass by the head and neck of the subject. On the other hand, the highlighted region in F_z shows increasing (in -ve direction) force while pulling down the clothing article to reach up to the torso of the subject.



(a) Dressing task trajectory

(b) Horizontal exploration

(c) Vertical exploration

Figure 5.23: Latent space exploration, showing robot configurations corresponding to four points on latent space inside a visualizer. Latent space contains the training trajectory (blue) and the exploration trajectory (red). The robot trajectories are shown inside the visualizer (yellow).

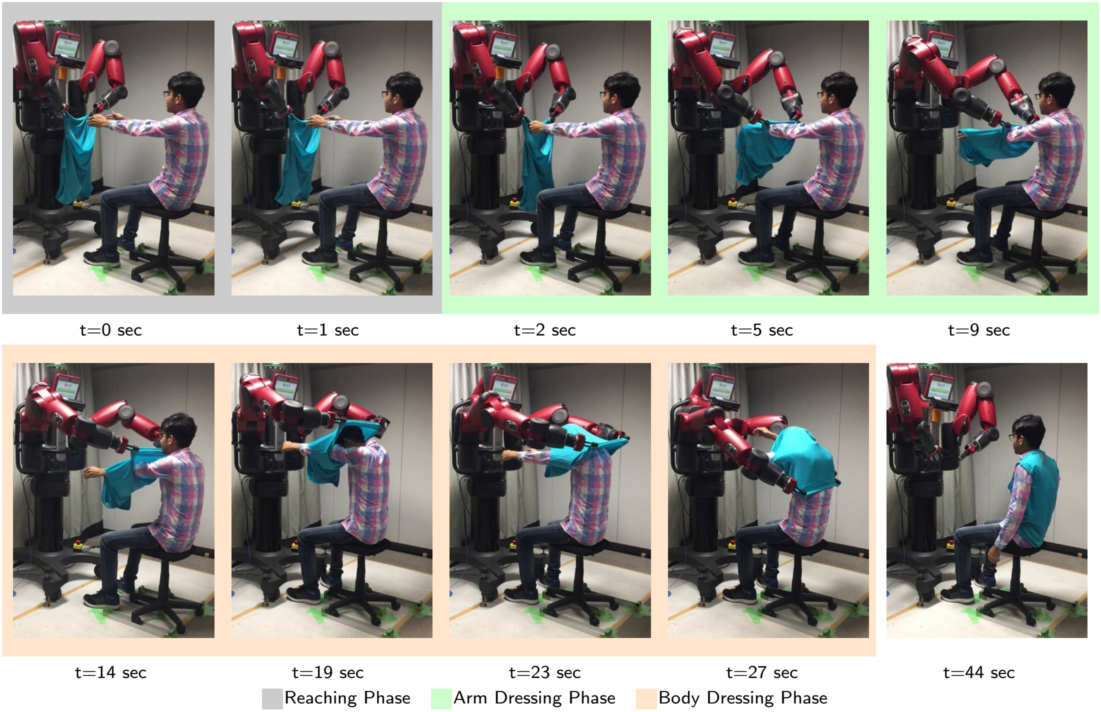


Figure 5.24: Robotic Clothing Assistance task shown at various timestamps. The clothing article, i.e., a sleeveless T-shirt which was initially held by the robot, is shown fully dressed on a human subject.

Table 5.1: Results of complete Robotic Clothing Assistance

Subject No.	Age	Height (cm)	Dressing			Dressing Time (sec)	
			Trials	Successful	Failed	Average	Std Dev
1	30	166	10	9	1	47	5
2	29	173	10	10	0	46	4
3	27	178	10	10	0	40	4
4	29	174	10	10	0	48	2
5	27	163	10	9	1	41	4
6	25	166	10	10	0	42	3
7	25	177	10	8	2	52	4
8	24	175	10	9	1	43	3
9	24	175	10	9	1	44	3
10	24	180	10	9	1	51	3
Total			100	93	7		

6 Whole-Body Robotic Simulator

We have developed a clothing assistance robot using dual arms and conducted many successful demonstrations with healthy people. It was, however, impossible to systematically evaluate its performance with a human subject because the posture of human arms can vary and also are invisible in a cloth during dressing. To address this problem, we propose to use another humanoid robot, Whole-Body Robotic Simulator of the Elderly [85, 86] that can simulate the pose and motion of the elderly persons during the dressing task. This robotic simulator is also referred as the robotic subject in this thesis. The trajectory of Baxter arms required for the dressing task is generated by using DMP. To adapt to perturbations generated by the subject's arm, we need to modify control points of DMP on the fly. In other words, we need to track control points which are the fingertips and elbows of the robotic subject. A potential solution to perform this tracking is by employing optical markers. These markers can be attached on desired points and can be tracked by optical cameras. However, this approach has a severe drawback. During the dressing task, due to the occlusion from clothing article and robot arms, marker tracking fails miserably. In this chapter, we apply forward kinematics on the robotic subject to determine the control points. Since the subject is a robotic mannequin, we are blessed with its capabilities. We can acquire all joints angles necessary for performing forward kinematics.

6.1 Setup of the Simulator

In robotic devices of non-wearable transfer aids and toileting aids, their underlying mechanism, movement, mechanistic performance such as safety and usability

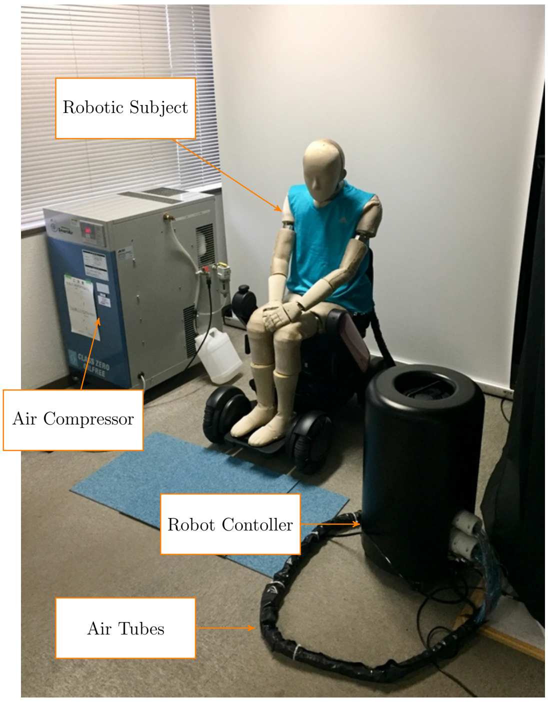


Figure 6.1: Setup of the robotic subject showing all the components of the system

were evaluated using the robotic simulator. In the monitoring system, the robotic simulator realized several poses to evaluate the detection performance of the device. Therefore, the detail specifications of the robotic simulator were decided based on these purposes of use.

The parameters of the robotic simulator are shown in Table 6.1. The shape, weight, and length of each body segment was determined to simulate the body of an elderly male in his 60s based on "Japanese Body Dimension Data" [87]. The developed robotic simulator and the position of the degree of freedom are shown in Fig. 6.2. The robotic simulator has 28 passive and 22 active joints that are controlled based on positional control. The actuator of an active joint is an air actuator, controlled using an air compressor (about 8 atmospheres) and valve units (22 channels), that are set outside the robotic simulator.

To contact the complex surface for nursing care, the robotic simulator is covered with a soft material. The distribution pressure patterns of the robotic simulator, human, and crash test dummy on a bed were measured.

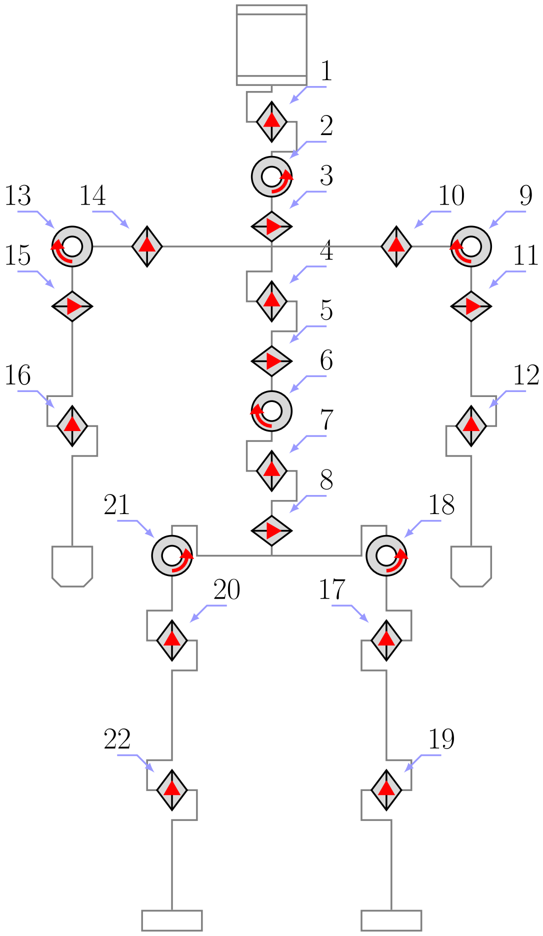


Figure 6.2: The skeleton model of robotic subject. It contains active and passive joints. However, in this figure, only active joints are shown. All the joints are revolute joints.

Table 6.1: Parameters of the robotic simulator

Length (mm)	1650
Weight (Kg)	50
Number of active joints	22
Number of positive joints	28

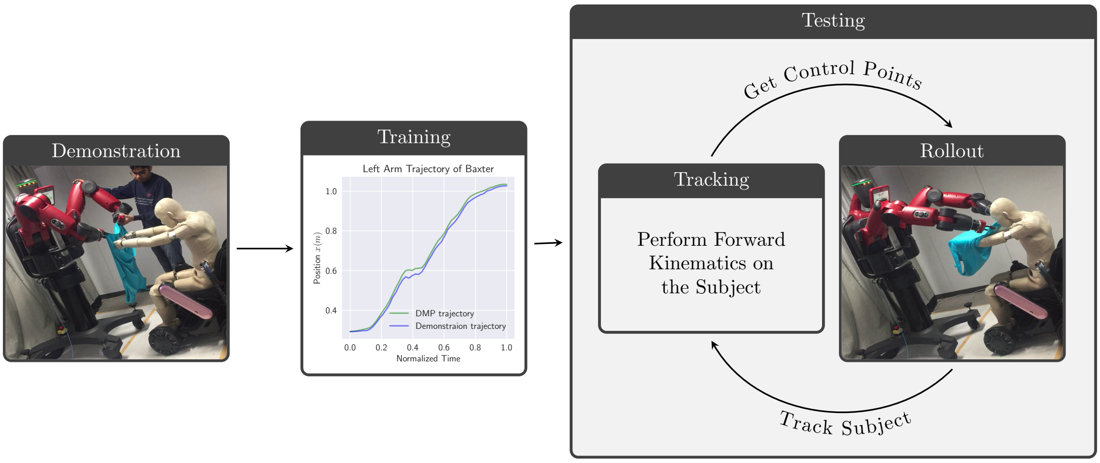


Figure 6.3: The framework for our method for evaluation. It consists of three stages named as “Demonstration”, “Training” and “Testing.”

6.2 Method

In this section, we are explaining our method used for the evaluation of clothing assistance. The framework of our method is shown in Fig. 6.3.

Our method contains three stages named as “Demonstration,” “Training” and “Testing.” As per the formulation described in Section 3.1, DMP can learn from the demonstration. Therefore we start by performing a kinesthetic demonstration with the robot controlled in gravity compensation mode, which is referred to as “Demonstration Stage” since, in this stage, an expert provides a demonstration of the dressing task while the robot is under gravity compensation. During the demonstration, the pose trajectory of end-effector is recorded using Baxter API and stored in a file. The term “pose” collectively refers to position in Cartesian space $p = (p_x, p_y, p_z) \in \mathbb{R}^3$ and orientation. The orientation is defined in terms of quaternion $q = (q_x, q_y, q_z, q_w) \in \mathbb{R}^4$. Once the demonstration is finished, the recorded trajectory is parameterized using DMP. This is termed as “Training Stage.” The parameterized DMP can represent all the characteristics of the original trajectory. Here, three DMP systems, one for each coordinate axis, i.e., x, y, and z are initialized for one arm. In this way, we have a totally six DMP systems, which can control both the arms of the Baxter robot. The orientation of the end-effector is not considered as a part of the DMP system and kept the same as it was at the time of “Demonstration Stage.” Now, we need to set the control points which are start and goal parameters of DMP as fingertip and elbow positions of the subject respectively shown in Fig. 6.4. The control points of DMP are retrieved by applying rigid body forward kinematics on the robotic subject. Joint angles of the robotic subject are retrieved and then used to calculate the position of control points in a Cartesian coordinate system. These coordinates derived by applying forward kinematics are referenced in the robotic subject frame. However, the Baxter robot has a different frame of reference. Hence, a coordinate calibration is done to transform the robotic subject frame into the Baxter robot frame. We prepared two experimental conditions/movement trajectories for the robotic subject as shown in Fig. 6.5. During the “Testing Stage,” these trajectories are applied to the robotic subject. At every timestamp, the control points are calculated and then set as the current start and goal parameters of DMP. Therefore, tracking of control points and rolling of DMPs are done at every timestamp. In this way, we have a DMP system, which can adapt accordingly while the arms of the robotic subject are in motion. To verify the adaptation, trajectories of Baxter and the robotic subject are recorded which are then analyzed in Section 6.3.

6.2.1 Experimental Setup

The experimental setup contains a compliant dual-arm humanoid robot Baxter. Each arm of the Baxter robot has 7 degrees of freedom (DOF). The setup of our system is shown in Fig. 6.6. We are also using an in-house developed whole-body robotic system to simulate the pose and motion of the elderly person during the dressing task. The robotic system is treated here as a subject for the evaluation of clothing assistance. We have used two finger electric gripper provided by Baxter. We designed soft fingertips that were plugged into these fingers tightly. These soft fingertips allow firm gripping by providing sufficient traction to hold the cloth. These soft fingertips are necessary for firm gripping of flexible clothing articles hence provides better cloth manipulation. These fingertips hold the clothing article. The cloth is put in the arms of the Baxter robot manually by a human assistant.

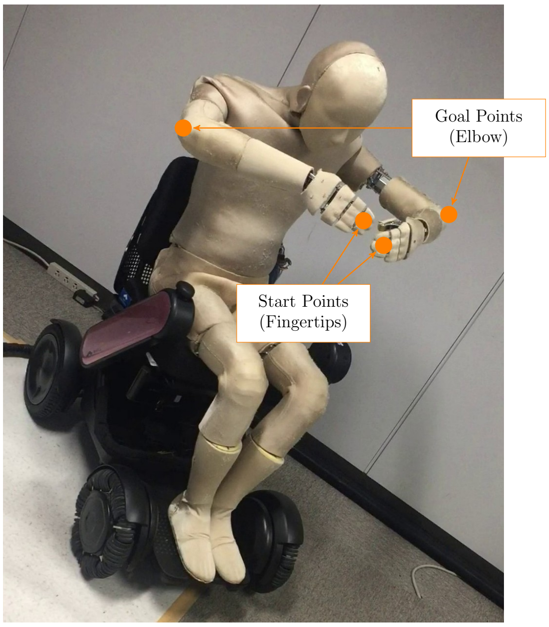
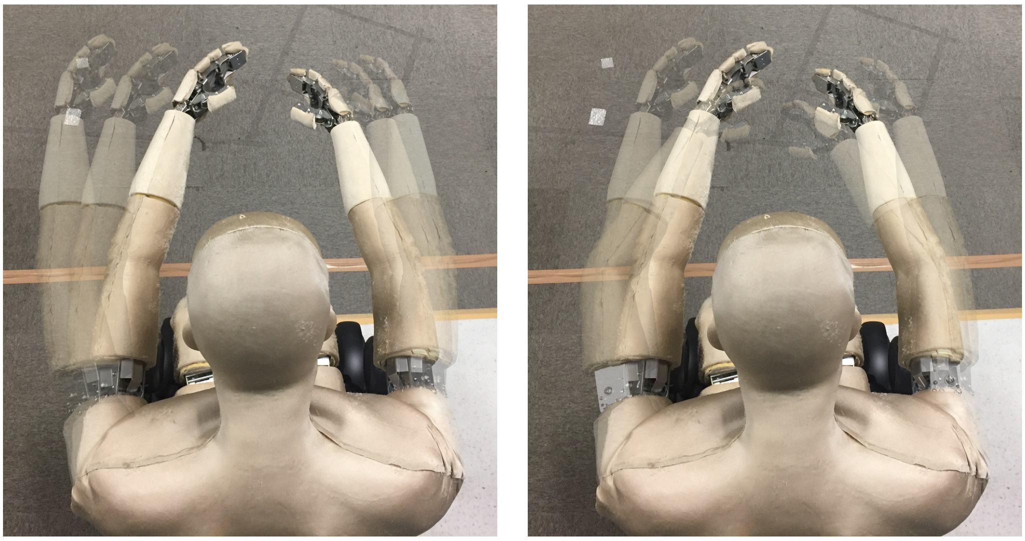


Figure 6.4: Control points for arm dressing task showing start and goal points of DMP system colored as orange



(a) The first type of movement

(b) The second type of movement

Figure 6.5: Movements defined for the arms of the robotic subject

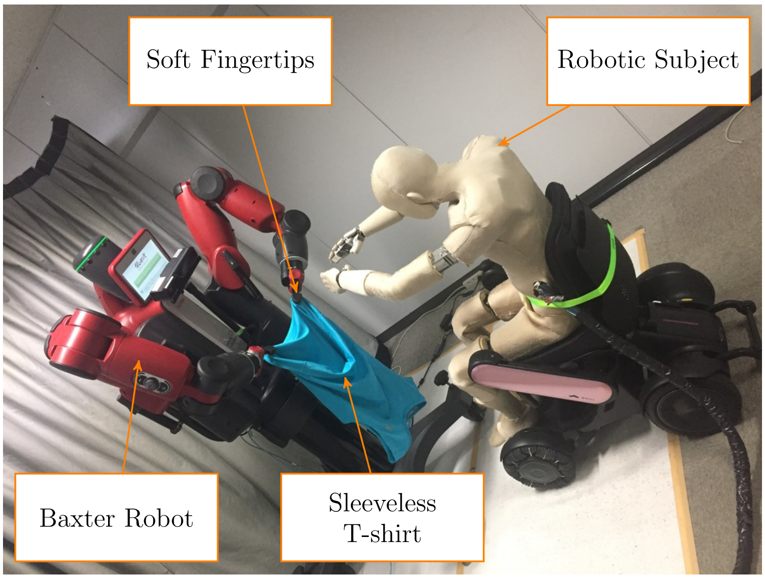


Figure 6.6: Setup of the evaluation task

6.3 Evaluation

We use Robot Operating System (ROS) to implement our framework in Ubuntu OS. Baxter robot is connected to the Ubuntu computer using an Ethernet cable. We used Ubuntu 14.04 LTS 64 Bit OS having 8 GB RAM on Intel Core i7, 3.40 GHz x 8 CPU for training and testing our framework. The clothing articles used in this study is 100% polyester (size L) sleeveless T-shirt. We have defined two types of movements for the arms of the robotic subject as shown in Fig. 6.5. These movements belong to day-to-day arm stretching movements and are defined empirically. More precisely, in these two movements, both the arms move in a horizontal plane. In the first motion, only the shoulder joint rotates. However, in the second motion, the elbow joint rotates primarily. The control points of DMP are set based on these movement trajectories.

The DMP system accomplished the arm dressing task. We defined a DMP for each coordinate axis and each arm. Hence we have a total of 6 DMPs, 3 for each arm. The demonstrated trajectory is parameterized using these DMPs. Fig. 6.7 shows the DMPs are well capable of learning the complex Baxter trajectories. We can see that DMPs are following the demonstrated trajectory.

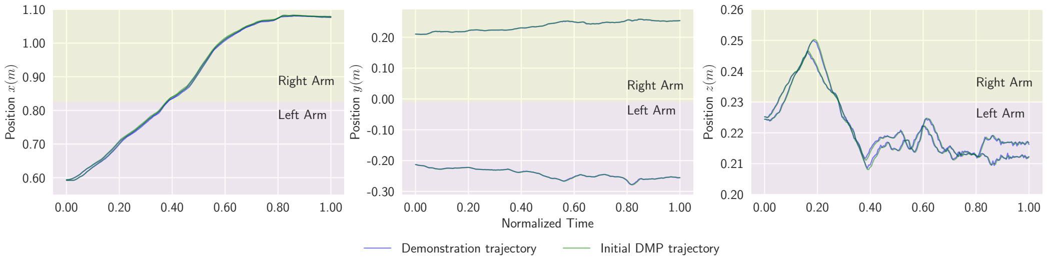
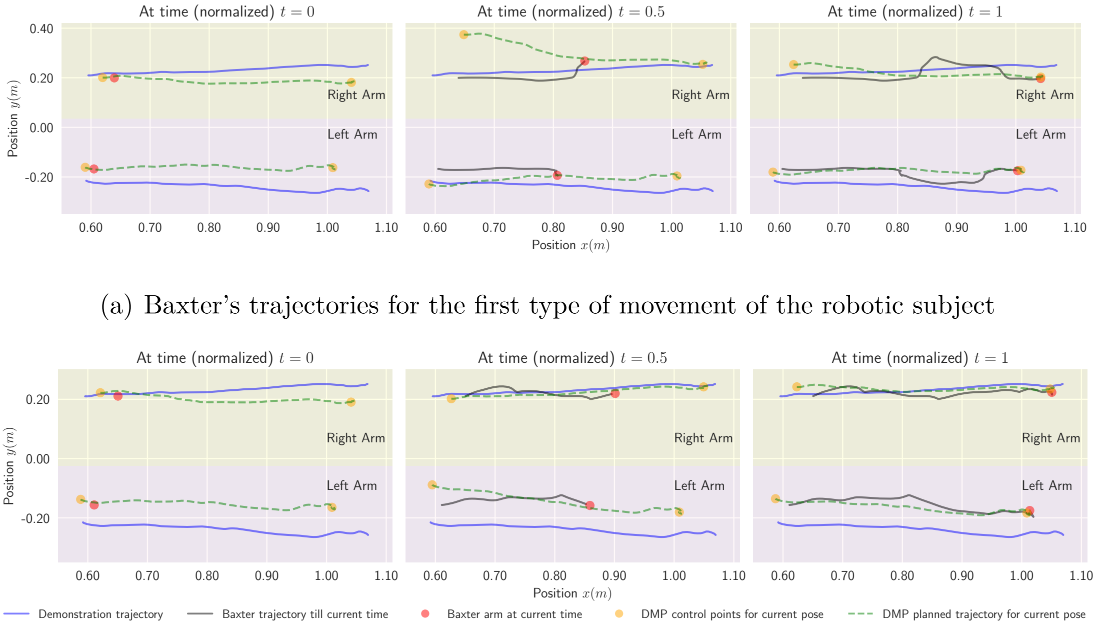


Figure 6.7: Baxter's arms trajectories for both the arms which are parameterized by using DMP. The time is normalized to [0, 1] range. The figure shows the DMPs are well capable of learning the complex Baxter trajectories.

Initial DMP is modified to accommodate new posture by changing start and goal parameters acquired from the forward kinematics of the robotic subject. The robot is commanded at each timestamp while setting the control points on the fly. During the movements of arms of the robotic subject, the robot adapts as shown in Fig. 6.8(a). This figure corresponds to the first type of arm movement as shown in Fig. 6.5(a). Baxter robot starts from the fingertips of the robotic subject. The time (t) is normalized to [0, 1] range for easier visualization. At $t = 0$, the fingertips of the robotic subject are parallel to the elbow of the robotic subject. As per the defined movement, the fingertips of the robotic subject start moving apart from each other. At $t = 0.5$, we can see that both the arms of the Baxter robot are adopting this change and moving away from each other. Baxter's end-effector corresponds to the left arm of the robotic subject is moving downwards whereas Baxter's end-effector corresponds to the right arm of the robotic subject is moving upwards. This motion is desired since Baxter needs to put the clothing article. Hence, it needs to expand the cloth in this situation. At $t = 1$, Baxter arms are approaching elbows of the robotic subject. The same behavior can be observed in Fig. 6.8(b). This figure corresponds to the second type of arm movement as shown in Fig. 6.5(b). At $t = 0.5$, we can see that the fingertips of the robotic subject are moving closer to each other. Hence Baxter immediately starts moving closer to the arms of the robotic subject. We can



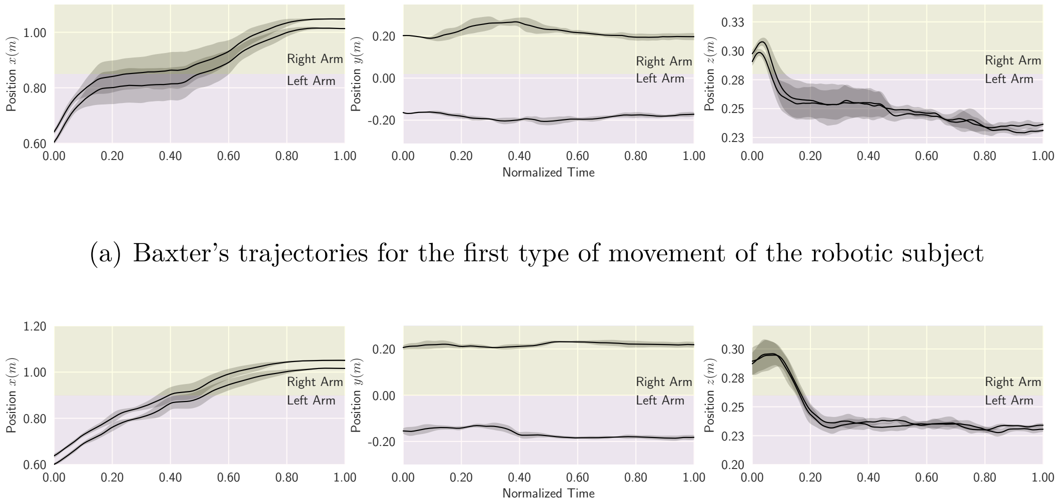
(a) Baxter's trajectories for the first type of movement of the robotic subject

(b) Baxter's trajectories for the second type of movement of the robotic subject

Figure 6.8: Trajectories of Baxter's arms while performing arm dressing task. The orange colored points showing control points of DMP, are moving as per the defined motion. The robot successfully adopts to the fast motions of the arms.

also see that even though the demonstrated trajectory is quite simple but Baxter trajectory turned out to be a complex one.

We ran the arm dressing task ten times for each type of arm movements and visualized the robot trajectory during the task. The visualization is shown in Fig. 6.9. The time is normalized to [0, 1] range. For each trajectory, the mean position is plotted with the black color, and the region $\mu \pm \sigma$ along the mean position is filled with the gray color. Even though, from the setup of our system, it appears that both the arms of the robotic subject are in symmetry. However, after looking at the y and z coordinate axis of Baxter's trajectories, it can be said that the arms are indeed not in symmetry. The z coordinate is in the vertical direction and corresponds to the height of arms of the subject. Moreover, the difference in z coordinate validates the unsymmetrical position of the arms.



(a) Baxter's trajectories for the first type of movement of the robotic subject

(b) Baxter's trajectories for the second type of movement of the robotic subject

Figure 6.9: Trajectories of Baxter's arms for the motions of the robotic subject. Baxter is run ten times for each type of movement. The time is normalized to [0, 1] range. For each trajectory, the mean position is plotted with the black color, and the region $\mu \pm \sigma$ along the mean position is filled with the gray color.

6.4 Discussion

In recent years, assistive robotic devices for nursing care have been developed and commercialized for such purposes. To make such devices accessible in the care facilities, we need to systematically evaluate the performance and the effects of the devices on the care receivers and caregivers.

We have developed a clothing assistance robot using Baxter and conducted many successful demonstrations mainly with healthy people. It was, however, impossible to systematically evaluate its performance with a human subject because the posture of human arms is invisible due to the cloth over them during dressing. To address this problem, we have proposed to use another humanoid robot, Whole-Body Robotic Simulator of the Elderly [85, 86] that can mimic the posture and movements of the elderly persons during the dressing task. In this study, we specifically evaluated our clothing assistance framework employing DMP for the arm dressing tasks with the robotic subject. The control points of DMP are determined by applying forward kinematics on the robotic simulator. We have performed a quantitative evaluation of arm dressing task by using forward kinematics for calculating the arm positions of the robotic simulator. We have shown the plausibility of our approach through the experiments where we defined two different arm movements, which were supposed to be disturbances, of the robotic subject during the arm dressing task.

Although, it appears from the setup of our task that both arms of the subject are required to be in symmetry. However, separate DMPs are employed to take care of each arm. During the task, arms are constrained due to the sleeveless T-shirt over them. In this situation, the arms cannot be moved beyond a limited range. Hence, both the arms are restricted to be parallel even though separate DMPs are employed for both arms.

7 Electric Wheelchair-Robot Collaboration

We have proposed a framework for the same by employing imitation learning from a human demonstration to a compliant dual-arm robot. As the robot has a limited workspace, this framework involves a manual movement of the wheeled chair by pushing it while coordinating with the robot to stay within the workspace of the robot. To avoid the manual push and coordination, we facilitate the automatic movement of the chair based on the trajectory of the robot's dual arms. In this chapter, we present an approach for the collaboration of an electric wheelchair and a humanoid robot to achieve the clothing assistance task. Our approach incorporates Manifold Relevance Determination (MRD) to learn an offline latent model from the simultaneous observations of the clothing assistance task as well as the movement of the wheelchair. We trained and tested the latent model on different human subjects by dressing a sleeveless T-shirt. Experimental results verify the plausibility of our approach. The important feature of this study is human-robot and robot-robot collaboration in service robotics.

7.1 Setup of the System

In this study, we present an approach to collaborate between the wheelchair and the humanoid robot to perform the clothing assistance task. The setup of our system is shown in Fig. 7.1. A human subject is sitting in an electric wheelchair and facing his hands towards the robot. The robot's arms have a limited workspace and need to reach the torso of the subject to perform the dressing task. Hence the chair must move forward to stay within the workspace of the robot during the dressing task. Therefore, it is empirical that the joint angles of the robot and the

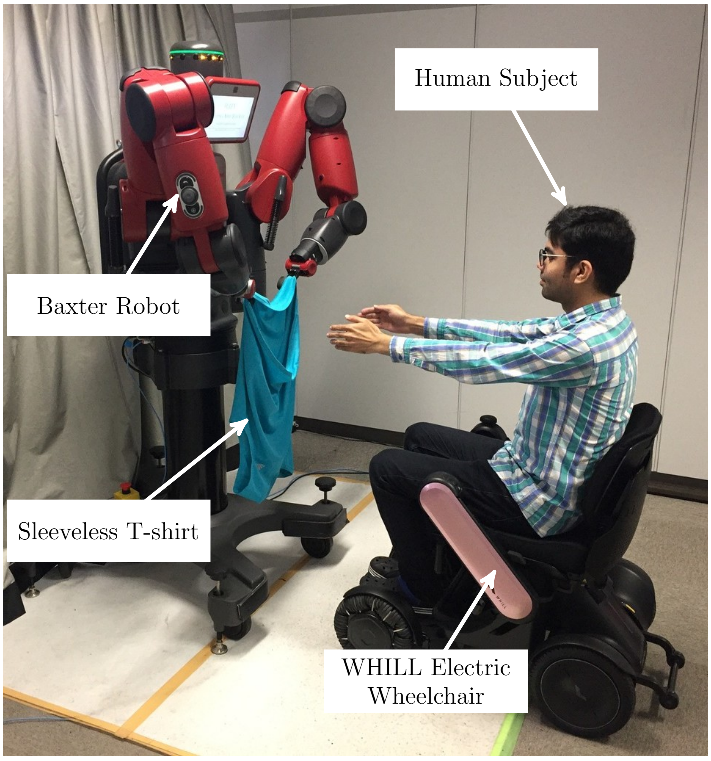


Figure 7.1: Setup of the task

movement of the chair share a common latent space. This is why we employed MRD to learn the latent space offline for the simultaneous observations from the clothing assistance task as well as the movement of the wheelchair. To the best of our knowledge, this is the first work addressing collaboration between wheelchair and robot to perform clothing assistance.

The experimental setup contains a compliant dual-arm humanoid Baxter robot and an electric wheelchair WHILL [88]. The Baxter robot has 7 degrees of freedom (DOF) in each arm, adding up to a total of 14 joint angles required to define a specific configuration of the robot. The Baxter robot is controlled using the Robot Operating System (ROS) in Ubuntu PC. It is connected to the PC using an Ethernet cable. We command the robot using the Baxter API, which is supported by ROS.

The WHILL wheelchair is also controlled using ROS, and our in-house developed API is used to command the movement [89]. The movement of the wheelchair refers to the tilt of the joystick and comprises of two parameters, forward tilt and sidewise tilt. The forward and sidewise tilt causes the forward and turn movement, respectively. The default joystick value is a tuple of forward and sidewise tilt, and it is (0, 0). The range of forward and sidewise tilt is [-100, +100]. Note that if the forward tilt is negative, i.e., the joystick is tilted backward, the wheelchair moves in the backward direction. In this study, we focus only on the forward movement of the wheelchair and keep the sidewise tilt to 0 always.

The wheelchair is kept approximately a meter away from the robot. To dress the sleeveless T-shirt, the subject needs to keep his hands stretched outwards and facing the robot. We used a sleeveless polyester T-shirt during the experiment. We used Ubuntu 14.04 LTS 64-bit Operating System having 8GB RAM on Intel Core i7, 3.40 GHz x 8 CPU for training and testing our method.

7.2 Method

In this study, we are using a sleeveless T-shirt as the clothing article. This task is performed using a compliant dual-arm Baxter robot while coordinating with WHILL, an electric wheelchair [88]. The cloth goes through the arms, then goes over the head, and finally reaches up to the torso of the subject. An overview of the proposed method is shown in Fig. 7.3. We define two observation spaces, i.e., Baxter joint angle space, and WHILL movement space. We perform a kinesthetic demonstration of the task while Baxter is controlled under gravity compensation mode. During the demonstration, an expert manipulates the arms of Baxter robot while a subject is sitting in the wheelchair, as shown in Fig. 7.2. At this stage, the wheelchair is controlled manually by using a joystick. Baxter joint angle space consists of the joint space trajectory of the robot, whereas the WHILL movement space consists of the movement given to the wheelchair during the demonstration. We apply MRD on both the observation spaces to discover shared dimensions in a 2D latent space. This latent space encodes the motor skills required to perform the clothing assistance tasks as well as to depict the wheelchair movement. During the inference, the mean trajectory is sampled from the latent space. This mean trajectory of the latent space is used to infer the joint space trajectory of the

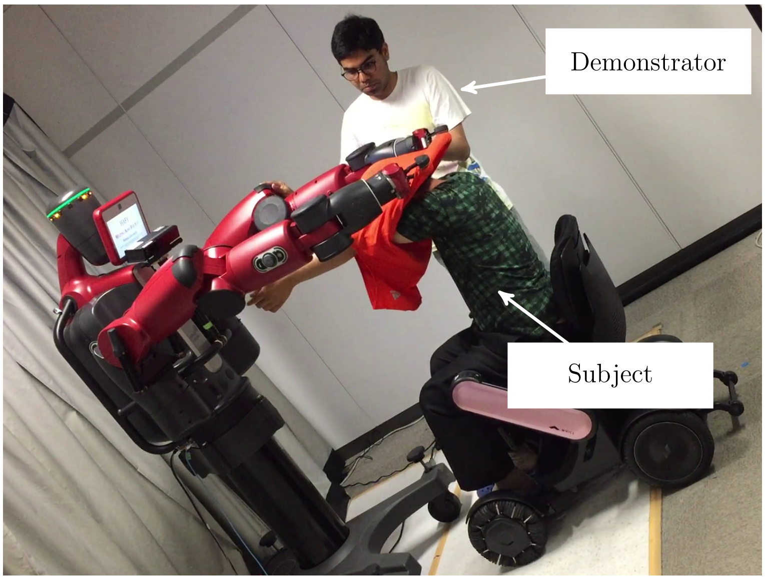


Figure 7.2: Demonstration of the task

Baxter robot. We then use the joint angles of Baxter to predict the movement of the wheelchair in real-time using the learned MRD model.

7.3 Shared Manifold Learning for Automated Wheelchair Movement

For training the MRD model, we collected data of joint angles of the Baxter robot and the corresponding wheelchair movement by performing a kinesthetic demonstration of the task on a subject. The collected data comprises the joint space trajectory of the Baxter robot and corresponding wheelchair movement throughout the clothing task. The data were preprocessed before being used for training the MRD model, as shown in Fig. 7.4. We used the median filter and cubic interpolation to preprocess the collected data. The WHILL movement refers to the forward tilt of the joystick as we have not considered sidewise tilt by keeping it 0 always.

We aim to learn a single latent space for the two observation spaces, i.e., joint angles of the Baxter robot $\left(\boldsymbol{Y} \in \mathbb{R}^{N\times 14}\right)$ and forward movement of the wheelchair

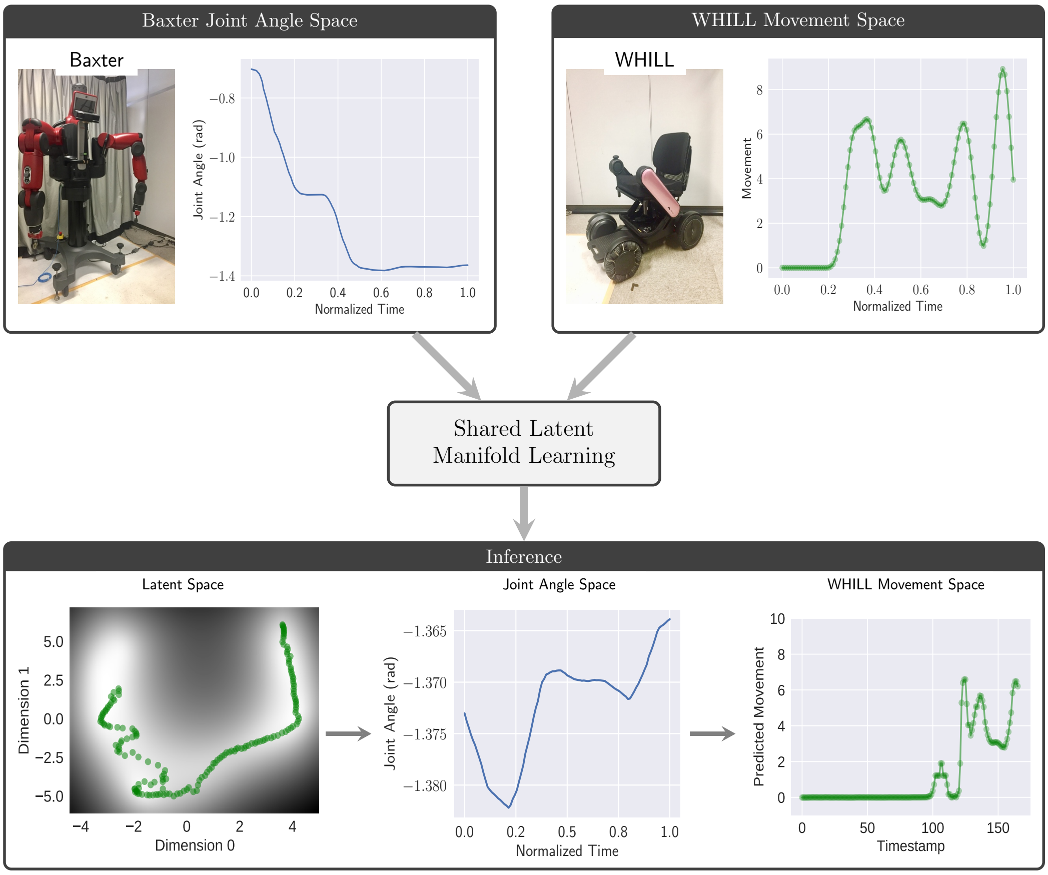


Figure 7.3: An overview of the proposed method for the collaboration between Baxter and WHILL. We defined two observation spaces, i.e., Baxter joint angle space (only 1 of the 14 joint angles of the robot is indicated in the graphs here) and WHILL movement space, and learned a shared latent space by employing MRD.

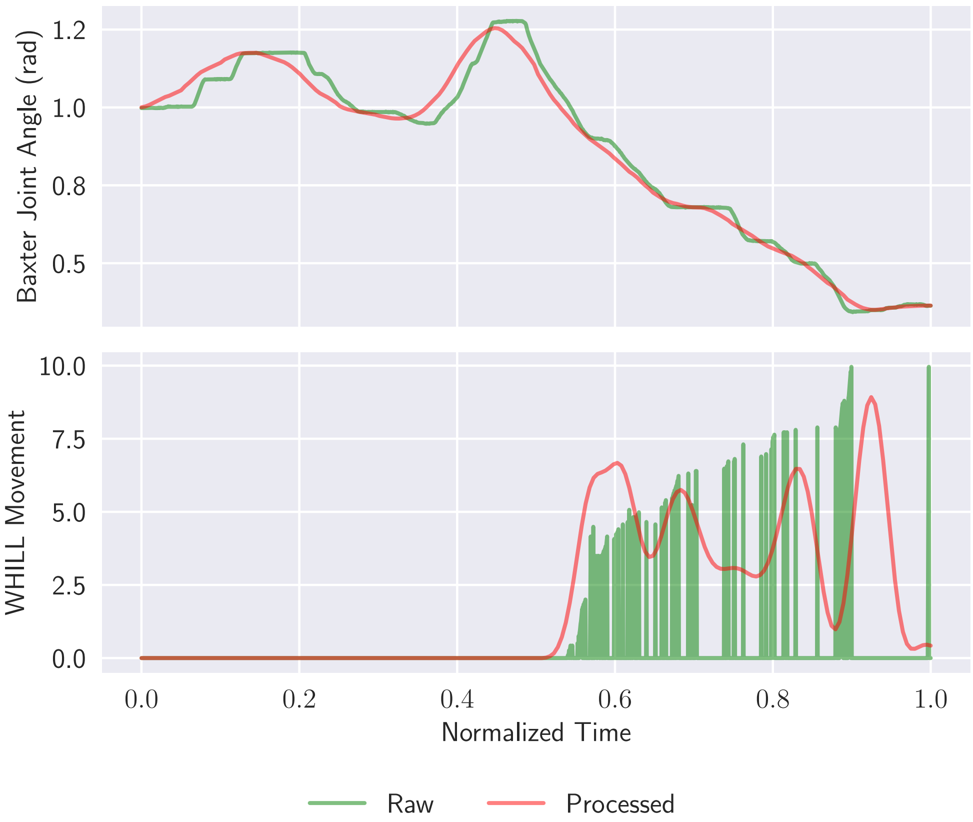


Figure 7.4: Preprocessing of the collected data. We use the median filter and cubic interpolation to preprocess the collected data. We are showing only one joint angle of the Baxter robot. The WHILL movement refers to the forward tilt of the joystick.

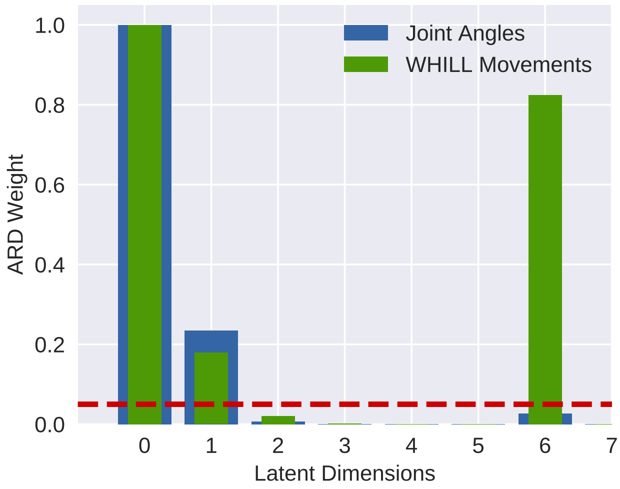


Figure 7.5: ARD weights of each latent dimension

$\left(\boldsymbol{Z} \in \mathbb{R}^{N\times 1}\right)$. Therefore, we train a MRD model using these two observation spaces. The MRD model was implemented using the GPy python library [84]. The latent variable $\boldsymbol{X}$ was initialized using Principal Component Analysis (PCA) from the preprocessed data. We have used 8 latent dimensions in this experiment, with 6 latent dimensions allocated for the joint angles of the Baxter robot and 2 latent dimensions for the movement of the wheelchair. ARD kernel and 100 inducing points were used to learn the MRD model. The model was trained in the following 3 steps:

1. For both observation spaces, the signal-to-noise ratio (SNR) was fixed to constrain the variance of Gaussian noise and Radial Basis Function (RBF) kernel. In this configuration, the model was optimized for ten iterations.

2. Each observation space, i.e., $\boldsymbol{Y}$ and $\boldsymbol{Z}$, was optimized individually for 200 iterations.

3. The model was trained without any constraints and optimized for 200 iterations.

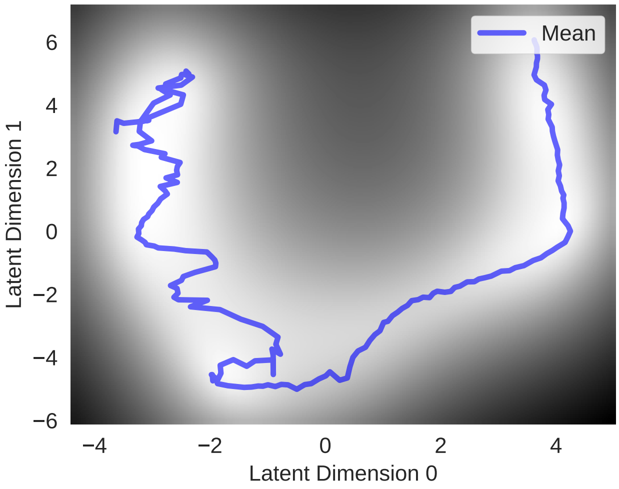
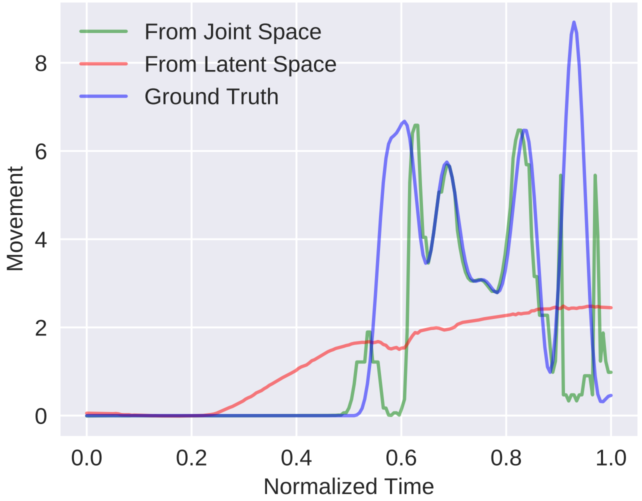


Figure 7.6: Latent Space generated using first two latent dimensions

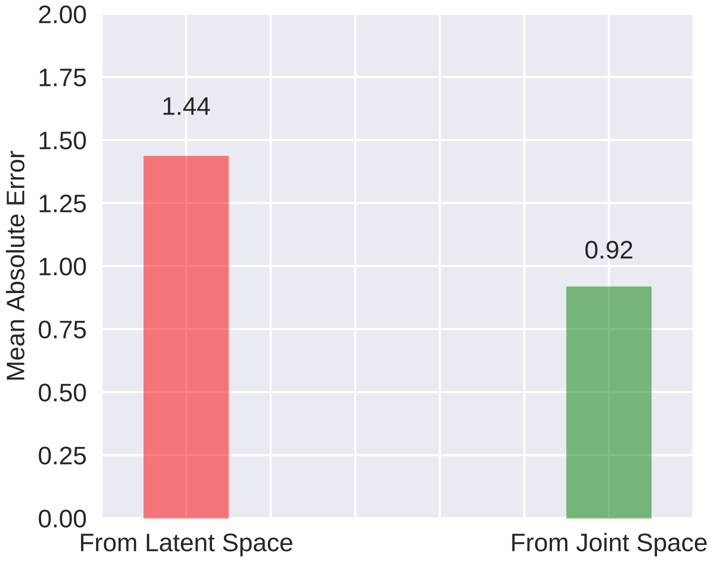
7.4 Results and Discussion

After training the MRD model, the ARD weights for each latent dimension are computed. We observed that there were two shared latent dimensions between the two observation spaces, namely latent dimensions 0 and 1. The ARD weights for each latent dimension are shown in Fig. 7.5. The latent dimensions 0 and 1 are following our intuition that the joint angles of the Baxter robot and the movement of the wheelchair share common latent dimensions. These two latent dimensions constitute the shared latent space ($\boldsymbol{X}_S$). The 2D latent space generated using the first two latent dimensions is shown in Fig. 7.6.

The latent space shown in Fig. 7.6 is used to infer the joint angles of Baxter. The corresponding WHILL movement is predicted in real-time from the joint angles of the Baxter robot using the learned MRD model through the inference process explained in Section 3.3. Alternatively, we can predict both the observation spaces directly from the latent space. However, we observe that the latent dimension 6 encodes the second-highest amount of information about the WHILL movement, as it can be noted from the ARD weights shown in green color in Fig. 7.5. The latent dimension 6 constitutes the private latent space of the WHILL movement, i.e., $\boldsymbol{X}_Z$. Hence, the prediction of WHILL movements



(a) Predicted WHILL movements. The WHILL movement refers to the forward tilt of the joystick.



(b) Comparison of the Mean Absolute Error (MAE)

Figure 7.7: Comparison of the predictions for WHILL movement

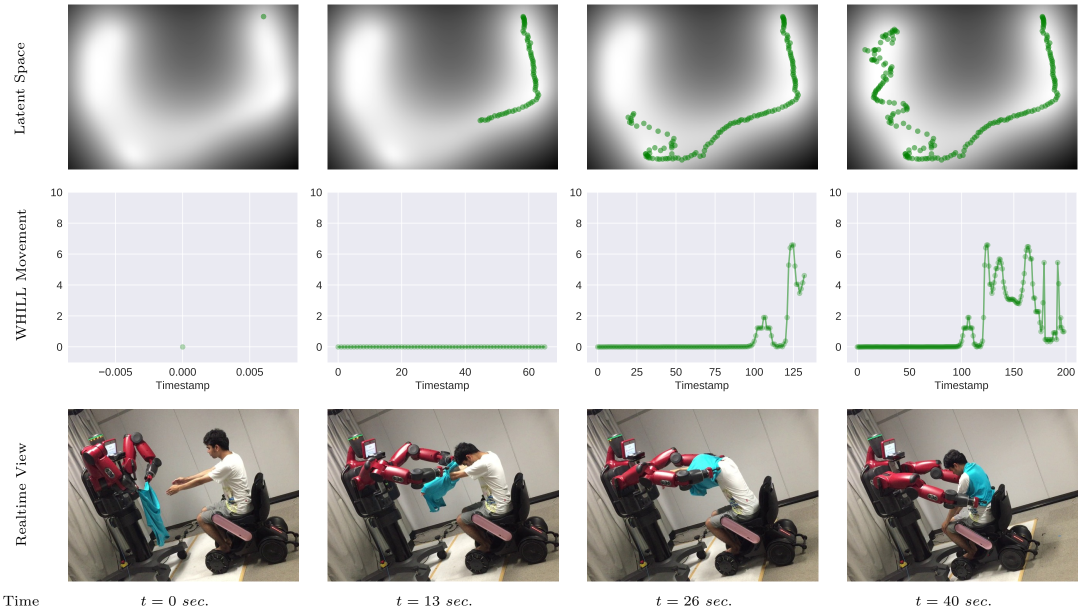


Figure 7.8: The clothing assistance task with Baxter and WHILL shown at various timestamps. The mean trajectory in latent space is sampled and shown in Green color in the latent space.

Table 7.1: Body physique information of the subjects

Parameter	Subject for Training	Subject for Testing
Height (cm)	166	173
Age (yrs)	30	29
Shoulder Width (cm)	45	46
Waist Size (cm)	92	86

from the latent space (Fig. 7.6) would lead to poor prediction accuracy due to the following reasons-

- The latent space is generated from latent dimensions 0 and 1.
- The latent space does not incorporate latent dimension 6.

Therefore, we predict the WHILL movement from the joint angles of Baxter using MRD inference described in Section 3.3.1.

Furthermore, to calculate the prediction accuracy of WHILL movements, we computed the Mean Absolute Error (MAE) by comparing the predicted movements with the ground truth. The ground truth is obtained by preprocessing the collected WHILL movement data. MAEs are computed for WHILL movement prediction from joint space and latent space. The predicted WHILL movements are shown in Fig. 7.7(a) and the corresponding MAEs are shown in Fig. 7.7(b).

The predictions from joint space are close to the ground truth, which can be verified by observing the corresponding MAE. We used the learned MRD model to perform the complete dressing of a sleeveless T-shirt. The robot starts moving from the home position. During the clothing assistance task, the Baxter robot collaborates with WHILL to successfully achieve the task. The dressing task at various timestamps is shown in Fig. 7.8. The complete task took 40 seconds to dress a sleeveless T-shirt. The body physique information of the subjects is given in Table 7.1.

7.4.1 Limitations

At present, there are limitations of our work, as listed below.

- We have only considered the forward movement of the wheelchair and ignored the sidewise rotation.

- In this study, wheelchair movement is defined by the tilt of the joystick, which is analogous to velocity control of the wheelchair. Thus, the current system needs proper synchronization with the wheelchair to prevent overshooting the boundaries and causing a collision with the robot base. Due to this limitation, we have used a fixed starting position for the wheelchair.

- We assume that there are no obstacles in front of the wheelchair, as the current system does not have any obstacle avoidance mechanism.

- As this is a preliminary study, the current system is trained and tested only on healthy young subjects. An evaluation using elderly subjects is yet to be done.

- We have used a sleeveless T-shirt and shown successful dressing of it using the proposed method. Dressing using other types of clothes, such as pajamas and a hospital gown, is yet to be done.

- The MRD model is trained using one demonstration trajectory only. Empirically, it makes the proposed method data-efficient. However, in practice, the generalization capability of the model is limited. To tackle this issue, we can use multiple demonstrations to train the model in the future.

8 Incremental Imitation Learning

Programming motions for Robotic Clothing Assistance is a challenging problem due to the complexity of the task. Prior work has shown that imitation learning provides an intuitive way for teaching complex robot motion by learning from demonstrations (LfD) [24] Thus imitation learning is considered a suitable procedure of teaching motor skills for Robotic Clothing Assistance. To obtain satisfactory performance in imitation learning, it is essential to collect the demonstration dataset that covers all the necessary conditions. However, it is challenging to collect the demonstration dataset that covers the necessary conditions before deploying the system. It is also the case for Robotic Clothing Assistance systems. Since the type of subjects is various, it is challenging to collect sufficient demonstration dataset that covers all the possible task conditions. Therefore, the functionality of incremental learning of dressing behavior is critical in practice.

Although the functionality of incrementally learning from physical HRI is important in clothing assistance, it has not been addressed in previous studies. To address this issue, we developed a system that incrementally learns whenever a new demonstration is performed and adjusts the behavior according to the new demonstration. Our incremental learning process allows correcting the robot behavior by performing physical interaction with the robot. Therefore, the contribution of this work is to develop an incremental learning method to overcome the issue of Robotic Clothing Assistance.

In our approach, a policy that plans a clothing assistance trajectory from a given task condition is learned from demonstrations performed by a human teacher. We assume that the demonstrations initially provided by the expert are not sufficient. To cope with the lack of sufficient demonstration data, we proposed a framework that collects the additional trajectory through physical HRI. Since trajectories obtained through physical HRI are not optimal in practice, we employ the trajectory update based on [31]. We evaluate our algorithm on a dual-arm robot by performing Robotic Clothing Assistance and empirically verify the performance of the proposed system. Experimental results show the improvement of the performance of autonomous clothing assistance through human-robot physical interaction.

The main contributions of this study are two-fold.

- We propose an algorithmic framework for incremental imitation learning for clothing assistance.
- In addition, we empirically analyze the behavior of the proposed system in the context of clothing assistance.

8.1 Setup of the System

We evaluated our algorithm on Robotic Clothing Assistance. The experimental setup uses a dual-arm compliant humanoid robot, Baxter as shown in Fig. 8.1. The human subject sits in front of the robot on a movable chair. Due to the limited workspace of the robot, the robot cannot reach the torso of the subject. Therefore, the subject is required to sit on a movable chair to deal with the limited workspace problem of the robot. During the dressing, when the T-shirt approaches the shoulders of the person, the person needs to move forward to complete the dressing. A Kinect v2 sensor is mounted in front of the LCD display of the robot. The Kinect sensor is tilted downwards, facing the subject and used to locate the body parts of the human subject. We designed custom soft fingertips and mounted on fingers of the electric gripper of Baxter robot. In this experiment, we are using a sleeveless T-shirt as the clothing item. Before starting the experiment, the shirt is put on the fingers of the robot by a human assistant. We marked two grasping points on the T-shirt to ensure consistent manual loading of the T-shirt. We used a PC with an Intel Core i7-7700 @ 3.60 Ghz x 8 CPU, 16 GB memory on Ubuntu 14.04 LTS 64 Bit OS for this research.

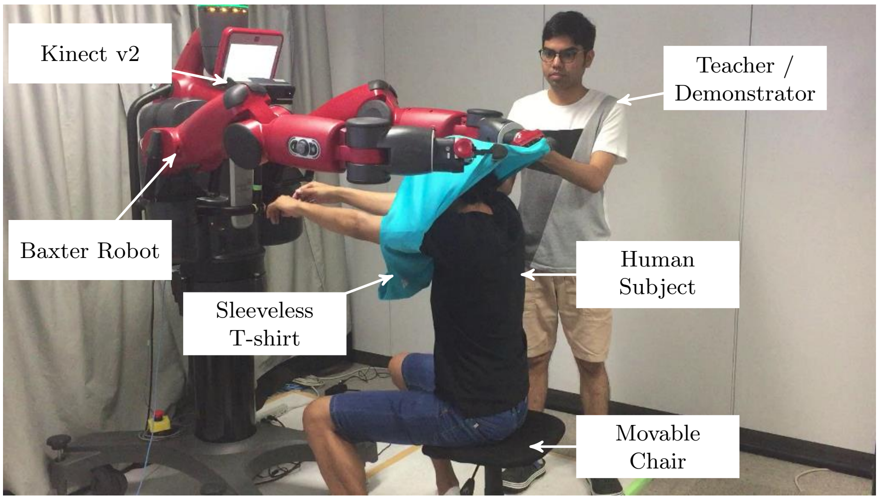


Figure 8.1: The experimental setup of the Robotic Clothing Assistance task

8.2 Method

Our algorithm consists of two steps, (a) Learning the distribution of robot trajectories, and (b) Incorporating the human physical feedback. In subsequent sections, we explain each of them in detail.

8.2.1 Learning Distribution of Robot Trajectories

We consider a problem of learning from the dataset of demonstrations $\mathcal{D} = \{(\boldsymbol{\tau}_i, \boldsymbol{c}_i)\}_{i=1}^{N}$. The dataset consists of pairs of a robot trajectory $\boldsymbol{\tau}$ and a task condition vector $\boldsymbol{c}$. The parameter N represents the total number of demonstrations in the dataset. We aim to learn a policy $\pi(\boldsymbol{\tau}|\boldsymbol{c})$ that generates a trajectory $\boldsymbol{\tau}$ for a given task condition $\boldsymbol{c}$. A trajectory $\boldsymbol{\tau}$ is given by $\boldsymbol{\tau} = [\boldsymbol{q}_0, \ \ldots, \ \boldsymbol{q}_T]$ where $\boldsymbol{q}_t$ is the robot configuration at the tth time step. The task condition vector $\boldsymbol{c}$ could be the position of the human subject. However, in our experiment, the fingertips and elbows of the subject are used to define $\boldsymbol{c}$. As in ProMP [28], we represent the trajectory as a linear combination of basis function given by

$$\boldsymbol{q}_t = \boldsymbol{w}^\top \boldsymbol{\phi}(t), \tag{8.1}$$

where $\boldsymbol{\phi}(t)$ is the basis function. The form of $\boldsymbol{\phi}(t)$ can be chosen based on the type of movement [28]. Since Robotic Clothing Assistance requires a point-to-point movement, we employ the Gaussian basis function given by

$$\boldsymbol{\phi}(t) = \frac{\exp\left(-\frac{(z(t)-c_i)^2}{2z_h}\right)}{\sum_{j=1}^{B}\exp\left(-\frac{(z(t)-c_j)^2}{2z_h}\right)}, \tag{8.2}$$

where B defines the number of basis function for each DOF of the robot, $z(t)$ is the phase variable, and z_h and z_c represent the width and centers of the basis function respectively. In our experiment, z_h is kept constant i.e., $z_h = 1/B$. The centers of the basis function c_i for $i = 1, \ldots, B$ are uniformly placed in $[0,\ 1]$ interval.

We initialize a weight matrix $\boldsymbol{W}$ where $\boldsymbol{W} = \{\boldsymbol{w}_i\}_{i=1}^{N} \in \mathbb{R}^{N \times P \cdot B}$. The parameter P defines the DOF of the robot. In this way, each DOF can be parameterized by B basis functions. Note that the weight matrix $\boldsymbol{W}$ is a 2D matrix containing N rows and $P \cdot B$ columns. The ith row of this matrix is computed by $\left(\boldsymbol{\phi}^\top \boldsymbol{\phi} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\phi}^\top \boldsymbol{\tau}_i$, where λ is a regularization factor and set to very small value i.e., $\lambda = 10^{-12}$. The parameter $\boldsymbol{I}$ denotes the identity matrix.

The policy $\pi_{\boldsymbol{\theta}}$ generates the trajectory parameter $\boldsymbol{w}$ given the task condition $\boldsymbol{c}$. We assume that the conditional distribution of $\boldsymbol{w}$ given $\boldsymbol{c}$ is a Gaussian distribution $\mathcal{N}\left(\boldsymbol{f}(\boldsymbol{c}), \Sigma\right)$ so that

$$\begin{aligned} \pi_{\boldsymbol{\theta}}(\boldsymbol{w}|\boldsymbol{c}) &= P(\boldsymbol{w}|\boldsymbol{c}) \\ &= \mathcal{N}\left(\boldsymbol{f}(\boldsymbol{c}), \Sigma\right) \end{aligned} \tag{8.3}$$

where $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{c})$ is a vector and represents mean of the Gaussian. $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{c})$ is given by

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{c}) = \psi(\boldsymbol{c})^\top \theta \tag{8.4}$$

where $\boldsymbol{\psi}$ is a feature function. Although the feature function is not limited to specific forms, we use exponential feature function to represent $\boldsymbol{\psi}$ as shown below

$$\boldsymbol{\psi}(\boldsymbol{c}) = \exp\left(\frac{\|\boldsymbol{c} - \boldsymbol{c}_s\|^2}{2{c_h}^2}\right) \tag{8.5}$$

where we refer to $\boldsymbol{c}_s$ as the basis condition. To obtain $\boldsymbol{\psi}(\boldsymbol{c}) \in \mathbb{R}^K$, we uniformly pick up K samples from the task conditions in the demonstration dataset and use them as $\boldsymbol{c}_s$. In this way, the number of the basis conditions i.e., K becomes the dimension of the feature function $\boldsymbol{\psi}$. Here, the parameter c_h is refereed to as the width of the feature function. It indicates that we cannot extrapolate more than c_h units away in the condition $\boldsymbol{c}$. The parameter c_h is set to 0.6 in our experiment.

The exponential feature function is similar to the kernel trick such as squared exponential kernel a.k.a. the Radial Basis Function (RBF) kernel, the Gaussian kernel. With the exponential feature function, we can represent the non-linear relationship of the conditions.

The policy parameter $\boldsymbol{\theta}$ can be obtained by minimizing the following mean squared error (MSE) of the residuals

$$\min \sum_{i=1}^{N} \|\boldsymbol{w}_i - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{c})\|^2$$

The minimizer of the above MSE can be obtained by computing

$$\boldsymbol{\theta}^* = \boldsymbol{\Psi}^\dagger \boldsymbol{W} \tag{8.6}$$

where $\boldsymbol{\Psi}$ is the feature matrix given by $[\boldsymbol{\psi}(\boldsymbol{c}_1)^\top, \ \ldots, \ \boldsymbol{\psi}(\boldsymbol{c}_N)^\top]^\top$ and $\boldsymbol{\Psi}^\dagger$ is the pseudo-inverse of $\boldsymbol{\Psi}$.

Theoretically, we are performing a regression in which we have $\boldsymbol{w}$ and feature vector $\boldsymbol{\psi}(\boldsymbol{c})$. We are projecting each dataset to the solution space and obtaining the linear function. The solution is linear in this space because we are using a non-linear kernel, i.e., exponential feature function.

The algorithm for learning the policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{\tau_w}|\boldsymbol{c})$ is given in Algorithm 2. We start by computing the weight matrix $\boldsymbol{W}$. Then we compute exponential features from basis conditions. Next, we obtain optimal $\boldsymbol{\theta}^*$. Finally, we calculate the covariance and ensure that it is a positive-definite matrix.

Algorithm 2 Learn policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{\tau}_{\boldsymbol{w}}|\boldsymbol{c})$

```
Input Dataset of demonstrations $\mathcal{D} = \{(\boldsymbol{\tau}_i, \boldsymbol{c}_i)\}_{i=1}^{N}$
Compute basis $\boldsymbol{\phi}$                                          ▷ Using Eq. (8.2)
Initialize $\boldsymbol{W} = \{\boldsymbol{w}_i\}_{i=1}^{N} \in \mathbb{R}^{N \times P \cdot B}$
for each trajectory $\boldsymbol{\tau}_i$ do
    Compute weight $\boldsymbol{w}_i \leftarrow \left(\boldsymbol{\phi}^\top \boldsymbol{\phi} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\phi}^\top \boldsymbol{\tau}_i$
Define basis conditions $\boldsymbol{c}_s \leftarrow$ uniform_selection$(\boldsymbol{c}, K)$
Compute features $\boldsymbol{\psi}(\boldsymbol{c}) \leftarrow f_{exp}(\boldsymbol{c}, \boldsymbol{c}_s, c_h)$      ▷ Using Eq. (8.5)
Compute theta $\boldsymbol{\theta} \leftarrow \boldsymbol{\Psi}^\dagger \boldsymbol{W}$
Compute $\boldsymbol{\Sigma}_{\boldsymbol{c}} \leftarrow \frac{1}{N}\left(\boldsymbol{W} - \boldsymbol{\Psi}\boldsymbol{\theta}\right)\left(\boldsymbol{W} - \boldsymbol{\Psi}\boldsymbol{\theta}\right)^\top$
if $\boldsymbol{\Sigma}_{\boldsymbol{c}}$ is positive-definite then
    $\boldsymbol{\Sigma}_{\boldsymbol{c}} \leftarrow$ nearest_positive_definite$(\boldsymbol{\Sigma}_{\boldsymbol{c}})$
Output $\boldsymbol{\phi}, \boldsymbol{c}_s, \boldsymbol{\theta}, \boldsymbol{\Sigma}_{\boldsymbol{c}}$
```

Algorithm 3 Prediction of the trajectory

```
Input A new condition $\boldsymbol{c}'$
Learn $\boldsymbol{\phi}, \boldsymbol{c}_s, \boldsymbol{\theta}, \boldsymbol{\Sigma}_{\boldsymbol{c}}$ from $\mathcal{D} = \{(\boldsymbol{\tau}_i, \boldsymbol{c}_i)\}_{i=1}^{N}$      ▷ Using Algorithm  2
Compute feature $\boldsymbol{\psi}(\boldsymbol{c}') \leftarrow f_{exp}(\boldsymbol{c}', \boldsymbol{c}_s, c_h)$
Compute weight $\boldsymbol{W}' \leftarrow \mathcal{N}(\mu, \boldsymbol{\Sigma}_{\boldsymbol{c}})$      ▷ $\mu \leftarrow \boldsymbol{\psi}(\boldsymbol{c}')^\top \boldsymbol{\theta}$
Compute trajectory $\boldsymbol{\tau}^{\text{plan}} \leftarrow \boldsymbol{\phi} \boldsymbol{W}'$
Output $\boldsymbol{\tau}^{\text{plan}}$
```

Given a new condition $\boldsymbol{c}'$, the predicted trajectory $\boldsymbol{\tau}^{\text{plan}}$ is calculated from the policy $\pi_{\boldsymbol{\theta}}$, is shown in Algorithm 3.

8.2.2 Incorporating Human Physical Feedback

The policy $\pi_{\boldsymbol{\theta}}$ is used to predict the trajectory of the robot, i.e., $\boldsymbol{\tau}^{\text{plan}}$. However, when the demonstration dataset does not cover all the necessary task conditions, the trajectory $\boldsymbol{\tau}^{\text{plan}}$ generated by $\pi_{\boldsymbol{\theta}}$ may not be optimal and require minor adjustment. While executing $\boldsymbol{\tau}^{\text{plan}}$, the operator can provide feedback by perturbing an arm of the robot in our system. However, since providing optimal feedback is challenging in practice, the feedback is often noisy and contains random fluctuations that are undesirable for the system. Therefore, instead of directly storing the trajectories obtained through physical HRI, we employ a procedure to smooth the trajectory from the human physical feedback presented in [31], which is inspired by CHOMP [90]. The objective of CHOMP is to obtain a trajectory that minimizes the cost function $\mathcal{C}(\boldsymbol{\tau})$. The update rule of CHOMP is given by

$$\boldsymbol{\tau}^{\text{new}} = \arg\min_{\boldsymbol{\tau}} \left\{ \mathcal{C}(\boldsymbol{\tau}^c) + \boldsymbol{g}^\top(\boldsymbol{\tau} - \boldsymbol{\tau}^c) + \frac{\eta}{2} \|\boldsymbol{\tau} - \boldsymbol{\tau}^c\|_M^2 \right\}$$

where $\boldsymbol{g} = \nabla\mathcal{C}(\boldsymbol{\tau})$, $\boldsymbol{\tau}^{\text{new}}$ is the updated plan of the trajectory, $\boldsymbol{\tau}^c$ is the current plan of the trajectory, η is a regularization constant, and $\|\boldsymbol{\tau}\|_M^2$ is the norm defined by a matrix M as $\|\boldsymbol{\tau}\|_M^2 = \boldsymbol{\tau}^\top M \boldsymbol{\tau}$. This update rule can be rewritten as

$$\boldsymbol{\tau}^{\text{new}} = \boldsymbol{\tau}^c - \frac{1}{\eta} M^\dagger \boldsymbol{g}, \tag{8.7}$$

where $M^\dagger$ is the pseudo-inverse of M. The matrix $M^\dagger$ in Eq. (8.7) plays a role in propagating the update to the entire trajectory smoothly in practice. Although M is not limited to a specific form, a popular implementation is to use

$$M = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 2 & -1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 2 \end{bmatrix}. \tag{8.8}$$

The update rule of CHOMP is interpreted as a variant of trust region optimization in which the trust region is defined by $\|\boldsymbol{\tau}\|_M^2$. The trust region optimization approach leads to a stable update of the trajectory since it regularizes the update based on the trust region and avoids a too big update. In practice, the trajectory indicated by the physical HRI is often not optimal even if it is better than the initially planned trajectory. Therefore, it is not preferable to totally rely on the trajectory obtained by physical HRI. Thus, the trust region trajectory update

Algorithm 4 Incremental Learning

```
  Input Dataset of demonstrations $\mathcal{D} = \{(\boldsymbol{\tau}_i, \boldsymbol{c}_i)\}_{i=1}^N$
  Learn policy $\pi_{\boldsymbol{\theta}}$ from $\mathcal{D}$                    ▷ Using Algorithm 2
  for Perform $\pi_{\boldsymbol{\theta}}$ for a new condition $\boldsymbol{c}'$ do
      Plan the trajectory $\boldsymbol{\tau}^{\text{plan}} \leftarrow \pi_{\boldsymbol{\theta}}$                    ▷ Using Algorithm 3
      Get $\boldsymbol{\tau}^{\text{exe}}$ by adding user modification while executing $\boldsymbol{\tau}^{\text{plan}}$
      Compute the difference $\Delta\boldsymbol{\tau} \leftarrow \boldsymbol{\tau}^{\text{exe}} - \boldsymbol{\tau}^{\text{plan}}$
      Smooth the trajectory $\boldsymbol{\tau}' \leftarrow \boldsymbol{\tau}^{\text{plan}} + \alpha M^{-1}\Delta\boldsymbol{\tau}$
      Increment the dataset $\mathcal{D} \leftarrow (\boldsymbol{\tau}', \boldsymbol{c}')$
```

like CHOMP is suitable in our framework since it will generate a trajectory that mixes the trajectory obtained by physical HRI and the initially planned one.

However, the gradient of the cost function $\boldsymbol{g}$ cannot be computed in our framework. Instead, we assume that $\boldsymbol{g}$ is indicated by the physical HRI. In particular, we record the trajectory $\boldsymbol{\tau}^{\text{exe}}$ executed through physical HRI, and compute the difference from the trajectory $\boldsymbol{\tau}^{\text{plan}}$ planned by $\pi_{\boldsymbol{\theta}}(\boldsymbol{\tau}|\boldsymbol{s})$. Subsequently, we apply the following rule to the trajectory obtained through human-robot physical interaction:

$$\boldsymbol{\tau}' = \boldsymbol{\tau}^{\text{plan}} + \alpha M^{-1}\Delta\boldsymbol{\tau}, \tag{8.9}$$

which can be obtained by replacing $\boldsymbol{g}$ in Eq. (8.7) with $\Delta\boldsymbol{\tau} = \boldsymbol{\tau}^{\text{exe}} - \boldsymbol{\tau}^{\text{plan}}$. The parameter α is determined empirically. We describe the effect of α in the results Section 8.4. We store the trajectory $\boldsymbol{\tau}'$ obtained by Eq. (8.9) to increment the trajectory dataset. The complete algorithm is given in Algorithm 4.

8.3 Dataset Preparation

Before preparing the dataset, we performed kinesthetic demonstrations with the robot controlled in gravity compensation mode. During the demonstration, one arm of the robot is used to puppet another arm. In this way, the teacher / demonstrator can easily control both the arms just by holding one arm while performing the dressing task on a human subject. Our dataset consists of pairs of robot trajectory $\boldsymbol{\tau}$ and a task condition vector $\boldsymbol{c}$. We are considering joint-space trajectory of the robot. The condition vector represents 3D positions of fingertips and elbows of both the arms of the human subject. Therefore, each condition vector contains 12 elements. Furthermore, the last joint of Baxter arm is ignored since it represents the rotation about the last axis. This rotation is unnecessary for our task. Since one arm of the robot is used to puppet another arm. This is why we consider the joint-space trajectory of only one arm of the robot. Hence, each trajectory contains 6 elements only. The robot trajectory comprises 200 time steps.

We collected 10 demonstrations beforehand. These demonstrations are used to train the model initially. The trained model is used to plan a new trajectory that we executed on the robot. In the beginning, corrections are given to the robot by physically moving the arm accordingly. The robot trajectory is smoothed as per the described algorithm in Section 8.2.2. The demonstration is added to the dataset so that it can be used later. Next, for a new task condition $\boldsymbol{c}'$, a new trajectory is planned and then executed by the robot. The human physical feedback is given at every demonstration. In this way, we repeat the experiment and perform 10 successive demonstrations.

8.4 Results

We prepared the dataset and train the model incrementally. We used 60 number of basis functions for each DOF of the robot, i.e., $B = 60$. Since we used puppet mode for collecting the initial dataset, we used the joint angles of one arm only in our model. Furthermore, the last joint represents the rotation along its axis, and it is not necessary for our task. Therefore, we ignored the last joint and modeled the first six joints of the robot. Fig. 8.2 shows a plot of initial and incremented trajectories of the robot used for incremental learning. This plot shows the trajectory of only one joint (joint name is left_w0) of the robot. It shows that both types of trajectories are following a similar trend and thus capable of performing the clothing assistance task properly.

We analyzed the distribution of task conditions used for incremental learning in the model. Fig. 8.3 illustrates the x-y projection of task conditions used in the

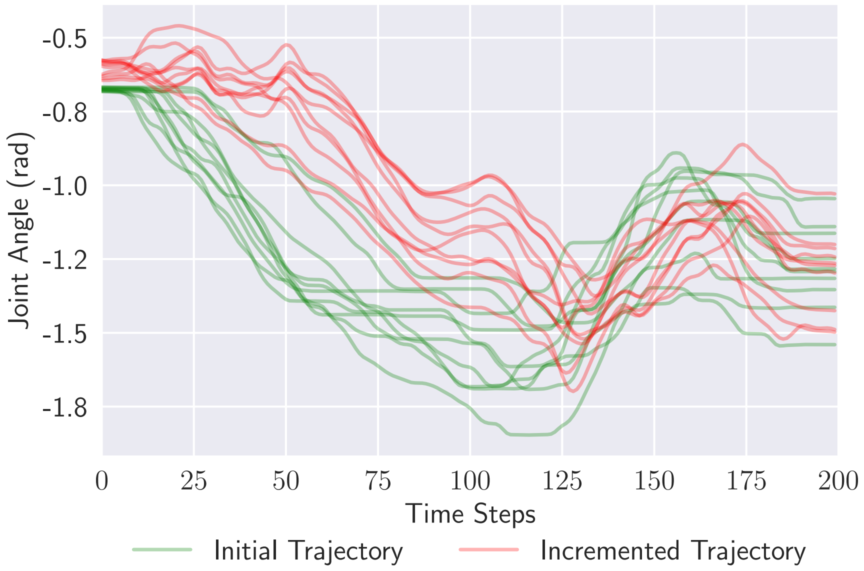


Figure 8.2: Initial and incremental trajectories of the robot in radians. To keep the plot clean and small, only one joint of the robot is shown. The joint name is left_w0.

model. It is clear from this figure that the initial demonstrations do not cover all necessary conditions.

In our model, the number of basis conditions, i.e., K is set to increase gradually. It is defined by the half of the number of training samples. However, the maximum allowed value for K is set to 15. It is done to avoid the overfitting of the data.

The parameter α plays an important role in smoothing the trajectory which contains some noise along with perturbations. We empirically run demonstration with various values of α shown in Fig. 8.4. A too-small value of α ignores the human physical feedback, whereas a too-large value of α accommodates noise as well. Therefore, it is very important to carefully choose the value of α. In our experiment, α is set to 0.0005.

We validated the performance of our model by calculating the amount of physical interaction performed, which is indicated by $\|\Delta\tau\|$. In Fig. 8.5, we have shown a plot of $\|\Delta\tau\|$ computed for one subject, i.e., subject 4. It shows that the interactions are higher in the beginning, whereas they decrease as more demonstrations are performed. Notice that the $\|\Delta\tau\|$ is not zero ever after performing

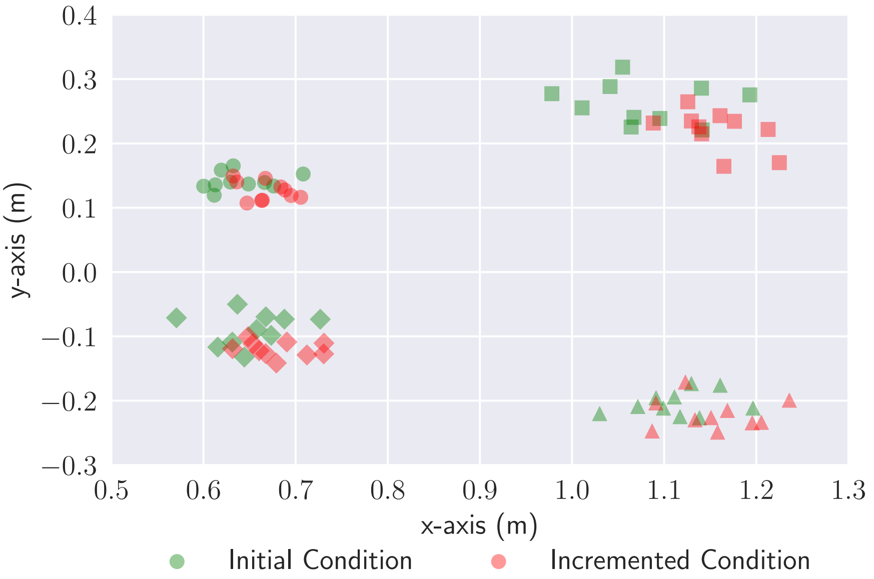


Figure 8.3: The x-y projection of initial and incremental task conditions in meters. The left fingertip, the left elbow, the right fingertip, and the right elbow is represented by a diamond, triangle, circle, and square shape respectively.

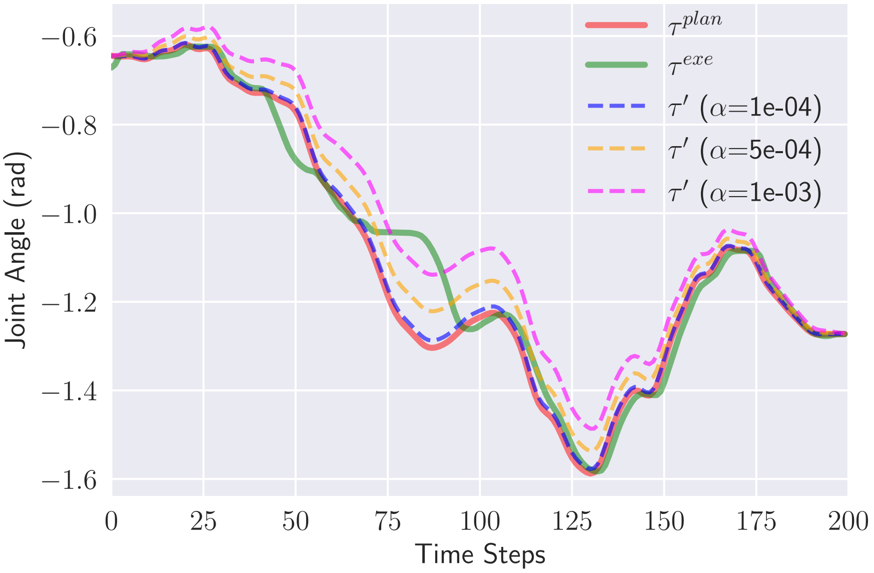


Figure 8.4: Effect of various values of α in a trajectory. The smoothed trajectories are displayed using the dashed line.

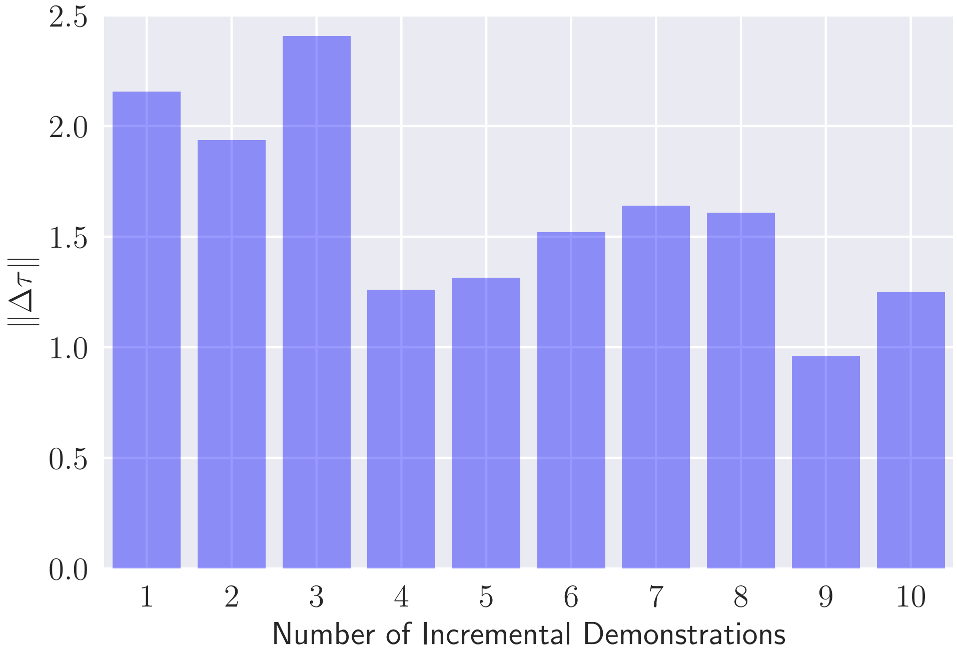


Figure 8.5: Amount of physical interaction $\|\Delta\tau\|$ required according to demonstrations. The amount of interaction is decreasing as more demonstrations are performed. Note that the decrement is non-monotonic due to the compliant nature of the robot arm.

10 demonstrations. Indeed, the $\Delta\tau$ cannot be zero even if the interaction is not required because the Baxter robot used in this research is a compliant robot. Therefore minor deviations from the commanded trajectory are expected in this robot which causes $\Delta\tau$ to stay non-zero. The compliant nature of the robot also causes $\|\Delta\tau\|$ to decrease non-monotonically.

We performed the clothing assistance task 10 times, each time our model learned incrementally. We noticed that the amount of physical interaction required to correct the trajectory is decreasing. Furthermore, we visually observed the improvement of the model, as well. The model required more interactions in the beginning, and after significant demonstrations, the required number of interactions were minimal. Fig. 8.6 shows the clothing assistance task at the various timestamp. At time $t = 10$ seconds, we noticed that the model had learned the behavior to move the robot's arm upward so that the subject can place his head

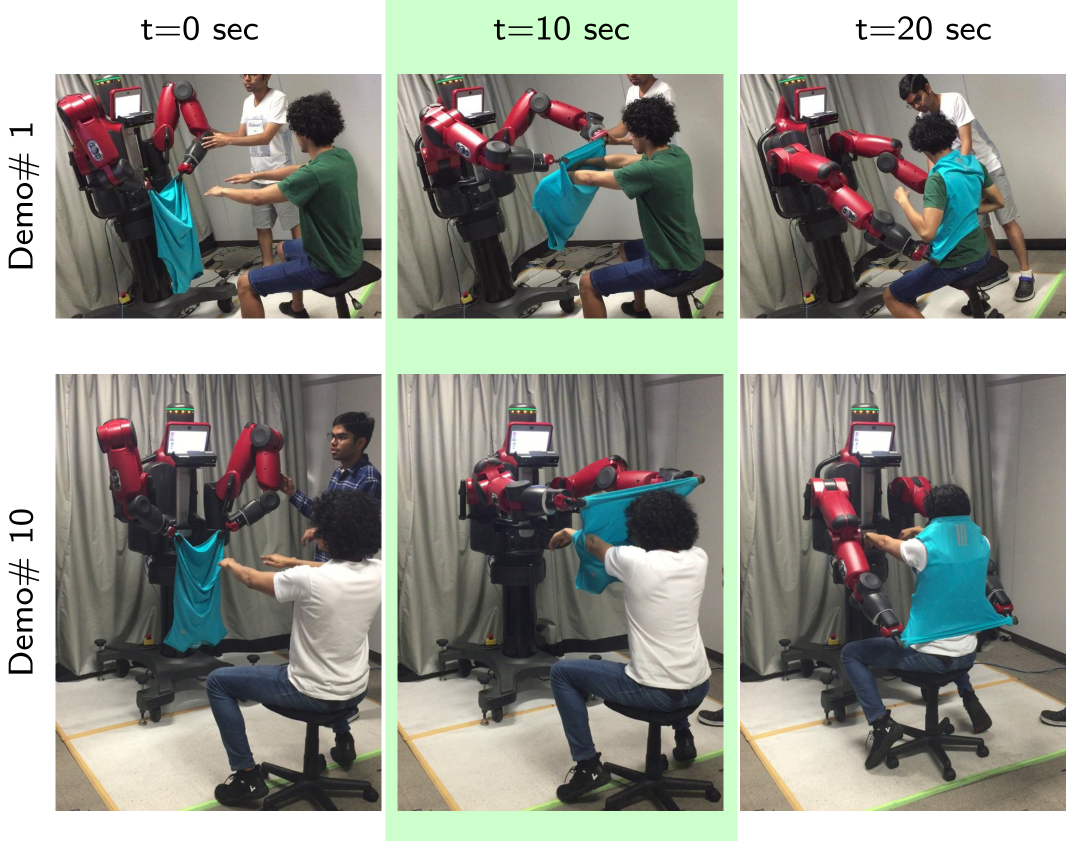


Figure 8.6: Pictures of the clothing assistance task at various timestamps. At time $t = 10$ seconds, we noticed that the model had learned the behavior to move the robot's arms upward so that the subject can place his head inside the shirt.

Table 8.1: Demographics information of all subjects

Parameter	Subject 1	Subject 2	Subject 3	Subject 4
Height (cm)	179	166	176	178
Age	26	32	23	27
Shoulder Width (cm)	49	47	47	45
Waist Size (cm)	81	90	86	83
Nationality	Filipino	Indian	Pakistani	Mexican

inside the shirt. Note that we do not include the position of the head in the task condition. Instead, the model learned the behavior from the demonstrations.

We performed trials on 3 subjects as shown in Fig. 8.7. The task took 20 seconds to finish. The demographics information of all subjects is given in Table 8.1.

8.5 Discussions

The initial and incremented task conditions used for incremental learning of the model shows that the initial demonstration could not cover all conditions. However, our model accommodates new demonstrations in the distribution. The model shows improvement in learning through human physical feedback. Thus our model can incorporate unforeseen scenarios by learning incrementally. Therefore the model can provide better generalization compared to previous studies [39, 43]. Furthermore, most of the previous studies of Robotic Clothing Assistance used a mannequin to perform experiments that cannot replicate more natural human arms movements [37].

We assume that the teacher / demonstrator behavior contains just one strategy, and we approximated the demonstrated trajectory distribution with the Gaussian distribution. In other words, the trajectory distribution contains only a single mode. For learning multiple types of behaviors, it will be necessary to employ a policy that can model multimodal distributions as in [91].

In this preliminary study, to define condition vector $\boldsymbol{c}$, we choose the 3D position of both elbows and fingertips of the arms of the subject. Theoretically, we

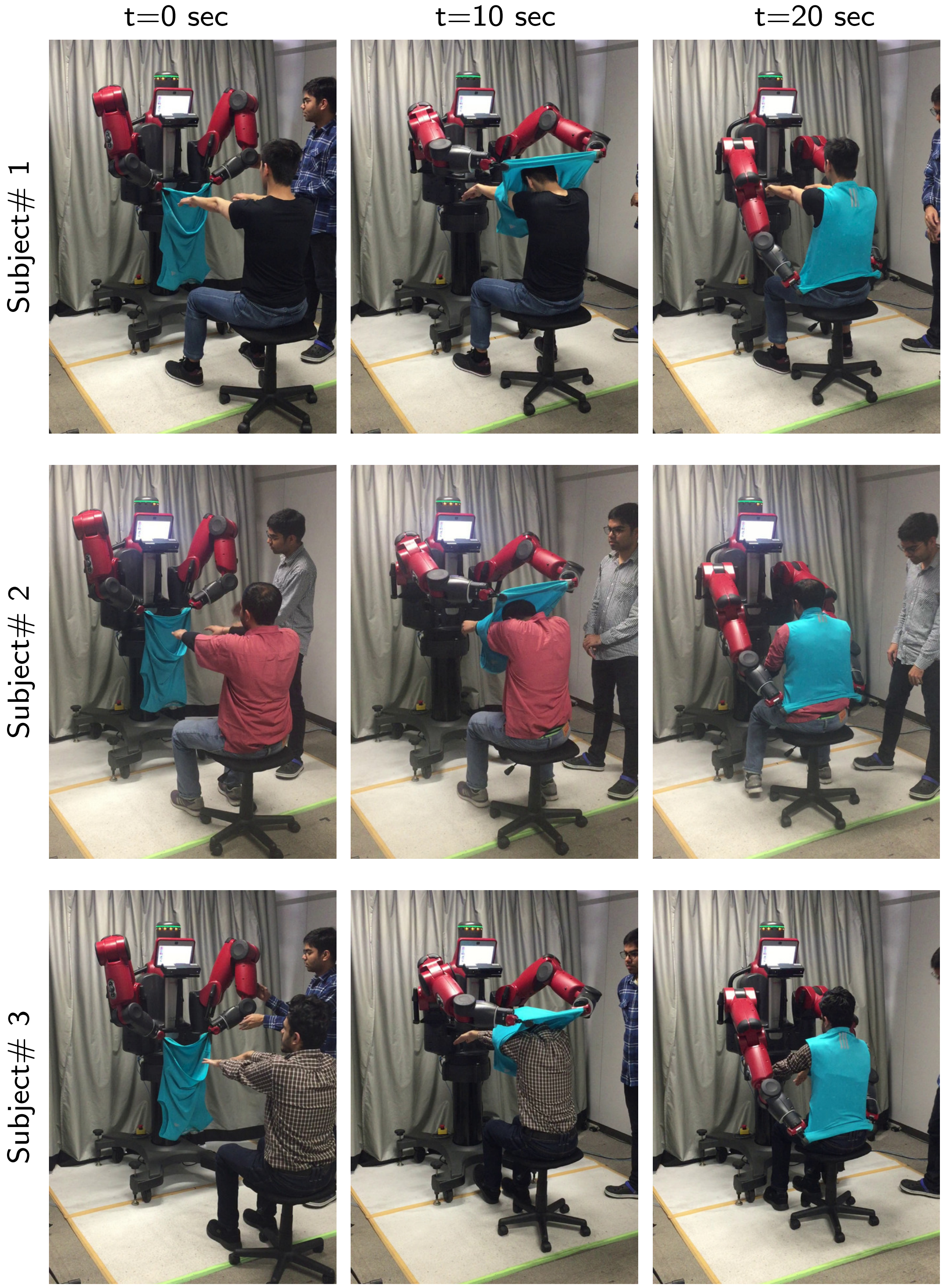


Figure 8.7: Sequential snapshots of the clothing assistance task performed on 3 subjects

can include the position of all upper body parts. However then the model will require more data to obtain satisfactory generalization ability.

In this work, we focused on the dressing of the sleeveless T-shirt. The algorithm can be applied to dress other types of clothes, such as pajamas and hospital gown. The required initial demonstrations must be collected accordingly. However, the proposed algorithm does not deal with the regrasping of garments.

9 Conclusion

Robotic Clothing Assistance can contribute to improving the quality of life of the elderly substantially and significantly reduce the burden on caregivers. Due to the inherent properties of clothes such as non-rigidity and flexibility, cloth manipulation is still limited to much simpler applications such as cloth folding tasks. Due to close cloth interaction with the human during dressing, Robotic Clothing Assistance is still a challenging and open problem. We have presented a framework for Robotic Clothing Assistance consisting of DMP and BGPLVM. We have shown that by dividing the entire trajectory into multiple parts, generalize using DMP and BGPLVM; Robotic Clothing Assistance can be performed. Our framework is trained by imitation learning from a human demonstration on a single demonstration which intuitively makes it data-efficient. It has been developed focusing on assisting the elderly hence expects minimal cooperation from the human subject concerning body movement. The robot was programmed to move slowly compared to its default speed so that any collision has minimal inertia and impact.

9.1 Discussion

The Robotic Clothing Assistance task deals with the manipulation of the clothing article, which contains complex dynamics due to the inherent non-rigid and flexible nature of clothes. Not to mention that a robot needs to perform the task close to humans, which further makes it complicated. In our experiment, Baxter's left and right arm trajectory depend on the posture of the person being assisted. During the arm dressing phase, it usually mirrors each other due to the symmetry of the posture of the subject's arms. In principle, we have employed separate DMPs for each arm. In the subsequent sections, we describe some of the problems faced in the experiment which can be considered as failure scenarios and public demonstration of the experiment at an international level robot exhibition followed by limitations of our current work.

9.1.1 Failure Scenarios

The task of clothing assistance inherits complex dynamics due to the presence of flexible clothing articles along with the human subject being assisted. Hence there can be various failure scenarios. At present, official API's of Kinect v2 depth sensor is being used for human pose estimation. The pose estimation is not robust to this environment as it fails during occlusion with Baxter arm and other objects such as clothing article. Another significant failure is caused by the clothing article. During the task, the clothing article undergoes severe deformations, and the shape of the clothing article keeps on changing, which affects the task settings. Therefore, a trajectory that was able to perform the task successfully once isn't guaranteed to work always. We noticed that there could be many other failure scenarios such as clothing article getting stuck to the fingers, clothing article not reaching up to the torso and clothing article getting slipped by end-effectors. Various failure scenarios are shown in Fig. 9.1.

9.1.2 Public Demonstrations

To examine the performance of the proposed framework, we demonstrated it to the public in iREX (International Robot Exhibition) 2017 held in Tokyo (Japan). People from academia, industry, healthcare and government officials visited, and saw the demonstration. We requested them to be a subject, and then we performed a clothing assistance task. After completing the task, we had conversations with them during that we asked them to provide feedback about their experience by filling a paper-based questionnaire. Unfortunately, due to the immense visitors during the exhibition, it was difficult to take feedback from each participant. Nevertheless, we managed to collect feedback from 35 participants. The feedback was mostly positive and encouraging. The comprehensive data is shown in Fig. 9.2.

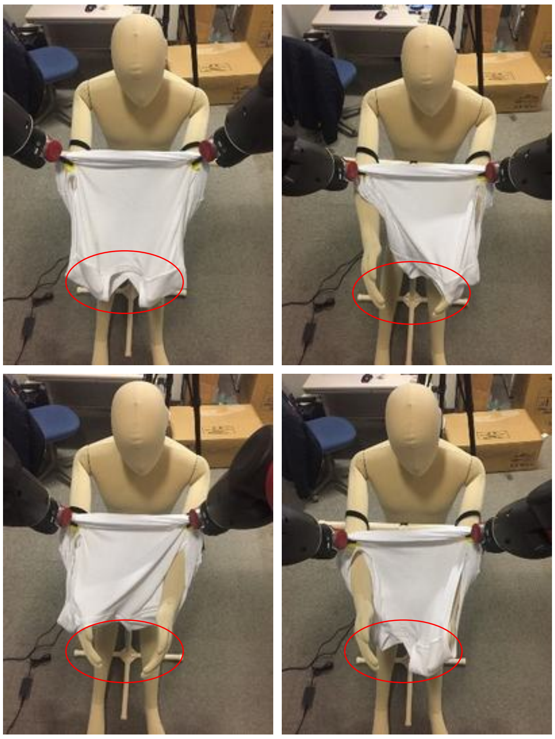


Figure 9.1: Various failure scenarios showing how a clothing article is difficult to manipulate because of inherent non-rigid and flexible nature.

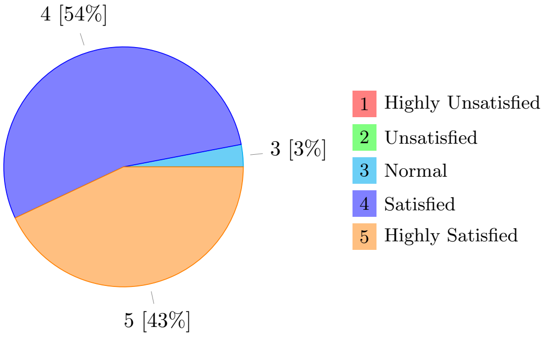


Figure 9.2: Feedback received from 35 participants during the public demonstration at iREX 2017. Most of the subjects were found excited and satisfied.

9.1.3 Limitations

At present, there are certain limitations of our work as listed below.

- The experiment is performed on healthy adults. Although we do have demonstrated this system to a few elderly at various exhibitions, a complete evaluation is yet to be done.

- During the arm dressing phase, arms are constrained due to the shirt over them. In this situation, the arms cannot be moved beyond a limited range. Hence, both the arms are restricted to be parallel even though separate DMPs are employed for both arms.

- Kinect v2 Sensor using Windows SDK is employed for human body recognition and tracking. All 26 joints defined by Kinect v2 are tracked. However, our algorithm uses only fingertips and elbow positions for both arms of the subject.

- The arm dressing phase is initialized by fingertips and elbow positions. Although, as per DMP formulation, it is possible to keep updating the fingertips and elbow positions during the dressing. Unfortunately, it is difficult to track these joints during the dressing due to the occlusion from cloth and robot arm. Hence, during the arm dressing phase, human arms are assumed to be stationary.

- During body dressing phase, due to the limited Baxter's workspace, the subject is requested to move towards the robot while sitting on the chair. This movement is difficult to achieve by using a mannequin. Hence, even though we performed the demonstration on a mannequin but we couldn't conduct the task on a mannequin.

- We have shown the dressing of a sleeveless T-shirt. However, our approach can be extended to other clothing articles such as pants, jackets and so on. Pant dressing can be incorporated from arm dressing. For each type of clothing articles, separate demonstration and training should be essential. However, in this work, our experiment is limited to the dressing of a sleeveless T-shirt.

9.1.4 Safety of the Subject

In this research, the safety of the subject is an essential requirement. Therefore, to ensure the safety of the subject, various measures are taken. These measures not only includes our own considerations but also considerations in the robot designing as enumerated below.

1. The latent space has learned a consistent set of motor skills. The manifold generated from this latent space exploration is found to be constrained. The constrained manifold assures that no trajectory other than what is required for the dressing can be produced. Thus the restriction to perform only the dressing task enables safe human-robot interaction.

2. The robot was programmed to move 50% slower than its default speed so that any collision has minimal inertia, thus reducing the impact.

3. The setup of the task contains fingertips made from a soft material, as shown in Fig. 4.2. These fingertips provide sufficient traction to hold the cloth firmly. Furthermore, all edges of the fingertips are rounded to avoid accidental scratch by the robot arm's touch during the dressing task.

4. Baxter robot is designed to work collaboratively with humans. It uses SEAs to provide passive compliance. All the joints are fully backdrivable.

5. Baxter meets the following safety standards- ISO 10218-1:2006, ISO 10218-2, ANSI RIA R15.06-2012, UL 60950-1. More safety information about Baxter is given in Section 4.1.1.

.

9.2 Future Work

In this thesis, we have presented a framework for assistance in dressing using imitation learning. In the subsequent paragraphs, we describe the future works of this research.

Although this thesis has focused only on sleeveless T-shirt dressing, our framework could enable robotic assistance in other dressing tasks such as the jacket, hospital gown, pajama. Intuitively, pajama dressing can be considered as a variation of the arm dressing phase. In the arm dressing phase, both the arm goes inside the shirt opening, whereas in pajama dressing, both the legs should go inside the pajama opening. Similarly, hospital gown dressing can be performed. For each type of clothing article, separate demonstrations and training are required. Meanwhile, it is crucial to identify the grasping points in each clothing article. In this thesis, these grasping points were determined empirically by observing the picking points of a human demonstrator while he was performing the dressing task.

The latent space generated by BGPLVM is constrained in task space and provides safe interaction with the human subject. At present, the latent space is generated from a single demonstration. Hence it suffers from limited generalizability. However, in the future, this latent space can be redesigned by providing multiple demonstration trajectories to accumulate the variability in human body shapes.

The framework provides a proper initialization for applying reinforcement learning. The generated latent space can be treated as a search space of a learning agent for policy search so that the robot can adapt to various failure scenarios in real-time due to constrained two-dimensional search space.

The proposed framework does not depend on the robot, specifically. Therefore it can easily be transferred to any dual-arm robot. Baxter robot has a narrow workspace. That is why a wheeled chair was utilized. It should be noted that a larger workspace or mobile dual-arm robot should be preferred to perform the task.

The robot can communicate with the user via visual representation using its LED screen. However, it lacks the speaking ability. We think that adding this ability may improve its chance of social acceptance.

Cloth manipulation requires dexterous skills. However, at present, two-finger grippers are used. This can be one of the future works to facilitate human-like cloth manipulation by using multiple fingers.

We would also like to perform detailed user evaluation in particular for acceptance by the elderly. We want to demonstrate this system to the elderly not only to examine it but also to understand their experience in connection with psychology.

9.3 Open Questions

In this thesis, we have tackled the problem of Robotic Clothing Assistance using a complaint dua-arm robot. Although the results verify the plausibility of our approach. There are several open questions as mentioned below-

Q1 How to define the state of a clothing article when it is entangled with a human?

Researchers have used point cloud or depth images of the clothing article to tackle specific problems. Thus, the scope of these works is still limited.

Q2 How to pick a clothing article to dress using a dual-arm robot?

Clothes are difficult to manipulate due to their non-rigid behavior. We humans always take advantage of our fingers by applying dexterous manipulation. Unfortunately, it is still a challenging problem to be solved by robots.

Q3 How to regrasp a clothing article in real-time?

Researchers in the past propose various regrapsing strategies. For example, they hang the cloth from one point and rotate. They take multiple images to predict a regrasping location. The process is slow, thus can not be used in Robotic Clothing Assistance.

Q4 How to incorporate other cloth types in a single model?

Previously, a collection of features is stored in a database that contains many movement primitives. The trajectory planner picks a suitable movement primitive and executes it.

Q5 How to estimate human pose reliably in the settings of Robotic Clothing Assistance?

Human pose estimation has been done for decades in the computer vision community. However, in the settings of Robotic Clothing Assistance, it becomes very complex due to severe occlusion from the robot arm and the clothing article.

Q6 How to perform a simulation of Robotic Clothing Assistance?

The task of Robotic Clothing Assistance is complicated. This is why, instead of applying traditional mathematical modeling, we used imitation learning to avoid the complexity of modeling. Unfortunately, the simulation can not be implemented by the same approach.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Tomohiro Shibata for the continuous support of my study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. His words of encouragement have inspired me to strive harder and to never give up when faced with difficulties. I could not have imagined having a better advisor for my study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Kazuo Ishii, Associate Prof. Hiroaki Wagatsuma and Associate Prof. Kazuto Takashima not only for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

My sincere gratitude to Associate Prof. Takayuki OSA, for his generous support in the latter part of this research related to incremental learning. I must say that the technical discussions and explanations were eye-opening and wonderful. I learned new ideas and enjoyed working under his supervision.

This research work was supported in part by the Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science (No. 16H01749).

I am very fortunate to have worked with wonderful people in the Shibata laboratory. I thank my fellow laboratory members for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had. I would like to specially thank my senior Nishanth Koganti for continuous guidance, laboratory member Tsuyoshi Shibata and lab secretory Ohta San and Iwashita San. Without these precious support it would not be possible to conduct this research.

Nevertheless, I would like to thank my family: my parents and to my brother for supporting me spiritually throughout my study and my life in general.

Finally, I would like to thank the Japan Government Scholarship Program (MEXT) for the support in my education and stay in Japan. Without the Japan Government Scholarship, my dream of studying in Japan would never have came true. Thanks a lot for this valuable opportunity.

References

[1] JR Allen, A Karchak, VL Nickel, and R Snelson. "The Rancho electric arm". In: *The 3rd Annual Rocky Mountain Bioengineering Symposium.* 1966, pp. 79–82.

[2] Holly A Yanco. "Wheelesley: A Robotic Wheelchair System: Indoor Navigation and User Interface". In: *Assistive Technology and Artificial Intelligence: Applications in Robotics, User Interfaces and Natural Language Processing.* Springer, 1998, pp. 256–268. DOI: `10.1007/BFb0055983`.

[3] Toshiharu Mukai, Shinya Hirano, Hiromichi Nakashima, Yo Kato, Yuki Sakaida, Shijie Guo, and Shigeyuki Hosoe. "Development of a Nursing-Care Assistant Robot RIBA That Can Lift a Human in Its Arms". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE. 2010, pp. 5996–6001. DOI: `10.1109/IROS.2010.5651735`.

[4] The Japan Times. *SoftBank unveils 'historic' robot.* Accessed: 2020-05-20. 2014. URL: `https://www.japantimes.co.jp/news/2014/06/05/business/corporate-business/softbank-unveils-pepper-worlds-first-robot-reads-emotions`.

[5] World Health Organization. *Facts about ageing.* [Online] Accessed: 2020-05-20. 2014. URL: `http://www.who.int/ageing/about/facts/en`.

[6] World Health Organization. *Long-term trends in fetal mortality: implications for developing countries.* [Online] Accessed: 2020-05-20. 2015. URL: `http://www.who.int/bulletin/volumes/86/6/07-043471/en`.

[7] Wikipedia. *Activities of Daily Living.* [Online] Accessed: 2020-01-30. 2020. URL: `https://en.wikipedia.org/wiki/Activities_of_daily_living`.

[8] Wan He, Daniel Goodkind, and Paul Kowal. *International Population Reports, An Aging World: 2015.* [Online] Accessed: 2020-05-20. 2016. URL: `https://www.census.gov/content/dam/Census/library/publications/2016/demo/p95-16-1.pdf`.

[9] European Commission. *EU-funded projects in ICT for Ageing Well.* [Online] Accessed: 2020-05-20. 2016. URL: `https://ec.europa.eu/digital-single-market/en/news/overview-eu-funded-running-projects-area-ict-ageing-well`.

[10] Brian J Dudgeon, Jeanne M Hoffman, Marcia A Ciol, Anne Shumway-Cook, Kathryn M Yorkston, and Leighton Chan. "Managing Activity Difficulties at Home: A Survey of Medicare Beneficiaries". In: *Archives of Physical Medicine and Rehabilitation* 89.7 (2008), pp. 1256–1261. DOI: `10.1016/j.apmr.2007.11.038`.

[11] The Japan Times. *Nursing care workers hard to find but in demand in aging Japan.* [Online] Accessed: 2020-03-28. 2016. URL: `https://www.japantimes.co.jp/news/2016/06/27/reference/nursing-care-workers-hard-to-find-but-in-demand-in-aging-japan`.

[12] Center for Disease Control and Prevention. *CDC HRQOL–14. Activity Limitations Module.* [Online] Accessed: 2020-06-01. 2020. URL: `https://www.stratasys.com/materials/search/agilus30`.

[13] EuroQoL Foundation. *EQ-5D-5L User Guide.* [Online] Accessed: 2020-06-02. 2020. URL: `https://euroqol.org/wp-content/uploads/2019/09/EQ-5D-5L-English-User-Guide_version-3.0-Sept-2019-secured.pdf`.

[14] Pol Monsó, Guillem Alenyà, and Carme Torras. "POMDP Approach to Robotized Clothes Separation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE. 2012, pp. 1324–1329. DOI: `10.1109/IROS.2012.6386011`.

[15] Kimitoshi Yamazaki, Ryosuke Oya, Kotaro Nagahama, and Masayuki Inaba. "A Method of State Recognition of Dressing Clothes Based on Dynamic State Matching". In: *IEEE/SICE International Symposium on System Integration (SII).* IEEE. 2013, pp. 406–411. DOI: `10.1109/SII.2013.6776728`.

[16] Yuji Yamakawa, Akio Namiki, and Masatoshi Ishikawa. "Dynamic Manipulation of a Cloth by High-speed Robot System using High-speed Visual Feedback". In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 8076–8081. ISSN: 1474-6670. DOI: `10.3182/20110828-6-IT-1002.00596`. URL: `http://www.sciencedirect.com/science/article/pii/S1474667016449074`.

[17] Yasuyo Kita, Fumio Kanehiro, Toshio Ueshiba, and Nobuyuki Kita. "Clothes handling based on recognition by strategic observation". In: *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE. 2011, pp. 53–58. DOI: `10.1109/Humanoids.2011.6100817`.

[18] Andreas Doumanoglou, Jan Stria, Georgia Peleka, Ioannis Mariolis, Vladimir Petrik, Andreas Kargakos, Libor Wagner, Václav Hlaváč, Tae-Kyun Kim, and Sotiris Malassiotis. "Folding Clothes Autonomously: A Complete Pipeline". In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1461–1478. DOI: `10.1109/TRO.2016.2602376`.

[19] Bryan Willimon, Stan Birchfield, and Ian Walker. "Classification of Clothing using Interactive Perception". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2011, pp. 1862–1868. DOI: `10.1109/ICRA.2011.5980336`.

[20] Yinxiao Li, Yonghao Yue, Danfei Xu, Eitan Grinspun, and Peter K Allen. "Folding Deformable Objects using Predictive Simulation and Trajectory Optimization". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 6000–6006. DOI: `10.1109/IROS.2015.7354231`.

[21] Marco Cusumano-Towner, Arjun Singh, Stephen Miller, James F O'Brien, and Pieter Abbeel. "Bringing Clothing into Desired Configurations with Limited Perception". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2011, pp. 3893–3900. DOI: `10.1109/ICRA.2011.5980327`.

[22] Stephen Miller, Mario Fritz, Trevor Darrell, and Pieter Abbeel. "Parametrized Shape Models for Clothing". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2011, pp. 4861–4868. DOI: `10.1109/ICRA.2011.5980453`.

[23] Eiichi Ono, Hidehiko Okabe, Hitoshi Akami, and Noboru Aisaka. "Robot Hand with a Sensor for Cloth Handling". In: *Journal of the Textile Machinery Society of Japan* 37.1 (1991), pp. 14–24. DOI: `10.4188/jte1955.37.14`.

[24] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. "An Algorithmic Perspective on Imitation Learning". In: *Foundations and Trends® in Robotics* 7.1-2 (2018), pp. 1–179. DOI: `10.1561/2300000053`.

[25] Stefan Schaal. "Dynamic Movement Primitives - A Framework for Motor Control in Humans and Humanoid Robotics". In: *Adaptive Motion of Animals and Machines*. Springer, 2006, pp. 261–280. ISBN: 978-4-431-31381-6. DOI: `10.1007/4-431-31381-8_23`.

[26] A. J. Ijspeert, J. Nakanishi, and S. Schaal. "Learning Attractor Landscapes for Learning Motor Primitives". In: *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2002, pp. 1547–1554. URL: `http://papers.nips.cc/paper/2140-learning-attractor-landscapes-for-learning-motor-primitives.pdf`.

[27] G. Maeda, M. Ewarton, T. Osa, B. Busch, and J. Peters. "Active Incremental Learning of Robot Movement Primitives". In: *Annual Conference on Robot Learning (CoRL)*. Vol. 78. PMLR, 2017, pp. 37–46. URL: `http://proceedings.mlr.press/v78/maeda17a.html`.

[28] A. Paraschos, C. Daniel, J. Peters, and G Neumann. "Probabilistic Movement Primitives". In: *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2013, pp. 2616–2624. URL: `http://papers.nips.cc/paper/5177-probabilistic-movement-primitives.pdf`.

[29] M. Ewerton, G. Maeda, G. Neumann, V. Kisner, G. Kollegger, J. Wiemeyer, and J. Peters. "Incremental Imitation Learning of Context-Dependent Motor Skills". In: *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. 2016, pp. 351–358. DOI: `10.1109/Humanoids.2016.7803300`.

[30] F. B. Farraj, T. Osa, N. Pedemonte, J. Peters, G. Neumann, and P. R. Giordano. "A Learning-based Shared Control Architecture for Interactive Task Execution". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 329–335. DOI: `10.1109/ICRA.2017.7989042`.

[31] Andrea Bajcsy, Dylan P Losey, Marcia K O'Malley, and Anca D Dragan. "Learning Robot Objectives from Physical Human Interaction". In: Proceedings of Machine Learning Research 78 (2017), pp. 217–226. URL: `http://proceedings.mlr.press/v78/bajcsy17a.html`.

[32] A. Bajcsy, D. Losey, M. O'Malley, and A.D. Dragan. "Learning from Physical Human Corrections, One Feature at a Time". In: *ACM/IEEE International Confernece on Human-Robot Interaction (HRI)*. 2018, pp. 141–149. DOI: `10.1145/3171221.3171267`.

[33] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. "Playing Atari with Deep Reinforcement Learning". In: *CoRR* abs/1312.5602 (2013). URL: `http://arxiv.org/abs/1312.5602`.

[34] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. "Human-level Control Through Deep Reinforcement Learning". In: *Nature* 518.7540 (2015), pp. 529–533. DOI: `10.1038/nature14236`.

[35] Jonathan Ho and Stefano Ermon. "Generative Adversarial Imitation Learning". In: *Advances in Neural Information Processing Systems (NIPS)*. Curran Associates, Inc., 2016, pp. 4565–4573. URL: `http://papers.nips.cc/paper/6391-generative-adversarial-imitation-learning.pdf`.

[36] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. *OpenAI Gym*. 2016. URL: `http://arxiv.org/abs/1606.01540`.

[37] Tomoya Tamei, Takamitsu Matsubara, Akshara Rai, and Tomohiro Shibata. "Reinforcement Learning of Clothing Assistance with A Dual-arm Robot". In: *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE. 2011, pp. 733–738. DOI: `10.1109/Humanoids.2011.6100915`.

[38] Adria Colomé, Antoni Planells, and Carme Torras. "A Friction-model-based Framework for Reinforcement Learning of Robotic Tasks in Non-rigid Environments". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 5649–5654. DOI: 10.1109/ICRA.2015.7139990.

[39] Nishanth Koganti, Tomoya Tamei, Kazushi Ikeda, and Tomohiro Shibata. "Bayesian Nonparametric Learning of Cloth Models for Real-Time State Estimation". In: *IEEE Transactions on Robotics* 33.4 (2017), pp. 916–931. DOI: 10.1109/TRO.2017.2691721.

[40] Zackory Erickson, Maggie Collier, Ariel Kapusta, and Charles C Kemp. "Tracking Human Pose During Robot-Assisted Dressing using Single-Axis Capacitive Proximity Sensing". In: *IEEE Robotics and Automation Letters* 3.3 (July 2018), pp. 2245–2252. ISSN: 2377-3766. DOI: 10.1109/LRA.2018.2812912.

[41] Aleksandar Jevtić, Andrés Flores Valle, Guillem Alenyà, Greg Chance, Praminda Caleb-Solly, Sanja Dogramadzi, and Carme Torras. "Personalized Robot Assistant for Support in Dressing". In: *IEEE Transactions on Cognitive and Developmental Systems* (2019), pp. 1–1. ISSN: 2379-8920. DOI: 10.1109/TCDS.2018.2817283.

[42] Fan Zhang, Antoine Cully, and Yiannis Demiris. "Probabilistic Real-Time User Posture Tracking for Personalized Robot-Assisted Dressing". In: *IEEE Transactions on Robotics* 35.4 (2019), pp. 873–888. DOI: 10.1109/TRO.2019.2904461.

[43] Nishanth Koganti, Tomohiro Shibata, Tomoya Tamei, and Kazushi Ikeda. "Data-efficient Learning of Robotic Clothing Assistance using Bayesian Gaussian Process Latent Variable Model". In: *Advanced Robotics* 33.15-16 (2019), pp. 800–814. DOI: 10.1080/01691864.2019.1610061.

[44] Alexander Clegg, Zackory Erickson, Patrick Grady, Greg Turk, Charles C Kemp, and C Karen Liu. "Learning to Collaborate From Simulation for Robot-Assisted Dressing". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 2746–2753. DOI: 10.1109/LRA.2020.2972852.

[45] Alexander Clegg, Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. “Learning to Dress: Synthesizing Human Dressing Motion via Deep Reinforcement learning”. In: *ACM Transactions on Graphics (TOG)* 37.6 (2018), pp. 1–10. DOI: 10.1145/3272127.3275048.

[46] Andreas Damianou, Carl Ek, Michalis Titsias, and Neil Lawrence. “Manifold Relevance Determination”. In: *International Conference on Machine Learning (ICML)*. ICML ’12. Edinburgh, Scotland, GB: Omnipress, July 2012, pp. 145–152. ISBN: 978-1-4503-1285-1.

[47] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. “Learning Control Policies For Movement Imitation and Movement recognition”. In: *Neural Information Processing Systems (NIPS)*. Vol. 15. 2003, pp. 1547–1554.

[48] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. “Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 2. IEEE. 2002, pp. 1398–1403. DOI: 10.1109/ROBOT.2002.1014739.

[49] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. “Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors”. In: *Neural Computation* 25.2 (2013), pp. 328–373. DOI: 10.1162/NECO_a_00393.

[50] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. “Learning and Generalization of Motor Skills by Learning from Demonstration”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2009, pp. 763–768. DOI: 10.1109/ROBOT.2009.5152385.

[51] Sethu Vijayakumar and Stefan Schaal. “Locally Weighted Projection Regression : An O(n) Algorithm for Incremental Real Time Learning in High Dimensional Space”. In: *International Conference on Machine Learning (ICML)*. Vol. 1. 2000, pp. 1079–1086.

[52] Neil Lawrence. “Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models”. In: *Journal of Machine Learning Research* 6 (Nov. 2005), pp. 1783–1816. ISSN: 1532-4435.

[53] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. en. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, Jan. 2006, p. 248. ISBN: 0-262-18253-X.

[54] Radford M. Neal. *Bayesian Learning for Neural Networks*. Vol. 118. Lecture Notes in Statistics. Springer Science & Business Media, 2012. ISBN: 9781461207450. URL: `https://books.google.co.jp/books?id=LHHrBwAAQBAJ`.

[55] Michalis Titsias and Neil D. Lawrence. "Bayesian Gaussian Process Latent Variable Model". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, 2010, pp. 844–851. URL: `http://proceedings.mlr.press/v9/titsias10a.html`.

[56] Neil D. Lawrence. "Learning for Larger Datasets with the Gaussian Process Latent Variable Model". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, 2007, pp. 243–250. URL: `http://proceedings.mlr.press/v2/lawrence07a.html`.

[57] Michalis Titsias. "Variational Learning of Inducing Variables in Sparse Gaussian Processes". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2009, pp. 567–574. URL: `http://proceedings.mlr.press/v5/titsias09a.html`.

[58] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. "ROS: An Open-source Robot Operating System". In: *ICRA Workshop on Open Source Software*. Vol. 3. 3.2. Kobe, Japan, 2009, p. 5.

[59] Rethink Robotics. *Baxter Research Robot SDK*. Accessed: 2020-04-20. 2012. URL: `https://github.com/RethinkRobotics/baxter`.

[60] Matthew M Williamson. *Series Elastic Actuators*. Tech. rep. Artificial Intelligence Laboratory, Massachusetts Institute Of Technology, 1995.

[61] Gill A Pratt and Matthew M Williamson. "Series Elastic Actuators". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Human Robot Interaction and Cooperative Robots. Vol. 1. IEEE. 1995, pp. 399–406. DOI: 10.1109/IROS.1995.525827.

[62] Nihon Binary. *Safety Compliance of the Baxter*. [Online] Accessed: 2020-06-01. 2008. URL: http://www.nihonbinary.co.jp/Products/Robot/baxter_reserch.html.

[63] Rethink Robotics. *Safety Compliance of the Baxter*. [Online] Accessed: 2020-06-01. 2008. URL: https://www.active8robots.com/wp-content/uploads/Baxter-and-Collaborative-Robot-Safety-Standards.pdf.

[64] Rethink Robotics. *Safety Compliance of the Baxter*. [Online] Accessed: 2020-06-01. 2008. URL: https://www.active8robots.com/wp-content/uploads/Baxter-Safety-and-Compliance-Overview.pdf.

[65] Stratasys Ltd. *Agilus30*. [Online] Accessed: 2020-07-14. 2020. URL: https://www.stratasys.com/materials/search/agilus30.

[66] *Microsoft Kinect Version 2 for Windows*. Accessed: 2020-04-20. 2014. URL: https://developer.microsoft.com/en-us/windows/kinect.

[67] Mark Draelos, Qiang Qiu, Alex Bronstein, and Guillermo Sapiro. "Intel RealSense = Real Low Cost Gaze". In: *IEEE International Conference on Image Processing (ICIP)*. IEEE. 2015, pp. 2520–2524. DOI: 10.1109/ICIP.2015.7351256.

[68] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. "Intel RealSense Stereoscopic Depth Cameras". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2017, pp. 1–10. DOI: 10.1109/CVPRW.2017.167.

[69] iPi Soft. *Depth Sensors Comparison*. [Online] Accessed: 2020-06-01. 2019. URL: http://docs.ipisoft.com/Depth_Sensors_Comparison.

[70] Rethink Robotics. *Safety Compliance*. [Online] Accessed: 2018-03-26. 2008. URL: http://www.rethinkrobotics.com/safety-compliance.

[71] Pieter Hintjens. *ZeroMQ: Messaging for Many Applications*. Oreilly and Associate Series. O'Reilly Media, Incorporated, 2013. ISBN: 9781449334062. URL: https://books.google.co.jp/books?id=KWtT5CJc6FYC.

[72] Shinji Umeyama. "Least-Squares Estimation of Transformation Parameters Between Two Point Patterns". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 13.4 (1991), pp. 376–380. DOI: `10.1109/34.88573`.

[73] Martin A Fischler and Robert C Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Communications of the ACM* 24.6 (1981), pp. 381–395. ISSN: 0001-0782. DOI: `10.1145/358669.358692`.

[74] Wikipedia. *HSL and HSV*. [Online] Accessed: 2020-04-20. 2020. URL: `https://en.wikipedia.org/wiki/HSL_and_HSV`.

[75] Radu Bogdan Rusu and Steve Cousins. "3d is here: Point cloud library (PCL)". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2011, pp. 1–4. DOI: `10.1109/ICRA.2011.5980567`.

[76] Lingzhu Xiang, Florian Echtler, Christian Kerl, Thiemo Wiedemeyer, Lars, hanyazou, Ryan Gordon, Francisco Facioni, laborer2008, Rich Wareham, Matthias Goldhoorn, alberth, gaborpapp, Steffen Fuchs, jmtatsch, Joshua Blake, Federico, Henning Jungkurth, Yuan Mingze, vinouz, Dave Coleman, Brendan Burns, Rahul Rawat, Serguei Mokhov, Paul Reynolds, P.E. Viau, Matthieu Fraissinet-Tachet, Ludique, James Billingham, and Alistair. *libfreenect2: Release 0.2*. 2016. DOI: `10.5281/zenodo.50641`.

[77] Thiemo Wiedemeyer. *IAI Kinect2*. Accessed June 12, 2015. University Bremen: Institute for Artificial Intelligence, 2014. URL: `https://github.com/code-iai/iai_kinect2`.

[78] Robot Operating System (ROS). *RViz*. [Online] Accessed: 2020-05-31. 2012. URL: `http://wiki.ros.org/rviz`.

[79] Rethink Robotics. *Baxter Hardware Specifications*. [Online] Accessed: 2020-04-20. 2015. URL: `http://sdk.rethinkrobotics.com/wiki/Hardware_Specifications`.

[80] Kevin S Chen. "Application of the ISO 9283 standard to test repeatability of the Baxter robot". PhD thesis. University of Illinois, 2015. URL: `http://hdl.handle.net/2142/88019`.

[81] Rethink Robotics. *Baxter Arm Calibration.* [Online] Accessed: 2020-04-20. 2015. URL: http://sdk.rethinkrobotics.com/wiki/Arm_Calibration.

[82] Point Cloud Library (PCL). *Aligning object templates to a point cloud.* [Online] Accessed: 2020-05-05. 2011. URL: https://pcl-tutorials.readthedocs.io/en/master/template_alignment.html.

[83] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. "Fast Point Feature Histograms (FPFH) for 3D registration". In: *IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2009, pp. 3212–3217. DOI: 10.1109/ROBOT.2009.5152473.

[84] GPy. *GPy: A Gaussian Process Framework in Python.* [Online] Accessed: 2020-04-04. 2012. URL: http://github.com/SheffieldML/GPy.

[85] Kunihiro Ogata, Yoshio Matsumoto, Isamu Kajitani, Keiko Homma, and Yujin Wakita. "Whole-Body Robotic Simulator of the Elderly for Evaluating Robotic Devices for Nursing Care". In: *International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management.* Springer. 2018, pp. 478–487. ISBN: 978-3-319-91397-1.

[86] Yoshio Matsumoto, Kunihiro Ogata, Isamu Kajitani, Keiko Homma, and Yujin Wakita. "Evaluating Robotic Devices of Non-Wearable Transferring Aids Using Whole-Body Robotic Simulator of the Elderly". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE. 2018, pp. 1–9. DOI: 10.1109/IROS.2018.8594022.

[87] AIST Digital Human Research Group. *Japanese Body Dimension Data.* [Online] Accessed: 2019-02-26. 2001. URL: https://unit.aist.go.jp/hiri/dhrg/ja/dhdb/97-98/index_e.html.

[88] WHILL. *WHILL: Model Ci.* [Online] Accessed: 2020-05-04. 2018. URL: http://whill.us/model-ci.

[89] Shibata Laboratory. *whillpy: Unofficial python package for WHILL Model CK control.* [Online] Accessed: 2020-05-04. 2018. URL: https://github.com/ShibataLab/whillpy.

[90] M. Zucker, N. Ratliff, A. Dragan, M. Pivtoraiko, M. Klingensmith, C. Dellin, J. A. Bagnell, and S. Srinivasa. “CHOMP: Covariant Hamiltonian Optimization for Motion Planning”. In: *The International Journal of Robotics Research* 32 (2013), pp. 1164–1193. DOI: `10.1177/0278364913488805`.

[91] T. Osa and M. Sugiyama. “Hierarchical Policy Search via Return-Weighted Density Estimation”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2018. URL: `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16736`.

Publication List

Journal Papers

[1] Ravi Prakash Joshi, Nishanth Koganti, and Tomohiro Shibata. "A Framework for Robotic Clothing Assistance by Imitation Learning". In: *Advanced Robotics* 33.22 (2019), pp. 1156–1174. DOI: 10.1080/01691864.2019.1636715. URL: https://doi.org/10.1080/01691864.2019.1636715.

Conference Papers

[1] Ravi Prakash Joshi, Takayuki Osa, and Tomohiro Shibata. "Incremental Imitation Learning through Physical Interaction and its Application to Clothing Assistance". In: *In Preparation for Frontiers in Robotics and AI*. 2020. URL: https://www.frontiersin.org/research-topics/14008.

[2] Ravi Prakash Joshi, Jayant Prasad Tarapure, and Tomohiro Shibata. "Intelligent Wheelchair–Humanoid Robot Collaboration for Clothing Assistance of the Elderly". In: *13th International Conference on Human System Interaction (HSI)*. IEEE. Tokyo, Japan, 2020, pp. 300–306. DOI: 10.1109/HSI49210.2020.9142645.

[3] Ravi Prakash Joshi, Tomohiro Shibata, Kunihiro Ogata, and Yoshio Matsumoto. "Quantitative Evaluation of Clothing Assistance using Whole-Body Robotic Simulator of the Elderly". In: *28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. New Delhi, India, Oct. 2019. DOI: 10.1109/RO-MAN46459.2019.8956308.

[4] Salvador Blanco Negrete, Ravi Prakash Joshi, Rollyn Labuguen, Jumpei Matsumoto, and Tomohiro Shibata. “Mouse Anatomical Cardinal Planes and Axes Towards Augmentation for Behavior Analysis”. In: *Joint 7th International Conference on Informatics, Electronics Vision (ICIEV) and 2nd International Conference on Imaging, Vision Pattern Recognition (icIVPR).* Kitakyushu, Japan, June 2018, pp. 195–199. doi: `10.1109/ICIEV.2018.8641000`.

[5] Krati Saxena, Rollyn Labuguen, Ravi Prakash Joshi, Nishanth Koganti, and Shibata Tomohiro. “A Study on Human-Robot Collaboration for Table-setting Task”. In: *IEEE International Conference on Robotics and Biomimetics (ROBIO).* Macau, China, Dec. 2017, pp. 183–188. doi: `10.1109/ROBIO.2017.8324415`.

[6] Ravi Prakash Joshi, Rithul Perathara, Rolyn Labuguen, Nishanth Koganti, and Tomohiro Shibata. “Estimating 3D Hand Location for Clothing Assistance Initialization Using Dynamic Movement Primitives”. In: *Proceedings of the 35th Annual Conference of Robotics Society of Japan.* Kawagoe, Japan, 2017. url: `http://rsj2017.rsj-web.org/RSJ2017_pg02.pdf`.

[7] Ravi Prakash Joshi, Nishanth Koganti, and Tomohiro Shibata. “Robotic Cloth manipulation for Clothing Assistance Task using Dynamic Movement Primitives”. In: *Proceedings of the Conference on Advances in Robotics.* AIR ’17. New Delhi, India: ACM, 2017, 14:1–14:6. isbn: 978-1-4503-5294-9. doi: `10.1145/3132446.3134878`.

[8] Nishanth Koganti, Ravi Prakash Joshi, Tomoya Tamei, Tomohiro Shibata, and Kazushi Ikeda. “Motor-skill Learning in Latent Spaces for Robotic Clothing Assistance”. In: *Proceedings of the 34th Annual Conference of Robotics Society of Japan.* Yamagata, Japan, 2016. url: `http://rsj2016.rsj-web.org/data/RSJ2016_pg.pdf`.

[9] Ravi Prakash Joshi, Nishanth Koganti, and Tomohiro Shibata. “Robotic Cloth Manipulation for Clothing Assistance Task using Dynamic Movement Primitives”. In: *4th International Symposium on Applied Engineering and Sciences (SAES).* Tobata, Japan, 2016. url: `http://www.life.kyutech.ac.jp/~saes2016/SAES3.html`.

[10] Vishal Gaurav, Nishanth Koganti, Riku Nakata, Ravi Prakash Joshi, and Tomohiro Shibata. “Cloth Extremity Detection from a Clutter of Clothes using Bayesian GPLVM”. In: *International session on Robotics, Mechatronics and Control, Robotics Society of Japan*. RSJ. Yamagata, Japan, 2016. URL: `http://rsj2016.rsj-web.org/data/RSJ2016_pg.pdf`.

Workshop Papers

[1] Vishal Gaurav, Nishanth Koganti, Riku Nakata, Ravi Prakash Joshi, and Tomohiro Shibata. “Extremity Detection from a Pile of Garments using Bayesian GPLVM”. In: *Workshop on Assistive and Service Robotics in a Human Envionment, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. Daejeon, Korea, 2016, pp. 1–6. URL: `https://research02.jimu.kyutech.ac.jp/html/100000703_knkyu_prsn_en.html`.

Poster Presentations

[1] Ravi Prakash Joshi, Jayant Prasad Tarapure, and Tomohiro Shibata. “A Study on the Electric Wheelchair-Humanoid Collaboration for Clothing Assistance of the Elderly”. In: *Late Breaking Results at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Macau, China, 2019. URL: `https://www.iros2019.org/LBRposter`.