TASK-CENTRIC OPTIMIZATION FOR ASSISTIVE MOBILE MANIPULATORS

A Dissertation
Presented to
The Academic Faculty

By

Ariel S. Kapusta

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering

Georgia Institute of Technology

August 2018

TASK-CENTRIC OPTIMIZATION FOR ASSISTIVE MOBILE MANIPULATORS

Approved by:

Dr. Charles C. Kemp, Advisor
School of Biomedical Engineering
Georgia Institute of Technology

Dr. Frank Hammond
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Lena Ting
School of Mechanical Engineering
Georgia Institute of Technology

Dr. C. Karen Liu
School of Interactive Computing
Georgia Institute of Technology

Dr. Greg Turk
School of Interactive Computing
Georgia Institute of Technology

Date Approved: June 7, 2018

It is not your responsibility to finish the work [of perfecting the world], but you are not free to desist from it either.

Pirke Avot 2:21

Dedicated to my wife, Chana, for being with me through the roller coaster that is a Ph.D, and to my son, Asher, for being a cute and extremely distracting bundle of baby.

ACKNOWLEDGEMENTS

I am grateful to finally complete and reflect on the almost six year journey that led me to completing this Ph.D. dissertation. There are many I would like to recognize for helping me reach this point.

I'll start by thanking my family. I thank my parents, who have been there for me from the beginning, always providing support, love, and occasional cheerleading. I thank my wife, Chana, who sacrificed and put up with much, to make this dream a reality. She supported me through the many, many ups, downs, spins, and twirls, that was my Ph.D. It was a wild roller coaster, but we made it. I thank my son, Asher, who helped me take breaks and cheered me up with giant smiles.

I thank my advisor, Dr Charlie Kemp, who has provided me guidance my efforts over these years. He brought me into the world of robotics and helped me navigate through the maze that is research, in interesting and exciting ways.

I also thank Henry and Jane Evans, who have been a part of long-term collaborative efforts to develop the methods and system presented in this dissertation. They have been patient, understanding, and demanding (when it comes to feedback on robotic systems), as I ran tests repeatedly in their home. I am certain that their involvement improved my work.

I thank the many great members of the Healthcare Robotics Lab, past and present, who helped me make it through. Your discussions, collaborations, and companionship were invaluable. In alphabetical order, I thank Tapomayukh Bhattacharjee, Yash Chitalia, Kevin Chow, Henry M. Clever, Zackory Erickson, Phillip M. Grice, Jeffrey Hawke, Marc D. Killpack, Daehyung Park, Ashwin A. Shenoi, Sarvagya Survy Vaish, and Joshua Wade.

I also thank Alex Clegg, Dr C. Karen Liu, Dr Greg Turk, and Wenhao Yu for the wonderful collaboration we've had in tackling robot-assisted dressing.

I am also grateful for the funding I have received from various sources, including the NSF via the project on robot-assisted dressing, NIDILRR TechSAge, and the DARPA M3

program.

I thank everyone who helped me make this PhD Dissertation a reality. Thanks!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Assistive mobile manipulators could enable people with disabilities to perform tasks for themselves which would otherwise be difficult or impossible. The robot's assistance has the potential to increase independence and quality of life. Through this dissertation we have explored methods to realize that assistance.

Although many groups have looked at how a robot could execute some specific tasks, few have considered where to place the robot to better provide assistance. We have observed that this problem arises frequently in real-world settings and solving the problem can be challenging, even for an expert user. We first present an answer to the question, "How should a robot choose a configuration of its base to be better able to provide assistance?" In answering this question we expand the problem to better match common scenarios in assistive robotics, where the task may be complicated and may take place in a bed or wheelchair. We present task-centric optimization of robot configurations (TOC), a method for addressing this question. We demonstrate how TOC can select one or more robot configurations for many assistive tasks that involve the robot moving a tool around a persons body. We additionally provide evidence that TOC outperforms baseline methods from literature. We present an assistive robotic system consisting of a robotic bed and a mobile manipulator that uses TOC to allow the two robots to autonomously collaborate to better provide assistance. We tested this system with a person with severe quadriplegia in his home, providing evidence of the feasibility of TOC and the robotic system for providing assistance to real people.

Through our work on assistive robotics, we recognized that an important activity of daily living (ADL), dressing, contains special challenges not fully addressed by TOC. We present task optimization of robot assisted dressing (TOORAD), a method for selecting actions for both the robot and the person that will result in successful dressing. We demonstrate the efficacy of TOORAD in a study with participants with disabilities receiving dressing

assistance from a mobile manipulator. In that study, we also administered surveys on habits, needs, capabilities, and views on robot-assisted dressing that we expect will provide guidance for future research.

CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 Overview

Societal challenges, such as aging populations, high health care costs, and shortages of health care workers found in the United States and other countries, suggest a need for innovative methods of care [1, 2]. Activities of daily living (ADLs) are important for independent living and quality of life [3, 4, 5]. Dressing is a prominent ADL, along with feeding, toileting, transferring, and hygiene [3]. Studies suggest that more older adults receive assistance from human caregivers with dressing than other ADLs besides bathing/showering, and over 80% of people in skilled nursing facilities require dressing assistance [6]. A number of specially designed assistive devices exist to help people maintain their independence. However, current assistive devices, such as those for dressing (e.g., reachers, dressing sticks, long-handled shoehorns, and sock aids), provide limited support and rely on the user having substantial cognitive, perceptual, and motor capabilities [7, 8]. Robots could potentially serve as more versatile assistive devices.

Specialized robots, such as desktop feeding devices, have been successful for a narrow range of assistive tasks when placed in fixed and designated positions with respect to the user [9, 10]. Others have used fixed-base general-purpose manipulators (a robot arm in a fixed location) [11, 12, 13, 14], again in designated positions. General-purpose mobile manipulators have the advantage of mobility, which may allow them to provide assistance across a wider range of tasks, users, and environments, and provide this assistance independent of any fixed-location [15]. However, this mobility introduces additional complexity and challenges.

1.2 Problem Description

Through this dissertation we work towards mobile manipulators that can provide effective assistance to people with disabilities, allowing the user to perform tasks which would otherwise be difficult or impossible. We present methods to inform a mobile manipulator on how it can assist a user with tasks, such as ADLs. We are guided by past work and observations on areas of assistive robotics that are more or less explored. Many groups have looked at how a robot could execute specific tasks, such as ADLs [16, 17, 18, 19, 20] or instrumental activities of daily living (IADLs) [21, 22]. However, in most of these works, the location of the robot's base is selected manually or with a simple inverse-kinematics (IK) solver. The choice of where to place such robots is important, as it can impact the robot's ability to provide effective assistance. For a mobile manipulator to provide assistance to a person, it may first need to get close enough to reach task-relevant locations. Furthermore, the problem of selecting the robot's base pose may arise repeatedly for mobile manipulators, as they move around performing tasks. In this dissertation, we often use the more general term, configuration, which includes the robot's base pose as well as other configurable parameters of the robot (e.g., the height of the robot's spine).

The question addressed in Chapters 2 and 3 is, "How should a robot choose a configuration of its base to be better able to provide assistance?" In answering this question we expanded the problem to better fit the scenario of assistive robotics, where tasks can be complicated, involve a person, and that person may be in an environment that can be leveraged to better provide assistance. Chapter 3 follows as a practical demonstration, showing a proof-of-concent assistive robotic system that makes use of the method presented in Chapter 2. The system consists of a mobile manipulator working in collaboration with a robotic bed. In continued exploration towards mobile manipulators that can provide assistance with tasks, we discuss in Chapter 4 our work on robot-assisted dressing, which modifies our previous methods to address the specific challenges of dressing.

1.3 Contributions

The contributions of this dissertation include the development of two methods: task-centric optimization of robot configurations (TOC) and task optimization of robot-assisted dressing (TOORAD). An additional contribution is a proof-of-concept assistive robotic system with a robotic bed and a mobile manipulator that uses TOC to improve assistance.

1.3.1 Task-centric Optimization of Robot Configurations (TOC)

We present task-centric optimization of robot configurations (TOC), which is an algorithm that finds configurations from which the robot can better reach task-relevant locations and handle task variation. TOC is suitable for quickly selecting one or more robot configurations for many assistive tasks, including some ADLs, that involve the robot moving a tool around a person's body. Notably, TOC can return more than one configuration that when used sequentially enable an assistive robot to reach more task-relevant locations. TOC performs substantial computation offline to generate a function that can be applied rapidly online to select robot configurations based on current observations. TOC explicitly models the task, environment, and user, and implicitly handles error using two representations of robot dexterity that we have developed. Key features of TOC are its selection of multiple configurations for a task, its task-centric approach, its explicit modeling of many task-specific parameters that are important to assistive tasks (possible because of the task-centric approach), its representations of robot dexterity, and its framework splitting offline and online computation. We evaluated TOC in simulation with a PR2 assisting a user with 9 assistive tasks in both a wheelchair and a robotic bed. TOC had an overall average success rate of 90.6% compared to 43.5%, 50.4%, and 58.9% for three baseline methods from literature. We additionally demonstrate how TOC can find configurations for more than one robot and can be used to assist in designing or optimizing environments. Details on TOC are presented in Chapter 2.

1.3.2 Application: A System for Bedside Assistance that Integrates a Robotic Bed and a Mobile Manipulator

As a platform for testing TOC in the real world with a person with disabilities, we present a proof-of-concept robotic system for bedside assistance that integrates a robotic bed and a mobile Manipulator (a PR2 robot made by Willow Garage), that work together to better provide assistance to a user with tasks around the user's body. Many assistive tasks depend on movements with respect to the person's body, and the complementary physical and perceptual capabilities of the robots help with respect to this general goal. To manage the system's complexity for common assistive tasks, our system provides autonomous functions with a coarse-to-fine approach. The system autonomously completes the "coarse" parts of the task, such as moving the mobile robot's base to the appropriate place and configuring the robotic bed, and then gives control back to the user for "fine" execution, the detailed performance of the task. This "coarse" setup can be challenging for a user, and we can reduce the user's overall workload by having it performed autonomously. The detailed performance of the task can be challenging to perform autonomously; for example, in a scratching task, the precise location of an itch can be difficult for the robot to find autonomously. The system makes use of the user's cognitive capabilities in providing fine control.

With this system, we introduce and investigate the potential for a robotic bed to collaborate with a mobile manipulator in order to provide more effective assistance to people in bed. We demonstrate that a robotic bed and a mobile manipulator have complementary physical and perceptual capabilities. The robotic bed can move the human body using a small number of degrees of freedom into positions that are more reachable or relevant to the task, such as sitting the person up to eat food. It can also help the mobile manipulator reach the human body by raising itself above the ground so that the mobile manipulator's base can go under it or give the manipulator better access to the person. Our mobile manipulator can dexterously manipulate a lightweight payload using a large number of degrees of freedom. The robotic bed can also perceive the human body via a pressure sensing mat, while the mobile manipulator would typically perceive the person using on-board line-of-sight sensors that can be obstructed by bedding and other objects around a bed, such as an overbed table or IV lines.

In an evaluation using a medical mannequin, we found that the robotic bed's motion and perception each improved the assistive robotic system's performance. The system achieved 100% success over 9 trials involving 3 tasks, but disabling bed movement or body pose estimation resulted in success in only 33% or 78% of the trials, respectively. We also evaluated our system with Henry Evans, a person with severe quadriplegia, in his home. In a formal test, Henry successfully used our system to perform 3 different tasks, 5 times each, without any failures. Henry's feedback on the system was positive regarding usefulness and ease of use, and he noted benefits of using our system over fully manual teleoperation. Overall, our results suggest that a robotic bed and a mobile manipulator can work collaboratively to provide effective personal assistance and the combination of the two robots is beneficial. Details on this assistive robotic system are presented in Chapter 3.

1.3.3 Task Optimization of Robot-Assisted Dressing (TOORAD)

We present task optimization of robot-assisted dressing (TOORAD), a method for selecting actions for both the robot and the person that will result in successful completion of a dressing task. TOORAD makes use of geometric, kinematic, and physics simulations of the person, robot, and garment in its optimization. It uses customized models for the person to model their geometries and physical capabilities. These models consider what the person is capable of doing, instead of what he or she typically does. With this approach, TOORAD is able to explore a wide range of actions for dressing in simulation, some of which might be challenging to test in the real world. Using a general-purpose mobile manipulator can mitigate some of the challenges in dressing by allowing the robot to move around to access different areas around the body, as explored by [23]. We consider the robot an important part of TOORAD's optimization, so TOORAD optimizes the robot's base pose to improve the robot's ability to adapt to unexpected changes. Additionally, our method provides computer-generated instructions for the user receiving assistance from the robot. We have used TOORAD to optimize the actions of a person and a PR2 robot (a mobile manipulator made by Willow Garage) to collaborate in pulling the two sleeves of a hospital gown onto the person's body. These features are notable differences from previous work on robot-assisted dressing We conducted a study with six participants with physical disabilities who have difficulty with dressing. In the first session of the study, we surveyed participants' needs, capabilities, and views on robot-assisted dressing to gain insight for future robot-assisted dressing research through better understanding of the target populations. For the second session of the study, four of the participants successfully received assistance from a robot-assisted dressing system in putting on both sleeves of a hospital gown. Two of the four participants dressed one sleeve independently and received assistance from the robot with only the second sleeve, and two participants received assistance with both sleeves. From participant feedback and survey responses we note paths forward for advancing robot-assisted dressing towards providing practical assistance to people with disabilities. Our results provide evidence that TOORAD can be used to select actions that will result in a robot and person with disabilities collaborating successfully to complete a dressing task. Details on TOORAD are presented in Chapter 4.

1.4 Organization of Dissertation

We organize the remainder of this dissertation as follows. Each of Chapters 2, 3, and 4 present their own comparisons to related works from literature. Chapter 2 presents TOC and Chapter 3 presents our work on a proof-of-concept assistive robotic system that uses TOC to inform the robotic system. Chapter 4 presents TOORAD, our work on robot-assisted dressing, and a study both on the the needs and capabilities, with respect to dressing, of people with disabilities, and on a system implementing TOORAD to coordinate the actions of a robot assisting people with disabilities with a dressing task. Chapter 5 discusses lessons learned, opportunities for future research extending our work, and finally summarizes the overall work and potential impact. In Appendix **??**, we provide the complete responses to the questionnaires administered in our study on robot-assisted dressing with participants with disabilities.

CHAPTER 2
TASK-CENTRIC OPTIMIZATION OF CONFIGURATIONS FOR ASSISTIVE ROBOTS

2.1 Introduction

Robotic assistance with activities of daily living (ADLs) [3] could potentially enable people to be more independent. This may improve quality of life [5, 4] and help address societal challenges, such as aging populations, high health care costs, and shortages of health care workers found in the United States and other countries [1, 2].

Many specialized assistive devices can help people with motor impairments perform ADLs on their own [7, 8]. Specialized robots, such as desktop feeding devices, have been successful for a narrow range of assistive tasks when placed in fixed and designated positions with respect to the user [9, 10]. The choice of where to place such robots is important, as it can impact the robot's ability to provide effective assistance. General-purpose mobile manipulators have the advantage of mobility, which may allow them to provide assistance across a wide range of tasks, users, and environments [15]. However, this mobility introduces additional complexity. For a mobile manipulator to provide assistance to a person, it may first need to get close and be able to reach task-relevant locations. For mobile manipulators, the problem of selecting the robot's base pose may arise repeatedly as it moves around performing tasks. In this chapter we use the more general term, configuration, which includes the robot's base pose as well as other configurable parameters selected for the task (e.g., the height of the robot's spine). Choosing the robot's configuration can be challenging because of robot complexity, task complexity, geometric constraints, and kinematic constraints. Additionally, more than one configuration of the robot may be necessary to complete some tasks or more than one robot may be involved. For example, [16] used two positions of a

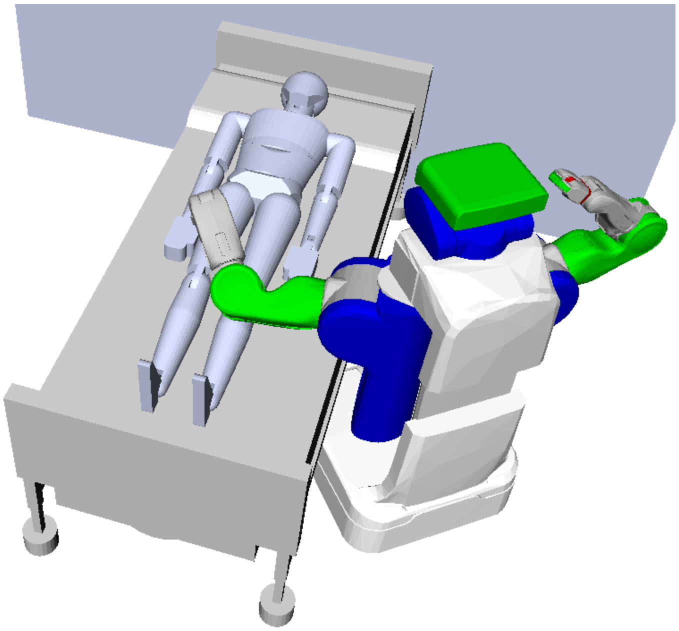


Figure 2.1: TOC can select a configuration for the PR2 and the robotic bed so the PR2 can better reach task-relevant locations. This figure shows the configuration for the task of a PR2 cleaning the legs of a person in a robotic bed.

PR2 (a mobile manipulator made by Willow Garage) to reach and shave the entirety of a user's face (both sides) for a user in a wheelchair.

Our approach to providing assistance to a user with a robot is to first find a good configuration for the robot from which it can perform the task. In this chapter, we focus exclusively on the problem of selecting the robot configuration, leaving task performance out of scope. Performance of assistive tasks is addressed in many other works (see Section 2.2.4). The questions we ask are: how can the the robot select a good configuration, and what makes a configuration good? We address these questions in Section 2.3. With a good configuration, the robot is more likely to be able to complete the task successfully.

We present a task-centric, optimization-based method to select one or more configurations for assistive robots, that we call task-centric optimization of robot configurations (TOC). TOC extends two previous works from [24, 25], with changes to framework, formulation of terms used within the methods, and modeling of the user and environment. Additionally, we have improved the optimization search and performed additional evaluations of our method

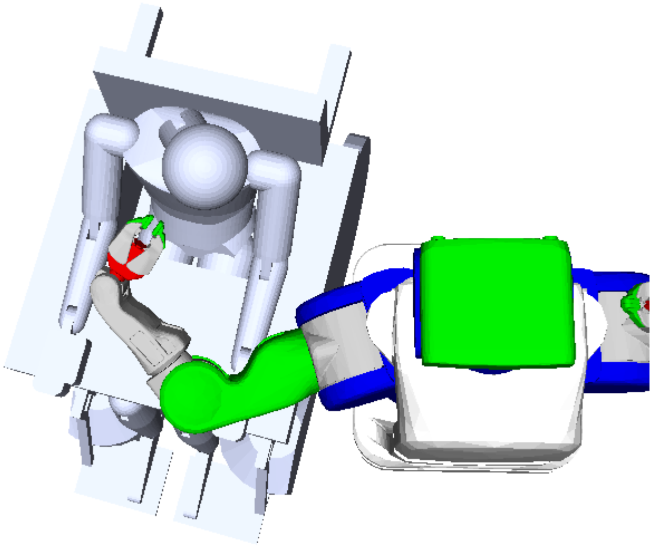


Figure 2.2: TOC can select a single configuration for the PR2 to shave a person in a wheelchair, shown here. TOC takes advantage of the person's physical capabilities, such being able to rotate his or her neck.

with thorough comparisons to other methods from the literature. TOC is suitable for quickly selecting one or more robot configurations for assistive tasks, including some activities of daily living (ADLs), that involve the robot moving a tool around a person's body. A task-centric approach is particularly applicable to assistive tasks where there may be important or common tasks that take place in environments that are known apriori. Key features of TOC are its selection of multiple configurations for a task, its task-centric approach, its explicit modeling of many task-specific parameters that are important to assistive tasks (possible because of the task-centric approach), its representations of robot dexterity, and its framework splitting offline and online computation.

TOC performs substantial offline computation to generate a function that can be applied rapidly online to select robot configurations based on current observations. These offline computations use explicit models of the task, environment, and user. Because offline modeling may be vulnerable to problems with error and mismatch between models and reality, TOC uses two representations of robot dexterity that we developed, task-centric

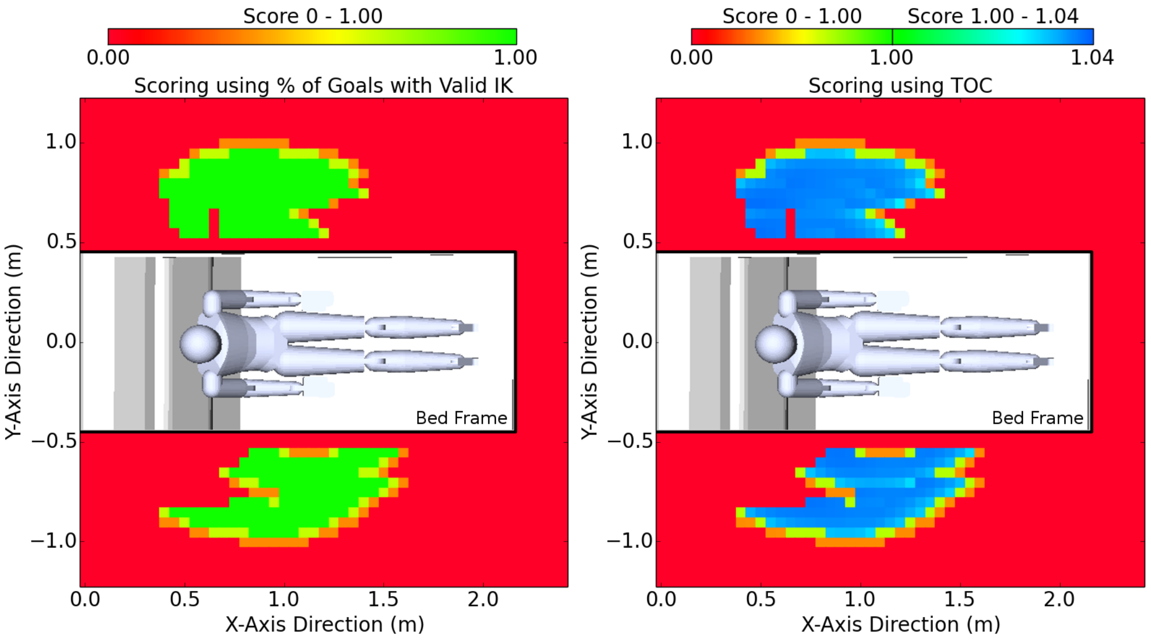


Figure 2.3: TOC differentiates between robot configurations that have collision-free IK solutions to all goal poses. This differentiation allows TOC to better select good robot configurations. The figure visualizes two scoring methods, showing the score for discretized PR2 base poses. Z-rotation is sampled every $45°$ from $0°$ to $360°$, and the best score is shown (left) using the percentage of goals with collision-free IK solutions and (right) TOC, for the mouth wiping task in the robotic bed environment, with the bed raised $20\,\mathrm{cm}$ and at $45°$. The color represents the best score for that 2D position of the PR2. 0 means no goals and ≥ 1 means all goals have collision-free IK solutions.

reachability (TC-reachability) and task-centric manipulability (TC-manipulability), in its objective function to help it select robot configurations that are implicitly robust to error. TOC searches the robot configuration space to maximize its objective function, a linear combination of TC-reachability and TC-manipulability, using a simulation-based, derivative-free optimization from literature, covariance matrix adaptation evolution strategy (CMA-ES). Figures 2.1 and 2.2 show the configurations selected for the cleaning legs task for a user in bed and the shaving task for a user in a wheelchair, respectively.

We evaluated TOC in simulation with a PR2 assisting a user with 9 assistive tasks in both a wheelchair and a robotic bed. For our evaluations we implemented three baseline algorithms from literature. The first used an inverse-kinematics (IK) solver to select a configuration with a collision-free IK solution to all task-relevant goal poses. The second and third baselines used a capability map from [26] to select a configuration with high capability score to the goal poses. The third baseline also checked that a collision-free IK solution existed for each goal pose. We ran Monte-Carlo simulations of pose estimation error and found that TOC's average success rate was higher than or comparable to baseline algorithms for each task. TOC had an overall average success rate of 90.6% compared to 50.4% for the IK solver, 43.5% for capability map, and 58.9% for capability map with collision checking. Additionally, we provide evidence that TOC's objective function is positively correlate with robustness to error, and we demonstrate how TOC can be used to assist in designing environments to improve robotic assistance.

2.2 Related Work

2.2.1 Representations of Robot Dexterity

Many metrics have been developed to quantify the kinematic dexterity of robot manipulators. These metrics can be broadly divided into those that use the manipulator's Jacobian, $\boldsymbol{J}(\boldsymbol{q})$ [27] and those that do not. These metrics can also be divided into those that find global measures (a metric for the robot irrespective of joint configuration) and those that find local, configuration-dependent measures of dexterity. Global dexterity metrics are often used for robot design [28, 29]. As we are focused on dexterity measures to assist in positioning existing robots, we will focus on discussing local metrics. [30] proposed the local Jacobian-based metric called measure of manipulability (or just, manipulability), $w(\boldsymbol{q})$, shown in Equation 2.1.

$$w(\boldsymbol{q}) = \sqrt{\det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)} \tag{2.1}$$

Geometrically, manipulability is proportional to the volume of the manipulability ellipsoid of the manipulator, which is the volume of Cartesian space moved by the end effector for a unit ball of movement by the arm's joints. This metric can be useful when assessing kinematic dexterity between different configurations of the same robot. However, its scale and order dependencies make comparison between different robot morphologies challenging.

Other dexterity measures were developed that address some of the issues of using manipulability [31]. [32] proposed another local Jacobian-based metric that we refer to in this chapter as kinematic isotropy, $\Delta(\boldsymbol{q})$, shown in Equation 2.2.

$$\Delta(\boldsymbol{q}) = \frac{\sqrt[a]{\det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)}}{\left(\frac{\operatorname{trace}(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)}{a}\right)} \tag{2.2}$$

Kinematic isotropy uses the manipulability term (shown in Equation 2.1) with an alteration to remove order dependency and divided by a term to remove scale dependency. Order is the size of the Cartesian space of interest. For a planar robot, the order would be three if translations are all in 2D in-plane and in-plane rotations are considered. For the case of tasks in 6D space (position and orientation), the order is six. Unlike manipulability, the values of kinematic isotropy always range from 0 to 1 they can be directly compared across robot platforms. In our work we modified kinematic isotropy, adding a weighting term to create what we call joint-limit-weighted kinematic isotropy (JLWKI). We use JLWKI in our task-centric manipulability (TC-manipulability) defined in Section 2.3.7. TC-manipulability represents the kinematic dexterity of the robot for a task (a set of goal poses) from a set of one or more positions of the robot.

An important limitation to some Jacobian-based measures of dexterity is their ignorance of many relevant features of the workspace, such as joint limits and collisions. The manipulability ellipsoid calculated from the Jacobian suggests that the end effector can move in ways that may be constrained by joint limits. Many researchers have proposed various ways to include these features (joint limits: [33, 34]; velocity limits: [35]; torque limits: [36]) into weighting terms. [37] created what they called an extended manipulability measure by modifying the Jacobian to include weights on joint limits, on proximity to self-collision, and on proximity to collision with the environment. Many of these methods apply their weighting terms directly to manipulability or indirectly by modifying the Jacobian, which is used in manipulability. JLWKI differs from other measures of dexterity, using a distinct weighting function on joint limits and applying it to kinematic isotropy. JLWKI, does not include costs on proximity to collision because we found that calculating these costs increases the computation time of TOC excessively.

[26] introduced a method for representing manipulator dexterity without using Jacobians, which they use to score the workspace of a robot, creating what they call a capability map. To create the capability map they discretize space around the robot into 3D points and discretize the range of possible orientations around each 3D point. The capability score (also known as reachability score) is the number of orientations for which the robot has a valid IK solution. A way to interpret the meaning of this score is, if a goal pose is located at the 3D location, the capability score is similar to the probability that the manipulator can achieve the pose.

2.2.2 Selecting Robot Configurations for Mobile Manipulation

Prior research has investigated how to select configurations for a mobile robot. A common method is to address the problem using IK solvers [38, 39, 40, 41]. The entire kinematic chain from end effector to the robot's base location may be solved using IK [42, 43]. Alternatively, sampling-based methods may be used to find robot base poses that have valid IK solutions, often as part of motion planning [44, 45, 46, 47, 48].

By relying solely on IK to ensure that the robot can reach the goals, these methods are dependent on accurate models. Many of these methods are fast, but may fail if there is modeling or state estimation error. Like these methods, TOC uses a sampling-based search to find a robot configuration with valid IK solutions. However, there are many such robot configurations, and they cannot be distinguished using only IK, as shown in Figure 2.3(left). All locations in green have collision-free IK solutions to all goals, but some may result in higher success rates than others. TOC uses task-centric manipulability to differentiate those configurations, as shown in Figure 2.3(right). We show in Section 2.4.4 that higher TOC score is correlated with improved performance for configurations that have collision-free IK solutions to all goals. Additionally, TOC can find more than one robot configuration for a task. We implemented a standard IK sampling-based method as a baseline for comparison, as described in Section 2.4.2.

A body of work is based on the capability map from [26]. Capability-map-based methods are robot-centric and task-agnostic; they are generated offline for the robot's manipulator and applied to tasks online. They typically select the robot base position by overlapping the capability map with end effector goal poses and maximizing the average capability score [49, 50, 51].

[52] altered the capability map by creating an orientation-based capability map and extending the map for tools on the robot's end effector. In contrast with our method, existing capability-map-based methods do not consider collisions with the environment in their offline computations because they do not model the environment. Collisions are only considered at runtime to eliminate robot base locations in collision or that lack collision-free IK solutions. Simply selecting the robot base location with highest capability map score is fast, but searching for a collision-free location can take more time. Capability-map-based methods also only find a single location for a robot for a task, but our method can find multiple robot configurations. We implemented two capability-map-based methods as baselines for comparison, based on [49], as described in Section 2.4.2.

Another body of work extends the capability map by inverting it, creating an inverse-reachability map. While a capability map scores end effector poses with respect to a robot base pose, the inverse-reachability map scores robot base poses with respect to an end effector pose. As with the capability map, the inverse-reachability map is generated offline for the robot's manipulator and can be used quickly online. For an end effector 3D position, discretized robot base poses are scored based on the capability map score to that 3D position. The inverse-reachability map is used to rapidly sample a robot base position that can reach a set of goal end effector poses [53, 54]. [55] used an alternative representation of the robot's dexterity from their previous work [37] that uses 6D poses in the workspace. They invert that workspace representation to create what they call an Oriented Reachability Map (ORM). ORM, like the inverse-reachability map, scores robot base poses based on the extended manipulability measure of each 6D pose in the manipulator's discretized workspace. When searching for a robot base pose for a task, they sample in series from the ORM using the map's score as a sampling weight. If the sampled robot base pose has collision-free IK solutions to the goal end effector poses, they use that base pose; if not they re-sample. They propose methods to incorporate task-specific information in their extended manipulability measure (and thus into the ORM), and to calculate the ORM map online through what they call lazy-ORM. Inverse-reachability-based methods and ORM differ from our work in a few ways. These methods are typically used in task-agnostic and robot-specific ways to facilitate applications of the robot to new tasks. Notably, these methods only find a single location for a robot for a given task, rather than multiple robot configurations. TOC also explicitly models features and parameters of the environment and user that may be important to assistive tasks. Details on this modeling is found in Sections 2.3.5 and 2.3.6.

Most previous task-centric methods use simulation of the task, with explicit error modeling, to evaluate robot base poses. [56] presented a task-specific method for selecting a place for an industrial robot manipulator to perform a series of tasks amidst clutter. They used randomized path planners to generate collision-free paths for the arm and they randomly perturb the robot position to find positions from which the tasks can be performed quickly. [57] present a task-centric method for finding areas in which to place a mobile manipulator where it can successfully perform a grasping task. They use Monte-Carlo simulation of error in the location of the object to be grasped to find base positions with high success rates. They simulate performance of the entire task including, navigation, motion planning and motion execution. For real-time base position selection, they convolve uncertainty in robot location with base position scores to provide an area of high-success probability. They used their method to select a 2D position of the robot base for a grasping task. These task-centric methods that explicitly model error and fully simulate task performance have only been used to select a few degrees of freedom in static environments, and can only select a single robot configuration for a task. In contrast, TOC uses faster, simpler simulation and implicitly handles error. TOC selects more degrees of freedom in configurable environments, and, again, can select multiple robot configurations for a task.

2.2.3 Human-robot Proxemics

Several bodies of work have examined the proxemics of human-robot interactions [58, 59, 60, 61, 62].

Proxemics is the study of the spatial requirements of humans (e.g., the amount of space that people feel it necessary to set between themselves and others). These works look at acceptable interpersonal distances between humans and robots in social settings. Various works have used the concepts of human-robot proxemics to inform a robot when performing tasks. These works couple task performance concepts with scoring methods based on proxemics to select base positions and paths for the robot and item handover locations [63, 64, 65, 66]. Proxemics might suggest that placing the robot in front of the person at some minimum distance is preferred over other locations.

[67] present a thorough survey of human-aware robot navigation. In contrast, TOC does not consider proxemics or social factors; it instead focuses on kinematic aspects of the task. However, inclusion of additional terms in TOC's objective function to include user comfort and proxemics is possible. While proxemics is often used to consider navigation problems, TOC focuses exclusively on selecting the configuration for the task, which may be the goal pose for navigation.

2.2.4 Assistive Robots

Researchers have investigated the use of mobile manipulators as assistive devices [68, 69, 70, 71, 72, 16].

We seek to further empower assistive mobile manipulators by autonomously selecting

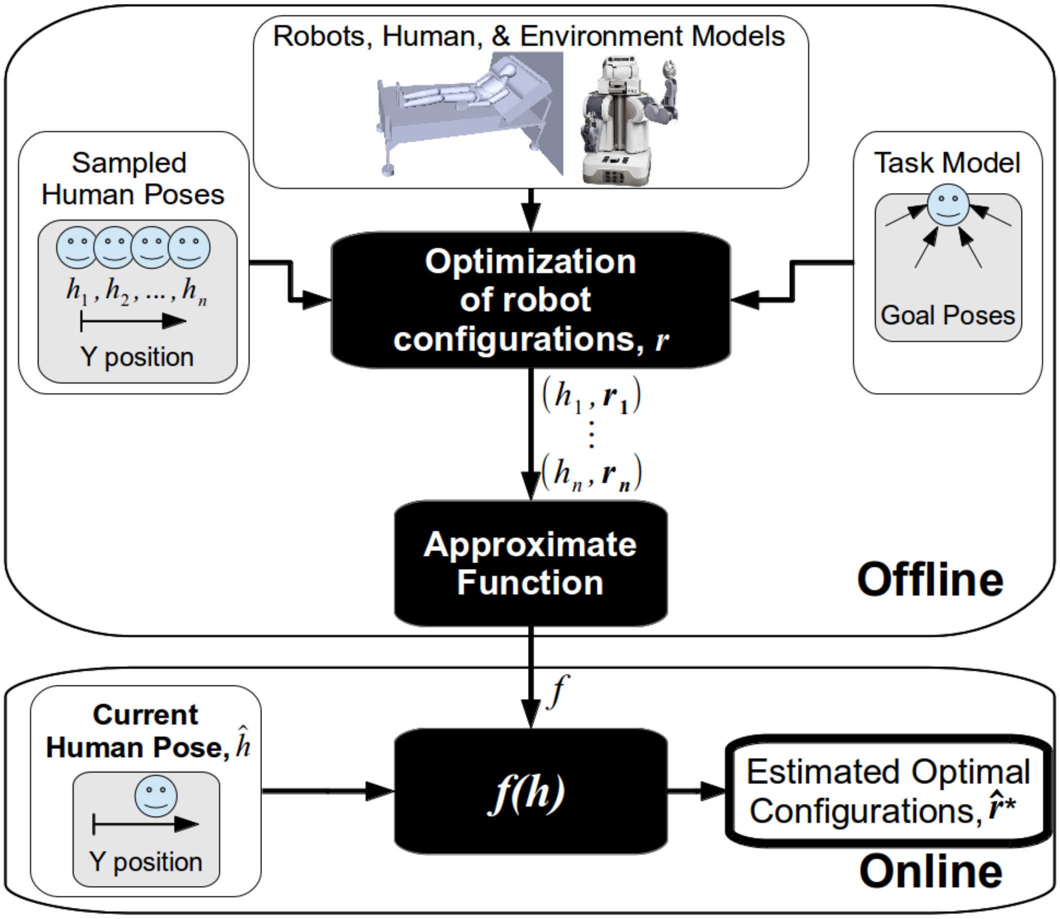

Robots, Human, & Environment Models
Sampled Human Poses
$h_1, h_2, \ldots, h_n$
Y position
Task Model
Goal Poses
Optimization of robot configurations, r
$(h_1, \boldsymbol{r}_1)$
$(h_n, \boldsymbol{r}_n)$
Approximate Function
Offline
f
Current Human Pose, $\hat{h}$
Y position
$f(h)$
Estimated Optimal Configurations, $\hat{\boldsymbol{r}}^*$
Online

Figure 2.4: The Framework used in TOC. The offline portion of TOC takes as input task-relevant models and samples of the uncontrollable parameters and outputs optimized robot configurations. It then approximates a function that is used online to estimate the optimal robot configurations given the current, observed uncontrollable parameters.

configurations from which they can better provide assistance. This autonomy can improve task performance and decrease cognitive workload for teleoperated assistive systems, as from [73]. In this work, we have used a model of a robotic bed that matches Autobed, a robotic modified hospital bed from [74]. We have shown how TOC can optimize the configuration of the bed, allowing improved assistance from a mobile manipulator as part of a collaborative assistive system, as from [75]. This capability is demonstrated in our evaluations in the robotic bed environment in Section 2.4.2.

2.3 Task-centric Optimization of Robot Configurations (TOC)

As mentioned in Section 2.1, key features of task-centric optimization of robot configurations (TOC) are its task-centric approach, representations of robot dexterity, selection of multiple configurations for a task, and framework that splits offline and online computation. TOC is suitable for situations when tasks and environment layouts are known beforehand and we would like to configure the robot for these tasks such that the robot is successful despite variations between models and reality. By taking a task-centric approach, TOC is able to use task-specific knowledge, such as explicit modeling of task-relevant parameters, to better select configurations. We will first explain the goal of TOC. Afterwards we describe the nomenclature used in the remainder the chapter. We then explain the framework of TOC, details of its features, and specifics of our implementation.

2.3.1 TOC Goal: Selecting Good Configurations

The goal of TOC is to select a good set of one or more configurations for a robot to perform a task without additional adjustments. But what constitutes a *good* robot configuration? In this chapter we use the term *robot configuration* as a more general term for the pose of the robot's base, so it can include additional relevant parameters. For example, for a PR2, the robot configuration might be the position and orientation of the robot's mobile base as well as the z-axis spine height of the robot. If the PR2 were operating in a room with a robotic bed, the degrees of freedom (e.g., the height of the bed) of the bed could be included in the robot configuration. We consider robot configurations in sets that can be of cardinality 1 or greater; the robot can complete the entire task by adopting all configurations in the set in any order. With a *good* set of configurations, the robot is more likely to be able to complete the task successfully. We judge the robot's ability to perform the task from a set of robot configurations with one measure: if it can reach all goal poses collision-free with its end effector. Various forms of error, such as modeling error or state estimation error, may cause the robot to be unable to perform the task. Because the robot does not know how the modeling and state estimation error will manifest apriori, from a good set of robot configurations, the robot should be able to perform the task despite such error.

2.3.2 Nomenclature

c: A task identifier

N_c: The number of goal poses for task c

$\boldsymbol{q}$: A joint configuration of the robot arm. $\boldsymbol{q} \in \mathbb{R}^n$, where n is the number of DoF of the arm

q_i: The value for joint i in joint configuration $\boldsymbol{q}$, $q \in \boldsymbol{q}$

$\boldsymbol{q}^-, \boldsymbol{q}^+$: A list of the minimum and maximum values, respectively, for the joints of a robot's arm.

q_i^-, q_i^+: The minimum and maximum values, respectively, for joint i of a robot's arm.

$\boldsymbol{r}$: A set of robot configurations of cardinality ≥ 1, $\boldsymbol{r} = \{r_1, r_2, ..., r_n\}$, where n is the number of robot configurations in set $\boldsymbol{r}$. We used $n \in \{1, 2\}$ in our implementation of TOC used in our ev.

$\hat{\boldsymbol{r}}^*$: The estimated optimal robot configurations given current observations, the output of the online portion of TOC.

$\boldsymbol{h}$: The set of uncontrollable parameters, discretized into $\{h_1, h_2, h_3, ...\}$

$\hat{h}$: The uncontrollable parameters observed and estimated at run-time, the input to the online portion of TOC.

$\boldsymbol{b}$: The set of free parameters, discretized into $\{b_1, b_2, b_3, ...\}$

$\boldsymbol{x}$: Set of position and orientation end effector goal poses $x \in \mathbb{R}^6$. $\boldsymbol{x}$ depends on c, h, and b, but we omit those for simplicity in writing. $\boldsymbol{x} = \{x_1, x_2, ..., x_{N_c}\}$.

$\boldsymbol{s}_{r,x}$: Set of IK joint configuration solutions to goal x from *robot configuration* r, $\boldsymbol{s}_{r,x} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_n\}$, where n is the number of IK solutions

a: The order of the robot arm. In our case, 6.

$\boldsymbol{J}(\boldsymbol{q})$ The Jacobian of the arm in joint configuration $\boldsymbol{q}$

$\Delta(\boldsymbol{q})$: The kinematic isotropy for the arm in joint configuration $\boldsymbol{q}$

f: A function that takes $\hat{h}$ as input and outputs $\hat{\boldsymbol{r}}^*$. TOC generates f offline and applies it online.

2.3.3 Framework

Figure 2.4 shows the framework of TOC for our implementation described in Section 2.4 for a person on a robotic bed. TOC performs most of its computation offline to approximate a function that can be used online to select robot configurations for a task. The optimization takes as input task-relevant models (e.g., task, robot, user, and environment models) and a sample of the uncontrollable parameters (e.g., the position of the person on the bed), h. It outputs an optimized robot configuration, $\boldsymbol{r}$, for that h. The inputs and outputs of the optimization are used to approximate the function, f that takes as input at run-time the observed estimated uncontrollable parameters, $\hat{h}$ and outputs the estimated optimal configurations, $\hat{\boldsymbol{r}}^*$.

2.3.4 Task Modeling

Our aim with task modeling is to create a representation that allows TOC to efficiently evaluate a robot's ability to perform a task. There are many tasks that consist of manipulation of small objects or tools around a person's body, that we expect can be well modeled by a set of goal poses (Cartesian positions and quaternion orientations) with respect to relevant reference frames. We manually model each task as a sparse set of poses for the robot's end effector. For example, Figure 2.5 shows the eight goal poses with respect to the person's head that make up our model of a shaving task. We assume that if the robot can reach all goal poses, it is likely to be able to perform the task. However, we expect there to be differences between the models and the real tasks. We consider these discrepancies to be a

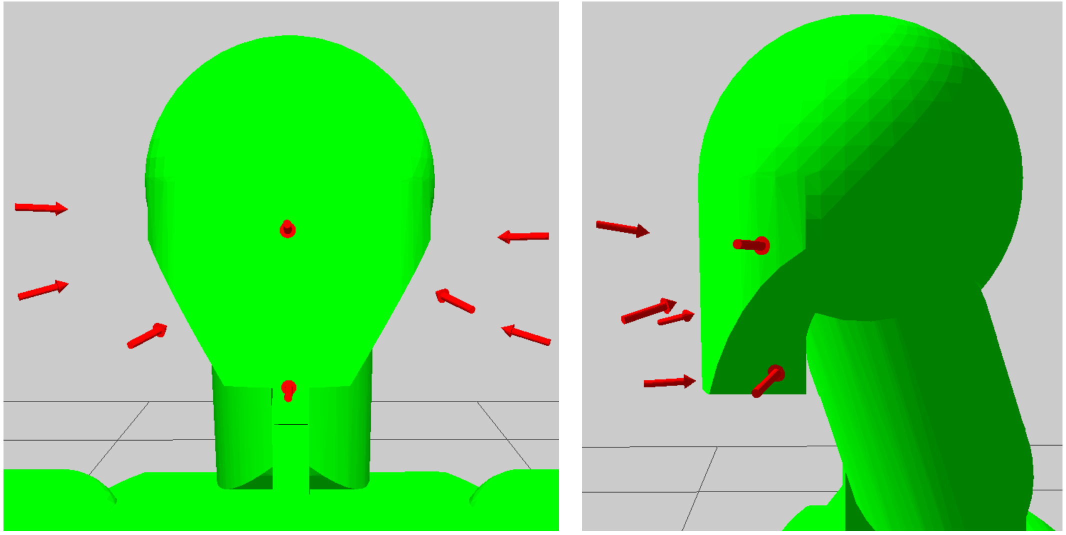


Figure 2.5: The manually selected goal poses for the shaving task. Each arrow represents a position and orientation, 6-DoF end effector goal pose, with respect to the head. This shows views from the front and side.

form of modeling error that TOC accounts for when selecting robot configurations.

2.3.5 Environment Modeling

Using its environment model, TOC finds robot configurations that avoid collision and unwanted interaction with obstacles, such as a bedside table or walls. TOC can use different resolutions for its environment model depending on the needs of the task. A room could be be represented simply as a wall behind the bed, as shown in Figure 2.1, or it could contain models of furniture and other potential obstacles. The resolution of each object model could range from a block to a detailed mesh. TOC uses three types of objects to model the environment:

- Fixed objects
- Controllable movable/configurable objects
- Uncontrollable movable/configurable objects

TOC treats fixed objects as static obstacles in the world, to be avoided by the robot. We add the configuration of the controllable objects to the robot configuration space that TOC optimizes. An example of a controllable object is a adjustable bed. For uncontrollable objects, TOC selects a robot configuration for a sample of the possible configurations of the objects. For each sample of its configuration, the object is considered static. An example of an uncontrollable object is a nightstand that is not movable by the robot, but could be moved somewhere by a person prior to the robot starting the task. Using controllable and uncontrollable movable objects, TOC can suggest alterations to the environment that may improve task performance. Uncontrollable movable objects can also be used to generate robot configurations for possible states of the environment. This may be beneficial for environments, such as hospitals, where there are a few possible room layouts, but the robot may not know which layout will be relevant until it reaches the room.

Because TOC does not include a cost in proximity to collision between the robot and the environment in its objective function, we include a margin of safety in the environment model by expanding the environment model (we used $\sim 3\,\mathrm{cm}$ in our evaluations). This safety margin reduces the risk of collision in the case of model or state estimation error, without having to explicitly include closeness to collision in the objective function.

2.3.6 User Modeling

TOC's user model can be customized for a user to better locate relevant parts of the body, and to allow more accurate collision-checking. In our evaluation of TOC we used a mesh model of a human designed around a 50 percentile male from [76], shown in Figure 2.1. TOC uses three types of parameters for the person's configuration:

- Environment-driven parameters
- Uncontrollable parameters
- Free parameters

Environment-driven parameters are set according to the state of the environment. For a chair, the user's body would be in a seated configuration. For a flat bed, the user's body would be in a supine configuration. For a bed with an adjustable back rest, the user's configuration would depend on the angle of the back rest. Uncontrollable parameters are treated similar to uncontrollable movable objects. TOC selects a robot configuration for a sample of the uncontrollable parameter. An example of an uncontrollable parameter is the position of the user on the bed, if the robot is unable to shift the body on the bed. Free parameters are used by TOC freely without including it in the robot configuration. An example of a free parameter used is the user's neck rotation. Figure 2.2 shows a configuration of the PR2 that takes advantage of the user's neck rotation to reach all goals for the shaving task in a wheelchair.

Just as with the environment model, we include a margin of safety in the human model by expanding the model.

Additional User Customization

TOC can consider additional customizations for the user's needs or preferences. For example, a user may prefer certain angles of the bed's head rest for feeding tasks. This preference can be represented as limitations or costs on the robot's configuration space.

2.3.7 Configuration Scoring

Implicitly handling variation and error is a key aspect of TOC, because its heavy computation is performed offline for models that may differ from reality. TOC uses two metrics that we have developed to estimate how well the robot will be able to perform the task from a set of configurations: task-centric reachability (TC-reachability) and task-centric manipulability (TC-manipulability).

Task-centric Reachability

Task-centric Reachability (TC-reachability), P_R, is the percentage of goal poses to which the robot can find a collision-free IK solution from robot configurations, $\boldsymbol{r}$, for a task c and uncontrollable parameters h, as shown in Equation (2.3).

$$P_R(\boldsymbol{r}, c, h) = (\frac{1}{N_c}) \sum_{k=1}^{N_c} \max_{r \in \boldsymbol{r}, b \in \boldsymbol{b}} W(r, x_k), \tag{2.3}$$

where

$$\begin{gathered} W(r, x_k) = 1 \quad \forall \boldsymbol{s}_{r,x_k} \neq \emptyset, \\ \text{and} \\ W(r, x_k) = \ 0 \quad \forall \boldsymbol{s}_{r,x_k} = \emptyset. \end{gathered} \tag{2.4}$$

Recall that $\boldsymbol{x}$ depends on c, h, and b, but we omit those for simplicity in writing and N_c is the number of goal poses for task c. Note that $\boldsymbol{s}_{r,x_k} \neq \emptyset$ means that the IK solver can find a collision-free solution to the goal pose x_k from robot configuration, r.

TC-Reachability is related to using an IK solver with collision checking, but with the additional functionality of evaluating sets of robot configurations.

Task-centric Manipulability

Task-centric Manipulability (TC-manipulability), P_M, is related to the average kinematic dexterity of the arm when reaching the goal poses. It is defined here differently from our previous works, such as from [24].

TC-manipulability score is based on kinematic isotropy [32], shown in Equation (2.2). Kinematic isotropy only considers the Jacobian of the arm in a configuration, ignoring potentially relevant properties of the robot arm, such as joint limits. When at a joint limit, the arm cannot move in one direction, effectively halving the movement of that joint. [36] used torque-weighted global isotropy index and torque-weighted kinematic isotropy to estimate the dexterity of a robotic arm given joint torques and torque limits. [37] investigated configuration-based weighting functions to create what they call an augmented Jacobian that they use in manipulability. We have similarly modified kinematic isotropy to consider joint limits by scaling the manipulator's Jacobian by an nxn diagonal joint-limit-weighting matrix $\boldsymbol{T}$, defined in Equation 2.5, where n is the number of joints of the manipulator.

$$\boldsymbol{T}(\boldsymbol{q},\boldsymbol{q}^{-},\boldsymbol{q}^{+}) \;=\; \begin{bmatrix} t_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & t_n \end{bmatrix} \tag{2.5}$$

t_i in $\boldsymbol{T}$ is defined as

$$t_i = 1 - \eta^{\kappa}$$

where

$$\kappa = \frac{q_i^r - |q_i^r - q_i + q_i^-|}{\zeta q_i^r} + 1 \tag{2.6}$$

and

$$q_i^r = \frac{1}{2}(q_i^+ - q_i^-).$$

We set $t_i = 1$ for infinite roll joints. The variable η is a scalar value that determines the maximum penalty incurred when joint q_i approaches q_i^+ or q_i^- and ζ determines the shape of the penalty function. We used a value of 0.5 for η and $\frac{1}{20}$ for ζ. This weighting function and the values for η and ζ were selected to halve the value of the kinematic isotropy at joint limits, have little effect in the center of the joint range, to begin exponentially penalizing joint values beyond 75% of the range, and to operate as a function of the percentage of the joint range. Fig. 2.6 shows the value of t_i as a function of the joint value as a percentage of its joint range.

We then define joint-limited-weighted kinematic isotropy (JLWKI) as

$$\text{JLWKI}(\boldsymbol{q}) = \frac{\sqrt[a]{\text{det}(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{T}(\boldsymbol{q},\boldsymbol{q}^{-},\boldsymbol{q}^{+})\boldsymbol{J}(\boldsymbol{q})^{T})}}{(\frac{1}{a})\text{trace}(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{T}(\boldsymbol{q},\boldsymbol{q}^{-},\boldsymbol{q}^{+})\boldsymbol{J}(\boldsymbol{q})^{T})}. \tag{2.7}$$

We use a function, F, to find the maximum value of JLWKI$(\boldsymbol{q})$ for robot configuration

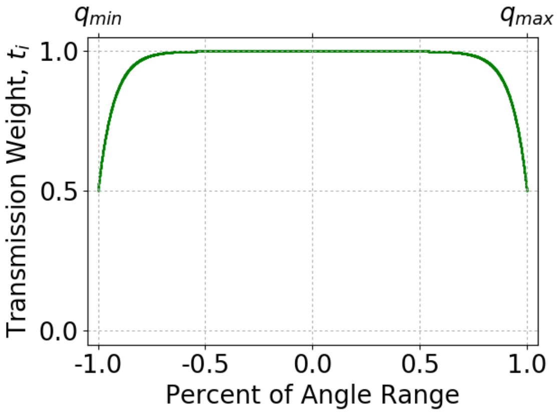


Figure 2.6: A plot of the joint-limit weighting function ranging from maximum joint value to the minimum joint value.

r and goal pose x_k, where

$$\begin{aligned} F(r, x_k) &= \max_{\boldsymbol{q} \in \boldsymbol{s}_{r,x_k}} \text{JLWKI}(\boldsymbol{q}) \quad \forall \boldsymbol{s}_{r,x_k} \neq \emptyset, \\ \text{and} & \\ F(r, x_k) &= \ 0 \qquad\qquad\qquad\quad \forall \boldsymbol{s}_{r,x_k} = \emptyset. \end{aligned} \tag{2.8}$$

We finally define TC-manipulability, P_M, as

$$P_M(\boldsymbol{r}, c, h) = (\frac{1}{N_c}) \sum_{k=1}^{N_c} \max_{r \in \boldsymbol{r}, b \in \boldsymbol{b}} F(r, x_k). \tag{2.9}$$

2.3.8 Optimization

TOC's optimization takes as input task-relevant models for task, c, and a sample of the uncontrollable parameters, h. It outputs an optimized set of robot configurations, $\boldsymbol{r}$. TOC runs this optimization for samples of the uncontrollable parameters for each task. TOC searches the robot configuration space to maximize its objective function, a linear combination of TC-Reachability and TC-Manipulability, shown in Equation 2.10.

$$\arg\max_{\boldsymbol{r}_i} \quad \alpha P_R(\boldsymbol{r}_i, h_i, c) + \beta P_M(\boldsymbol{r}_i, h_i, c) \tag{2.10}$$

Both TC-reachability and TC-manipulability range from 0 to 1, allowing them to be directly compared in the optimization. We selected a value of 1 for α. We chose to define β as

$$\beta(\boldsymbol{r}) = (0.1)(0.95)^{n-1} \tag{2.11}$$

where n is the cardinality of $\boldsymbol{r}$. In our implementation of TOC the cardinality of $\boldsymbol{r}$ was 1 or 2. These definitions of α and β emphasize the importance of reaching goals over being able to reach around goals, and includes a small penalty in the objective function's value for using more configurations. There are often many configurations that can reach all goals. TC-manipulability is used to differentiate between these configurations. As an example, we compare TOC to a standard method from literature: using an IK solver with a collision checker to find a robot configuration that can reach all goals. Figure 2.3 shows the difference in scoring between using the existence of IK solutions for scoring and using TOC for scoring for a task for a user in bed. Figure 2.3 (left) shows that many poses of the robot's base have the same score, each having collision-free IK solutions to all goals. Figure 2.3 (right) shows scoring using TOC, where TC-manipulability allows additional differentiation between robot base poses that can reach all goals. Higher TC-manipulability is correlated with mean accuracy (percentage of goals that are reachable), as we show in Section 2.4.4.

Search Method

The space of the objective function can be highly nonlinear and challenging to search. There are several derivative-free, simulation-based optimization methods that could be applied to this problem. A simple method would be to uniformly sample the space and select the configuration with highest objective function value. However, we found that uniform sampling had difficulty finding good configurations for tasks where the solution space was small. [77] used Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to design a controller for articulated bodies moving in a hydrodynamic environment, which inspired our use of CMA-ES (from `https://pypi.python.org/pypi/cma`) for our optimization. We used a heuristic when both P_R and P_M are zero that pushes the search toward configurations that may have non-zero P_R and P_M. All values of the heuristic are less than 0.

2.3.9 Approximate Function

Offline, TOC approximates a function that takes as input an estimation of the uncontrollable parameters, $\hat{h}$, such as the location of the person on the bed, and outputs the estimated optimal robot configurations, $\hat{\boldsymbol{r}}^*$. At run-time, TOC applies this function to the observed, estimated uncontrollable parameters. We used K-nearest neighbor (K-NN) with $K = 1$ (hence, 1-NN) as the function, f. We trained the 1-NN algorithm with a set of (h, r) pairs and it returns as $\hat{\boldsymbol{r}}^*$ the r for the h that is closest to $\hat{h}$. In our implementations of TOC we trained the 1-NN on fewer than 20 (h, r) pairs for each task and found that the 1-NN would return $\hat{\boldsymbol{r}}^*$ in less than 1 second.

2.4 Evaluation

2.4.1 Implementation

We manually created models for 9 assistive tasks: shaving, feeding, wiping the mouth, cleaning both arms, cleaning both legs, and scratching the left/right upper arm, and left/right knee (each scratching task was considered separately). Previous work has noted that these types of tasks may be useful for those with severe motor impairments [15]. As described in Section 2.3.4, task models consisted of a set of goal poses, each of which was a position and orientation goal for the robot's end effector. We defined each goal pose with respect to a relevant reference frame (e.g., the head for shaving, or the shoulder for scratching the upper arm), so they move appropriately as the model parameters change (e.g., the height of the bed). We chose these tasks as representative of various activities for which a robot like the PR2 may be able to provide assistance to a user with motor impairments.

For example, Figure 2.5 shows the eight goal poses with respect to the person's head that

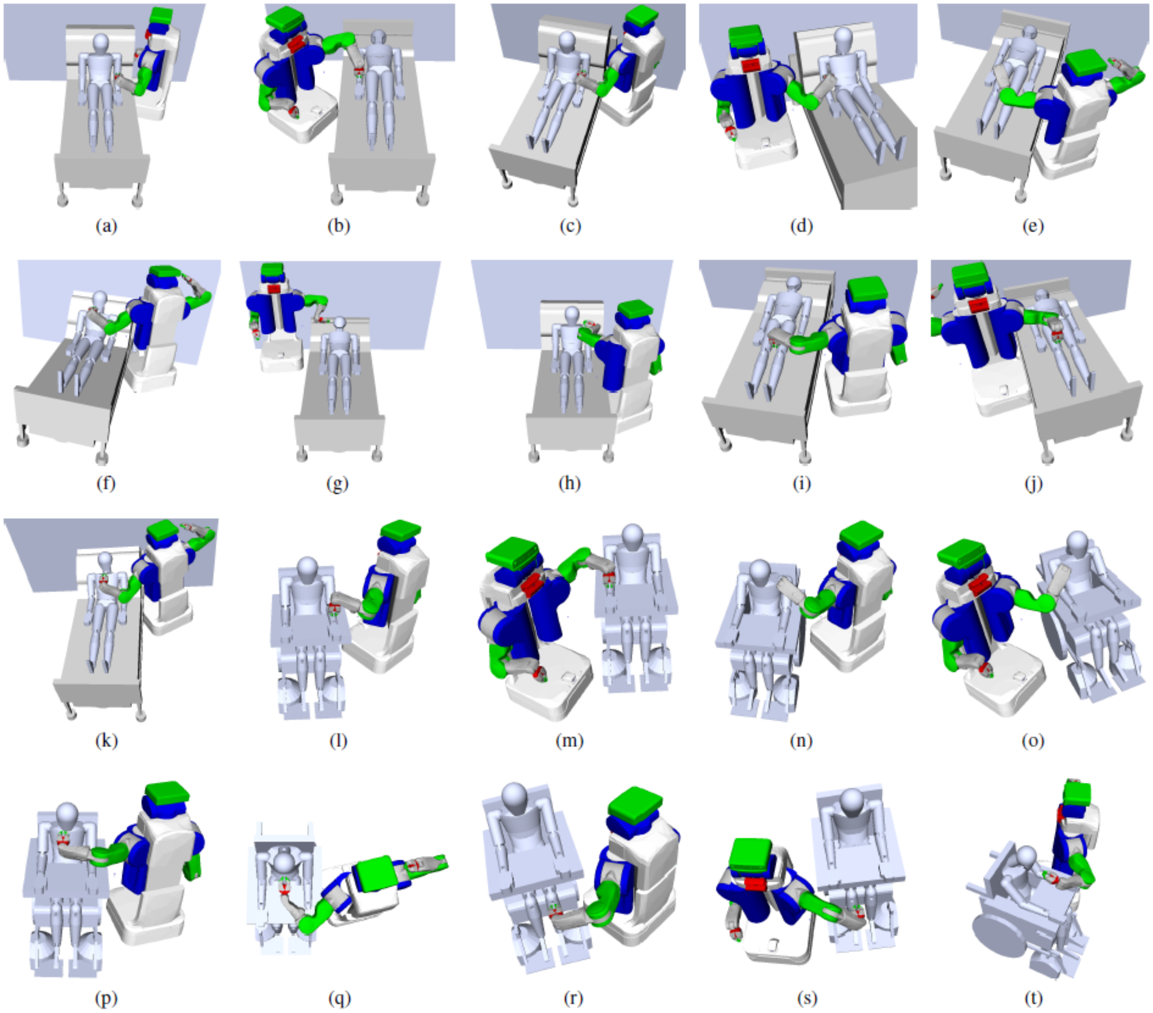


Figure 2.7: Visualization of the robot configurations selected by TOC for each task. Two images are used when TOC selected two configurations for a task task. The images show for the robotic bed environment: (a) cleaning arms config #1 (b) cleaning arms config #2 (c) scratching left upper arm (d) scratching right upper arm (e) cleaning legs (f) wiping mouth (g) shaving config #1 (h) shaving config #2 (i) scratching left knee (j) scratching right knee (k) feeding. The images show for the wheelchair environment: (l) arm cleaning config #1 (m) arm bed config #2 (n) scratching left upper arm (o) scratching right upper arm (p) wiping mouth (q) shaving (r) scratching left knee (s) scratching right knee (t) feeding.

we selected to model the shaving task. For simplicity, we limited tasks to one-handed tasks and used only the robot's left arm in our evaluations. In our implementation we allowed TOC to search for sets of robot configurations of cardinality 1 or 2. When exploring multiple robot configurations for a task, we assume the robot can move from one configuration to another.

We ran all simulations in OpenRAVE (`http://www.openrave.org/` from [53]), for which we created environment models with a PR2 robot and a model of an average male human placed either in a wheelchair or in a robotic bed. The human model dimensions come from [76]. The PR2 is a mobile manipulator made by Willow Garage with two 7-DoF arms. The models we created for the robotic bed and the wheelchair match Autobed, a modified Invacare 5401IVC full electric hospital bed [74] and a Sunrise Medical Quickie 2 wheelchair with overlap table, respectively. The casters on the bed and wheelchair are represented by swept volumes. For the wheelchair, we removed the part of the casters' swept volumes that extends to the sides of the chair to increase free space around the chair. We assume that the user would ensure that the casters are not pointing out from the chair.

Figure 2.7 shows the configurations selected by our implementation of TOC for each task, given the observation, $\hat{h}$, that the person was positioned in the center of the bed or wheelchair.

The robotic bed environment has the bed in front of a wall, to emulate how beds are often positioned in rooms. The robotic bed can raise up to $25\,\mathrm{cm}$ and can increase the angle of its head rest up to $75°$. For the wheelchair environment we gave the human the ability to rotate its neck up to $45°$ in either direction about the Z-axis. Figure 2.2 shows the neck rotated $45°$. These two environments (robotic bed and wheelchair) were selected to demonstrate many of the functionalities of the TOC framework. The robotic bed environment demonstrates how TOC can select configurations for multiple robots in the environment, how TOC can handle controllable and uncontrollable parameters of the environment, and how it handles uncontrollable and environment-driven parameters of the user. TOC treats additional robots the same as controllable objects, adding their parameters to the robot configuration, as it does with the robotic bed's parameters. TOC considers the position of the person on the bed as an uncontrollable parameter and considers the other parameters of the user (the configuration of the person's joints) as environment-driven parameters. The wheelchair environment demonstrates how TOC can make use of free parameters in the user model.

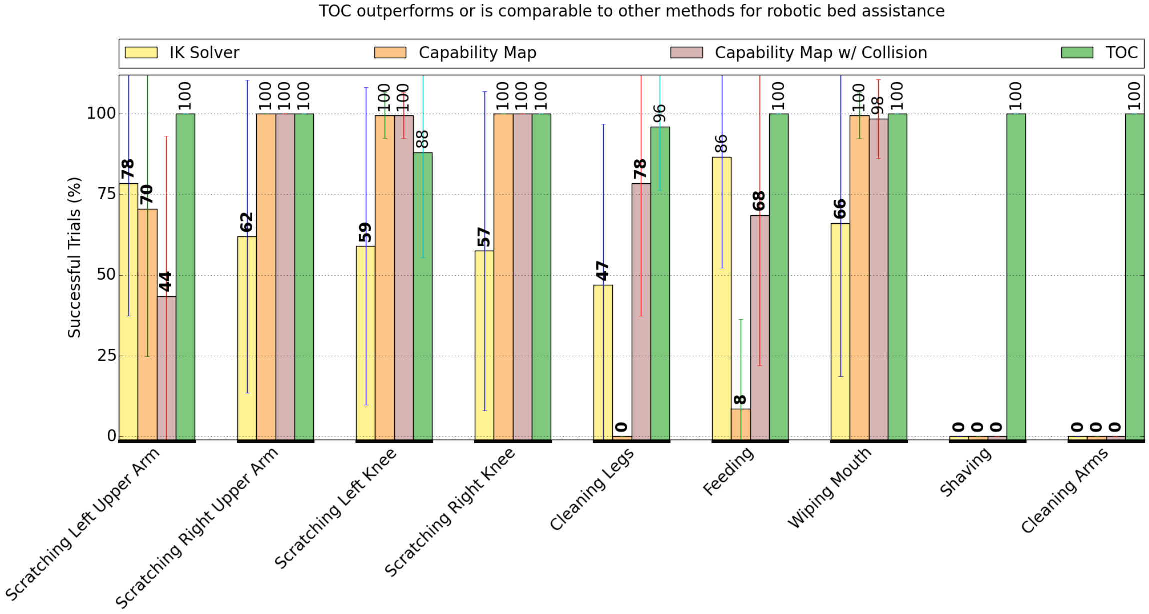


Figure 2.8: Comparison of performance between TOC and three baseline methods averaged over 200 Monte-Carlo simulations of state estimation error for tasks in the robotic bed environment. Bold numbers have statistically significant ($p < 0.01$) difference from the TOC result in a Wilcoxon Rank-Sum test. Error bars show one standard deviation. TOC chose to use a single configuration for all tasks other than the shaving and cleaning arms tasks in this environment. Baseline methods could only select a single configuration.

TOC considers the rotation of the person's neck as a free parameter.

2.4.2 Evaluation Against Baselines

We compared the performance of TOC against three baseline methods in Monte Carlo simulations of Gaussian error introduced in the person's position (e.g., translating around on the bed while the bed remains stationary) and in the robot's base pose (e.g., translating and rotating the robot from the selected robot configuration). Each method estimated an optimal set of robot configurations, $\hat{\boldsymbol{r}}^*$, given the observation, $\hat{h}$, that the person was positioned in the center of the bed or wheelchair. Because the goals are sparse and represent a more complicated task, we considered a trial successful if, from the robot configurations selected by the method, the PR2 could reach all goal poses despite the error introduced in the Monte Carlo simulation. Otherwise, the trial was a failure. We performed this evaluation for all 9 modeled tasks in both environments except the cleaning legs tasks for the wheelchair

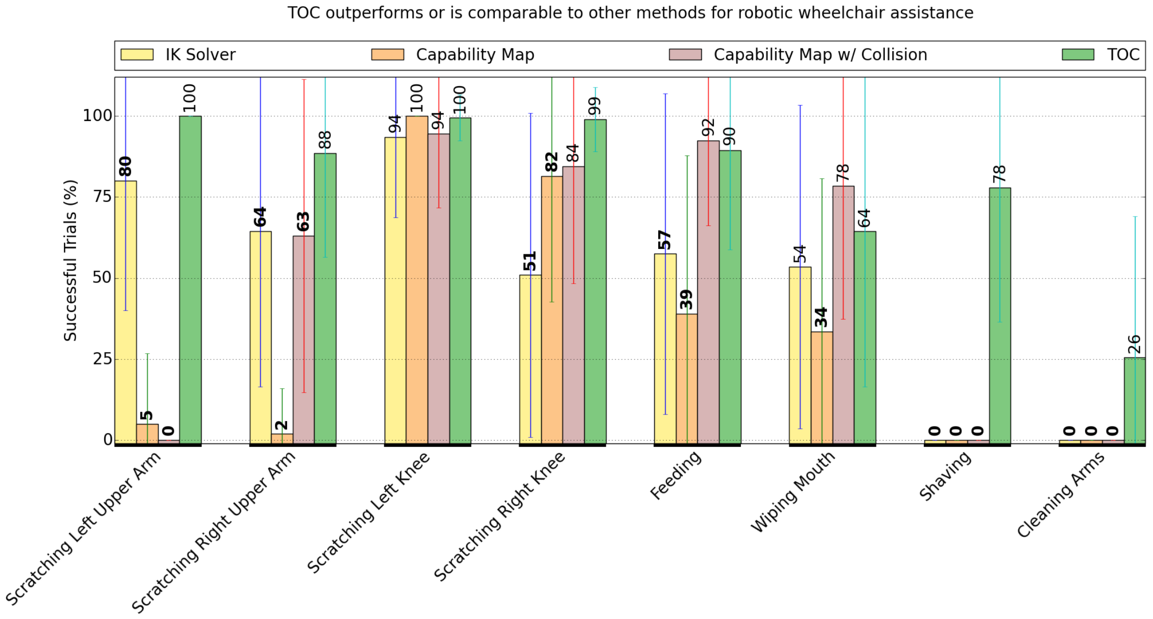


Figure 2.9: Comparison of performance between TOC and three baseline methods averaged over 200 Monte-Carlo simulations of state estimation error for tasks in the wheelchair environment. Bold numbers have statistically significant ($p < 0.01$) difference from the TOC result in a Wilcoxon Rank-Sum test. Error bars show one standard deviation. TOC chose to use a single configuration for all tasks other than the cleaning arms task in this environment. Baseline methods could only select a single configuration.

environment because the overlap table blocks access to the thighs, preventing successful performance of the task.

All introduced error was normally distributed around 0. For the robotic bed environment, the standard deviation for the human's pose was $2.5\,\mathrm{cm}$ translation in the global X direction and $5.0\,\mathrm{cm}$ translation in the global Y direction. Rotations of the human in bed were not considered because small rotations about the head results in large movements of the legs. For the wheelchair environment, the standard deviation for the human's pose was $2.5\,\mathrm{cm}$ translation in the global X direction, $5.0\,\mathrm{cm}$ translation in the global Y direction, and $5°$ rotation about the human head's Z axis. The standard deviation for the PR2's position was $1.0\,\mathrm{cm}$ in the global X and Y directions and $5°$ rotation about the robot's Z axis. We selected these error distributions from typical error in human pose estimation and PR2 servoing in our previous work [75]. As described in Sections 2.3.5 and 2.3.6, models used for selecting configurations had a small safety margin of $\sim 3\,\mathrm{cm}$. Models used for testing had no safety margin.

For fair comparison, all methods in this evaluation were given matching seeds for their optimization via CMA-ES as well as for their error in Monte-Carlo simulation. For the CMA-ES optimization, all methods were given a population size of 40, a maximum number of iterations of 1000, and the opportunity to restart with double the population if the optimization ran out of iterations before converging within some tolerance. All methods also had the same heuristics for driving the search towards configurations that may have collision-free IK solutions. We assigned appropriate bounds on parameters based on the environment (e.g., slightly beyond reach of the bed). We initialized the parameters to aid coverage in the search, giving two initial locations, one position on one side of the bed or wheelchair and one on the other side. Baseline methods were allowed two searches, one for each initialization, and selected the single best configuration. TOC jointly optimized its two configurations from their respective initialization locations. Thus, all methods were given the comparable initializations and bounds.

Baselines

We implemented three baselines from literature to compare against TOC, one based on IK and two based on the robot capability map. An overview of these and other related methods from literature can be found in Section 2.2. These methods selected a single configuration for both the PR2 and the robotic bed and made use of the human's free parameter (neck rotation) in the wheelchair environment.

Inverse-Kinematics (IK) Solver-based Baseline IK solver-based methods to select a robot base pose for a task are common in literature. The method we implemented uses CMA-ES to search for a robot configuration where the robot has a collision-free IK solution to all goal end effector poses. We used the ikfast module within the OpenRAVE simulation

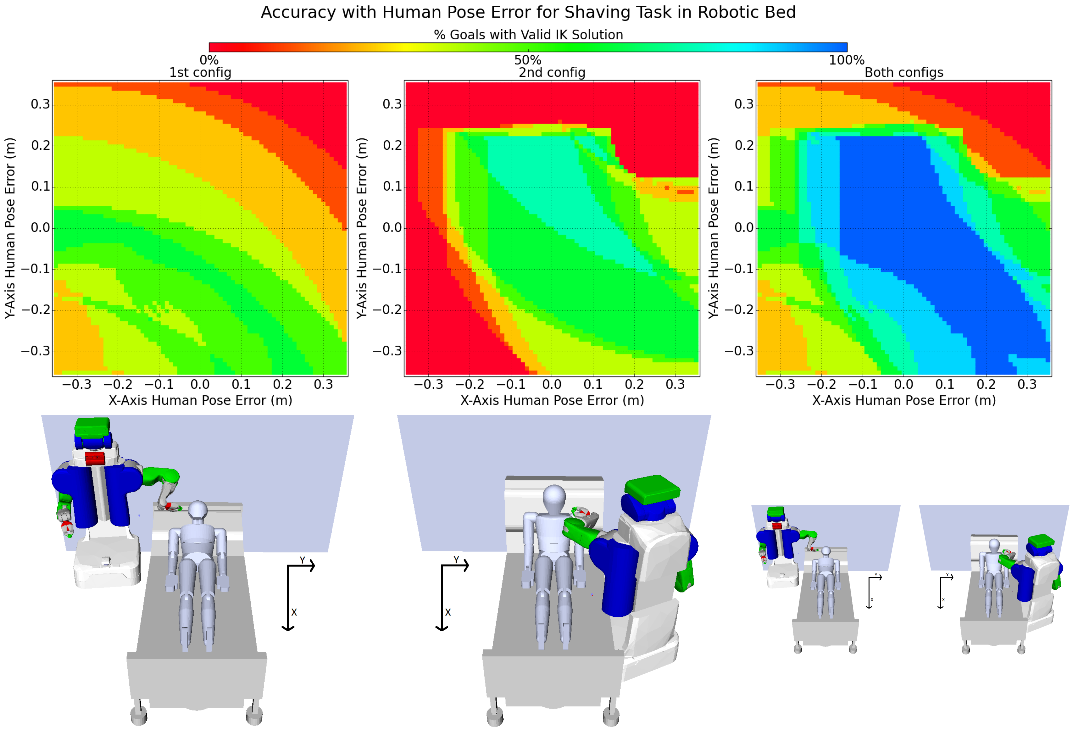


Figure 2.10: Visualization of the robustness of TOC's selected configurations for the shaving task in the robotic bed. (Top) Percentage of goals reached from the first, second, and both configurations for error in 1cm increments in the x-y position of the person. (bottom) The first and second configurations of the PR2, and the two configurations combined on the right. Color is necessary to interpret this figure. The blue region represents when all goals can be reached.

environment to determine if a collision-free IK solution existed for each robot configuration.

Capability Map-based Baselines Various methods from literature use the capability map [26]. We implemented two baseline methods roughly based on [49]. For these methods we first created a capability map using OpenRAVE's kinematic reachability module. To create the capability map, the module discretized 3D space around the robot's arm into 3D points and discretized the range of possible orientations around each 3D point. The capability score (also known as reachability score) for each point is the percentage of orientations for which the robot has a valid IK solution. These scores are calculated offline and saved. These two methods use CMA-ES to search for a robot configuration that maximizes the average capability score for all goal poses. The score of a goal pose is the score of the

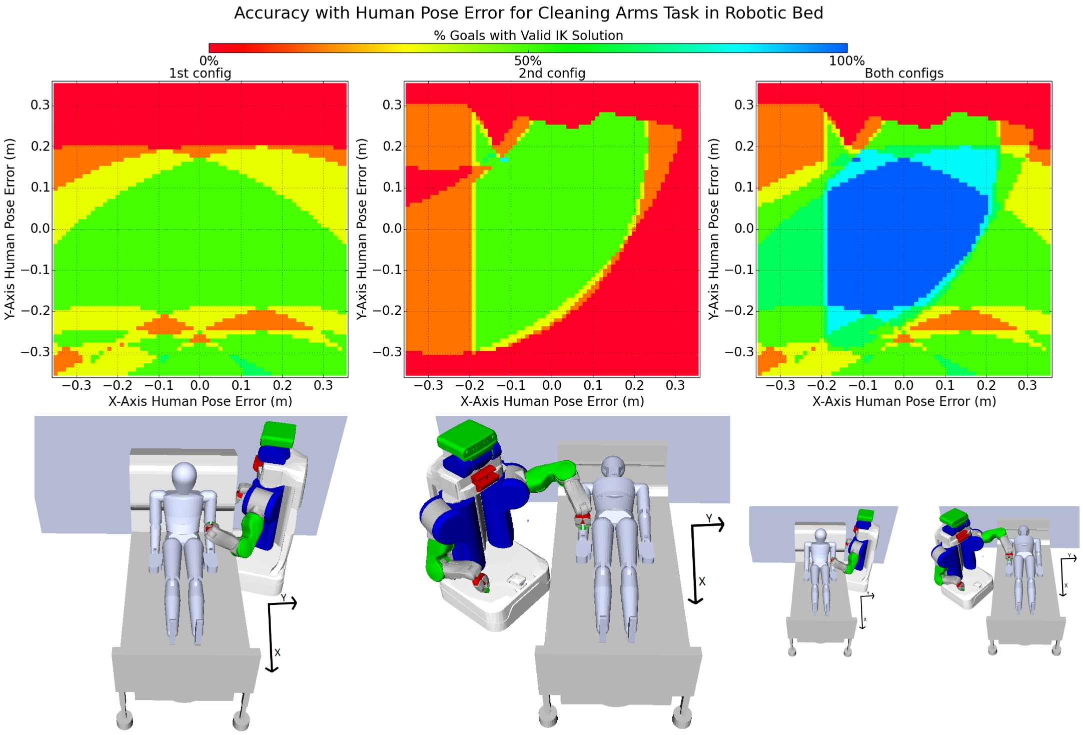


Figure 2.11: Visualization of the robustness of TOC's selected configurations for the arm cleaning task in the robotic bed. (Top) Percentage of goals reached from the first, second, and both configurations for error in 1cm increments in the x-y position of the person. (bottom) The first and second configurations of the PR2, and the two configurations combined on the right. Color is necessary to interpret this figure. The blue region represents when all goals can be reached.

closest 3D point from the capability map. The first capability map-based baseline considered capability scores without regard to the environment. The second gave goal poses a 0 score if a collision-free IK solution could not be found to that pose in the environment.

Results

The results for each task for the robotic bed and wheelchair are shown in Figures 2.8 and 2.9, respectively. TOC's average success rate was higher than or comparable to baseline methods in all tasks. Statistically significant difference from the TOC result ($p < 0.01$ in a Wilcoxon Rank-Sum test) is indicated in the figures with bold numbers.

TOC had an overall average success rate of 90.6%, compared to 50.4% for IK, 43.5% for capability map, and 58.9% for capability map with collision checking. The overall

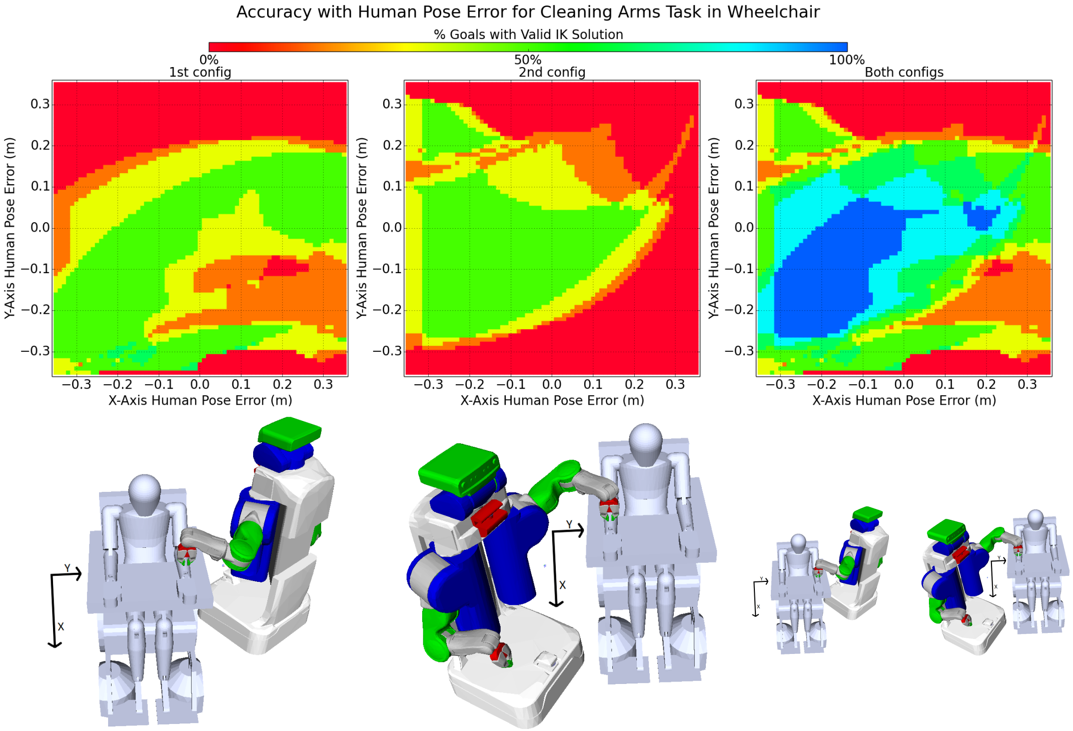


Figure 2.12: Visualization of the robustness of TOC's selected configurations for the arm cleaning task in the wheelchair. (Top) Percentage of goals reached from the first, second, and both configurations for error in 1cm increments in the x-y position of the person. (bottom) The first and second configurations of the PR2, and the two configurations combined on the right. Color is necessary to interpret this figure. The blue region represents when all goals can be reached.

differences between baseline results and TOC are statistically significant ($p < 0.01$) in a Wilcoxon Rank-Sum test. TOC chose to use a single configuration for all tasks other than the shaving and cleaning arms tasks in this environment in the bed environment and it used a single configuration for all but the cleaning arms task in the wheelchair environment. TOC achieved higher average success rates both for tasks for which it used one and two configurations. This result suggests that benefit from TOC comes from more than just from using 2 configurations over one. For tasks that require more than one robot configuration, baseline methods failed; they could only select a single configuration. TOC jointly optimizes 2 configurations, allowing it to succeed in these challenging tasks.

2.4.3 Quantifying Robustness

In Figures 2.10, 2.11 and 2.12 We visualize the robustness of robot configurations selected by TOC for the shaving and cleaning arms tasks in the two environments. These figures show the percentage of goal poses that have collision-free IK solutions (indicated by the color) for varying error in the person's position on the bed or wheelchair (the X-Y axes). Notable in these figures is the success region in blue, where all goals are reachable, as well as how the two configurations combine to reach all goals. For pose estimation error in the success region, the PR2 would still be able to successfully perform the task. TOC opted to use two configurations for each of the tasks shown. The success region is large and surrounds the origin for shaving and cleaning arms in bed, which is why 100% of the trials were successful for these tasks in Figure 2.8. The success region is less centered around the origin for the cleaning arms task in the wheelchair, hence its lower percentage of successful trials in Figure 2.9. The Monte Carlo simulations randomly sampled in these, as well as other, degrees of freedom and sampling outside the success region results in a failed trial. These figures suggests that the task may be easier for the robot to perform for a person in bed. Closer observation of the task shows that, because the wheelchair is tall, the goals poses for the cleaning arms task are vertically higher in the PR2's workspace and the arm have relatively low JLWKI when reaching those goals.

2.4.4 Evaluation of TOC Objective Function

TOC searches for a set of robot configurations that maximizes its objective function, which we will call its score for simplicity. The assumption therein is that higher values of the score are correlated with better robot configurations, that are more robust to error. To test this assumption, we evaluated the relationship between the TOC score and the accuracy (the percentage of goals that are reachable) for robot configurations in the same evaluation described in Section 2.4.2. Note that we chose to compare against accuracy in this evaluation because it can convey more information than success, which is binary. A similar correlation

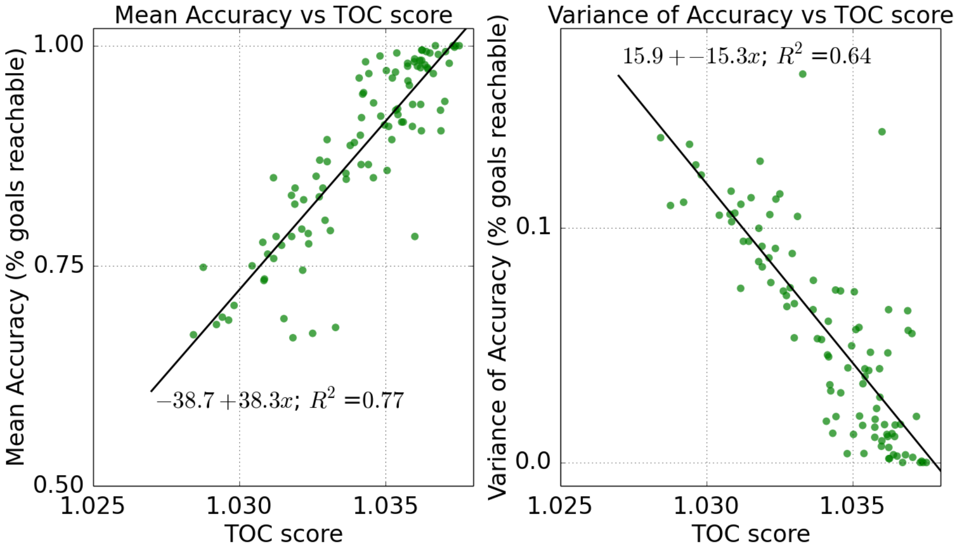


Figure 2.13: Increasing TOC score is correlated to accuracy and inversely correlated to variance in accuracy. TOC score above 1.0 means a collision-free IK solution exists to all goals. The amount above 1.0 is the weighted TC-manipulability score for the configuration.

can be seen for success. For the wiping mouth task in the robotic bed, we sampled robot configurations with TC-reachability of 1 (i.e., all goal poses have collision-free IK solutions) and compared their mean and variance in accuracy over 200 Monte Carlo simulations with their TOC score. Figure 2.13 shows the results of the analysis. Higher TOC score is correlated with accuracy and inversely correlated with variance in accuracy.

2.5 Discussion

We have shown results with a single type and source of error. However, we expect TOC to be able to deal with many types and sources of error. If we knew all sources of error apriori, we could explicitly model them, to improve selection of robot configurations. Such a method would be similar to those from [56] and [57], described in Section 2.2.2. It should be noted, however, that movement of the person (e.g., pose estimation error) does not directly translate into motion of the robot's base, so directly using these evaluations as part of a method to select the robot's configuration can be difficult. Our results provide evidence that TOC can often select robot configurations that are more robust to error in the person's pose than baseline methods. This is seen in many of the tasks and environments we examined, but is most pronounced in more challenging tasks that may need more than a single robot configuration to complete.

We observed that baseline methods work well for many tasks and they selected configurations, on average, faster than TOC. For its offline computations, TOC took on average 70 minutes for each task running in a single thread on a a 64-bit, 14.04 Ubuntu operating system with 8 GB of RAM and a 3.40 GHz Intel Core i7-3770 CPU. The IK solver baseline we implemented runs faster, on average 6 minutes and often less than 1 minute for each task. These results suggest that this baseline may work well for many tasks that only require a single robot configuration and where there is little error between model and reality. The capability map with collision detection also performed well on many tasks and took on average 34 minutes to select a configuration. Note that this is the time to run these algorithms either offline within our framework or online. As in assistive robotics, in many robotics applications the speed of online selection is important and 6 minutes, 34 minutes, and 70 minutes are all too long for a user to wait for a robot to decide where to move. Additionally, there may be specifically important or common tasks that take place in environments that are known beforehand. In this case, it can be valuable to take additional time offline to find solutions, using our framework, to take advantage of prior knowledge and speed up the responsiveness of the online process.

We sought to handle fairly the comparison between TOC and baseline methods by using the same sampling for each. However, because of the nature of the search problem, using other search methods, additional heuristics, or other meta-parameters for the search (e.g., different population size) may find better configurations than those found in our evaluations.

We observed a trade-off when selecting robot configurations. In general, moving the PR2 closer tends to improve the arm's dexterity, but tends to make collisions more likely. An assumption in this work is that contact is bad and should be avoided. However, [78] found that contact can be both beneficial and acceptable during robotic assistance. Allowing contact can increase the space of reachable poses, and there are methods for controlling contact safely [79].

2.5.1 Design Application

TOC may also be used to assist in the design of environments. For example, in our evaluations with the robotic bed, TOC selected a configuration for the bed by including the bed's DoF in the robot configuration. TOC could similarly optimize many other continuous controllable parameters of the environment. Comparison of TOC scores between a robotic bed and a standard, static bed may demonstrate the value added by a robotic bed. By recognizing that for some tasks a lower bed height improves performance and for other tasks a higher bed height improves performance, we may recognize that an adjustable bed may be preferable to allow the bed to reconfigure to the desired height for each task. As detailed in Section 2.3.5, TOC can be used to assist in design for parameters with discrete choices by including it as an uncontrollable parameter in the optimization. In this way, you might decide to put the bed against the wall, 0.5 m away from the wall, or 1 m away from the wall. For example, we have found that for the shaving task, if there is sufficient space behind the bed, the PR2 can perform the task from a single configuration instead of requiring two configurations. Figure 2.14 shows the configuration found by TOC for shaving in the robotic bed without a wall and visualizes the robustness to error in the pose of the person.

2.5.2 Limitations

There are some limitations to our work with TOC and our evaluation. Although the framework of TOC allows it to search for sets of robot configurations of cardinality larger than 2, we limited it to two in our evaluation. We made this choice because we found that more than two robot configurations were not needed for any of the tasks examined.

We hand designed the task models for our evaluation in simulation, but have not investigated how well the task models actually represent the tasks. In addition, all tasks in this

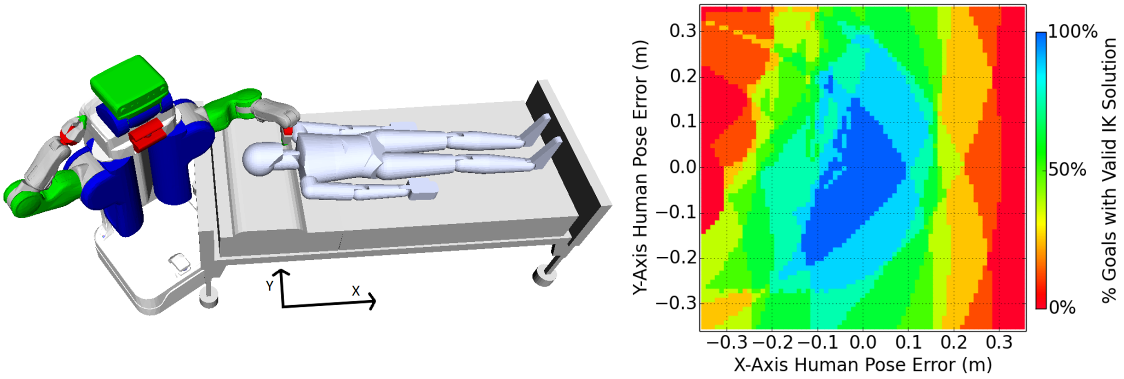


Figure 2.14: The PR2 can perform the shaving task on the person in bed from a single location if there is no wall behind the bed. (Left) the configuration TOC selected (right) a visualization of the robustness to error in the person's pose on the bed for this configuration.

work were defined with full 6-DoF goal poses, although some tasks, such as sponge baths, have position requirements and few orientation requirements. We also did not address tasks with complex motions in which the trajectory between goals poses is important, such as dressing, or tasks with high strength requirements, such as lifting or ambulating.

Joint-limit-weighted kinematic isotropy, the foundation of our TC-manipulabilty score, does not account for environmental constraints. There may be value in preferring joint configurations away from obstacles. We have mitigated the risk of collisions using a safety margin on the environment and user models at the cost of decreasing the valid search space and eliminating valid solutions. Explicitly penalizing proximity to collisions in the objective function, as was done by [37], may be another way to mitigate this issue at the cost of additional computation time. TOC also does not determine if there are valid paths to reach collision-free IK solutions, which may be problematic when there are environmental constraints. A motion planner could be used to check for valid paths at the cost of computation time. Although computation time is often a less critical concern for offline processes, it must remain reasonable. We elected to ignore proximity to collisions and the question of the existence of valid paths to achieve lower computation time.

2.6 Conclusion

In this chapter, we have presented task-centric optimization of robot configurations (TOC), a method to select one or more configurations for robots to assist with tasks around a person's body. TOC uses TC-reachability and TC-manipulability, metrics that we have developed, to represent the robot's dexterity, and implicitly handle error. TOC is particularly suitable for assistive tasks, where there are a set of desired tasks known apriori that can be modeled as a set of end effector poses with respect to relevant reference frames. We have shown that TOC can determine a set of one or two robot configurations from which the robot can perform a task well. TOC performs the bulk of its computation offline using models of the task, robot, environment, and person to generate a function that rapidly (≤ 1 second) estimates the optimal set of robot configurations for a task given observations at runtime. We provide evidence that configurations selected by TOC are robust to state estimation errors between the models used offline and observations at runtime. We created 9 models of assistive tasks to test our system in simulation and showed that for each task TOC's average success rate was higher than or comparable to three baseline algorithms from literature. TOC had an overall average success rate of 90.6% compared to 50.4%, 43.5%, and 58.9% for baseline methods.

CHAPTER 3
A SYSTEM FOR BEDSIDE ASSISTANCE THAT INTEGRATES A ROBOTIC BED AND A MOBILE MANIPULATOR

3.1 Introduction

Illnesses, injuries, long-term disabilities, and other situations can result in people receiving physical assistance while in bed. For example, patients in hospitals and people with severe disabilities living at home may spend substantial time in bed. Some robotic systems exist, often using fixed-base manipulators (a robot arm in a fixed location) that can provide assistance to a user in bed [11], but this setting comes with challenges. The width of the bed can make it difficult for a manipulator (a robotic arm) to reach task-relevant locations around the human body. Moreover, people who spend substantial time in bed often use overbed tables and other nearby furniture that can interfere with perception and manipulation. A mobile manipulator (a robot with at least one manipulator arm and a mobile base) can mitigate some difficulties by allowing the robot to move around to access different areas around the body, but it increases complexity. In our previous work, using a teleoperated system with a mobile manipulator without autonomous functions, Grice et al. [73] observed that a significant amount of the time spent on each task was dedicated to moving the mobile robot into an appropriate position and moving the robot's gripper near the task area. A robotic bed could collaborate to help the mobile manipulator with providing assistance, but it further increases the degrees of freedom of the combined system. Finding a good pose for the base of the mobile robot and a configuration for the robotic bed can be challenging due to complex geometry of the bed and robot, a large number of degrees of freedom (DoF), task complexity and other factors, as discussed in our previous work [23].

In this chapter, we present a robotic system to provide physical assistance to a person in bed, which extends our prior work [75, 25, 73]. The system consists of a robotic bed and a mobile manipulator, specifically a PR2 robot, that collaborate to assist with tasks around the user's body. To manage the system's complexity for common assistive tasks, our system provides autonomous functions with a coarse-to-fine approach. The system autonomously completes the "coarse" parts of the task, such as moving the mobile robot's base to the appropriate place and configuring the robotic bed, and then gives control back to the user for "fine" execution, the detailed performance of the task. In our previous work, using a teleoperated system without autonomous functions, Grice et al. [73] observed that a significant amount of the time spent on each task was dedicated to moving the mobile robot into an appropriate position and moving the robot's gripper near the task area. This "coarse" setup can be challenging for a user, and we can reduce the user's overall workload by having it performed autonomously. The detailed performance of the task can be challenging to perform autonomously; for example, in a scratching task, the precise location of an itch can be difficult for the robot to find autonomously. The system makes use of the user's cognitive capabilities in providing fine control.

When a user commands the system to assist with a task, the system creates and executes a plan. The plan typically begins with the mobile robot finding the relative location of the bed. Then the robotic bed estimates the position of the user's body using a pressure sensing mat and reconfigures itself to position the person's body for the task. The mobile manipulator moves to a position with respect to the person's body from which it can reach task-relevant areas. Then, the manipulator reaches out autonomously to the task area if it can do so easily and safely. At this point the autonomous function ends and the user takes control of the robot via a web-based interface.

With this system, we introduce and investigate the potential for a robotic bed to collaborate with a mobile manipulator in order to provide more effective assistance to people in bed. Dental hygienists, barbers, and other professionals who perform tasks around the human body sometimes position peoples' bodies using adjustable furniture. By doing so,

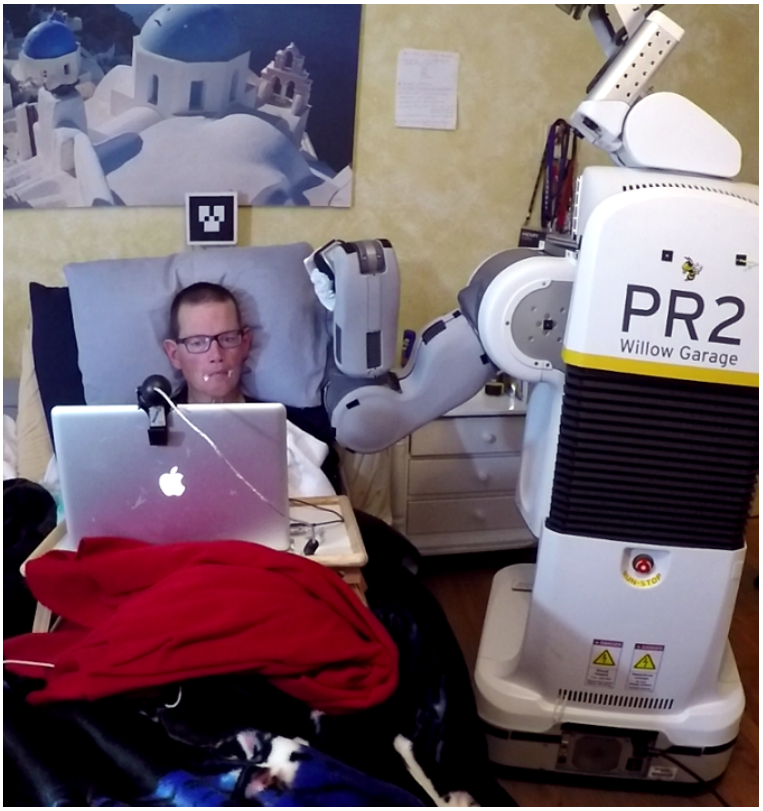


Figure 3.1: The system in use in a person's home. Henry Evans, a person with severe quadriplegia, used our system in his own home to wipe yogurt from his mouth, visible as white dots on his face.

the professional can improve ergonomics and the quality of the services they perform. The two robots in our system coordinate in an analogous manner. We demonstrate that a robotic bed and a mobile manipulator have complementary physical and perceptual capabilities. The robotic bed can move the human body using a small number of degrees of freedom into positions that are more reachable or relevant to the task, such as sitting the person up to eat food. It can also help the mobile manipulator reach the human body by raising itself above the ground so that the mobile manipulator's base can go under it or give the manipulator better access to the person. Our mobile manipulator can dexterously manipulate a lightweight payload using a large number of degrees of freedom. The robotic bed can also perceive the human body via a pressure sensing mat, while the mobile manipulator would typically perceive the person using on-board line-of-sight sensors that can be obstructed by bedding and other objects around a bed, such as an overbed table or IV lines.

We performed three evaluations of our system. We conducted this research with approval from the Georgia Institute of Technology Institutional Review Board (IRB), and obtained written informed consent from all participants. First, we conducted a study with 8 able-bodied participants to evaluate the ability of our robotic bed to estimate the position of the human body on the bed. We observed in our second and third evaluations that the position estimation error in the direction of the width of the bed, 5.00 cm $\pm$ 2.54 cm (mean ± std), is sufficiently small for our system to still succeed in tasks without the user moving the mobile manipulator's base or the bed's configuration from that attained autonomously. In our second evaluation, we used a medical mannequin to evaluate the extent to which the collaboration between the robotic bed and the mobile manipulator improved performance. We found that the each of the complementary capabilities of the robotic bed improve the system's ability to successfully reach task-relevant poses. In our third evaluation, we tested the system with Henry Evans, a person with severe quadriplegia, in his home in California, USA (see Fig 3.1). We personalized the system for him by creating models of his bedroom, his body, and the accessible user interface he uses with his laptop computer. Henry successfully performed 3 tasks 5 times each in a formal evaluation. Henry's feedback on the system was positive, including the statement "It works well and is very easy."

3.2 Related Work

3.2.1 Assistive Mobile Manipulation

Researchers have long explored the idea of assistive robots, especially for people with motor impairments [80, 81, 82, 83]. Some researchers have investigated the use of mobile manipulators as assistive devices, such as in [68, 69, 71, 72, 16]. Recently, several studies have introduced general-purpose mobile manipulators for various assistive robotic tasks, including shaving, picking-and-placing, and guiding tasks [15, 16, 84, 70, 85, 86]. However, we are unaware of other research that has investigated the potential for a mobile manipulator to collaborate with a robotic bed while providing assistance.

3.2.2 Selecting Robot Configurations for Mobile Manipulation

Much prior research has investigated how to find good configurations for a mobile robot. A common method is to address the problem using inverse-kinematics (IK) solvers [38, 39, 40, 41]. IK solvers typically use kinematics equations and analytical or numerical methods to find a set of joint angles for a robot arm that result in the robot's end effector being in a desired pose. Often, sampling-based methods are used to find robot base poses that have valid IK solutions, often as part of motion planning [44, 45, 46, 48]. Much work is based on the capability map presented by Zacharias et al.[26]. To create the capability map they discretize space around the robot into 3D points and discretize the range of possible orientations around each 3D point. The capability map is the scoring of discretized 3D points using the percentage of discretized orientations around each point for which the robot has a valid IK solution. If a goal pose is located at the 3D location, the capability score is similar to the probability that the robot can achieve the pose. Capability maps are task generic and robot specific, facilitating applications of the robot to new tasks. Methods using capability maps typically select the robot base position by overlapping the capability map with end effector goal poses and maximizing the average capability score [49, 50, 51]. In this work, we use task-centric optimization of robot configurations (TOC), a method from [25] to select one or two robot configurations for the mobile manipulator and bed to perform each task. A robot configuration is the position and orientation of the robot's mobile base, the z-axis spine height of the mobile robot, and the degrees of freedom of the bed. TOC, unlike capability maps, is task specific, robot specific, and user specific. These properties allow our system to be customized to a specific user and environment. Other methods to select configurations for robots include using data-driven simulation [56, 57] and inverse-reachability maps [53, 54, 55]. These data-driven simulation-based methods explicitly model error and fully simulate task performance, but they have only been used to select a few degrees of freedom in static environments and can only select a single robot configuration for a task. Inverse-reachability-based methods are typically used in task-agnostic and robot-specific ways to facilitate applications of the robot to new tasks. Notably, these methods only find a single base pose for a robot for a given task. In contrast, TOC is designed specifically for assistive robotics. TOC explicitly models features and parameters of the environment, user, and task, that may be important to task performance. TOC implicitly handles error and can be used to select more one or two configurations consisting of more degrees of freedom.

3.2.3 Collaborative Robots

Many investigations have explored multi-robot, collaborative systems [87, 88], including heterogeneous multi-robot systems [89, 90]. Some surgical robots feature collaboration between heterogeneous robots, such as robotic surgical arms and a robotic surgical table [91, 92]. Our approach is similar, applying the idea of heterogeneous multi-robot collaboration to the field of assistive robotics.

3.2.4 Robotic Beds for Physical Assistance

Several groups have developed robotic beds that assist with or prompt a user to roll over using rollers or actuators parallel to the longitudinal axes of the body [93, 94]. Others have constructed novel robotic beds with greater control of the position of the body [95, 96]. In this work, we have used Autobed [74], a modified hospital bed, as an agent in a collaborative heterogeneous multi-robot system.

3.2.5 Body Pose Estimation from Bed Sensors

A challenge in providing assistance with ADLs for a person in bed lies in perceiving the position and orientation of various body parts, which are often occluded by bedding. Researchers have fit 3D models to the output of pressure-sensing arrays to estimate the pose of a person lying on a mattress [97, 98]. Liu et al. used pictorial structures to identify the locations of body parts of a person lying on a pressure-sensing array [99]. The current version of our system uses a simplified body model in a fixed orientation, a fixed distance from the back of the bed, and a manually selected posture. It positions the model across the width of the bed using the center of mass of the pressure image.

3.2.6 Our Previous Work

We previously described elements of our system in two workshop papers [75, 73]. Our current system differs substantially and incorporates improvements to the web-based interface designed for operators with motor impairments, a planning domain definition language (PDDL) framework for assistive-task planning, modeling of the user's physical capabilities, a new body position estimation method, and user customization (including modeling the user's body size and his or her human-computer interface). We also conducted completely new evaluations based on [75], and for the first time have evaluated our system with a person with disabilities.

3.3 System Description

As illustrated in Fig 3.2, our system integrates a number of components. It uses the collaboration between two robots, a PR2 and Autobed, and a human user, leveraging the strengths of each. We focus particularly on the cognitive, physical, and perceptual collaboration between the three.

With respect to cognitive collaboration, the system leverages the user's cognitive capabilities to provide high-level direction and oversight during the autonomous setup for the task. The PR2 generates and executes a sound high-level task plan, and identifies a good configuration of the PR2 and bed (and therefore user) for completing the task. The user also provides control input for the fine control to perform the task, which may be difficult to complete autonomously. Although the entire task could be performed through

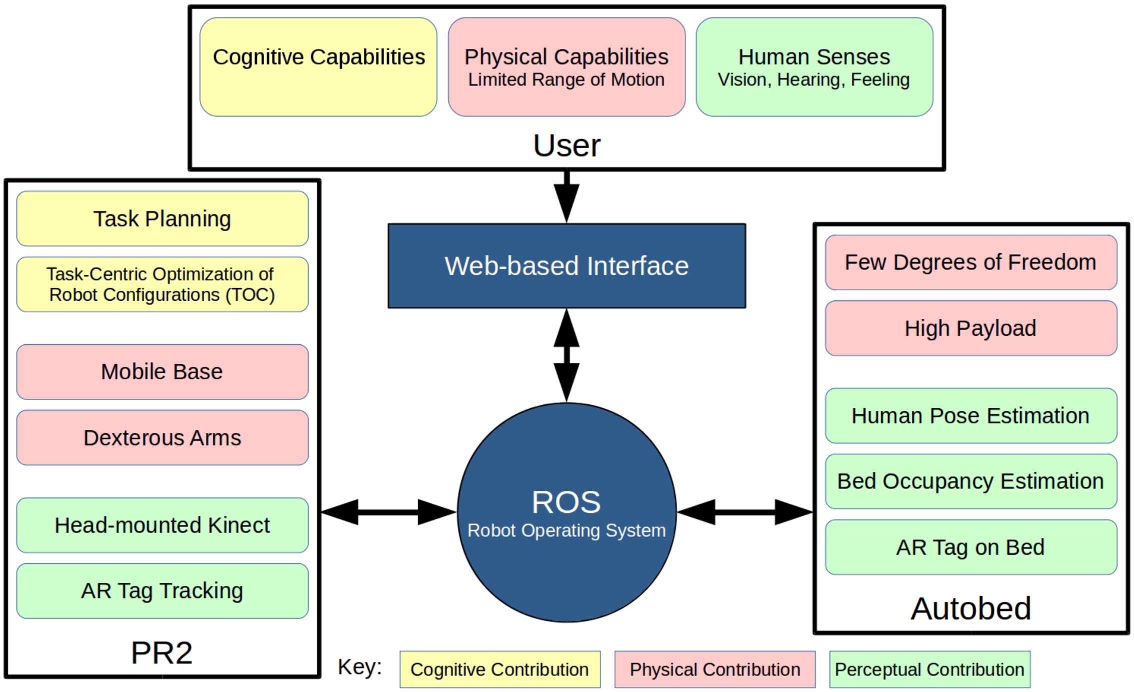


Figure 3.2: System architecture. Shows contributions of each component.

teleoperation, some aspects may be challenging. For example, Hawkins et al. observed that some assistive tasks require that a mobile manipulator use multiple base positions, and that manually choosing positions for some tasks can be difficult [16]. Grice et al. observed that a significant amount of the time spent on each task was spent moving the mobile robot into an appropriate position and moving the robot's gripper near the task area [73] . These autonomous functions allow the user to offload parts of the task, only requiring the user to oversee correct operation, thus reducing cognitive load. The goal of the autonomous functions are to configure the robots such that the task can performed with manually moving Autobed or the PR2 base.

With regard to physical collaboration, Autobed has few degrees of freedom and an immobile base, but it has a high payload capacity, allowing it to position itself and the body of the user. The PR2 has a mobile base, high dexterity, and many degrees of freedom, but a low payload capacity. This allows the PR2 to bring lightweight, task-appropriate tools to the necessary locations around the user. Although the human user's motor impairments

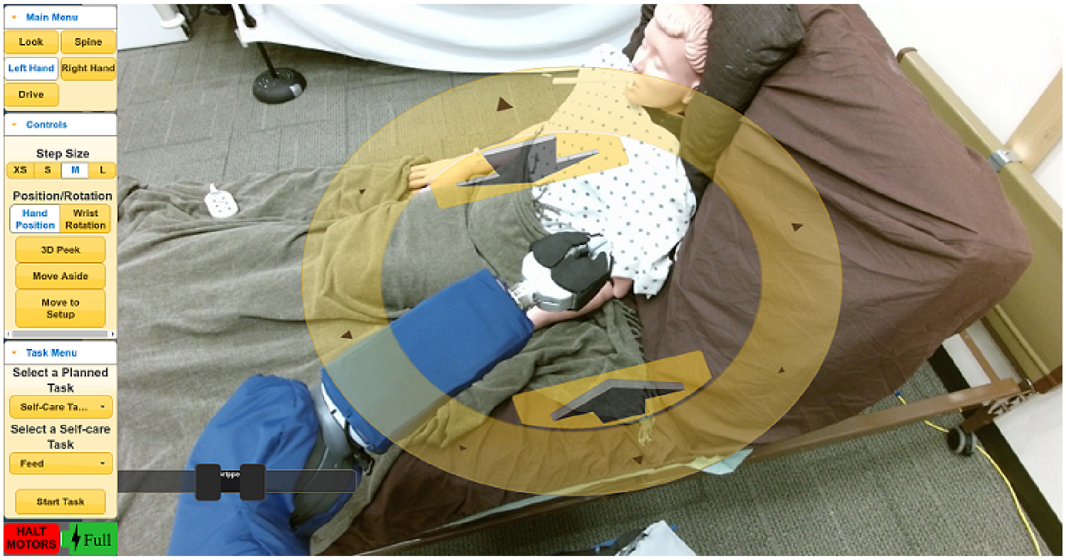


Figure 3.3: The web-based user interface. This interface enables low-level teleoperation by the user as well as the selection of autonomous planned tasks, with optional parameters, selected from a drop-down menu.

may limit certain motions, they may have high dexterity within those limits, allowing them to perform the subtle motions necessary for task performance once the bed and PR2 have appropriately positioned a tool.

With respect to perceptual collaboration, Autobed's joint encoders and pressure-sensing mat estimate the position and configuration of the bed and the human on the bed. The PR2's head-mounted Kinect v2 RGB-D camera visually locates Autobed via an AR Tag. The user supplements their own senses with views provided by the PR2's Kinect through the web-based interface shown in Fig 3.3 and described in [73], which allows for low-level teleoperation or directing autonomous execution of planned tasks.

3.3.1 Autobed

Based on a commercial actuated hospital bed (Invacare Full-Electric Homecare Bed, product ID: 5410IVC), we have developed the robotic bed, Autobed, extending from previous work [74]. The bed has 3 degrees of freedom: it can adjust its height, the angle of its head est, and the angle of the leg rest. In our autonomous functions we did not consider the leg rest for simplicity. By adding additional hardware between the remote control and the bed's motor drivers, we are able to send commands directly to the bed using a Robot Operating System (ROS) interface. We use a Raspberry Pi single-board computer to interface with the bed actuators and we run the other Autobed functions on an adjacent computer.

Autobed Control

Autobed uses accelerometers to measure the angle of the bed's head rest with respect to gravity. We mounted a Hokuyo laser scanner to the underside of Autobed, pointing towards the floor to measure the height of the bed. Autobed fits a line to the scanner's measurements using RANSAC to produce a height estimate height robust to some obstructions (e.g., a foot). Autobed's actuators are capable of changing the height of the bed, the angle of the head rest, and the angle of the leg rest when a human is lying on the mattress. Autobed runs a simple on-off controller with a deadband to reach commanded configurations.

Human Pose Estimation

We have equipped Autobed with a pressure-sensing mat, manufactured by Boditrak (`http://www.boditrak.com/` Model # BT3510), to measure the pressure distribution of the person lying on the bed. We placed the pressure-sensing mat on the top side of the mattress and below a fitted sheet. The pressure-sensing mat returns a pressure value for each of its 1728 tactile pixels (taxels) at 5Hz. Autobed sums the pressure values to estimate the total weight on the pressure-sensing mat. When this estimate exceeds a threshold (we used ∼20 kg), it reports that the bed is occupied.

Autobed uses a center of mass estimator from scikit-image (`http://scikit-image.org/`) to estimate the position of the body on the pressure-sensing mat. The center of mass is used to position a human model on the bed in the direction of the width of the bed (Y-axis) and assumes the head's position near the top of the bed. The system assumes the human is lying on the bed parallel to the long edge of the bed (X-axis), with arms at its sides. We

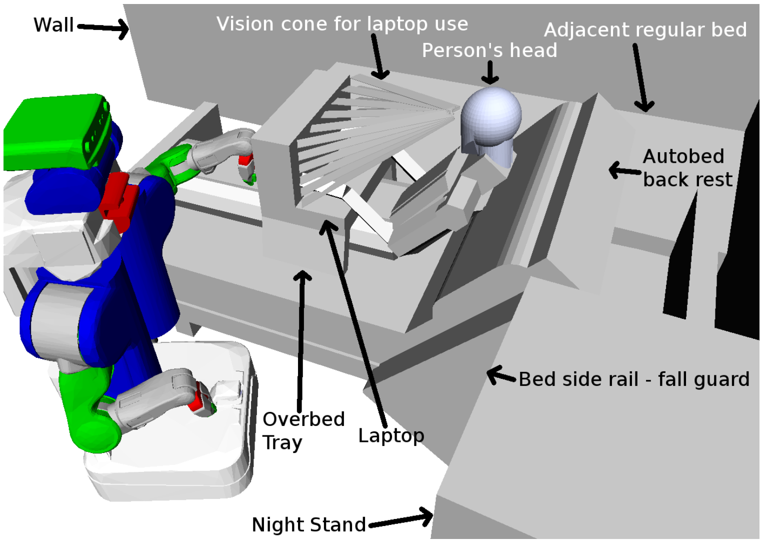


Figure 3.4: The customized model of Henry's bedroom used by the system. This is the simulation environment used by TOC, our system's configuration selection method. As an example of the customization, Henry's vision requirement for use of his laptop is modeled as rays from the laptop screen to the eyes of the human model.

constructed this rough model of a human from geometric primitives in the Unified Robot Description Format (URDF) file format, which allows the system to be easily customizable by scaling the model from a 50 percentile male to users using Xacro. Xacro stands for XML Macros, a language with which URDF files can be generated from macro'ed properties. Fig 3.4 shows the simulation environment with an example bed and human configuration. Coordinate frames associated with the model's body parts (e.g., knees, feet) are included in the model, although those parts are not visibly distinguishable.

This method differs from that used previously in [75], because we found that at higher angles of the bed's head rest, the head is not easily visible on the pressure-sensing mat. At high angles, the weight of the head is mostly downward, with relatively little weight on the back rest. On average, the human pose estimation algorithm takes < 1 milliseconds to run on an external machine (Intel Core i7-3770, 3.40GHz). Fig 3.5 shows the estimated head position for a person lying on Autobed.

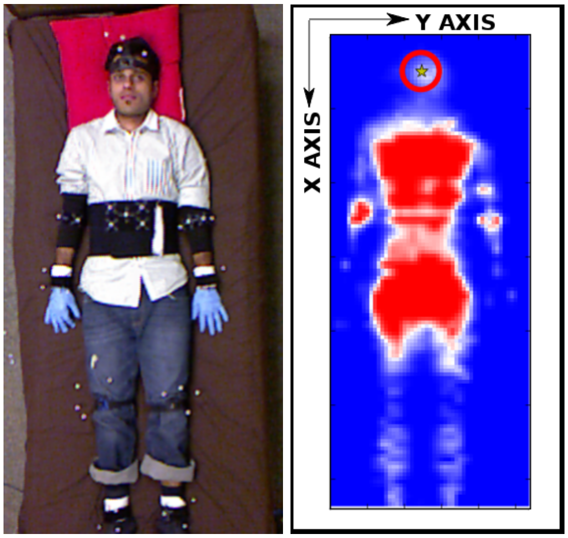


Figure 3.5: Autobed and the estimated pose of a person in bed. Left: View of a participant wearing various infrared reflective markers lying on Autobed. Right: Visualization of the pressure-sensing mat measurements from the participant with the estimated head position marked by a star inside a circle. The position in the Y direction is estimated from the pressure map. The position in the X direction is assumed to be ~ 25 cm from the top of the mat.

3.3.2 PR2

Our system uses a PR2 robot, a general-purpose mobile manipulator from Willow Garage. The PR2 has a mobile base, two 7-degree-of-freedom (DoF) arms with grippers, a pan-tilt head, and a head-mounted Kinect v2. The arms have high dexterity, but a low payload capacity of 1.8 kg.

PR2 Base Servoing

The system uses the AR Tag servoing from [16] to move the PR2 base directly from its current position to a goal position and orientation defined with respect to Autobed's AR Tag. This is accomplished via straight-line path and assumes that there are no obstacles to block PR2 movement. Our system uses the code package ar_track_alvar (`http://wiki.ros.org/ar_track_alvar`) to track an AR Tag mounted on the back-board of Autobed, and thereby to locate Autobed. Fig 3.1 shows the AR Tag as mounted on the bed. The PR2 moves its head to keep the AR Tag in the center of its head-mounted Kinect's view.

Haptic MPC Control of PR2 Arm

Our system uses a newer version of the model predictive controller (MPC) described in [78], with low stiffness, in both manual control mode and autonomous functions, to move the PR2's arm to end effector poses (position and orientation) or joint configurations (when resetting the arm configuration between trials). Note that our current system does not use the fabric-based skin or tactile sensing from that work.

3.3.3 Task-centric Optimization of Robot Configurations (TOC)

Hawkins et al. observed that some assistive tasks require that a mobile manipulator use multiple base positions, and that manually choosing those positions can be difficult [16]. In order for the system to autonomously move to an appropriate configuration for a task, we first select the configuration of the PR2 and Autobed using Task-centric Optimization of robot Configurations (TOC). TOC is based on that from [24] and [25], with additions to the algorithm customized to the user. For example, we incorporate a model of the person's physical capabilities.

TOC performs substantial offline computation to generate a function that can be applied rapidly online to select robot configurations based on the current user's body size and observations of the user's position on the bed. Offline, for each task, TOC jointly optimizes one or two 6-DoF system configurations, each of which consists of a 4-DoF configuration for the PR2 (X-Y base position, base orientation, and Z-axis height) and a 2-DoF configuration for Autobed (Z-axis height and head-rest angle). TOC models each task as a sparse set of 6-DoF end-effector goal poses. TOC runs the optimization for samples of the person's position on the bed and sizes of the person's body, given robot, person, and environment models, and then it interprets the optimization results to see if a single configuration of the two robots is sufficient for the task, or if there is value in using two configurations. Online, TOC applies the generated function for the desired task and current observations to inform the system what configurations to use for the task.

Fig 3.4 shows the simulation environment used by TOC, demonstrating a configuration for the *blanket adjustment* task.

User Customization

We customize the task models used by TOC based on user preference and needs. TOC models the user's physical capabilities (e.g., can rotate head up to 60 degrees), the user's human-computer interface (e.g., a laptop, a head-tracker, and the visibility lines needed to use them), and important features of the user's environment (e.g., a nightstand by the bed). TOC also uses a human model of similar size to the user and models the user's environment. These customizations allow TOC to better select robot configurations.

3.3.4 Task Planning

The system uses the task-level planning system from [73], based on the PDDL [100]. When possible, the planner produces a correct, minimal sequence of actions to complete the task. This sequence is then used to produce and execute a Hierarchal Finite State Machine (HFSM), implemented using the ROS SMACH framework (`http://wiki.ros.org/smach`).

In our two studies, all 6 tasks followed the same plan, but may have had differences in their internal states (e.g., reaching the PR2's to the head or knee) depending on the task. As the system moves through the plan, it may require assistance from the user to advance to the next state (e.g. using teleoperation commands to locate AR Tag above the bed, which is then tracked autonomously).

Fig 3.6 shows a typical task plan from start to completion of task. In the state in the plan, CHECK BED OCCUPANCY, the system checks if a person is in the bed. In FIND AR TAG, the interface switches into a Looking Mode so the user can move the robot's head until the Autobed AR Tag is in view. In FIND HUMAN POSE, the system checks

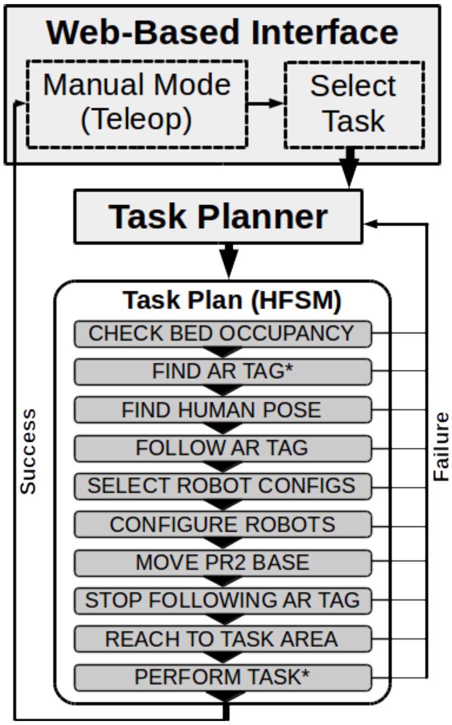


Figure 3.6: The system use process. From manual mode, the user can either teleoperate the robot or select a task. The task planner generates and then executes a task plan. Each state in the plan either succeeds and continues to the next state, or causes the planner to replan. The plan ends by returning the system to manual control mode. ∗Indicates a subtask requiring user input

Autobed's pose estimate for the user. In FOLLOW AR TAG, the PR2 begins following the AR Tag with its head. In SELECT ROBOT CONFIGS, the system uses TOC to select a configuration for the robots for the task based on the user's current pose on the bed. In CONFIGURE ROBOTS, Autobed moves to the desired configuration and the PR2 moves to the desired height and moves its arms to an initial configuration. In MOVE PR2 BASE, the PR2 uses AR Tag servoing to the desired configuration for its base relative to the Autobed. In STOP FOLLOWING AR TAG, the PR2's head stops tracking the AR Tag, returning control to the user. In REACH TO TASK AREA, the PR2 moves its gripper to the task area. In PERFORM TASK, the system switches to the manual Arm Control Mode of the user interface for the user to provide the fine control to complete the task. Each state either succeeds and proceeds to the next state, or fails and causes the system to re-plan.

3.4 Evaluation of Head Position Estimation

We conducted a study with 8 able-bodied participants to evaluate the performance of our method of human position estimation.

Participants' weights ranged from 52 to 95 kg and heights from 1.60 to 1.87 m. For the experiment, we placed Autobed in a motion capture room. We asked the participants to lie on Autobed in a supine configuration comfortable to them, keeping their heads looking straight, while wearing infrared reflective marker arrays on their bodies and heads (see Fig 3.5 Left). The bed was in a flat configuration for this evaluation. We designated the projection of the center of the forehead marker array onto the plane of the bed as the ground truth head position.

We selected 50 pressure distribution images from each participant while they were lying on the bed looking straight to form our test dataset of (400 total pressure distribution images). The images for each participant spanned on average 28 seconds. We estimated the head as aligned with the center of mass at an assumed position in the X direction (see Fig 3.5 Right). We compared the estimated head position with ground truth, but only consider error along the Y-axis, which the system uses to place the human body model. The error in the Y direction was 5.00 cm $\pm$ 2.54 cm (mean $\pm$ std). Previous work has demonstrated in simulation that this amount of error is manageable by using TOC to select robot configurations [25]. We further show evidence in our other evaluations that, despite error in pose estimation, the system was able to successfully reach task-relevant poses and perform desired tasks.

3.5 Evaluation of Robot Collaboration

We investigated the effectiveness of our system at configuring the PR2 and Autobed such that the PR2 could reach task-relevant poses. Additionally, we investigated the value of the physical and perceptual collaboration between the two robots. For each task and for a mannequin in different locations on the bed, we examined the percentage of goal poses the

PR2 could reach on the mannequin.

3.5.1 Implementation Details

In these experiments we used a weighted medical mannequin ($\sim$48 kg). We included into TOC's environment model a wall behind the bed, which matches our test environment. We roughly modeled the geometries of the bed and expanded the bed model by 4 centimeters to provide a safety margin. TOC searched for up to two configurations for each task on any side of the bed. TOC uses covariance matrix adaptation evolution strategy [101](CMA-ES from `https://pypi.python.org/pypi/cma`), to perform its optimization search, and we used a population size of 3000 with 10 iterations as the meta-parameters for the optimization for all tasks. When running the optimization to find configurations for no physical collaboration, we decreased the number of parameters to exclude the bed's degrees of freedom. For one of the tasks, the feeding task, we required that the back rest be ≥ 55 from horizontal to emulate typical eating posture in humans.

Throughout our experiments, the PR2 used its left arm to perform the task and kept its right arm to its side. For some difficult to reach task areas, the TOC chose to use two configurations for the task. In these cases, the experimenters ran the system for each configuration separately, starting the PR2 on the appropriate side of Autobed.

3.5.2 Experimental Protocol

An able-bodied experimenter sitting at a nearby desk used the web-based interface to simulate performing three tasks on the mannequin lying on Autobed. The tasks were feeding, bathing the lower legs, and dressing with a hospital gown. These tasks are representative of classical ADLs (feeding, hygiene, and dressing).

For each trial, we started the PR2 on the same side of the bed as the goal position, $\sim$1 m away and facing Autobed, and looking at the AR Tag mounted on Autobed. Using the web-based interface, the experimenter commanded the system to begin the task. The system

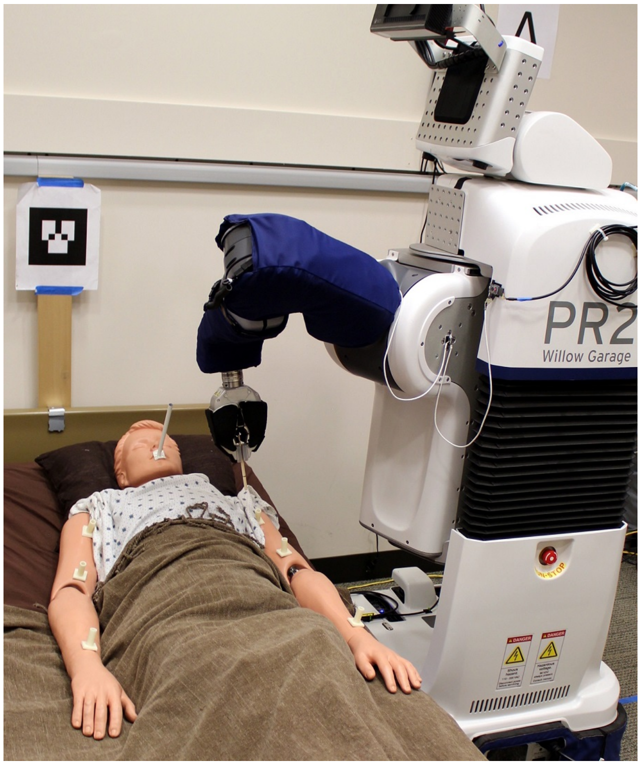


Figure 3.7: The system after execution of the autonomous functions for the arm hygiene task. The PR2 and Autobed have configured themselves for the task and the PR2 has reached its arm to the task area on the mannequin.

then autonomously generates and executes a plan, configuring the PR2 and Autobed for the task, and moving the left end effector to the task area. Fig 3.7 shows a configuration reached using the system for the *mouth wiping* task with the mannequin on the left side of the bed.

At this point in the experiment, the experimenter deactivated the PR2's motors and manually moved its left end effector to several task-specific goal poses around the task area (8 for bathing legs, 6 for arm hygiene, 1 for feeding). We affixed a hollow tube to the mannequin at each goal location. The PR2 held a small cylindrical tool in its left end effector. If the experimenter could reach the base of the tube with the PR2's held cylinder without touching the walls of the tube, we considered the PR2 as being able to reach that goal (see Fig 3.8). The tubes for bathing legs and arm hygiene were 3 cm long. The tube for feeding was 15 cm long. Note that the PR2 has infinite-roll wrists making the goal poses 5-DoF.

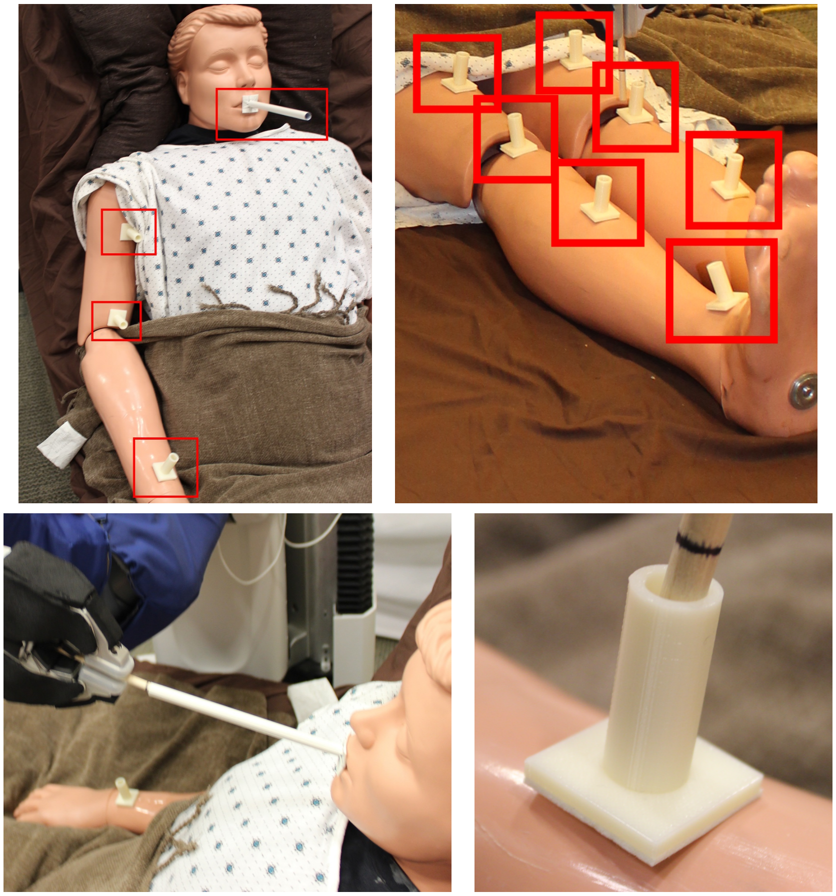


Figure 3.8: The evaluation of the system with a mannequin. Top: Mannequin with the physical 5-DoF goals highlighted for the feeding task and the right arm of the arm hygiene task (left) and for the leg bathing task (right). The left and right side had symmetric goals. Bottom: (Left) Example of the PR2 successfully reaching the goal pose for the feeding task, inserting a 20 cm long cylindrical tool into the 15 cm tube on the mannequin's mouth. (Right) Example of failing to reach a goal pose. The tool could not reach the bottom of the tube without touching the walls of the tube.

For the bathing legs task, the 8 goals poses were along the tops of the legs, ranging from 10 cm above the knee to the ankles. For the arm hygiene task, the 6 goal poses were along the front or top of the arms, ranging from the wrist to the shoulder. For the feeding task, the single goal pose was a attached to the center of the mouth. The goal locations are shown in Fig 3.8. If the PR2 could reach all goals for a task from the one or two configurations selected by TOC, without the user having to move Autobed or the PR2 base the system was considered to have succeeded. Otherwise, the system was considered to have failed. Because the goals are sparse and represent a more complicated task, we deemed reaching

Table 3.1: Overall performance, percentage of successful tasks by the system with and without bed movement or human position estimation

Condition	*Overall Performance*
Full System	**100%** (9/9)
No Bed Movement	**33%** (3/9)
No Position Estimation	**78%** (7/9)

all goals necessary for success. For feeding task, the pose of the person's body matters; it is typically comfortable to eat while seated upright. We required that the back rest be ≥ 55 and that the goal be reached for the feeding task to be judged successful.

To evaluate the physical collaboration between the two robots, we ran the experiment with Autobed fixed in its lowest, flattest configuration. TOC was given this un-actuated model of the bed, so the system could plan for this change. To evaluate the perceptual collaboration, we ran the experiment and informed the system that the mannequin was in the center of the bed.

3.5.3 Results

Table 3.1 shows overall performance and Table 3.2 shows the task outcome and number of reached goal poses for each task as we translated the mannequin from the center of the bed in the Y direction by -15, 0, and 15 cm. Using all parts of the system, the PR2 was successful in 100% of the tasks, reaching all goals.

Physical Collaboration Results

The two robots collaborated physically to allow the PR2 to better perform the task, by adjusting the Autobed configuration to give the PR2 better physical access around the mannequin and to adjust the mannequin's configuration. For example, Figure 3.9 shows a solution frequently used by the system, which was to raise the bed and to move the PR2's base under it. Without physical collaboration between the two robots, the system succeeded in 33% (3/9) of the tasks. In the feeding task, physically moving was a necessity to achieve

Table 3.2: System performance with and without physical or perceptual collaboration. Task is successful if all goals reached.

		Full System		No Bed Movement		No Position Estimation	
Task	*Body Shift*	*Outcome*	*Poses Reached*	*Outcome*	*Poses Reached*	*Outcome*	*Poses Reached*
Feeding	-15 cm	success	100% (1/1)	failure	N/A	success	100% (1/1)
	0 cm	success	100% (1/1)	failure	N/A	success	100% (1/1)
	15 cm	success	100% (1/1)	failure	N/A	success	100% (1/1)
Arm Skin Care	-15 cm	success	100% (6/6)	failure	83% (5/6)	success	100% (6/6)
	0 cm	success	100% (6/6)	failure	50% (3/6)	success	100% (6/6)
	15 cm	success	100% (6/6)	failure	83% (5/6)	failure	83% (5/6)
Bathing Legs	-15 cm	success	100% (8/8)	success	100% (8/8)	failure	88% (7/8)
	0 cm	success	100% (8/8)	success	100% (8/8)	success	100% (8/8)
	15 cm	success	100% (8/8)	success	100% (8/8)	success	100% (8/8)

the task goal of having the person sitting upright ($\geq$ 55 degrees from horizontal). When the system could not move the bed, it could not succeed at this task.

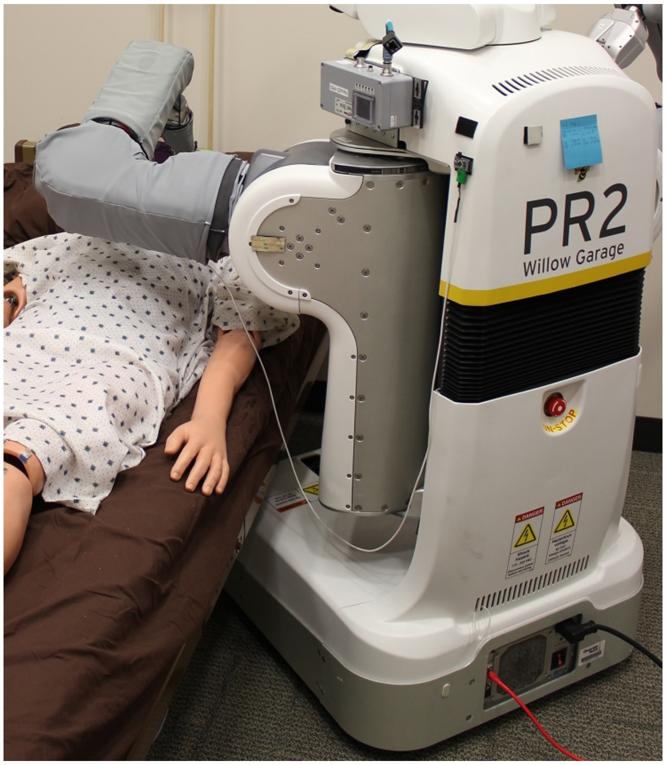


Figure 3.9: Physical collaboration between the two robots. A solution frequently used by the system was to raise the bed and to move the PR2's base under it, so the PR2 could better reach task areas.

Perceptual Collaboration Results

The robots collaborated perceptually to estimate the mannequin's position and configuration, allowing better initial configuration selection for the task, and also to inform the PR2 when autonomously moving its end effector to the task area. Without perceptual collaboration between the robots, the PR2 succeeded in 78% of the tasks, since it was sometimes too far from the mannequin to reach the goal poses.

3.6 Evaluation in the Home of a Person with Severe Quadriplegia

We tested if a real user with severe motor impairments could use our system to perform real tasks in a non-controlled setting. We had a person with motor impairments, Henry Evans, who is also a frequent collaborator with us, use the system in his home to repeatedly perform three assistive tasks around his body. Henry has severe quadriplegia and is mute as the result of a brain-stem stroke, and has only limited movement in his head and left arm and hand, although he has full sensation. Henry regularly uses a computer via a head-tracking mouse from Madentec.

3.6.1 Implementation Details

We brought a PR2 to Henry Evans' home and converted his standard electronic bed from Invacare into Autobed. During late night trials we introduced an additional light in the room to aid in visual perception of the AR Tag, but otherwise we did not alter his bedroom. Figures 3.1 shows the room.

Henry's particular form of human-computer interface (HCI) requires that he keep sight of his laptop screen and the head-tracking device attached to it. We gave TOC a coarse model of Henry's bedroom, his laptop, and the necessary cone of vision in the form of rays from the human model's eyes to all parts of the screen (see Fig 3.4). TOC used a human model that included Henry's ability to move his head and that was adjusted to roughly his height. As in the mannequin experiments, we expanded the bed model by 4 centimeters to provide a safety margin and the PR2 used its left arm to perform the tasks, keeping its right arm at its side.

Because only one side of Henry's bed is accessible, we allowed TOC to optimize only a single configuration for each task, limited its search space to the accessible side of the bed, and required the left arm's workspace to at least partially overlap the bed. We also limited the optimization of the head rest angle to stay within the range in which Henry could comfortably operate the laptop, 40 to 55 degrees. Because of the tight space around the head, the system did not autonomously reach to the task area for tasks around Henry's head. The PR2 left its arms in their initialization pose when it initiating the manual control part of the task plan.

3.6.2 Experimental Protocol

For each trial, we started the PR2 $\sim$1 meter away from the bed, roughly midway down the bed, with its base facing the bed, its arms to its sides, and its head pointing at the AR Tag mounted on Autobed. Autobed started in its lowest setting with its back rest at 45 degrees and leg rest down. Henry started each trial lying on the bed; we did not alter his position from where his caretaker lay him. Henry controlled the system using a laptop on an overbed tray.

For each task, Henry was required to use the system's autonomous function associated with that task and then only use the interface's left arm control to perform the task. The system used TOC to select one configuration for the PR2 and Autobed for the task, based on the estimated position of Henry's body. Henry made use of his own physical capabilities to assist in task performance. For example, for the tasks around his head, Henry would move the towel held in the PR2's gripper to an appropriate location and then wipe his face on the towel.

The tasks performed were:

(a) Pulling a blanket down from knees to feet

(b) Wiping yogurt from mouth

(c) Wiping lotion from forehead

Previous work has noted that these three tasks may be useful for those with severe motor impairments [15] and Henry has indicated that these tasks would be useful.

For each task, Henry was given instructions on how to use the interface and was given practice with the system until he felt comfortable. He then performed the task 5 times in a row. An experimenter set up each task's trials by a) covering the feet and knees with a blanket; b) applying yogurt to both sides of Henry's mouth; or c) applying dabs of lotion to the center of Henry's forehead. For the wiping tasks, a fresh, rolled towel was placed in the PR2's left gripper before each performance of the mouth wiping task. The goal of each task was explained to Henry, and he decided if a task was completed successfully. He used his laptop's camera to inspect his face to check for yogurt and lotion and used the PR2's camera to inspect the state of the blanket. Trials were considered failed if any interruptions in the system's performance, such as an experimenter intervention, occurred, or if Henry deemed that manual movement of the PR2 base or of Autobed was necessary to complete the task. For each trial, the success (or failure) and time to completion was recorded and free-form feedback was solicited.

Fig 3.10 shows sequences of images of the performance of the three tasks by the system. Table 3.3 shows the percentage of successful trials. It took 3 minutes and 33 seconds on average to complete each trial across all tasks. During the blanket adjustment task, Henry said "It works well and is very easy" and "Almost no mental effort. That's how I like it. This would be great for a first time user." After the fourth trial of the forehead wiping task, Henry said "It required a lot of mental effort at first, but now it's easy.' After the last mouth wiping task, Henry said "Most complicated task yet." Wiping the yogurt from the mouth seemed particularly challenging because wipes tended to smear yogurt around the face. During the

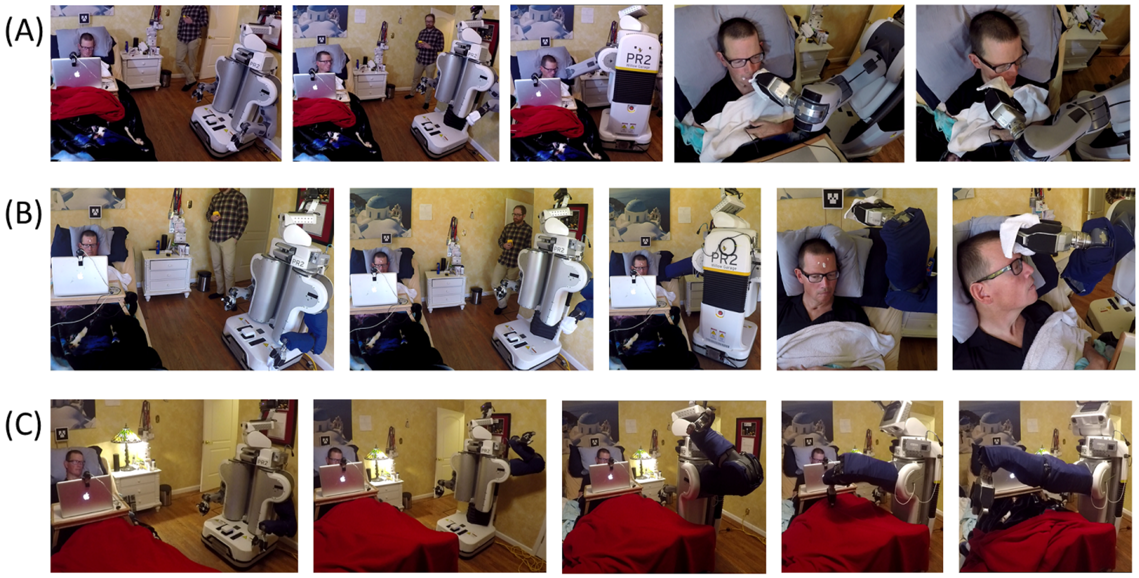


Figure 3.10: Image sequences from the three tasks performed by Henry. Using this system, Henry Evans was able to (A) wipe yogurt from his mouth, (B) wipe lotion from his forehead, and (C) pull a blanket down from his knees to his feet.

Table 3.3: Performance of our system used by a person with severe quadriplegia in his own home to perform three real tasks. Movement of the PR2 base and of Autobed was performed only by the autonomous portion of the system.

Task	% Trials Successful
Wipe mouth	100% (5/5)
Wipe forehead	100% (5/5)
Adjust blanket	100% (5/5)

autonomous portion of each task, Henry exerted little effort, simply watching the task plan execute. He seemed to particularly like the autonomous reaching to the task area in the blanket task.

3.7 Discussion and Limitations

We sought to handle fairly the comparison between the full system and the system without allowed bed movement by using the same parameters (e.g., search time) for the optimization. However, because of the nature of the problem, a non-convex optimization in a large search space, searching longer may have found better configurations than we used in our experiments.

We found that our system was capable of reaching many or all goal poses even with 15 cm of error in human position estimation. This was particularly noticeable in the feeding task, where the system without position estimation succeeded in all tasks. Because Autobed is narrow, we could not feasibly position the mannequin more than 15 cm from the center of Autobed. Estimation of the person's pose might be more important on wider beds where position error could be larger.

The repeated successful use of the system by Henry Evans provides evidence that a person with severe motor impairments can use our system to perform assistive tasks around their body. The system was customized for Henry, and we expect that similar customization could be implemented for other users. However the question of whether the system can be equally successfully applied to other users in other environments remains open. Henry's feedback was particularly promising, suggesting that beyond the system's ability to successfully perform tasks, he found its assistance valuable. Henry seemed to find that the autonomous functions made the tasks easier and require less mental effort. These positive responses address two main block in the technology acceptance model (TAM) [102], perceived usefulness and perceived ease of use, and suggest the possibility of adoption of this sort of assistive technology by people with severe motor impairments.

3.8 Conclusion

We have presented a robotic system designed to provide physical assistance to people who are in bed, and potentially increase user independence and reduce caregiver burden. The system consists of a robotic bed and a mobile manipulator that collaborate to assist with tasks around the user's body. For a task, the robotic bed configures itself and the mobile manipulator moves to a position, so that the mobile manipulator can reach task-relevant locations with high manipulability. Then, the mobile manipulator can either reach out autonomously to these locations or the user can take control of the robot via a web-based interface. We showed evidence that the physical and perceptual collaboration between the robots each improves the system's performance. We further evaluated our system with a person with severe quadriplegia, in his home. We personalized the system with coarse models of his bedroom, body, and user interface. Henry successfully used our system to perform 3 different tasks, 5 times each, without failure and provided positive feedback about the system's usefulness and ease of use.

CHAPTER 4

TASK OPTIMIZATION OF ROBOT-ASSISTED DRESSING

4.1 Introduction

Robotic assistance with activities of daily living (ADLs) [3] could increase independence for people with disabilities. This may improve quality of life and help address societal challenges, such as aging populations, high healthcare costs, and shortages of healthcare workers found in the United States and other countries [1, 2]. A number of specially designed assistive devices exist to help people maintain their independence. However, many current assistive devices, such as those for dressing (e.g., reachers, dressing sticks, long-handled shoehorns, and sock aids), provide limited support and rely on the user having substantial cognitive, perceptual, and motor capabilities [7, 8]. Robots could potentially serve as more versatile assistive devices.

Specialized robots are commercially available for a variety of ADLs [9, 10], such as desktop feeding devices for feeding tasks, but robotic assistance for dressing remains in early stages of research. Studies suggest a need for robot-assisted dressing, with more older adults receiving assistance with dressing and bathing/showering than other ADLs, and over 80% of people in skilled nursing facilities requiring dressing assistance [6]. Dressing tasks are complicated, involving complex physical interactions between garments and the person's body. Robot-assisted dressing tasks differ from robotic assistance with many other ADLs because dressing involves more complex physics, more complex cooperation between the person and robot, and can involve a wide variety of clothing. Determining how the robot and person can collaborate to complete the task is challenging, especially for people with disabilities. Disabilities can make dressing more difficult both for the participant to dress themselves and for the robot to provide assistance. For example, disabilities may limit the actions that the person can take in collaboration with the robot. Additionally, other assistive tools the person may use, such as a wheelchair, may impede the robot's workspace. Customization for individuals can be important as disabilities can vary greatly.

We present task optimization of robot-assisted dressing (TOORAD), a method for finding actions for a person and robot that will likely result in successful dressing. TOORAD makes use of geometric, kinematic, and physics simulations of the person, robot, and garment in its optimization. It uses customized models for the person to model their geometries and physical capabilities. These models consider what the person is capable of doing, instead of what he or she typically does. With this approach, TOORAD is able to explore a wide range of actions for dressing in simulation, some of which might be challenging to test in the real world. Using a general-purpose mobile manipulator can mitigate some of the challenges in dressing by allowing the robot to move around to access different areas around the body, as explored by [23]. We consider the robot an important part of TOORAD's optimization, so TOORAD optimizes the robot's base pose to improve the robot's ability to adapt to unexpected changes. Additionally, our method provides computer-generated instructions for the user receiving assistance from the robot. We have used TOORAD to optimize the actions of a person and a PR2 robot (a mobile manipulator made by Willow Garage) to collaborate in pulling the two sleeves of a hospital gown onto the person's body. These features are notable differences from previous work on robot-assisted dressing, as described in Section 4.2.1.

We conducted a study with six people with disabilities who require assistance with dressing to learn more about the habits, needs, and capabilities of some members of this population, as well as their views on robot-assisted dressing. Their responses are summarized in Section 4.4.4 and discussed in Section 4.5. The results from this study may help guide future research in robot-assisted dressing. Additionally, we evaluated TOORAD on a dressing system with a PR2 robot, which we tested with four human participants with disabilities whose capabilities were matched to the assistance our system was able to

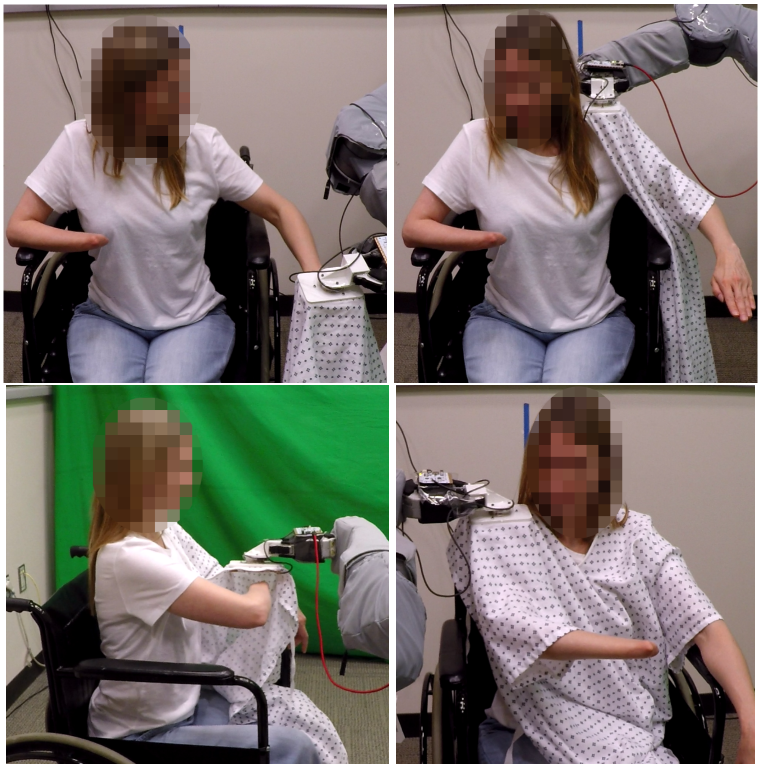


Figure 4.1: Through our system implementation of TOORAD, four participants with disabilities were able to dress themselves with assistance from the robot. (left-to-right) Prior to dressing the left arm; end of dressing the left arm; prior to dressing the right arm; end of dressing the right arm. After successful dressing, both of the participant's arms are in the hospital gown.

provide. The system was able to successfully assist all four participants in pulling on both sleeves of a hospital gown. Two of the four participants dressed one sleeve independently and received assistance from the robot with only the second sleeve, and two participants received assistance with both sleeves. Figure 4.1 shows one of the participants receiving dressing assistance from the robot. Our results provide evidence that TOORAD can be used to select actions that will result in a robot and person with disabilities collaborating successfully to complete a dressing task.

4.2 Related Work

4.2.1 Robot-assisted Dressing

Many researchers who have previously investigated robot-assisted dressing have focused on using kinematics and vision. [12] presented work on user modeling during robot-assisted dressing. They identify the pose of the user, model the movement space of the upper body joints, and select where to place the openings of a sleeveless jacket. More recently, [103] introduced a stochastic path optimization method to optimize a dressing trajectory for the robot using the estimated human pose and haptic data from the sensor attached at the robot's end effector. Their method updates the dressing trajectory over several attempts of the task. [104] proposed a method using reinforcement learning for a robot to learn trajectories to dress a mannequin in a t-shirt, focusing on topological relationships between the mannequin and shirt. Their experiments were initiated with the mannequin's arms in the shirt sleeves. [105] have also presented methods for learning dressing motions using Bayesian nonparametric latent space learning. [106] present a method for determining the trajectory to pull a knit hat on a mannequin head using a head-centric policy-space . [107] presented a system for putting on a slipper-style shoe onto a person's foot using voice commands and gestures from the user to pick up, maneuver, and put on the shoe. [108] introduced an approach for a robot manipulator to coordinate with a human during assistive dressing and to learn the person's physical limitations using a vision module. They demonstrated the approach by having a Baxter robot place a hat on two human participants. They represented dressing tasks as a sequence of goal poses with respect to the user. The robot requested that the user reposition themselves if the robot determined the goal as infeasible. The robot modeled the user's constraints to determine where to reposition the user. [109] use learning from demonstration by a person to teach a robot how to pull the sleeve of a jacket onto a person's arm. [110] presented a method to determine dressing errors and clothing types when dressing a human participant in various poses. [111] present methods that uses a dataset of the tracked poses of a person's body to develop a model of a person's movement capabilities in addition to force-feedback during dressing to customize dressing trajectories for a person's movement capabilities and in realtime for movement during task execution.

We note three main differences between our work and other work from the literature. First, many other methods have focused on solely geometric and kinematic simulation, and have optimized trajectories using real-world trials. We instead make use of geometric, kinematic, and physics simulation of the person, robot, and garment to optimize the robot and person actions in simulated environments. This allows TOORAD to optimize through more iterations than are feasible with real-world training. Additionally, TOORAD is able to explore actions in simulation that might be challenging to test in the real world. Second, we consider the robot an important part of our optimization. While most other works focus on a fixed robot pose, we use a mobile manipulator and optimize the robot's configuration to improve the robot's ability to adapt to unexpected variations between simulation and the real world. Third, we evaluate our method in a study with six participants with disabilities, while previous works have focused on able-bodied participants or mannequins. We customize the models in simulation so they match the person's geometries and physical capabilities. Through our study we provide evidence that our method can select actions for the robot and person that result in successful dressing of people with disabilities. Additionally we present insights from our target population to guide future research.

4.2.2 Physics Simulation in Robotics

Physics simulation has been used to test algorithms before use with real robots [112, 113, 114]. In addition, a few works have used physics simulation as part of learning or control algorithms. [115] presented an interactive game of cooking pancakes using rigid-body and fluid simulation. They collected data from a human performing the cooking task and trained classifiers on the data to predict failure outcomes. In a more recent work by [116], they demonstrated predicting failure cases for the same task on a real robot, in which they learned a classifier by constructing envelopes for the data collected from the interactive game. We use the PhysX simulator for human-cloth physics simulation in a way similar to our previous work in which we trained classifiers on force measurements in simulation to predict the task outcomes for robot-assisted dressing on a real robot [117].

4.2.3 Selecting Robot Configurations for Mobile Manipulation

Much prior research has investigated how to select configurations for a mobile robot. By the term robot configuration, we mean the more general term for robot base pose, which may include other degrees of freedom such as the robot's spine height. A common method is to address the problem using inverse-kinematics (IK) solvers [38, 39, 40, 41]. IK solvers typically seek a single joint configuration of the robot, although often many solutions exist or no solutions exist. The entire kinematic chain from end effector to the robot's base location can be solved using IK [42, 43]. Alternatively, sampling-based methods can be used to find robot base poses that have valid IK solutions, often as part of motion planning [44, 45, 46, 47, 48].

By relying solely on the existence of IK solutions to ensure that the robot can reach the goals, these methods are dependent on accurate models. Many of these methods are fast, but task execution may fail if there is modeling or state estimation error. Like these methods, TOORAD uses a sampling-based search in its optimization of the robot's configuration. However, there are often many robot configurations with valid IK solutions to all goals, and they cannot be distinguished using only IK. TOORAD uses measures of the robot's dexterity based on task-centric manipulability from [23] to differentiate between those configurations. In that work, we show that higher task-centric manipulability is correlated with improved performance for configurations that have collision-free IK solutions to all goals in a task. In this chapter, we are looking at dressing tasks, which contain more structure than the generic assistive tasks addressed by [23]. The structure in dressing, for example, requires that the robot reach goal poses in a specific order. Section **??** describes the details of the robot configuration optimization used in this chapter.

4.3 Task Optimization of Robot-Assisted Dressing (TOORAD)

With our method, task optimization of robot-assisted dressing (TOORAD), we seek to find a sequence of actions for both the assistive mobile manipulator and the person receiving assistance that are likely to result in successful dressing. Key features of TOORAD are its simulation of cloth-human physics, its simulation of human and robot kinematics, its representations of robot dexterity, its selection of multiple actions for a task, and its selection of actions for both the robot and person using an optimization-based approach. TOORAD is an offline process and it is suitable for situations when the garment and person can be modeled beforehand, when we would like to customize the actions for the person's capabilities, and when we would like to configure the robot such that the task is successful despite variations between models and reality. We first define the problem addressed by TOORAD and important assumptions it makes, followed by the details of the optimization. We provide examples from our implementation of TOORAD used in our evaluation, which is for the task of pulling the two sleeves of a hospital gown onto a person's arms.

4.3.1 Problem Definition

TOORAD aims to find, within the space of the sequence of all actions that the person being assisted and the assistive robot can perform, $\mathcal{U}$, "What is a sequence of actions that will result in successful dressing?". Notably, TOORAD considers what humans *are capable of doing*, rather than what humans *typically do*. This choice follows from the notion that humans are currently more adaptable than robots.

With this approach, we avoid the challenges in modeling actions a person might take in given circumstances. Instead, we model the person's physical and kinematic capabilities. By taking this approach, we can take advantage of the robot's strengths and capabilities that may differ from a human assistant. For example, we can equip the robot with sensors, such as a capacitive distance sensor, that can estimate the distance to the person's arm in a way a human caregiver may find difficult. Additionally, instead of basing the robot actions on those from human caregivers, TOORAD explores strategies for dressing in a simulated environment that might be specific to robotic assistance. Some solutions can be difficult for caregivers or robotics experts to identify without computer assistance to due the high number of degrees of freedom, solutions that may be distinct from human practice, and the complexity of human disabilities that can vary greatly between individuals. The non-anthropomorphic kinematics of the robot presents challenges for approaches based on learning from demonstration or other forms of user control.

We can formulate the problem as optimizing the sequence of actions for the human and robot, $\boldsymbol{A}_{\mathrm{h,r}}$, such that

$$\underset{\boldsymbol{A}_{\mathrm{h,r}} \in \mathcal{U}}{\arg\max} \quad R_{\mathrm{d}}(\boldsymbol{A}_{\mathrm{h,r}}) \tag{4.1}$$

where $\mathcal{U}$ is the domain of feasible robot and human actions and R_d is an objective function for dressing that we define in Eq 4.3.

A challenge to solving this problem is that the space of all possible human and robot actions is large. To achieve tractability, we apply constraints on the action space that structures how interaction between the person and robot will take place. We constrain this space by limiting the robot's end effector to linear trajectories and alternating between the actions of the person and robot. Additionally, the person and robot base hold still as the robot moves the garment onto the person's body. This constrained search space, $\mathcal{U}^{\mathrm{c}}$, simplifies the actions of the robot and the person. It also allows TOORAD to consider only the person's static physical capabilities, instead of his/her dynamic capabilities. These simplified actions are also easier to convey to people who are receiving assistance from the robot.

We limit the sequence of actions to this constrained space, such that $\boldsymbol{A}_{\mathrm{h,r}} \in \mathcal{U}^{\mathrm{c}}$, and modify the optimization as shown in Eq 4.2. $\boldsymbol{A}_{\mathrm{h,r}}$ can then be defined as a sequence of

$$\underset{\boldsymbol{A}_{\text{h,r}} \in \mathcal{U}^{\text{c}}}{\arg\max}\, R_{\text{d}}(\boldsymbol{A}_{\text{h,r}}) \tag{4.2}$$

$$\begin{aligned} R_{\text{d}}(\boldsymbol{A}_{\text{h, r}}) = &- \psi C_{\text{n}}(N) \\ &+ \frac{1}{N} \sum_{i=1}^{N} \Big(- \zeta C_t(\boldsymbol{c}_{\text{h},i}) - \eta C_s(\boldsymbol{c}_{\text{h},i}, \boldsymbol{t}_{\text{r},i}) \\ &+ \gamma \big(\alpha R_{reach}(\boldsymbol{c}_{\text{h},i}, \boldsymbol{c}_{\text{r},i}, \boldsymbol{t}_{\text{r},i}) + \beta R_{manip}(\boldsymbol{c}_{\text{h},i}, \boldsymbol{c}_{\text{r},i}, \boldsymbol{t}_{\text{r},i}) \big) \Big) \end{aligned} \tag{4.3}$$

Subject to:

- Collision constraints
- Garment stretching constraints
- Range-of-motion constraints of the person

Where:

- $\boldsymbol{A}_{\text{h,r}} = \big\{ \{\boldsymbol{c}_{h,1}, \boldsymbol{c}_{r,1}, \boldsymbol{t}_{r,1}\}, ..., \{\boldsymbol{c}_{h,N}, \boldsymbol{c}_{r,N}, \boldsymbol{t}_{r,N}\} \big\}$
- N is the number of subtasks, $N = \text{length}(\boldsymbol{A}_{\text{h,r}})$.
- C_n is a cost on the number of subtasks.
- C_t is a cost on torque experienced at the person's shoulder.
- C_s is a cost on stretching the garment.
- R_{reach} is a reward for how much of the trajectory the robot can reach.
- R_{manip} is a reward for the dexterity of the robot arm along the trajectory.
- $\psi, \zeta, \eta, \gamma, \alpha, \beta$ are weights for the terms in the objective function.

paired actions by the human and robot,

$$\boldsymbol{A}_{\text{h,r}} = \big\{ \{\boldsymbol{a}_{\text{h},1}, \boldsymbol{a}_{\text{r},1}\}, ..., \{\boldsymbol{a}_{\text{h},N}, \boldsymbol{a}_{\text{r},N}\} \big\}, \tag{4.4}$$

where $\boldsymbol{a}_{\text{h},i}$ and $\boldsymbol{a}_{\text{r},i}$ are actions for the robot and human, respectively, for subtask i. Each pair of actions is performed in sequence. We call each pair of actions a subtask. N is the length of the sequence of actions, the number of subtasks for the dressing task. The actions of the person and robot are taken in that order, first the person, then the robot.

TOORAD does not automatically segment the task into subtasks. Instead, we provide a small set of candidate subtask sequences to the optimization. For example, a candidate sequence we used in our evaluation in Section 4.4.4 was to first dress the entire left arm and then dress the entire right arm. Other candidate sequences used included splitting each arm into two subtasks, first dressing the forearm and then dressing the upper arm. These subtasks define a trajectory policy for the robot end effector. The policy gives a trajectory along which the robot should move the garment to complete the subtask. Therefore, we provide TOORAD with a small set of candidate sequences of trajectory policies.

We narrow our definitions of the human and robot actions, based on our limited action space. The assumptions we use to define these actions are described in Section 4.3.2. We model the human's actions as only the pose he or she holds and we ignore the movement of that pose. Therefore, we define the human's actions $\boldsymbol{a}_{\mathrm{h},i}$ as

$$\boldsymbol{a}_{\mathrm{h},i} = \boldsymbol{c}_{\mathrm{h},i}, \tag{4.5}$$

where $\boldsymbol{c}_{\mathrm{h},i}$ is a configuration of the human body that the person holds while the robot performs its action for the subtask, i. For the task used in our evaluation of pulling sleeves onto a person's arms, we used $\boldsymbol{c}_{\mathrm{h},i} \in \mathbb{R}^4$ (3 DoF at the shoulder, 1 DoF at the elbow). We model the robot's actions for subtask i as the pose of the robot's base, $\boldsymbol{c}_{\mathrm{r},i}$, and the trajectory of the robot's end effector, $\boldsymbol{t}_{\mathrm{r},i}$. We ignore the movement of the robot to achieve the base pose. Therefore, we define the robot's actions, $\boldsymbol{a}_{\mathrm{r},i}$, as

$$\boldsymbol{a}_{\mathrm{r},i} = \{\boldsymbol{c}_{\mathrm{r},i}, \boldsymbol{t}_{\mathrm{r},i}\}. \tag{4.6}$$

We hence refer to $\boldsymbol{c}_{\mathrm{r},i}$ as the robot configuration, which is a more general term for robot base pose as it may include other degrees of freedom such as the robot's spine height. In our evaluation we used $\boldsymbol{c}_{\mathrm{h},i} \in \mathbb{R}^4$ (3 DoF for the robot base pose, 1 DoF for the robot spine height). The robot action is taken in sequence: first attaining the robot configuration and then executing the end effector trajectory.

We can substitute these definitions of the human and robot actions into Eq 4.4, to obtain

$$\boldsymbol{A}_{\text{h,r}} = \big\{\{\boldsymbol{c}_{h,1}, \boldsymbol{c}_{r,1}, \boldsymbol{t}_{r,1}\}, ..., \{\boldsymbol{c}_{h,N}, \boldsymbol{c}_{r,N}, \boldsymbol{t}_{r,N}\}\big\}. \tag{4.7}$$

As before, the actions within each subtask occur in order: first the person then robot attains their configurations, then the robot executes the end effector trajectory.

The goal of TOORAD is to find $\boldsymbol{A}_{\text{h,r}}$, as defined in Eq 4.7, that maximizes the value of the dressing objective function, R_{d}, as defined in Eq 4.3. We discuss how we formulate the optimization architecture to maximize R_{d}, in Section 4.3.3. We define and discuss the details of the R_{d}, in Sections 4.3.6, 4.3.7, and 4.3.8.

4.3.2 Assumptions

We make five simplifying assumptions in addressing the problem described above. First, we do not explore dexterous manipulation and grasping of cloth by the robot, as this problem has its own challenges and is being addressed by others [118, 119, 120, 121, 122, 123]. We instead assume that the robot is able to grasp and re-grasp the cloth, which we achieve during experiments through human intervention and special tools. Second, we assume that the participant receiving assistance is collaborative. That is, the participant will move his/her body to the extent of his/her ability in support of the task. Third, we assume the robot can estimate the pose of the participant's body prior to dressing. Fourth, we assume that the person and robot can achieve the desired configurations with negligible effect on the task. Fifth, we assume that human is able to hold the pose for the entire duration of the subtask.

$$\arg\max_{\boldsymbol{\pi}} -\psi C_{\mathrm{n}}(N) + \frac{1}{N}\sum_{i=1}^{N}\big(F(\pi_i)\big) \qquad \begin{array}{l}\text{Top-level optimization}\\ \text{to select subtasks}\end{array} \tag{4.8}$$

Where:

$$\begin{aligned} F(\pi_i) = \max_{\boldsymbol{c}_{\mathrm{h},i}} \Big(& -\zeta C_t(\boldsymbol{c}_{\mathrm{h},i}) - \eta C_s(\boldsymbol{c}_{\mathrm{h},i}, \pi_i(\boldsymbol{c}_{\mathrm{h},i})) && \text{Mid-level optimization} \\ & + \gamma R_{\mathrm{r}}(\boldsymbol{c}_{\mathrm{h},i}, \pi_i(\boldsymbol{c}_{\mathrm{h},i}))\Big) && \text{of human configuration} \end{aligned} \tag{4.9}$$

$$\begin{aligned} R_{\mathrm{r}}(\boldsymbol{c}_{\mathrm{h},i}, \pi_i(\boldsymbol{c}_{\mathrm{h},i})) = \max_{\boldsymbol{c}_{\mathrm{r},i}} \Big(& \alpha R_{reach}(\boldsymbol{c}_{\mathrm{h},i}, \boldsymbol{c}_{\mathrm{r},i}, \pi_i(\boldsymbol{c}_{\mathrm{h},i})) && \text{Lower-level optimization} \\ & + \beta R_{manip}(\boldsymbol{c}_{\mathrm{h},i}, \boldsymbol{c}_{\mathrm{r},i}, \pi_i(\boldsymbol{c}_{\mathrm{h},i}))\Big) && \text{of robot configuration} \end{aligned} \tag{4.10}$$

And where:

- $\boldsymbol{\pi} = \{\pi_1, ..., \pi_N\}$
- $\boldsymbol{t}_{\mathrm{r},i} = \pi_i(\boldsymbol{c}_{\mathrm{h},i})$
- $N =$ length($\boldsymbol{\pi}$)

All subject to:

- Collision constraints
- Garment stretching constraints
- Range-of-motion constraints of the person
- C_n is a cost on the number of subtasks.
- C_t is a cost on torque experienced at the person's shoulder.
- C_s is a cost on stretching the garment.
- $\boldsymbol{c}_{\mathrm{h},\boldsymbol{i}} \in \mathcal{H}$
- $\boldsymbol{\pi} \in \boldsymbol{\Pi}$

Figure 4.2: The optimization performed by TOORAD is split into three levels.

4.3.3 Optimization Architecture

In order to increase the efficiency of TOORAD, we chose an optimization architecture that takes advantage of dependencies between the human configuration, robot configuration, and trajectory that we intend to optimize. The robot configuration and the trajectory of the robot's end effector depend on the configuration of the human. Both human and robot configurations and the robot trajectory depend on the subtask being optimized.

We organize the optimization into three levels, shown in Figure 4.2. We uses layers of optimization instead of joint optimization to reduce computational cost. The top-level optimization is used to select the optimal sequence of subtasks. The mid-level optimization is used to select the optimal human configuration. The lower-level optimization is used to select the optimal robot configuration.

An implication of this structure is that the nested optimizations heavily influence the computational requirements. Instead of optimizing the trajectory of the robot end effector in a low-level optimization, we use previously selected trajectory policies for each subtask. The policy returns a trajectory of the robot end effector given the human configuration. Trajectory optimization for dressing is complicated, as it involves complex physical interaction between the garment and the person's body, which can be computationally expensive to simulate. We trade accuracy in exchange for a policy that can rapidly be applied to many human configurations; the trajectory policy only gives a coarse approximation of the optimal trajectory.

Optimization Algorithms

We provide TOORAD a small set of candidate subtask sequences to optimize, and we use brute-force optimization to consider all options with the set. The space of the human and robot optimization objective functions can be highly nonlinear and challenging to search, and the parameters space is large. These objective functions do not have an analytical gradient and estimating their gradients can be computationally expensive. We perform the

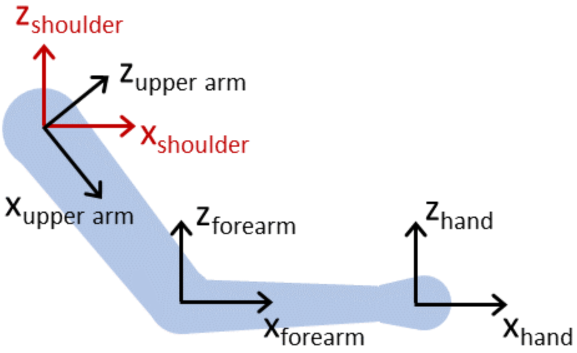


Figure 4.3: The axes used in defining the trajectory policy overlaid on a diagram of an arm.

optimizations of the human and robot configurations for each subtask using covariance matrix adaptation evolution strategy (CMA-ES) [124, 125], which works well for derivative-free, simulation-based, local optimization. We have observed that CMA-ES often performs better for local optimization than global optimization, and that starting with a good initialization often improves its performance.

4.3.4 Selecting Candidate Trajectory Policies

We chose to use a simple, manually defined trajectory policies for each dressing subtask. A rectangular tool holding the sleeve pulls the sleeve first along the forearm, then along the upper arm, and finally moves to the top of the shoulder. Each linear trajectory is defined with respect to a coordinate frame at the base of the link being dressed, with its X-axis along the axis of the link and the Y-axis parallel to the ground plane. Figure 4.3 shows these axes overlaid on a diagram of the arm. The trajectories waypoints written as (x, y, z) in meters, were $(0.1, 0.0, 0.1)$, $(-0.03, 0.0, 0.1)$, $(-0.05, 0.0, 0.1)$, and $(0.0, 0.0, 0.1)$, with respect to the hand, forearm, upper arm, and shoulder, respectively. The policy is fixed for each subtask. Policies for different subtasks are created using these waypoints. For example, dressing the whole arm would consist of moving through all four waypoints and dressing the forearm would consist of moving through the hand and forearm waypoints.

Using a simulator of cloth-person physics, we verified that our chosen policy succeeds in simulation for many configurations of the arm. In simulation we also estimate the space,

$\mathcal{H}_i$, of human configurations, $\boldsymbol{c}_{\text{h},i}$, where the policy succeeds. For example, Figure 4.4 shows the results in simulation of attempting different trajectories for pulling a sleeve onto a person's forearm for different angles of the forearm. The figure also shows what a successful outcome looks like in the simulator as well as what it looks like when the sleeve catches on the arm. The center of mass of the successful trials for angles between 30 and -30 degrees is consistently near 10 cm above the axis of the forearm. This result supports our selected policy. At higher angles of the forearm (from horizontal) the sleeve's opening deforms unpredictably.

The policy π_i gives a trajectory of the robot end effector, $\boldsymbol{t}_{\text{r},i}$, for pulling the sleeve onto the person's body, for subtask i, for any human configurations that lie within $\mathcal{H}_i$, as

$$\boldsymbol{t}_{\text{r},i} = \pi_i(\boldsymbol{c}_{\text{h},i}) \quad \forall \boldsymbol{c}_{\text{h},i} \in \mathcal{H}_i. \tag{4.11}$$

TOORAD uses a fixed-radius neighbors model[1] to quickly estimate if a human configuration, $\boldsymbol{c}_{\text{h},i}$ lies within $\mathcal{H}_i$.

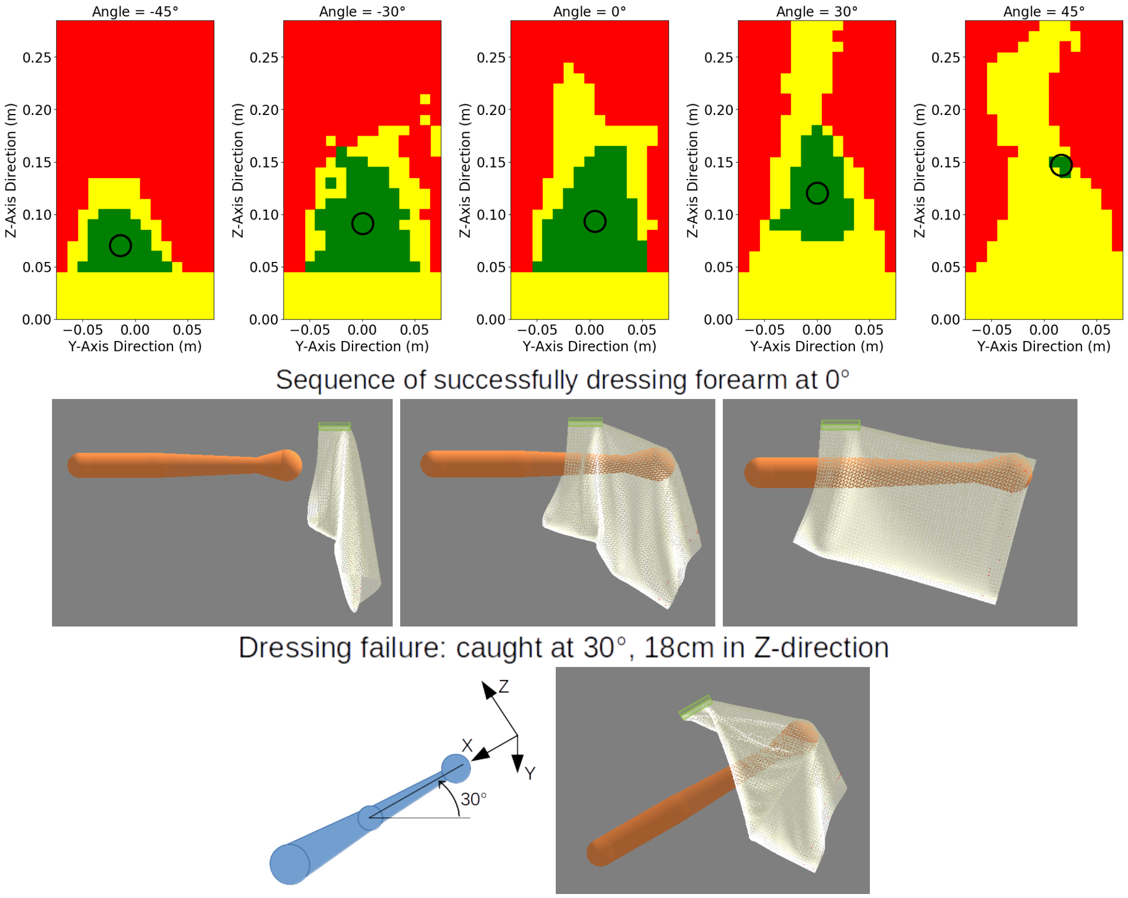
Determining the Human Configuration Space of the Policy

To determine the human configuration space, $\mathcal{H}_i$, where the trajectory policy will succeed for a given configuration of the human body, $\boldsymbol{c}_{\text{h},i}$, TOORAD uses both geometric and physics simulations. We verify that the following criteria hold true:

- The person's body is not in self-collision.
- The robot's tool does not collide with the person.
- The trajectory is successful in the cloth-person physics simulator.

If all criteria are satisfied, then we consider $\boldsymbol{c}_{\text{h},i} \in \mathcal{H}_i$. Otherwise, $\boldsymbol{c}_{\text{h},i} \notin \mathcal{H}_i$. The simulations used to determine the human configuration space where the policy is successful

[1] from `http://scikit-learn.org`



Sequence of successfully dressing forearm at 0°

Dressing failure: caught at 30°, 18cm in Z-direction

Figure 4.4: (Top) Plots showing the outcomes when pulling the sleeve onto a forearm for different poses of the arm. The sleeve is pulled by a tool along the axis of the forearm with varying start position with respect to the arm. Green represents the forearm successfully going into the sleeve, yellow represents the arm getting caught on the sleeve, and red represents the arm missing the opening of the sleeve. The circle represents the centroid of the green area. For Z of ≤ 0.05 m, the tool holding the gown collides with the person's arm. (Middle) A sequence of images showing the sleeve successfully being pulled onto the forearm with the arm at $0°$ from horizontal and the tool moving 10 cm above the axis of the arm. The tool holding the sleeve is colored green. (Bottom-left) A diagram showing the forearm at 30 degrees and the axes of the trajectory. X is along the axis of the forearm, Y is out of the plane, and Z is orthogonal. (Bottom-right) A view in simulation of the sleeve caught on the fist with the arm at $30°$ from horizontal with the tool moving at 18 cm in the Z direction and 0 cm in the Y direction.

are shown in Figure 4.5. The space $\mathcal{H}_i$ estimated once for each subtask using a generic human model based on a 50 percentile male from [76] and a generic cylindrical sleeve of similar dimension to a hospital gown sleeve, and it is referenced for all human users.

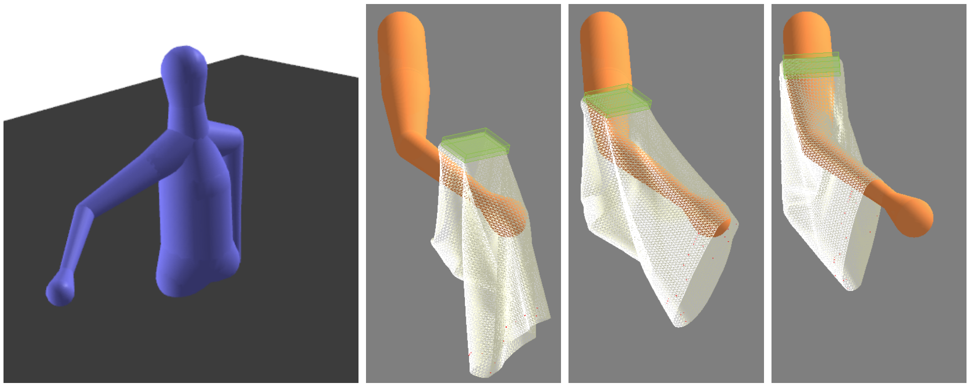


Figure 4.5: Simulation is used to verify that our selected policy for the trajectory of the garment is reasonable and to find the space of human configurations, $\mathcal{H}$, where the policy succeeds. The simulators we used are described in Section 4.4.1. (Left) The DART simulator is used to verify collision constraints. Here we visualize the human model (without legs) in the configuration being evaluated. (Right) The PhysX simulator used to simulate cloth physics when pulling the sleeve onto an arm in isolation in the same configuration (from a different perspective). The upper arm length is extended. The tool holding the sleeve is in green.

4.3.5 Constraints

The optimization of the human and robot configurations uses customized models of the person, wheelchair (when applicable), and garment. Using a geometric and kinematics simulator and a cloth-person physics simulator along with these models, TOORAD enforces six constraints in its optimization. These constraints are:

- Stretching limits of the garment.
- The person's range of motion.
- The person's body is not in self-collision.
- The robot does not collide with the person, wheelchair or garment.
- The garment does not experience interference from the wheelchair.
- $\boldsymbol{c}_{\mathrm{h},i} \in \mathcal{H}$

As the dressing task proceeds and subtasks are completed, TOORAD adds stretching constraints as necessary. These constraints are based on measurements of the real garment.

For example, the maximum stretch between the two shoulders of the hospital gown we used in our evaluation was 0.5 m. Once the person's left arm is in the sleeve of the gown, we assumed the left sleeve would remain in place. Therefore, for the subsequence subtask of pulling on the right sleeve, the right sleeve can move at most 0.5 m from the top of the person's left shoulder. These stretching constraints allow some dependency between subtasks without having to optimize them jointly.

We modeled the person's range of motion as limits to the range of joint angles on a joint-by-joint basis with a 3-axis Euler-angle-joint at the shoulder and a single axis joint at the elbow. We also applied range of motion constraints on the pose of body parts. For example, a range of motion constraint on the pose might be that the upper arm cannot be raised above parallel to the ground.

4.3.6 Subtask Optimization

For the candidate sequences of subtasks we provide, TOORAD performs the optimization in Eq 4.8. The optimization is given a set $\mathbf{\Pi}$ of sequences of trajectory policies $\boldsymbol{\pi} = \{\pi_1, ..., \pi_N\}$. The top-level optimization uses an exhaustive search to select the best sequence of trajectory policies. We define C_{n} as a cost for each additional subtask,

$$C_{\text{n}}(N) = N. \tag{4.12}$$

The function F is the objective function for the mid-level optimization of the human configuration shown in Eq 4.9. We manually selected $\psi = 0.1$ to be small, $\sim 100\times$ smaller, compared to values of F for for human configurations that result in successful dressing.

4.3.7 Human Optimization

Our formulation of F allows TOORAD to simultaneously consider the comfort of the person, the capabilities of the person, the kinematics of the robot, and the physics involved in manipulating garments onto a persons' body. We formulate F in Eq 4.9 as

C_t is the cost on torque due to gravity experienced at the person's shoulder in the configuration $\boldsymbol{c}_{\mathrm{h},i}$, C_s is the cost on exceeding a soft constraint on stretching the garment, and R_{r} is the score of how well the robot can execute the trajectory (see Section 4.3.8). C_t is normalized to $0-1$. C_s has units of distance we allowed it at most a value of 0.04 (meters). R_r is unit-less and ranges from $0-11$. With these ranges, we manually selected 0.5, 5.0, and 1.0 for ζ, η, and γ, respectively. These gains emphasize functionality of the configuration, with more emphasis placed on the value of R_{r} while also considering the other costs.

Cost on Torque

To encourage human arm configurations that are more comfortable for the person, we have included a cost on the torque at the person's shoulder due to gravity. This cost ignores torques due to external forces, such as from the garment, robot, or wheelchair. We divide the torque at the current configuration by the maximum possible torque, to create a normalized torque cost. The maximum torque is when the arm is straight, parallel to the ground. This torque cost is independent of the weight and scale of the person's arm. We use the body mass values for an average male[2]. The cost on torque follows as

$$C_t(\boldsymbol{c}_{\mathrm{h},i}) = \frac{\text{torque}(\boldsymbol{c}_{\mathrm{h},i})}{\text{maximum possible torque}} \tag{4.13}$$

Cost on Stretching the Garment

In addition to the hard constraint on stretching the garment described in Section 4.3.5, we added a soft boundary cost to encourage human configurations where the garment is stretched slightly less. This cost is focused on stretching caused by following the trajectory. We avoid stretching the garment due to the garment become caught by using the trajectory

[2] from `https://msis.jsc.nasa.gov/sections/section03.htm`

policy. For some small distance before the hard constraint, the objective function receives a penalty defined as

$$C_s(\boldsymbol{c}_{\text{h},i}, \pi_i(\boldsymbol{c}_{\text{h},i})) = d_{\text{exceeded}}, \tag{4.14}$$

where d_{exceeded} is the maximum amount by which the soft boundary has been exceeded for all points along the trajectory. This value comes from the kinematics simulator based on the human configuration and the trajectory of the garment (from the trajectory policy). For our hospital gown, based on its physical limits, we set a hard constraint on the maximum distance between the two shoulders of the gown to 0.5 m. We set a soft constraint of 0.46 m, resulting in a maximum value for C_s of 0.04.

4.3.8 Robot Optimization

The reward function R_r estimates how well the robot can execute the trajectory for the given human configuration. TOORAD calculates it using the lower-level optimization of the robot configuration, $\boldsymbol{c}_{\text{r},i}$, shown in Eq 4.10. This optimization is based on [23] with modifications that are relevant to dressing tasks. This objective function uses two measures that we have developed to estimate how well the robot will be able to perform the dressing task: R_{reach}, which is based on TC-reachability, and R_{manip}, which is based on TC-manipulability (from [23]). These two terms are related to task-centric (TC) robot dexterity.

Robot Dexterity Measures

TC-reachability and TC-manipulability are measures that consider the average values of reachability and manipulability across all goals in a task without order. Because dressing has a specific order that the robot should move to goals in its trajectory, we have added a graph-based search to the calculation of these terms to approximate this order and structure. Goal poses for the robot are determined from sampling the trajectory of the robot end effector. For each goal pose, TOORAD uses OpenRAVE's ikfast [126], to find a sample of collision-free IK solutions. These IK solutions are found for the PR2's 7-DoF arms by creating an analytical IK solver for discretized values of one of its joints. We used the robot's forearm roll joint discretized at 0.1 radian intervals. An IK sample is obtained for each discretized value of the forearm roll joint.

From these IK solutions for each goal pose, TOORAD creates a layered directed acyclic graph where a layer is created for each goal pose and each node is a joint configuration of the robot arm. Nodes in a layer are connected to nodes in the next layer if the maximum difference between each robot joint is less than a threshold (we used 40 degrees). This threshold is used to predict that a path exists between the two joint configurations without the time cost of running a motion planner. To facilitate use of a standard graph search algorithm, we add a *start* node connected to each node in the first layer (IK solutions for the first goal pose in the trajectory) an *end* node connected to each node in the last layer. Once it creates the graph, TOORAD performs a uniform cost search [**dijkstra1959note**] through the graph to find a path from start where the cost to each node is the joint-limit-weighted kinematic isotropy (JLWKI) for that node's joint configuration.

Joint-limit-weighted Kinematic Isotropy (JLWKI)

JLWKI is presented in [23] and is based on kinematic isotropy, $\Delta(\boldsymbol{q})$, from [32], shown in Equation 4.15.

$$\Delta(\boldsymbol{q}) = \frac{\sqrt[a]{\det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)}}{\left(\frac{\mathrm{trace}(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)}{a}\right)} \tag{4.15}$$

Geometrically, kinematic isotropy is proportional to the volume of the manipulability ellipsoid of the manipulator, which is the volume of Cartesian space moved by the end effector for a unit ball of movement by the arm's joints. It is based on manipulability from [30], with a modification to remove order dependency and scale dependency. This metric can be useful when assessing kinematic dexterity between different configurations of the same robot. The values of kinematic isotropy are always in the range of 0 to 1 so they can be more directly compared across robot platforms.

JLWKI modifies kinematic isotropy to consider joint limits by scaling the manipulator's Jacobian by an $n \times n$ diagonal joint-limit-weighting matrix $\boldsymbol{T}$, defined in Equation 4.16, where n is the number of joints of the manipulator.

$$\boldsymbol{T}(\boldsymbol{q}) = \begin{bmatrix} t_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & t_n \end{bmatrix} \tag{4.16}$$

t_i in $\boldsymbol{T}$ is defined as

$$t_i = 1 - \phi^{\kappa} \tag{4.17}$$

where

$$\kappa = \frac{q_i^r - |q_i^r - q_i + q_i^-|}{\lambda q_i^r} + 1 \tag{4.18}$$

and

$$q_i^r = \frac{1}{2}(q_i^+ - q_i^-). \tag{4.19}$$

We set $t_i = 1$ for infinite roll joints. The variable ϕ is a scalar that determines the maximum penalty incurred when joint q_i approaches its maximum and minimum joint limits, q_i^+ or q_i^-, and λ determines the shape of the penalty function. We used a value of 0.5 for ϕ and 0.05 for λ. This weighting function and the values for ϕ and λ were selected to halve the value of the kinematic isotropy at joint limits, have little effect in the center of the joint range, to begin exponentially penalizing joint values beyond 75% of the range, and to operate as a function of the percentage of the joint range. JLWKI is then defined as

$$\text{JLWKI}(\boldsymbol{q}) = \frac{\sqrt[a]{\text{det}(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{T}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)}}{(\frac{1}{a})\text{trace}(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{T}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T)}. \tag{4.20}$$

Scoring Metrics

The graph-based search returns either a path from *start* to *end* or the longest path achievable towards *end*. The metric, R_{reach}, is related to the percentage of goal poses along the trajectory to which the robot can find a collision-free IK solution from robot configuration, $\boldsymbol{c}_{\text{r},i}$. R_{reach} is defined as

$$R_{reach}(\boldsymbol{c}_{\text{h},i}, \boldsymbol{c}_{\text{r},i}, \pi_i(\boldsymbol{c}_{\text{h},i})) = (\frac{N_p}{N_{\text{total}}}), \tag{4.21}$$

where N_p is the number of nodes in the path and N_{total} is the total number of goal poses. R_{manip} is then defined as

$$R_{manip}(\boldsymbol{c}_{\text{h},i}, \boldsymbol{c}_{\text{r},i}, \pi_i(\boldsymbol{c}_{\text{h},i})) = (\frac{1}{N_{\text{total}}}) \sum_{i=1}^{N_p} JLWKI(\boldsymbol{p}_i), \tag{4.22}$$

where $\boldsymbol{p}_i$ is the IK solution for the ith goal pose.

4.4 TOORAD Implementation Details

4.4.1 Simulators

TOORAD uses two simulators, one for simulation of human-cloth physics, and a second for simulation of human-wheelchair-robot-garment geometries and robot kinematics. We customized the human and wheelchair models for each participant, and we customized the simulated garments based on the hospital gown dimensions.

We chose to use the cloth simulator in Nvidia's PhysX [127] to simulate human-cloth physics due to its efficiency and robustness at handling large contact forces (e.g., when the sleeve opening is caught on the hand). PhysX is based on position-based dynamics (PBD), which directly calculates position changes instead of through force integration. By avoiding solving and integrating forces and instead modifying position directly, PBD can be more stable, controllable, and efficient than alternative methods. We modified PhysX to add additional functionality and improved accuracy of friction handling, as in the work by [117]. We manually selected the parameters of the simulator so the modeled fabric would behave similarly to the real-world hospital gown used in this work. We modeled the person in PhysX using capsules, or pairs of spheres connected by a conical frustum, for simplicity: these are primitives in both PhysX and DART. We modeled the robot's tool holding the garment as a parallel jaw gripper that holds the shoulder region of the gown. We did not model any other part of the robot. This setup corresponds to the tool we used with the real-world PR2. Figure 4.5 (right) shows the PhysX simulation environment with a sleeve being pulled onto an arm.

TOORAD simulates the human-wheelchair-robot-garment geometries in the DART [128] simulation environment. In this environment, TOORAD performs collision detection using Bullet [129] and uses OpenRAVE's ikfast [126] to get inverse-kinematics for the robot. We set up this second simulation environment because DART has native support for robot and human models. DART uses the same human model as in PhysX, and we also add a customized model of the person's wheelchair within DART, and a coarse geometric representation of the garment. Figure 4.6 shows this simulation environment (the garment model is not shown).

4.4.2 Practical Additions to the Optimization

In our implementation of TOORAD, we used several methods to speed up computation and to improve results. As mentioned in Section 4.3.3, we observed that the optimization algorithm we used, CMA-ES, works better when given good initialization. We used grid-search with a resolution of $2.5°$ for of the 4 DoF of the human arm to quickly and coarsely find candidate human configurations to initialize the optimization of the human configuration (which used CMA-ES). This search found human configurations that are "feasible": where $\boldsymbol{c}_{\mathrm{h},i} \in \mathcal{H}$ and where constraints described in Section 4.3.4 were satisfied. We used a K-means clustering algorithm to choose the candidate human configurations from the feasible configurations. We selected $K = 3$ manually based on the number of regions of feasible human configuration we often observed. We used L_2 distance on the 4 DoF configuration space of the human's arm as the distance metric for the K-means algorithm. We used a population size of 20

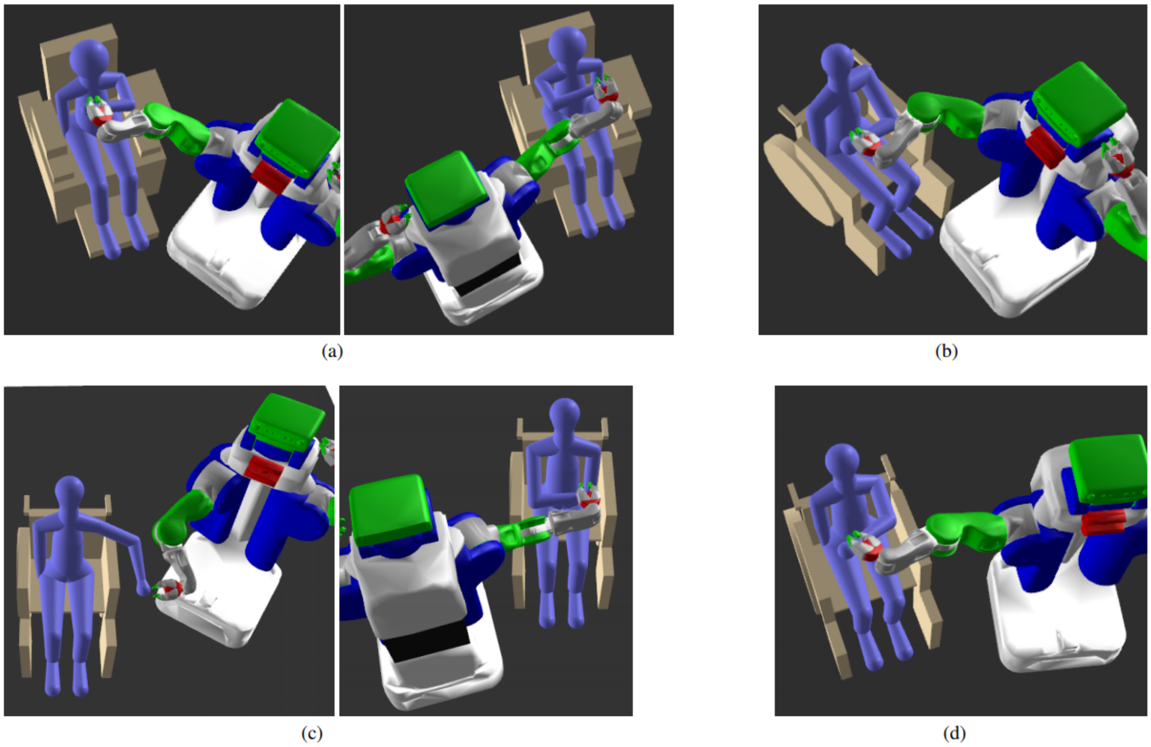


Figure 4.6: The initial robot and human configurations selected by TOORAD for the four participants who interacted with the robot. The models have been customized for the participants. The visualization comes from the DART simulator. The solutions found by TOORAD was shown to participants to provide instruction on their expected arm poses. The configurations are grouped by participant. (b) and (d) show the configurations for the participants who received assistance with only one sleeve.

and 50 iterations for the CMA-ES optimization of the human configuration. We manually selected five candidate robot configurations around the person to initialize the optimization of the robot configuration (which used CMA-ES). These candidate robot configurations were to the front and sides of the person, with the robot's arm facing the person.

In TOORAD, the robot configuration optimization takes place in a lower level. To decrease computation cost of this lower-level optimization, we limited the number of iterations used by CMA-ES to 20 iterations with a population of 20. To compensate, we added a step after the main optimization that, in which TOORAD more finely optimizes the robot configuration. This step used a fixed human configuration from the main optimization, and used the optimized robot configuration from the main optimization as an initialization.

This final step used smaller step-sizes and more iterations (80) for CMA-ES, but otherwise used the same framework.

The grid search took ~ 30 minutes on 18-cores of a 64-bit, 14.04 Ubuntu operating system with 60 GB of RAM and a 2.9 GHz Intel Xeon E5-2666 v3. The main optimization took ~ 10 hours on the same machine. The fine-tuning of the robot configuration took ~ 10 minutes on a single core of the same machine.

4.4.3 System Implementation

We implemented a system to test TOORAD, with sensing and control schemes for the PR2 to allow it to execute actions in collaboration with the participants. In this evaluation we limited the PR2 to using only its right arm. TOORAD found, for each subtask of the dressing task, a configuration for the participant's arm, a pose of the PR2's base, a height of the PR2's spine, a trajectory for the PR2's end effector, and joint configurations for the PR2's arm sampled along the trajectory. In this case, $\boldsymbol{c}_{\mathrm{h},i} \in \mathbb{R}^4$ (3 DoF at the shoulder, 1 DoF at the elbow), and $\boldsymbol{c}_{\mathrm{r},i} \in \mathbb{R}^4$ (3 DoF in the robot base pose, 1 DoF in the robot spine height). We evaluated if, using our system, a participant and the PR2 could collaborate to successfully complete the dressing task.

A challenge when moving from simulation to the real world is that there are often discrepancies from simulation, and sometimes even small differences can result in errors and task failure. We made use of several sensing and control schemes to allow the robot to both recognize and adjust to differences from simulation.

Grasping

When manipulating the hospital gown, the PR2 grasped a tool in its right gripper that held the top of the sleeve of the hospital gown. The tool was instrumented with a capacitive distance sensor and a force-torque sensor. The experimenters attached the tool to the gown before the start of each subtask.

Sensing

We used various sensors on the PR2 and the tool to estimate the pose of the person and track arm motion in real time. Using the PR2's head-mounted Kinect, the system perceives an AR Tag mounted in a known position with respect to the person's wheelchair. The model customized for the wheelchair and the participant was then used to coarsely estimate the pose of the person with respect to the robot and help position the robot appropriately.

As we have noted in previous work [130], visual perception of the person's body during dressing can be challenging as the body becomes occluded by the garment, so we use force sensing and capacitive sensing to make inferences about the state of the dressing task and to more finely estimate the pose of the person's arm. The PR2 used the capacitive sensor on the tool to measure the distance between the tool and the person's arm during dressing and to track along the arm (maintaining a desired distance), as presented by [131]. This sensor's range is roughly from 0 to 10 cm, which works well for this purpose. With the capacitive sensor, the PR2 was able to move the garment along the arm with little contact despite movement in the arm pose or arm poses that did not match the expected pose. Our system also used force sensing for safety and anomaly detection. We additionally used a force-torque sensor in the tool holding the garment for safety and anomaly detection, stopping the PR2's movement if the magnitude of forces reached 15 Newtons.

Control

We implemented a proportional-derivative (PD) Cartesian controller running at 20 Hz on the PR2, to move the tool and gown along the person's arm at a distance of 5 cm. We used TracIK [39] to feed joint-space input to the PR2's low-level PID controllers. The X axis points in the direction of tool (the direction of the sleeve's opening) and the Z axis orthogonal to the capacitive sensor, pointing away from the person's arm. These axes translated and rotated with the tool. The robot obtained waypoints for the movement of the end effector from the end points of the linear trajectory generated by TOORAD with its trajectory policy

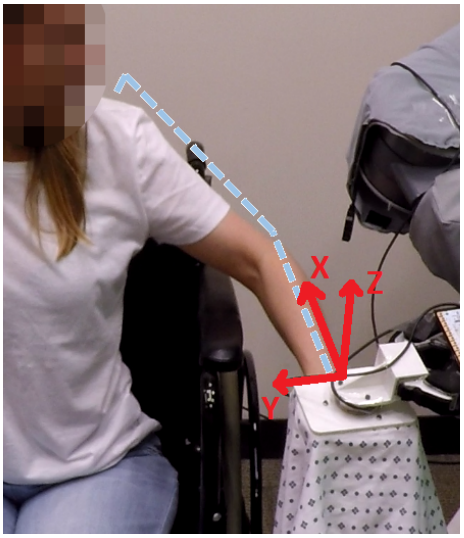


Figure 4.7: The axes used by the controller are shown in red. The axes translate and rotate with the tool. The trajectory for moving along the forearm, upper arm, and shoulder are illustrated in light blue.

for each subtask. For example, for the subtask of dressing the entire left arm, the robot would obtain four waypoints: at the hand, at the elbow, at the proximal end of the upper arm, and at the top of the shoulder. Figure 4.7 shows the axes and the trajectory for dressing the entire left arm for one of the participants. The controller calculated error in the X and Y directions using the waypoints, and calculated error in the Z direction using the measured distance between the tool and the person's arm (desired distance was 5 cm). The desired end effector orientation was to have the X axis pointing along the trajectory and the Y axis parallel to the ground; orientation error was calculated as difference from the desired orientation.

4.4.4 Study with Participants with Disabilities

All studies with participants were conducted with approval from the Georgia Institute of Technology Institutional Review Board (IRB), and we obtained informed written consent from all participants. We conducted a study with six participants with disabilities. The inclusion/exclusion criteria were: all participants must have difficulty putting both arms through the sleeves of a jacket without assistance, be able to raise at least one arm against gravity, have no cognitive impairments, and be ≥ 18 years old. Three of the participants were female and three were male. The age range was 23 - 69. In the first session of the study, we administered questionnaires on the participant's current habits, needs, and physical capabilities as related to dressing and their views on the potential for robots to assist with dressing. We additionally took measurements of their bodies (e.g., arm length) and wheelchair (we provided them a wheelchair if they did not use one), and we measured the comfortable range of motion of their arms.

We invited all participants whose physical capabilities were matched to the capabilities of our system back for a second session of the study in which the participant and the robot collaborated to pull the two sleeves of the hospital gown onto the participant's body. Two of the six participants did not match this criteria because they could not comfortably hold their arms up against gravity. Four of the six participants took part in the second session. For these tests, TOORAD was run on a model customized for the participant in advance of their arrival.

Participant Details

Because the responses, needs, capabilities, and views of participants varied greatly depending on their disabilities, we describe the disabilities of each participant as is relevant to dressing. These descriptions come from the administered questionnaire in the first session of the study.

Participant 1 Participant 1 has cerebral palsy that causes involuntary movements and makes precise movements challenging. He/she uses a wheelchair, which we modeled as shown in Figure 4.6(a). He/she has a full range of motion of both arms and can attain and hold most poses with both arms, but the trajectories taken to reach poses can vary and are not fully voluntary. The participant said it can be hard to get his/her hands in to the right spot, holes, etc, and said that the hardest part of dressing is jackets, buttons, zippers, shoes, and tasks requiring fine motor control. This participant received assistance from the robot with dressing both arms. The configurations selected by TOORAD for this participant are shown in Figure 4.6(a).

Participant 2 Participant 2 is an older adult who is post-polio. He/she has weakness in his/her arms and legs and has no feeling in hands. He/she has limited range of motion with pain near the edges of the range, and cannot raise his/her arms to parallel with the ground and cannot move his/her right arm in front of his/her body. He/she is quickly tired and pained by raising arms and holding arms raised. This participant did not take part in the experiment with the robot because he/she could not comfortably raise his/her arm.

Participant 3 Participant 3 has partial paralysis of the right arm due to a C3-C8 spinal cord injury. Paralysis is of the arm distal from the elbow and of the bicep. He/she has no other impairments or limitations. This participant used his/her unimpaired arm to dress his/her impaired arm. The participant then received assistance from the robot in dressing his/her unimpaired arm. The configurations selected by TOORAD for this participant are shown in Figure 4.6(b).

Participant 4 Participant 4 has ALS that has caused muscle weakness in the shoulders, arms, hands, and legs. He/she can briefly raise arms against gravity, but cannot hold them up. He/she has weakness in his/her wrists and hands that make grasping and pulling on garments, especially pants, difficult. This participant did not take part in the experiment with the robot because he/she she could not comfortably hold up his/her arm.

Participant 5 Participant 5 has an amputation of the dominant right hand, located in the right forearm, near the wrist, and has a torn right rotator cuff. He/she experiences tensing of the left shoulder from overwork that is uncomfortable and that can sometimes involuntarily pull his/her head, neck, and left shoulder together. He/she experiences discomfort reaching the left arm across the body. He/she has a prosthetic but does not consider it a helpful tool for dressing because it cannot grip sufficiently dexterously or strongly to be helpful. This participant received assistance from the robot with dressing both arms. The configurations selected by TOORAD for this participant are shown in Figure 4.6(c).

Participant 6 Participant 6 has cerebral palsy whose symptoms include motor impairments, fatigue and balance issues. Impairments are primarily of the right side of the body. Lacks certain motor control in the right arm, specifically of the right elbow and wrist. His/her right shoulder has limited range of motion and decreased control. His/her flexion of the right elbow is not directly controlled and is dependent on the pose of the right shoulder. His/her right wrist is always flexed. He/she sometimes experiences involuntary spasms, most typically of the right arm. His/her left arm has some minor and limits in range of motion and he/she experiences balance issues when seated when reaching across body from left side to right side. This participant used his/her less impaired arm to dress his/her impaired arm and received assistance from the robot with dressing his/her unimpaired arm. We found that our method of modeling range of motion was not well matched to the motion constraints of the participant's right arm. The configurations selected by TOORAD for this participant are shown in Figure 4.6(d).

Questionnaire on Habits, Needs, Capabilities, and Thoughts on Robot-Assisted Dressing

The purpose of this survey was to gain insight for future robot-assisted dressing research through a better understanding of target populations. In this section, we describe the types of questions asked in the questionnaire. A discussion of take-aways from the questionnaire responses is in Section 4.5.

We asked a series of questions regarding typical dressing habits, such as typical dressing scenarios, number and types of dressing tasks, and time-to-complete, to learn how a robot might fit into a daily routine. We further inquired about which dressing tasks and articles of clothing are most challenging and how participants might prefer a robot to help meet their dressing needs. We included questions to survey participants' physical capabilities, as

Table 4.1: Responses to seven-point Likert type questionnaire items from participants with disabilities. Pre-experiment questions were administered in the first session of the study. Post-experiment questions were administered after interaction with the robot in the second session of hte study. The second column provides the average and standard deviation of scores with 1=strongly disagree, 2=disagree, 3= slightly agree, 4=neither, 5=slightly agree, 6=agree, and 7=strongly agree.

Pre-experiment Questions ($N = 6$)	Score: mean (std)
When I select clothing, the ease of don and doff is important.	6.3 (0.74)
If a robot could assist me with dressing to the same effectiveness as my current methods, I would allow the robot to assist me with dressing.	6.3 (0.74)

Post-experiment Questions ($N = 4$)	Score: mean (std)
The system successfully accomplished tasks	7.0 (0.0)
I felt comfortable using the robot-assisted dressing system.	6.3 (1.15)
I would prefer receiving dressing assistance from a robot than another human	4.7 (1.53)
I would like to receive this form of dressing assistance from a robot in the future	5.3 (1.53)

related to dressing, including range-of-motion, involuntary movements, and fatigue. These responses might help with the design of methods or systems for specific target populations in the future. We discussion the responses to the questionnaire in Section 4.5.

Table 4.1 summarizes the participants answers to two Likert type questionnaire items asked during this questionnaire. Participants were generally positive towards receiving assistance from a robot with dressing.

Experimental Protocol

Before the start of the dressing task, each participant was given verbal instructions by the experimenter and shown a visualization of the DART simulator with the poses expected by the robot and the trajectory the robot would move along. The experimenter provided each participant multiple view points of the poses in the simulator. Figure 4.6 shows snapshots from that visualization for each of the participants.

Each participant started the experiment seated in a wheelchair. Participants who did not

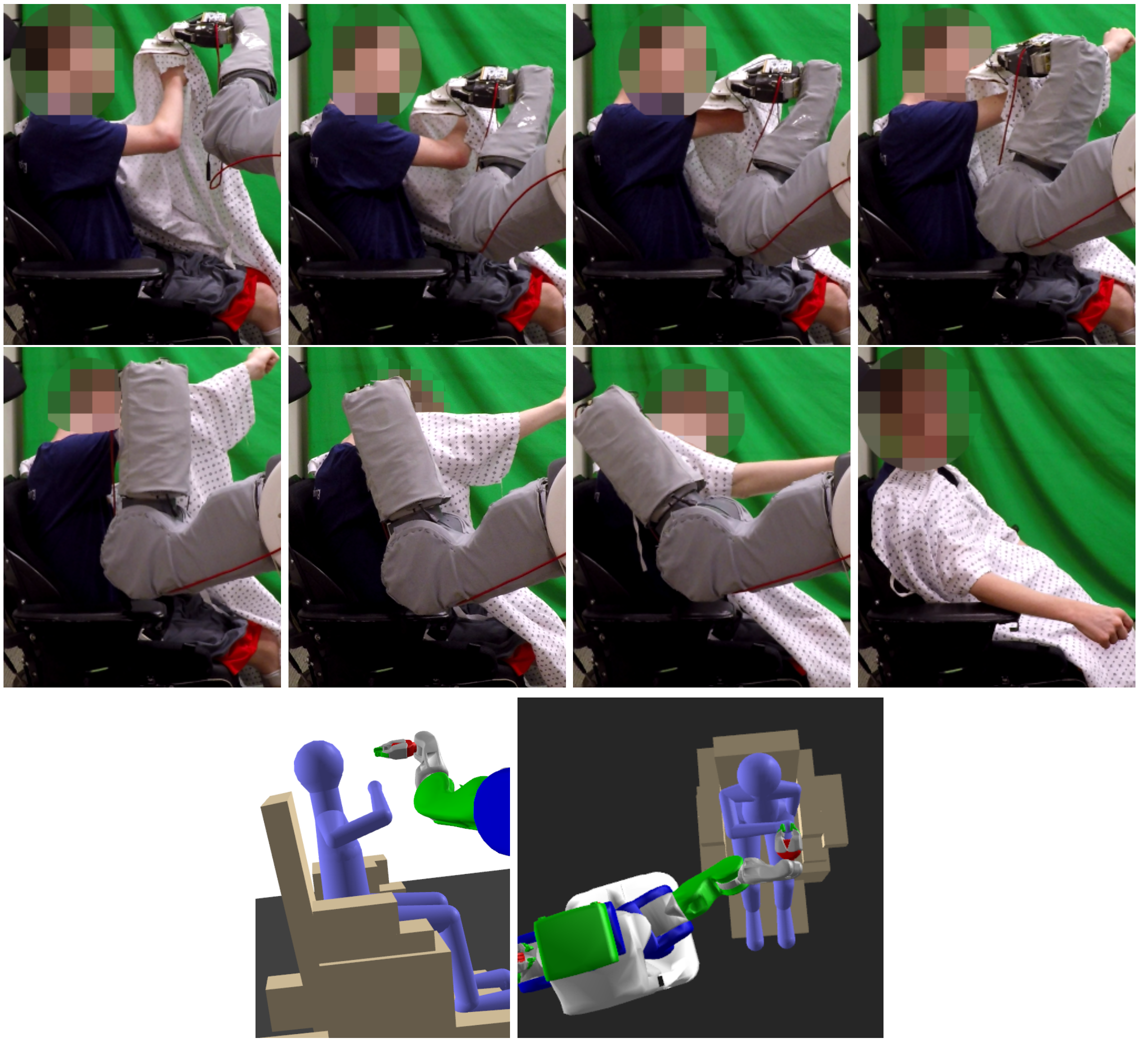


Figure 4.8: (top) A sequence of images from a participant receiving dressing assistance with the right arm of the hospital gown. The robot successfully pulled the sleeve up the arm as the participant's arm moved involuntarily. (Bottom) Views from the DART simulator showing the initial configurations of the person and robot for dressing the right arm.

bring their own, were provided a wheelchair by the experimenter. TOORAD used a model of the wheelchair in its simulations.

The goal of participants was to dress themselves, pulling on the two sleeves of the hospital gown, with assistance from the robot. Depending on the participant's level of impairment and the assistance they desired, participants either received assistance from the robot with one or both sleeves. Two of the participants had one impaired arm and one unimpaired arm. These participants pulled the gown unaided onto their impaired arm and then received assistance from the robot with dressing their unimpaired arm. Two of the participants received assistance with pulling on both sleeves. TOORAD was used to select the actions of the participant and the robot for coordinating the assistance. For each sleeve with which the participant received assistance, the participant held his/her arm in a pose as the robot moved the sleeve along the arm. Figure 4.8 shows a sequence of images from dressing the right arm of a participant as well as the expected pose of the person and robot in simulation. For all four participants, TOORAD's optimized sequences of actions consisted of a single subtask for dressing each arm. The human and robot configurations selected by TOORAD for each of the four participants is shown in simulation in Figure 4.6.

The PR2 began the experiment by approaching the participant. An experimenter then attached the PR2's tool to the gown. For participants receiving assistance with both sleeves, the PR2 started with the gown held in its tool. For participants receiving assistance with only a single sleeve, the experiment started with the gown being handed to the participant to independently dress his/her impaired arm. For all participants, once the robot was in position, the participant started the task by moving his/her hand within 5 cm of the capacitive sensor on the tool.

Once the participant started the task, the PR2 moved the sleeve slowly up the participant's arm until the robot completed its trajectory or the 15 Newton threshold was reached. At that point the experimenter judged if the subtask was successful by examining if the arm was in the sleeve, the sleeve was pulled up the participant's shoulder and the sleeve was not caught on anything. If successful, the robot would move on to the next dressing subtask. If the subtask was unsuccessful, the entire dressing trial was deemed a failure and the experimenters would set up the next trial. Between subtasks, the experimenter would detach and reattach the PR2's tool from the gown so the robot could drive its base around freely without affecting the gown or the participant.

All movement of the robot's base was controlled via teleoperation by the experimenter. The experimenter moved the robot to the base pose selected by TOORAD.

We gave the opportunity to participants to practice receiving assistance from the robot with putting on the sleeves of the hospital gown up to five times prior to the official test. During practice, the experimenter provided feedback to the participant on how to improve coordination with the robot. For example, if the participant was not attaining the correct arm pose, the experimenter demonstrated the arm pose selected by TOORAD. The experimenter also responded to questions from the participant. After the practice trials, the participants attempted to dress themselves four times with the robot's assistance. During the official test, the experimenters responded to questions, but otherwise did not provide feedback. At the end of the experiment, we administered a questionnaire with 7-point Likert type questionnaire items as well as three open-ended questions.

Results: Robot-Assisted Dressing System

All four participants successfully dressed both of their arms using assistance from the robot four times consecutively. Success criteria was: both arms were in the sleeves, the sleeves were pulled up to the shoulders, and the gown did not get caught on the person or wheelchair. Figure 4.9 shows a sequence of images from a participant successfully collaborating with the robot to dress her right arm.

One participant experienced a failed trial in addition to the four successful trials due to the experimenter moving the robot to the wrong base position. Another participant experienced a failed trial in addition to the four successful trials due to an erroneous IK solution that asked the robot to spin one of the roll joints of its arm 360° during the task. The two failed trials were retried and subsequently were successful.

Table 4.1 summarizes a few of the participants' answers to Likert type questionnaire items asked after interaction with the robot-assisted dressing system. Participants strongly agreed that the system successfully accomplished tasks, but were more mixed on comfort and future assistance from robots. In open-ended questions, participants were positive about the robot's ability to move along their arm, tracking their arm as it moved up or down.

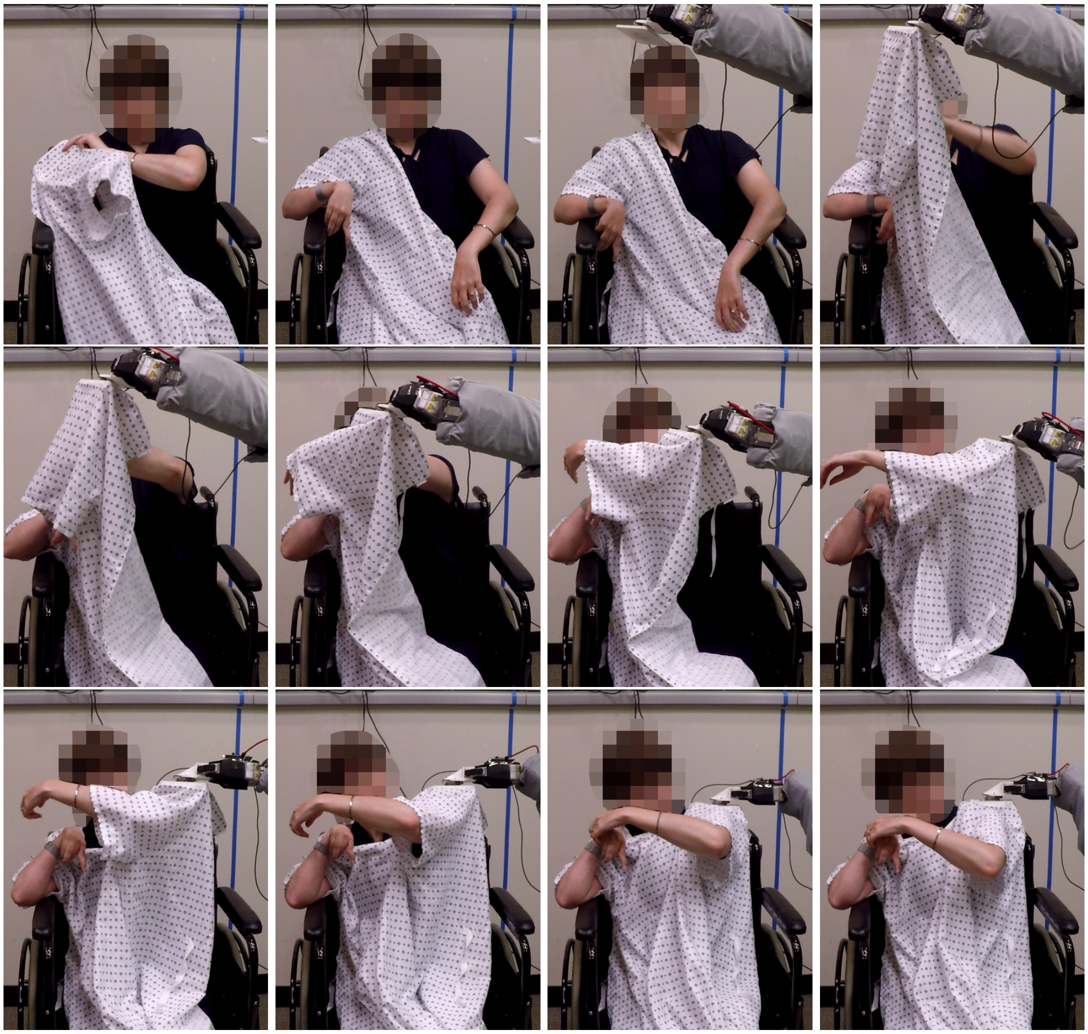


Figure 4.9: A sequence of images (going left-to-right, top-to-bottom) showing a participant successfully putting on the sleeves of the hospital gown with assistance from the robot. This participant independently put on the right sleeve of the gown onto his/her more impaired arm and received assistance with putting on the left sleeve onto his/her less impaired arm. The expected configurations of the participant's body is shown in Figure 4.6(d). The participant's arm pose differed from the expected configuration.

This tracking was particularly important for one participant who experiences involuntary movement of his/her arms. Figures 4.8 and 4.9 show sequences of images from two participants receiving dressing assistance from the robot, in which the participants moved their arms during the execution of the task.

Some of the feedback from participants was positive, including the following quotes: ”This is pretty kick-ass”, ”You are not following the robot, the robot is following you”, and

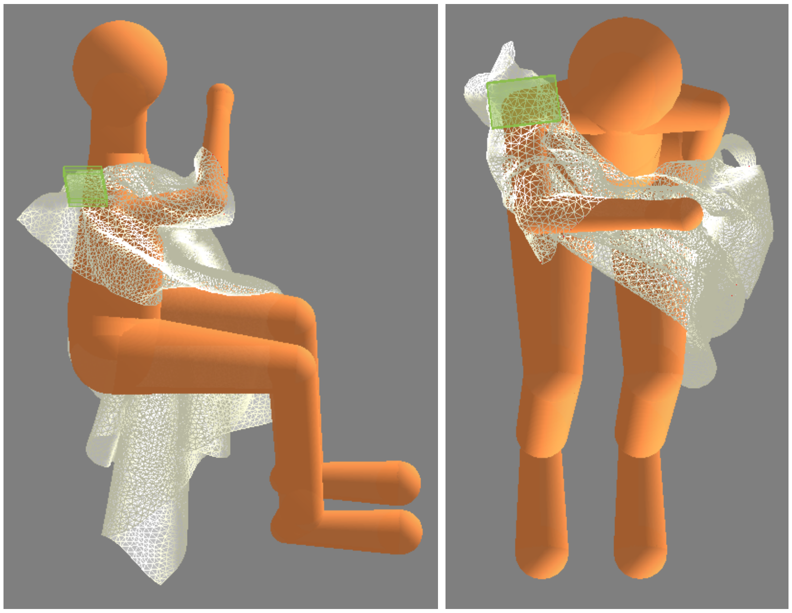


Figure 4.10: The gown successfully pulled onto the right arm of a model of one of the participants using the robot end effector trajectory and human configuration selected by TOORAD. This test was run in the PhysX simulator. The dimensions of the torso, legs, and feet were not customized to the participant. (Left) View from the side. (Right) View from the top.

”It was pretty intuitive”. One participant expressed discomfort with the start position of the robot’s arm: ”I was uncomfortable with how close it was to my face”. One participant suggested that ”Other garments would be great”, in addition to hospital gowns.

Results: Confirmation in Simulation

We also confirmed that the results of the optimization could be used for successful dressing assistance in simulation. This could potentially be used to check solutions prior to testing with people or as part of a process to further improve the solutions. We executed the trajectories of the robot end effector holding a simulated gown in the PhysX simulator using the customized models for each participant, pulling each of the two sleeves onto the arms independently. In all cases, the sleeve of the gown was successfully pulled onto the human model’s arm. Success was judged through manual visual inspection in the PhysX simulator. Figure 4.10 shows an example of having successfully pulled the right sleeve of the gown onto the right arm of a model of one of the participants.

4.5 Discussion

Although we only tested TOORAD on with a PR2 providing dressing assistance for the sleeves of a hospital gown, we expect aspects of the approach may generalize to other problems. We expect that the overall method, the equations, the modeling tools, and the instructional visualization, and parts of the system would generalize to other garments, tasks, robot end effectors, and robots. Other robot models can be swapped in for the PR2 in DART and ikfast, requiring few other changes. Other robot end effectors would require new tool models in PhysX and DART, and might require modifications of the trajectory policies for the end effector. Other garments and dressing tasks would require new garment models in PhysX and the selection of new trajectory policies for subtasks. Many garments share properties with the hospital gown: many require that a tube of cloth be pulled over an appendage and can have the dressing task split into subtasks. These garments would likely use some subtasks and trajectory policies similar to the ones used for dressing the sleeves of the hospital gown.

The TOORAD's distinctive features from robot-assisted dressing literature is its optimization of the robot configuration and its use of geometric and physics simulation to perform its optimization offline with many iterations. It's objective functions allows TOORAD to select configurations in which the robot can adapt to changes in the trajectory. TOORAD could potentially be used in conjunction with methods from literature. Many robot-assisted dressing methods from literature use learning from real-world trials to perform trajectory optimization and improve dressing performance. These methods could be used to refine the trajectories selected by TOORAD after interaction with users.

4.5.1 Limitations

There are some limitations to our work with TOORAD and our evaluations. Although the problem formulation starts with exploring the whole space of possible actions by the robot and person, we constrain that space to achieve tractability as described in Section 4.3.1. However, actions that we removed from our actions space may be valuable for dressing. Feedback from participants suggests that our constrained action space may be preferred by for some people, but that different actions may be preferred by others.

We implemented a robot-assisted dressing system with the purpose of testing TOORAD in the real world with participants with disabilities. Many avenues exist to improve that system. Use of the system required experimenter intervention to attach and reattach the robot's tool to the gown, move the robot's base, and configure the robot's arm prior to starting tasks. Previous work has addressed many of these actions [75, 23], and could be added to the system.

Although this method of modeling range-of-motion constraints worked sufficiently well for many of our participants, it may not work for all people. For example, one of our participants experiences dependencies between joints where the range of motion of one joint depends on the configuration of another joint. This type of range of motion limitation can be difficult to capture using our method for modeling range of motion. We applied range-of-motion limitations to the optimization as a binary function: if the person's configuration is out of range, the configuration is discarded. Other binary functions for checking range-of-motion, such as that by [**akhter2015pose**] and [**jiang2017data**], could replace the range of motion modeling currently used in TOORAD.

4.5.2 Participant Survey Highlights and Lessons Learned

We expect that the responses to the survey of needs, habits, physical capability, and thoughts on robot-assisted dressing of the six participants with disabilities can guide future research. Here we present some key take-aways and lessons learned from conducting the study with participants with disabilities.

Dressing assistance needs varied

The need for assistance for each type of garment varied by participant, with one exception. All participants indicated that they would like assistance with buttons (and many also with zippers), because they require fine motor control of both hands. Otherwise, participants wanted assistance with different garments and wanted different forms of assistance. All participants suggested that the robot should be able to help with many garments and dressing tasks.

Participants indicated consideration of the difficulty of dressing is important when selecting the garments they choose to wear. Several also suggested that if specialized garments would allow them to receive dressing assistance from a robot instead of from a human caregiver, they might be open to wearing such garments.

A robust task-execution system is helpful

We observed that having a system that could adapt the robot's movements to a dynamic environment using some sort of feedback mechanism was important for successful execution of the dressing tasks. For example, as noted in Section 4.4.4, one participant experienced large involuntary arm movements. Using a capacitive sensor as input in a feedback controller, the robot was able to follow the participant's arm as it moved, and successfully completed the task. Other forms of feedback and control to improve the robustness of task-execution may be valuable. We note that aspects of TOORAD increases the robustness of the solution configurations and trajectories of the person and robot. In particular, the representations of robot dexterity used in the robot configuration optimization helps select a robot configuration where the robot is likely to be able to reach poses near the selected trajectory. Feedback control without a good configurations of the person and robot might not work as well.

Keep users physically involved in the task

Feedback from all participants suggests that it is important for users to make use of their capabilities and to feel more involved. Some participants suggested that a high level of involvement would make them feel empowered and more like they are independently dressing themselves with aid from a robot. They presented this feeling in contrast to feeling dependent on dressing assistance from a human caregiver. Participants with fine control of the larger movements of their arms and body preferred that the robot hold a sleeve out for them to put their arm through, over pulling the sleeve onto their arm. Participants with decreased control of the gross movements of their arms preferred that the robot pull a sleeve onto their arm as they hold up their arm, over having the robot manipulate their body.

Customization is important

In our study with six participants we observed large variation in the needs and capabilities of participants. We found that our robot-assisted dressing system was only well matched to the capabilities of four of the six participants. The other two participants were unable to comfortably hold their arms raised, and the system could not manipulate the person's body to achieve a pose of the arm. Participants also differed on their preferred level of involvement in the task, and suggested that they might want a different level of involvement depending on the day, their fatigue, and the task. It may be valuable to have different modes of assistance that both depend on the needs of the particular user, and also on the current preference of the user. It may also be valuable to select a target population and design robot-assisted dressing systems and methods for that population.

Participants were interested in general-purpose mobile robots

A frequent question from participants was if the robot used to assist with dressing might be able to assist with other tasks around the house. Participants motivated this question both with a practical perspective (they desired assistance with many other tasks), as well as an economic perspective (if they were to spend money on a robot, they would want it to do many things). This view supports our approach of using a general-purpose mobile manipulator for robot-assisted dressing, which is more likely to be able to do other tasks than a fixed-base robot or a robot that has been customized solely for dressing.

Robot capabilities may limit possible assistance

Our system successfully provided dressing assistance to four of the participants by pulling one or two two sleeves of a hospital gown onto the participants' bodies. However, our dressing system and our initial action policy constraints were not well suited to some participants. We suggest a few changes that may improve the ability for a dressing system to assist more participants. Our system used a PR2 robot that has weak arms with a max payload of ~ 1 kg. With this limit, it is challenging for the robot to manipulate a user's body. A stronger robot might allow other forms of assistance. The PR2 also has a relatively limited workspace. We found that workspace sometimes causes the robot to require multiple base locations to perform a task or to be unable to perform a task. A larger workspace may be beneficial.

Consideration of the entire dressing task is hard but may be valuable

TOORAD breaks up dressing tasks into mostly independent subtasks for computational efficiency. It additionally limits the action space of the person to holding still as the robot pulls on the garment. We observed that these two choices can limit the feasibility of some tasks. For example, stretching constraints for dressing the second arm after one is dressed may not be satisfiable for some garments or some people with limited range of motion. Dressing both arms simultaneously or alternating dressing parts of each arm (e.g., dress left forearm, then right forearm, then left upper arm, etc), might make dressing possible for more garments and people. We observed that simultaneously dressing both arms may be challenging for a robot with a limited workspace, like the PR2. Alternating dressing parts of each arm may require joint optimization of the two arms, which may present computational challenges.

Additional subtasks are sometimes necessary

For all participants with which we tested our system, TOORAD found that the optimal sequence of subtasks was a single subtask for pulling the sleeve onto each arm. However, we found that for some human models, such as those with longer arms, more subtasks were necessary. In part, this is because it can be difficult to fit the trajectories for dressing an entire arm into the robot's workspace at a single base location.

4.6 Conclusion

In this chapter, we have presented task optimization of robot-assisted dressing (TOORAD), a method to use optimization and simulation to select actions for a robot and a person to collaborate in a dressing task. TOORAD makes use of geometric, kinematic, and physics simulation of the person, robot, and garment in its optimization. It uses customized models for the person to match their geometries and physical capabilities and selects configurations of the robot to allow the task to be successful despite errors between simulation and the real world. These models consider what the person is capable of doing, instead of what he or she typically does. With this approach, TOORAD is able to explore a wide range of actions for dressing in simulation, some of which might be challenging to test in the real world, or might differ from what human caregivers do. We used TOORAD to optimize the actions of a person and a PR2 robot to collaborate in pulling the two sleeves of a hospital gown onto the person's body. We performed a study with six participants with disabilities who require assistance with dressing tasks to survey their needs, capabilities, habits, and views on robot-assisted dressing to guide current and future research in robot-assisted dressing. We implemented a robot-assisted dressing system using a PR2 robot and tested the actions selected by TOORAD by dressing four of the six participants in a hospital gown. The system used capacitive sensing to allow it to modify the trajectory selected by TOORAD to match unexpected error in the person's pose or movement of the person's arm, and TOORAD's optimization selected a robot configuration that allowed the robot to reach these end effector poses in the modified trajectory. From our study and the participants' responses to questionnaires we provided suggestions for future on robot-assisted dressing for people with disabilities.

CHAPTER 5
CONCLUSION

In this chapter we discuss level lessons learned, present recommendations for researchers in this, and highlight opportunities for future research to extend these results. Additionally, we review the results and the potential impact of this work as a whole.

5.1 Lessons Learned

We learned various lessons along the process of performing this work, not described previously. These include design recommendations for the development of future systems.

5.1.1 Assistive Mobile Manipulation

Good placement of the robot is important

As described in Chapter 2.2, much work has explored ways to choose the placement of a robot's base for a task. The placement of the robot is important to the performance of the task. Nevertheless, this problem is sometimes trivialized and is most frequently addressed using a simple IK solver. In our work in Chapter 2, we compare our method, TOC, to baseline algorithms, including a simple IK solver. Simple IK solvers are often used because they are fast, online, simple to use (off-the-shelf tools exist), simple to set up (many standard robot platforms come off-the-shelf), and are task-generic. As an example for comparison, TOC is task-specific, and requires computation offline for each task.

In our tests both in simulation (Chapter 2) and in the real world (Chapter 3), we observed that good placement of the robot is both difficult and important. State estimation is often noisy, so it is important that the robot be able to perform the task despite noise. This is particularly important when following the common robotic-manipulation paradigm of performing manipulation from a static base pose. It is potentially even more important when choosing where to place a fixed-base manipulator whose base position cannot easily be altered. We observed that base poses chosen using simple IK solvers are brittle: they only say that if the real world exactly matches the IK solver's model, the robot will be able to reach the goal poses. They might be able to reach nearby poses, but have no relevant metric.

Mobile manipulation is challenging but powerful

In our interactions with participants, we frequently received feedback in favor of generic mobile manipulators over fixed-base and task-specific robots. Although the technology is not yet there, participants expressed hope that a robot that helps them with any one task will also be able to provide assistance with many other tasks around their body and around the home. We found that creating assistive robotic systems with a mobile manipulator, or specifically, making use of the mobile base of the mobile manipulator, introduces many problems that might otherwise be ignored. In much work from literature, the base pose of the robot is ignored, selected using a simple IK solver, or manually selected by the researcher or experimenter. Making mobility a feature of assistive robotic systems is exciting for the user and can allow the robot to assist with many more tasks in more places. However, it creates new questions that must be solved: "To where should the robot move?"

Accuracy, errors, and small details can make or break a system

When deploying robots in the real world, ensuring high accuracy and accounting for small details is critical for success. When reaching to wipe a user's nose, a centimeter of pose estimation error can result in collision with the nose. In early development of a robot-assisted dressing system, we observed that when executing a trajectory open-loop to pull a sleeve onto a person's arm, very small deviations of the person's arm from the expected pose could result in failure. We later used a capacitive sensor to track the person's arm. Use of accurate, local sensors and feedback controllers to compensate for errors can help to make a system robust to inevitable discrepancies between simulation/expectation and reality.

5.1.2 Optimization

Optimizations optimize whatever is in their objective function

This statement seems both obvious and a classic source of challenge in using optimizations. The objective function is optimized, no matter the intention or the desired results of the programmer. Sometimes the results are interesting, finding solutions the programmer may not have expected. Other times, the optimization exploit some undesired loophole in the objective function. In our work on TOC, the optimization found some unexpected configurations that we found intriguing. We expected that the shaving task would require two configurations of the robot, one of each side of the person. However, for a person in bed without a wall behind the bed, the optimization found the robot could shave the person from a single configuration if placed behind the bed. Early in our work on TOORAD, the optimization found actions for the person and the robot where the robot snaked its arm around the person's arm. Those actions would not be feasible in practice, because the robot's arm would interfere with the garment. We later modified the models and objective function to ensure that the optimization would discard such actions.

Understand your optimization algorithm

When using an optimization algorithm from literature, it is important to understand the strengths and weaknesses of your chosen optimization algorithm. For example, some algorithms work better for global search than local search and some work better with simpler objective-function-landscapes. In our work, we primarily used CMA-ES. We found that CMA-ES works well for local optimization of highly non-linear and derivative-free objective functions, but we found it to be less well suited for global optimization. To improve performance of TOC and TOORAD, we took steps to give the CMA-ES optimizations good initializations. We found that grid-search (or brute-force optimization) can work well for searching the entire configuration space to find initializations for CMA-ES.

5.1.3 Design Recommendations

Provide reasonable defaults and allow customization

Designing systems for human users is challenging, as preferences and capabilities can span a wide range. Additionally, these preferences and capabilities can change over time. We have found, especially for novice users and those who fall in some "typical" category, default behaviors and models can work well. For example, in preliminary informal testing of our robot-assisted dressing system, we observed that we could provide assistance to a wide range of sizes of able-bodied participants when trained on a 50-percentile male model. Customization becomes more important as either the participant diverges from the default model or preferences arise in the user. We found with our work with Henry Evans, an expert user of the PR2 for assistive tasks, that allowing customization for preference can be important for user satisfaction. Customization to the capabilities of the user is particularly important for people with disabilities, where it may be valuable to take an individualized approach to design.

Allow users access to low-level control

In Chapters 2 and 3 we investigated methods to provide some level of autonomy to an assistive robotic system. The autonomy can reduce the complexity for the user of using the system, improving performance and usability. However, we recommend giving access to low-level control of all aspects of the robot in some way. Once deployed, there are countless ways that an autonomous system can fail or that a user can attempt to leverage the autonomy in ways the designer does not expect. Providing the user with methods to customize the autonomy, to override behavior, and to wrest control of the robotic system is important to allow the system to handle all cases to the user's satisfaction. These capabilities can mitigate user frustration when an autonomous action does not match the user's desires.

We made use of this feature in our assistive robotics system in Chapter 3, where we gave the user low level control to perform the task after autonomy moved the robots into place for the task. Low level control could allow the user to use an autonomous function for wiping the mouth and then instead use the robot to wipe the nose, scratch the chest, etc.

5.1.4 Design Process Recommendations

Engage with varied users of the target population early and often.

Our work on TOC and the assistive robotic system implementing TOC received feedback primarily from Henry and Jane Evans' and we tailored our approach and the systems we developed to meet his needs and capabilities. Although they often pushed us toward systems and features that could apply to other people with disabilities, their experience is limited to that of two people. In the later study on robot-assisted dressing with people with disabilities, we caught a hint of how varied disabilities can be and how innovative people can be in managing disabilities. Each participant in that study had different capabilities, needs, habits, and views on how they would want a robot to provide assistance with dressing.

We did not conduct that study with people with disabilities until 3 years into our work on robot-assisted dressing. When it became time to test TOORAD with a target population, we first had difficulty identifying a target population whose capabilities and needs were well matched to TOORAD's approach. In retrospect, earlier consideration of the target population may have aided in the research process and making our work more easily applicable and testable. Earlier engagement with varied users of the target population may have benefits of allowing a user-centered design process, iterative improvement of systems, and ultimately more practically useful outputs.

Receiving Feedback

Receiving feedback: Sometimes users are right and sometimes they are less right

Feedback from a variety of sources, especially users of the target population, tends to significantly improve the resultant system. However, incorporation of feedback into a system design can be challenging as no single design will satisfy all users. Additionally, The system designer may possess expertise or a vision of the system that users do not share. We found that user feedback may disagree with the thoughts of the designer.

It is necessary, to carefully consider all feedback, no matter how absurd it may seem. Consider the content of the feedback, context of the feedback, rationale for the feedback. Ask for further explanation to better understand the feedback. In cases where user satisfaction is most important, it may be necessary to cater to the whims of the user despite doubts on the part of the designer. In many cases, the user identifies important and relevant issues that the designer may not have considered or may have assigned less significance. Other times, the user's feedback may be based on experiences or preferences that are not applicable, or they may be based on a lack of expertise.

We received much feedback from Henry Evans during our development of the assistive robotic system. All of it was extremely beneficial and helped improve the system. From that feedback, I will give one example where the user was "less right", but the feedback was still valuable. During initial development of the assistive robotic system, Henry would sometimes demonstrate how he would teleoperate the PR2 to perform an assistive task around his body. He would often completely straighten one of the PR2's arms, point it at his own body, and begin driving the robot's base towards himself. Similarly, when teleoperating manipulation around his body, he would often keep the PR2's arm straight and use the base movements to position the arm's end effector. When the firsts tests of our system failed spectacularly, he suggested making the assistive system copy the actions and configurations he demonstrated. However, his demonstrations contradict both the configurations selected by TOC and "common knowledge" of robot safety that a fully straightened robot on a mobile base is not a great idea, as it can directly transfer force from the movement of the robot's base. With a straight arm, manipulability is low, and we aimed to minimize base movement, not use it as a replacement for arm movement. In this case, the rationale for Henry's actions was that base movements were much faster than arm movements on our system for teleoperation of the PR2. This feedback led us to include in our autonomous functions movement of the arms to the task area to simplify and speed up the task for the user.

5.1.5 Robot Deployment Recommendations

Success rates and robustness are important for building user trust

In the years spent developing the assistive robotic system presented in Chapter 3, we experimented with many different modules and methods for providing the user experience we were ultimately able to provide. Some of the modules of the system worked most of the time, say with a 90% success rate. When testing that module alone it seemed like a manageable success rate. However, when combined in a system with 5 other modules each with a 90% success rate, we had a system that often failed. Along that process we learned that the success rate for individual modules in a complex system must be very high.

When testing these early systems with Henry Evans we learned the importance of system robustness in maintaining the trust of users. Users often want the system to "just work." A few system failures can quickly erode that trust and lead to a negative view of the system. When deploying a robot, the user should be considered an important stakeholder and management of their trust should be a priority.

Test in the real world for real-world systems

When preparing a system for use in an environment, it is important to test in that same environment. We discovered on many occasions that parts of our system that had been extensively tested in our lab failed to work when deployed in Henry Evans' home in California. Lighting, floors, walls, room dimensions, environmental obstacles, etc, can all cause systems to fail. For example, the tests of the assistive robotic system took place in Henry Evans' bedroom. In early tests of the system, our method for autonomously driving the PR2 to a location would collide the robot's arm with the wall, because of the limited space in the room.

5.2 Future Work

The work we have presented in this dissertation suggests numerous opportunities for further research and development. We explore and provide recommendations for future work.

5.2.1 General Assistive Robotics

In continuing research on general assistive robotics (in contrast to robot-assisted dressing), we recommend some areas for future investigation.

Continued Development of TOC

In our work on TOC we identified potential future avenues of research. In its current state, TOC does not use a distance metric in its objective function to avoid configurations where the robot is very close to collision with the person or environment. In selecting the base pose of a robot, we observed there is a trade-off between how close the robot gets to the person receiving assistance. In general, the closer the robot is placed, the higher the TC-manipulability score. However, the closer the robot is placed, the higher the likelihood that the robot will collide with the person or environment when reaching poses around the person. We noted in our discussion of related work that others have used the minimum distance between the robot and obstacles as an additional term for evaluating robot dexterity. To improve the speed of TOC's computation, we instead used expanded models of the person and environment. Inclusion of a distance-to-collision term in TOC's objective function might improve its performance.

Continued Development of Assistive Robotic System

There remain many potential improvements to the assistive robotic system. We recommend user-centered design with iterative improvement and frequent feedback from users as an approach to this area. Much of the user experience remains overly complicated, with few autonomous functions to simply task execution. Additional autonomous functions, including some utility functions would be valuable. Two possible utility functions would be picking up or swapping tools and driving away to an out-of-the-way location to recharge the robot's battery.

Assistive Robotic Devices

The robotic bed, Autobed, could be improved in many ways. In its form used in the assistive robotic system, it used a Hokuyo scanning laser rangefinder to measure the height of the bed, and an accelerometer to measure the angle of the head rest. Both sensors were attached to the bed using velcro. Design of the bed's sensing capabilities could be improved to be cheaper, better mounted, and more accurate. Additionally, the method we used for estimating the pose of the person in bed was very simple, using the center-of-mass in the pressure image to fit a 3D human model. The level of accuracy of our pose estimation limited the ability for the PR2 to autonomously perform tasks. With sufficiently accurate pose estimation, autonomous task execution may be feasible.

The benefits we found from using a robotic bed in collaboration with a mobile manipulator suggest the potential for developing additional assistive robotic devices. A robotic wheelchair could provide similar capabilities to a robotic bed, but for users in wheelchairs. A robotic lamp might provide illumination for improved perception in poorly lit areas. A robotic nightstand might be able to move to provide better access for a mobile manipulator. A second mobile or fixed manipulator could also provide valuable capabilities to a collaborative robotic system.

Application to Other Robots

The design of TOC and the robotic system for bedside assistance are such that other robots, other robotic beds, and other environment can be used in place of those we investigated. The requirements to make such changes are that the models used be swapped. For example, if a different robot model were loaded into OPENRave in place of the PR2, TOC could be used to select configurations for that new robot.

5.2.2 Robot-Assisted Dressing

The field of robot-assisted dressing remains largely open, but we can recommend a few notable areas for future investigation based on our work.

Extensions from TOORAD

TOORAD limits the space of actions of the person and the robot to achieve tractability. We take the approach of having the person move to and hold a configuration of their arm for each subtask of the dressing task as the robot moves the garment onto the person's body. We additionally constrained the garment's trajectories to linear paths, one for each major limb segment (e.g. forearm, upper arm). TOORAD could easily be extended by altering the space of possible actions. Some participants expressed that they would prefer the robot hold the sleeve of the gown by their arms so they could put their arms through.

Participants also highlighted several other garments and dressing tasks with which they would particularly like assistance. Those include T-shirts, jackets, and closing buttons. Application of TOORAD to other garments might be a valuable extension of the work.

Integration of Methods

TOORAD is an offline process for helping select actions for the robot and person. Other researchers have presented methods for improving trajectories over several attempts by the robot. Such methods could be integrated with TOORAD to improve the trajectories over time.

Execution-side Methods

We observed that small discrepancies between simulation/expectation and reality could result in failure of the dressing task. This can be due to human error (it can be very challenging for a person to hold a pose with high precision), involuntary movements from a disability, or other reasons. We used capacitive sensing to allow the robot to adapt to differences in the pose of the person's arm along one axis. We had previously attempted force-based feedback control and found that using only force, the sleeve would often get caught on the arm. We recommend additional investigations into control and sensing methods for allowing the robot to adapt better to the unexpected during dressing.

Continued System Development

There are many areas of potential improvement with the robot-assisted dressing system we presented to test TOORAD. The system required experimenter intervention between each subtask. A new tool that allows the robot to grasp the garment without experimenter intervention would be an important step towards a system that can be used independently. Additionally, the system would benefit from methods for accurately perceiving the user and the environment and from a method for autonomously moving the robot into position for each subtask.

5.2.3 Commercial Product Development

There are no current commercially available mobile manipulators of a reasonable price for an average person. We hope that the work presented in this dissertation suggests potential uses and markets for such devices. We have shown the feasibility of using mobile manipulators for providing assistance to users with disabilities with assistive tasks that involve moving a tool around the user's body as well as with dressing tasks. There is great potential for development of commercial products that perform a similar function.

5.3 Final Thoughts

Assistive mobile manipulators have the potential to increase independence and quality of life for people with disabilities. The robot's assistance could enable users to perform tasks which would otherwise be difficult or impossible. Through this dissertation we have explored methods to realize that assistance.

Although many groups have looked at how a robot could execute some specific tasks, few have considered where to place the robot to better provide assistance. We have observed that this problem arises frequently in real-world settings and solving the problem can be challenging, even for an expert user. We sought to answer the question "How should a robot choose a configuration of its base to be better able to provide assistance?" In answering this question we expanded the problem to better fit common scenarios in assistive robotics, where tasks are complicated, involve a person with disabilities, and the person is in an real-world environment that must be considered. We presented task-centric optimization of robot configurations (TOC), a method for addressing this question. We demonstrated how TOC can select one or more robot configurations for many assistive tasks that involve the robot moving a tool around a person's body. We additionally provided evidence that TOC outperforms baseline methods from literature. We demonstrated an assistive robotic system with a robotic bed and a mobile manipulator that used TOC to allow the two robot to autonomously collaborate to better provide assistance. We tested this system with a person with severe quadriplegia in his home, providing evidence of the feasibility of TOC and the robotic system for providing real assistance to real people. Our results also suggest that: (1) a robotic bed and mobile manipulator can work collaboratively to provide effective personal assistance, and (2) the combination of the two robots is beneficial.

Through our work on assistive robotics, we recognized that an important ADL, dressing, both was under-explored and contains special challenges not fully addressed by our previous methods. We presented task optimization of robot assisted dressing (TOORAD), a method for selecting actions for both the robot and a person that will result in successful dressing. We demonstrated the efficacy of TOORAD in a study with participants with disabilities receiving dressing assistance from a mobile manipulator. In that study, we also administered surveys on habits, needs, capabilities, and views on robot-assisted dressing that we expect will provide guidance to future research. Although much work remains to develop robot-assisted dressing methods and systems to the same level effectiveness as my human caregivers, TOORAD represents an important step along that path.

Together, the contents of this dissertation constitute a step towards our longer-term goal for assistive robots. We would like to enable robots to provide effective assistance with any task to people with a wide range of needs and capabilities.

Appendices

APPENDIX A
CODE, VIDEO, AND DATA RELEASE

A.1 Code

The code for the TOC, the robotic system for bedside assistance, and TOORAD has been made available through following public repositories:

- **TOC**:
  - `https://github.com/gt-ros-pkg/hrl-assistive/tree/ari_dart_devel/hrl_base_selection`
- **Robotic System for Bedside Assistance**:
  - `https://github.com/gt-ros-pkg/hrl-assistive/tree/integrated_system_evaluation/hrl_base_selection`
  - `https://github.com/gt-ros-pkg/hrl-assistive/tree/integrated_system_evaluation/assistive_teleop`
  - `https://github.com/gt-ros-pkg/hrl-assistive/tree/integrated_system_evaluation/hrl_task_planning`
- **TOORAD**:
  - `https://github.com/gt-ros-pkg/hrl-assistive/tree/ari_dart_devel/hrl_dressing`
  - Code that runs PhysX is currently held privately, but will be released on request.

A.2 Video

Videos for the execution monitoring and the robot-assisted feeding are available at:

- A System for Bedside Assistance that Integrates a Robotic Bed and a Mobile Manipulator: `https://youtu.be/9yg-NpY1HRE`

- TOORAD: Robot-Assisted Dressing: `https://youtu.be/BJSPlucGwmE`

A.3 Data

The training and testing data for estimating the pose of a person using a pressure mat has been also made available through following file server, `ftp://ftp-hrl.bme.gatech.edu`.

REFERENCES

[1] Institute of Medicine, *Retooling for an aging america: Building the health care workforce*. The National Academies Press, 2008.

[2] H. J. Goodin, "The nursing shortage in the United States of America: An integrative review of the literature," *J Adv Nurs*, 2003.

[3] J. M. Wiener, R. J. Hanley, R. Clark, and J. F. V. Nostrand, "Measuring the activities of daily living: Comparisons across national surveys," *Journal of Gerontology: SOCIAL SCIENCES*, vol. 45, no. 6, S229–237, 1990.

[4] C. K. Andersen, K. U. Wittrup-Jensen, A. Lolk, K. Andersen, and P. Kragh-Sørensen, "Ability to perform activities of daily living is the main factor affecting quality of life in patients with dementia," *Health and quality of life outcomes*, vol. 2, no. 1, p. 52, 2004.

[5] M Vest, T Murphy, K Araujo, M Pisani, *et al.*, "Disability in activities of daily living, depression, and quality of life among older medical ICU survivors: A prospective cohort study," 2011.

[6] T. L. Mitzner, T. L. Chen, C. C. Kemp, and W. A. Rogers, "Identifying the potential for robotics to assist older adults in different living environments," *International Journal of Social Robotics*, pp. 1–15, 2013.

[7] (2015). Canadian Partnership for Stroke Recovery. "Stroke Engine". Accessed: 2018-01-01, (visited on 05/27/2016).

[8] (2017). The Wright Stuff, Inc. "The Wright Stuff healthcare products that make life easier". Accessed: 2018-01-01.

[9] M. Topping and J. Smith, "The development of Handy 1, a rehabilitation robotic system to assist the severely disabled," *Industrial Robot*, vol. 25, no. 5, pp. 316–20, 1998.

[10] M. Topping, "The development of handy 1. a robotic system to assist the severely disabled.," in *Int. Conf. Rehab. Robo.*, 1999, pp. 244–249.

[11] (). Robot arms - Kinova, (visited on 06/02/2015).

[12] Y. Gao, H. J. Chang, and Y. Demiris, "User modelling for personalised dressing assistance by humanoid robots," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 1840–1845.

[13] V. Maheu, P. S. Archambault, J. Frappier, and F. Routhier, "Evaluation of the JACO robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities," in *Rehabilitation Robotics (ICORR), IEEE International Conference on*, IEEE, 2011, pp. 1–5.

[14] S. Schröer, I. Killmann, B. Frank, M. Völker, L. Fiederer, T. Ball, and W. Burgard, "An autonomous robotic assistant for drinking," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, 2015, pp. 6482–6487.

[15] T. L. Chen, M. Ciocarlie, S. Cousins, P. M. Grice, K. Hawkins, *et al.*, "Robots for humanity: A case study in assistive mobile manipulation," 2013.

[16] K. P. Hawkins, P. M. Grice, T. L. Chen, C.-H. King, and C. C. Kemp, "Assistive mobile manipulation for self-care tasks around the head," in *Computational Intelligence in Robotic Rehabilitation and Assistive Technologies (CIR2AT), 2014 IEEE Symposium on*, IEEE, 2014.

[17] S. Schrer, I. Killmann, B. Frank, M. Vlker, L. Fiederer, T. Ball, and W. Burgard, "An autonomous robotic assistant for drinking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6482–6487.

[18] C. Copilusi, M. Kaur, and M. Ceccarelli, "Lab experiences with larm clutched arm for assisting disabled people," in *New Trends in Mechanism and Machine Science: From Fundamentals to Industrial Applications*, P. Flores and F. Viadero, Eds. Cham: Springer International Publishing, 2015, pp. 603–611, ISBN: 978-3-319-09411-3.

[19] Y. Takahashi and S. Suzukawa, "Easy human interface for severely handicapped persons and application for eating assist robot," in *2006 IEEE International Conference on Mechatronics*, Jul. 2006, pp. 225–229.

[20] D. Park, Y. K. Kim, Z. Erickson, and C. C. Kemp, "Towards assistive feeding with a general-purpose mobile manipulator," in *IEEE International Conference on Robotics and Automation - workshop on Human-Robot Interfaces for Enhanced Physical Interactions*, 2016.

[21] J. Silvrio, L. Rozo, S. Calinon, and D. G. Caldwell, "Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sep. 2015, pp. 464–470.

[22] D. Leidner, A. Dietrich, M. Beetz, and A. Albu-Schäffer, "Knowledge-enabled parameterization of whole-body control strategies for compliant service robots," *Autonomous Robots*, vol. 40, no. 3, pp. 519–536, 2016.

[23] A. Kapusta and C. C. Kemp, "Task-centric optimization of configurations for assistive robots," *ArXiv preprint arXiv:1804.07328*, 2018.

[24] A. Kapusta, D. Park, and C. C. Kemp, "Task-centric selection of robot and environment initial configurations for assistive tasks," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 1480–1487.

[25] A. Kapusta and C. C. Kemp, "Optimization of robot configurations for assistive tasks," in *Proceedings of Robotics: Science and Systems (RSS 2016) Workshop on Planning for Human-Robot Interaction: Shared Autonomy and Collaborative Robotics*, 2016.

[26] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: Representing robot capabilities," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Ieee, 2007, pp. 3229–3236.

[27] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley New York, 2006, vol. 3.

[28] L. Stocco, S. Salcudean, and F. Sassani, "Matrix normalization for optimal robot design," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, IEEE, vol. 2, 1998, pp. 1346–1351.

[29] F. Hammond and K Shimada, "Multi-objective weighted isotropy measures for the morphological design optimisation of kinematically redundant manipulators," *International Journal of Mechatronics and Manufacturing Systems*, vol. 4, no. 2, pp. 150–170, 2011.

[30] T. Yoshikawa, "Analysis and control of robot manipulators with redundancy," in *Robotics research: The first international symposium*, Mit Press Cambridge, MA, USA, 1984, pp. 735–747.

[31] C. A. Klein and B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 72–83, 1987.

[32] J.-O. Kim and K Khosla, "Dexterity measures for design and control of manipulators," in *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, IEEE, 1991, pp. 758–763.

[33] M.-J. Tsai, “Workspace geometric characterization and manipulability of industrial robots,” PhD thesis, The Ohio State University, 1986.

[34] T. F. Chan and R. V. Dubey, “A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 286–292, 1995.

[35] J. Lee, “A study on the manipulability measures for robot manipulators,” in *Intelligent Robots and Systems, 1997. IROS’97., Proceedings of the 1997 IEEE/RSJ International Conference on*, IEEE, vol. 3, 1997, pp. 1458–1465.

[36] F. L. Hammond III and K. Shimada, “Improvement of kinematically redundant manipulator design and placement using torque-weighted isotropy measures,” in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, IEEE, 2009, pp. 1–8.

[37] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, “Manipulability analysis,” in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, IEEE, 2012, pp. 568–573.

[38] R. Diankov, “Automated construction of robotic manipulation programs,” 2010.

[39] P. Beeson and B. Ames, “Trac-ik: An open-source library for improved solving of generic inverse kinematics,” in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, IEEE, 2015, pp. 928–935.

[40] S. Kumar, N Sukavanam, and R Balasubramanian, “An optimization approach to solve the inverse kinematics of redundant manipulator,” *International Journal of Information and System Sciences (Institute for Scientific Computing and Information)*, vol. 6, no. 4, pp. 414–423, 2010.

[41] R. Smits. (2006). KDL: Kinematics and Dynamics Library. Accessed: 2018-01-01.

[42] M. Gienger, H. Janssen, and C. Goerick, “Task-oriented whole body motion for humanoid robots,” in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, IEEE, 2005, pp. 238–244.

[43] M. X. Grey, C. R. Garrett, C. K. Liu, A. D. Ames, and A. L. Thomaz, “Humanoid manipulation planning using backward-forward search,” in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, IEEE, 2016, pp. 5467–5473.

[44] M. Elbanhawi and M. Simic, “Sampling-based robot motion planning: A review,” *IEEE Access*, vol. 2, pp. 56–77, 2014.

[45] M. Stilman and J. J. Kuffner, “Navigation among movable obstacles: Real-time reasoning in complex environments,” *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 479–503, 2005.

[46] S. R. Lindemann and S. M. LaValle, “Current issues in sampling-based motion planning,” *Robotics Research*, pp. 36–54, 2005.

[47] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, “Backward-forward search for manipulation planning,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 6366–6373.

[48] R. Diankov, N. Ratliff, D. Ferguson, S. Srinivasa, and J. Kuffner, “Bispace planning: Concurrent multi-space exploration,” *Proceedings of Robotics: Science and Systems IV*, vol. 63, 2008.

[49] F. Zacharias, W. Sepp, C. Borst, and G. Hirzinger, “Using a model of the reachable workspace to position mobile manipulators for 3-d trajectories,” in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, IEEE, 2009, pp. 55–61.

[50] O. Porges, T. Stouraitis, C. Borst, and M. A. Roa, “Reachability and capability analysis for manipulation tasks,” in *ROBOT2013: First Iberian Robotics Conference*, Springer, 2014, pp. 703–718.

[51] D. Leidner, A. Dietrich, F. Schmidt, C. Borst, and A. Albu-Schaffer, “Object-centered hybrid reasoning for whole-body mobile manipulation,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, 2014, pp. 1828–1835.

[52] J. Dong and J. C. Trinkle, “Orientation-based reachability map for robot base placement,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 1488–1493.

[53] R. Diankov and J. Kuffner, “Openrave: A planning architecture for autonomous robotics,” *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34*, vol. 79, 2008.

[54] F. Burget and M. Bennewitz, “Stance selection for humanoid grasping tasks by inverse reachability maps,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, 2015, pp. 5669–5674.

[55] N. Vahrenkamp, T. Asfour, and R. Dillmann, “Robot placement based on reachability inversion,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, IEEE, 2013, pp. 1970–1975.

[56] D. Hsu, J.-C. Latcombe, and S. Sorkin, "Placing a robot manipulator amid obstacles for optimized execution," in *Assembly and Task Planning, 1999.(ISATP'99) Proceedings of the 1999 IEEE International Symposium on*, IEEE, 1999, pp. 280–285.

[57] F. Stulp, A. Fedrizzi, and M. Beetz, "Learning and performing place-based mobile manipulation," in *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*, IEEE, 2009, pp. 1–7.

[58] M. L. Walters, M. A. Oskoei, D. S. Syrdal, and K. Dautenhahn, "A long-term human-robot proxemic study," in *RO-MAN, 2011 IEEE*, IEEE, 2011, pp. 137–142.

[59] M. L. Walters, K. Dautenhahn, R. Te Boekhorst, K. L. Koay, D. S. Syrdal, and C. L. Nehaniv, "An empirical framework for human-robot proxemics," *Procs of New Frontiers in Human-Robot Interaction*, 2009.

[60] J. Mumm and B. Mutlu, "Human-robot proxemics: Physical and psychological distancing in human-robot interaction," in *Proceedings of the 6th international conference on Human-robot interaction*, ACM, 2011, pp. 331–338.

[61] L. Takayama and C. Pantofaru, "Influences on proxemic behaviors in human-robot interaction," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, IEEE, 2009, pp. 5495–5502.

[62] M. L. Walters, K. Dautenhahn, R. Te Boekhorst, K. L. Koay, C. Kaouri, S. Woods, C. Nehaniv, D. Lee, and I. Werry, "The influence of subjects personality traits on personal spatial zones in a human-robot interaction experiment," in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, IEEE, 2005, pp. 347–352.

[63] E. A. Sisbot, L. F. Marin-Urias, X. Broquere, D. Sidobre, and R. Alami, "Synthesizing robot motions adapted to human presence," *International Journal of Social Robotics*, vol. 2, no. 3, pp. 329–343, 2010.

[64] J. Mainprice, E. A. Sisbot, L. Jaillet, J. Cortes, R. Alami, and T. Simeon, "Planning human-aware motions using a sampling-based costmap planner," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 5012–5017.

[65] E. A. Sisbot, L. F. Marin, R. Alami, and T. Simeon, "A mobile robot that performs human acceptable motions," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, IEEE, 2006, pp. 1811–1816.

[66] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, “A human aware mobile robot motion planner,” *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 874–883, 2007.

[67] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, “Human-aware robot navigation: A survey,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[68] P. Dario, E. Guglielmelli, C. Laschi, and G. Teti, “Movaid: A personal robot in everyday life of disabled and elderly people,” *Technology and Disability*, vol. 10, no. 2, pp. 77–93, 1999.

[69] C Schaeffer and T May, “Care-o-bot-a system for assisting elderly or disabled persons in home environments,” *Assistive technology on the threshold of the new millenium*, 1999.

[70] B. Graf, U. Reiser, M. Hägele, K. Mauz, and P. Klein, “Robotic home assistant care-o-bot® 3-product vision and innovation platform,” in *Advanced Robotics and its Social Impacts (ARSO), 2009 IEEE Workshop on*, IEEE, 2009, pp. 139–144.

[71] Z. Bien, M.-J. Chung, P.-H. Chang, D.-S. Kwon, D.-J. Kim, J.-S. Han, J.-H. Kim, D.-H. Kim, H.-S. Park, S.-H. Kang, *et al.*, “Integration of a rehabilitation robotic system (kares ii) with human-friendly man-machine interaction units,” *Autonomous robots*, vol. 16, no. 2, pp. 165–191, 2004.

[72] A. Jain and C. C. Kemp, “El-e: An assistive mobile manipulator that autonomously fetches objects from flat surfaces,” *Autonomous Robots*, vol. 28, no. 1, pp. 45–64, 2010.

[73] P. Grice and C. C. Kemp, “Assistive mobile manipulation: Designing for operators with motor impairments,” in *Proceedings of Robotics: Science and Systems (RSS 2016) Workshop on Socially and Physically Assistive Robotics for Humanity,*, 2016.

[74] P. Grice, Y. Chitalia, M. Rich, *et al.*, “Autobed: Open hardware for accessible web-based control of an electric bed,” RESNA, 2016.

[75] A. Kapusta, Y. Chitalia, D. Park, and C. C. Kemp, “Collaboration between a robotic bed and a mobile manipulator may improve physical assistance for people with disabilities,” 2016.

[76] A. R. Tilley, *The measure of man and woman: Human factors in design*. New York: Wiley, 2002.

[77] J. Tan, Y. Gu, G. Turk, and C. K. Liu, “Articulated swimming creatures,” in *ACM Transactions on Graphics (TOG)*, ACM, vol. 30, 2011, p. 58.

[78] P. M. Grice, M. D. Killpack, A. Jain, S. Vaish, J. Hawke, and C. C. Kemp, "Whole-arm tactile sensing for beneficial and acceptable contact during robotic assistance," in *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, IEEE, 2013, pp. 1–8.

[79] M. D. Killpack, A. Kapusta, and C. C. Kemp, "Model predictive control for fast reaching in clutter," *Autonomous Robots*, vol. 40, no. 3, pp. 537–560, 2016.

[80] H. M. Van der Loos, J. J. Wagner, N. Smaby, K Chang, O. Madrigal, *et al.*, "Provar assistive robot system architecture," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, IEEE, vol. 1, 1999, pp. 741–746.

[81] M. Topping and J. Smith, "The development of handy 1, a rehabilitation robotic system to assist the severely disabled," *Industrial Robot: An International Journal*, vol. 25, no. 5, pp. 316–320, 1998.

[82] V. Kumar, T. Rahman, and V. Krovi, "Assistive devices for motor disabilities," *Wiley Encyc. of Electrical and Electronics Eng.*, 1997.

[83] S. W. Brose, D. J. Weber, B. A. Salatin, G. G. Grindle, H. Wang, J. J. Vazquez, and R. A. Cooper, "The role of assistive robotics in the lives of persons with disability," *American Journal of Physical Medicine & Rehabilitation*, vol. 89, no. 6, pp. 509–521, 2010.

[84] M. Ciocarlie, K. Hsiao, A. Leeper, and D. Gossow, "Mobile manipulation through an assistive home robot," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, 2012, pp. 5313–5320.

[85] H. Nguyen, C. Anderson, A. Trevor, A. Jain, Z. Xu, and C. C. Kemp, "El-e: An assistive robot that fetches objects from flat surfaces," in *Robotic helpers, int. conf. on human-robot interaction*, 2008.

[86] A Kargov, T Asfour, C Pylatiuk, R Oberle, H Klosek, S Schulz, K Regenstein, G Bretthauer, and R Dillmann, "Development of an anthropomorphic hand for a mobile assistive robot," in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, IEEE, 2005, pp. 182–186.

[87] A. R. Mosteo and L. Montano, "A survey of multi-robot task allocation," *Instituto de Investigación en Ingenierıa de Aragón, University of Zaragoza, Technical Report No. AMI-009-10-TEC*, 2010.

[88] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.

[89] L. E. Parker, “Multiple mobile robot systems,” in *Springer Handbook of Robotics*, Springer, 2008, pp. 921–941.

[90] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, “Market-based multirobot coordination: A survey and analysis,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[91] (). Intuitive Surgical. “Intuitive Surgical and Hill-Rom Announce U.S. Clearance for Integrated Table Motion”, (visited on 06/06/2016).

[92] (). First Dayton Cyberknife. “Cyberknife: Beat Cancer Without Surgery”, (visited on 06/06/2016).

[93] B. Roy, A. Basmajian, and H. H. Asada, “Maneuvering a bed sheet for repositioning a bedridden patient,” in *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on*, IEEE, vol. 2, 2003, pp. 2224–2229.

[94] H. M. Van Der Loos, N. Ullrich, and H. Kobayashi, “Development of sensate and robotic bed technologies for vital signs monitoring and sleep quality improvement,” *Autonomous Robots 15.1*, 2003.

[95] S.-W. Peng, F.-L. Lian, and L.-C. Fu, “Mechanism design and mechatronic control of a multifunctional test bed for bedridden healthcare,” *Mechatronics, IEEE/ASME Transactions on*, vol. 15, no. 2, pp. 234–241, 2010.

[96] R. Portillo-Velez, E. Vzquez-Santacruz, C. Morales-Cruz, and M. Gamboa-Ziga, “Mechatronic design and manufacturing of an affordable healthcare robotic bed,” *Journal of Rehabilitation and Assistive Technologies Engineering*, vol. 3, pp. 1–13, 2016.

[97] T. Harada, T. Sato, and T. Mori, “Pressure distribution image based human motion tracking system using skeleton and surface integration model,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, IEEE, vol. 4, 2001, pp. 3201–3207.

[98] R. Grimm, J. Sukkau, J. Hornegger, and G. Greiner, “Automatic patient pose estimation using pressure sensing mattresses,” in *Bildverarbeitung für die Medizin 2011*, Springer, 2011, pp. 409–413.

[99] J. J. Liu, M.-C. Huang, W. Xu, and M. Sarrafzadeh, “Bodypart localization for pressure ulcer prevention,” in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, IEEE, 2014, pp. 766–769.

[100] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins, “Pddl-the planning domain definition language,” 1998.

[101] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[102] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw, “User acceptance of computer technology: A comparison of two theoretical models,” *Management science*, vol. 35, no. 8, pp. 982–1003, 1989.

[103] Y Gao, H. Chang, and Y Demiris, “Iterative path optimisation for personalised dressing assistance using vision and force information,” IEEE, 2016.

[104] T. Tamei, T. Matsubara, A. Rai, and T. Shibata, “Reinforcement learning of clothing assistance with a dual-arm robot,” in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, IEEE, 2011, pp. 733–738.

[105] N. Koganti, T. Tamei, K. Ikeda, and T. Shibata, “Bayesian nonparametric motor-skill representations for efficient learning of robotic clothing assistance,” in *Workshop on Practical Bayesian Nonparametrics, Neural Information Processing Systems, 2016*, 2016, pp. 1–5.

[106] L. Twardon and H. J. Ritter, “Learning to put on a knit cap in a head-centric policy space,” *IEEE Robotics and Automation Letters*, 2018.

[107] A. Jevtić, A. F. Valle, G. Alenyà, G. Chance, P. Caleb-Solly, S. Dogramadzi, and C. Torras, “Personalized robot assistant for support in dressing,” *IEEE Transactions on Cognitive and Developmental Systems*, 2018.

[108] S. D. Klee, B. Q. Ferreira, R. Silva, J. P. Costeira, F. S. Melo, and M. Veloso, “Personalized assistance for dressing users,” in *Social Robotics*, Springer, 2015, pp. 359–369.

[109] E. Pignat and S. Calinon, “Learning adaptive dressing assistance from human demonstration,” *Robotics and Autonomous Systems*, vol. 93, pp. 61–75, 2017.

[110] G. Chance, A. Jevtić, P. Caleb-Solly, and S. Dogramadzi, “A quantitative analysis of dressing dynamics for robotic dressing assistance,” *Frontiers in Robotics and AI*, vol. 4, 2017.

[111] F. Zhang, A. Cully, and Y. Demiris, “Personalized robot-assisted dressing using user modeling in latent spaces,” in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, IEEE, 2017, pp. 3603–3610.

[112] S. Ha and C. K. Liu, “Multiple contact planning for minimizing damage of humanoid falls,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 2015, pp. 2761–2767.

[113] Y. Bai and C. K. Liu, “Dexterous manipulation using both palm and fingers,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1560–1565.

[114] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.

[115] A. Haidu, D. Kohlsdorf, and M. Beetz, “Learning task outcome prediction for robot control from interactive environments,” in *Int. Conf. on Intelligent Robots and Systems (IROS)*, Chicago, USA, 2014.

[116] ——, “Learning action failure models from interactive physics-based simulations,” in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015.

[117] W. Yu, A. Kapusta, J. Tan, C. C. Kemp, G. Turk, and C. K. Liu, “Haptic simulation for robot-assisted dressing,”

[118] N. Fahantidis, K Paraschidis, V. Petridis, Z. Doulgeri, L. Petrou, and G. Hasapis, “Robot handling of flat textile materials,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 34–41, 1997.

[119] F. Osawa, H. Seki, and Y. Kamiya, “Unfolding of massive laundry and classification types by dual manipulator.,” *JACIII*, vol. 11, no. 5, pp. 457–463, 2007.

[120] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, “Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, 2010, pp. 2308–2315.

[121] S. Miller, J. Van Den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, “A geometric approach to robotic laundry folding,” *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 249–267, 2012.

[122] C. Bersch, B. Pitzer, and S. Kammel, “Bimanual robotic cloth manipulation for laundry folding,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, IEEE, 2011, pp. 1413–1419.

[123] D. Berenson, “Manipulation of deformable objects without modeling and simulating deformation,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, 2013, pp. 4525–4532.

[124] CMA. (2018). The CMA evolution strategy. Accessed: 2018-01-01.

[125] N. Hansen, “The cma evolution strategy: A comparing review,” in *Towards a new evolutionary computation*, Springer, 2006, pp. 75–102.

[126] OpenRAVE. (2018). Open robotics automation virtual environment (OpenRAVE). Accessed: 2018-01-01.

[127] PhysX. (2011). PhysX physics engine. Accessed: 2018-01-01.

[128] DART. (2018). Dynamic animation and robotics toolkit (DART). Accessed: 2018-01-01.

[129] E. Coumans and Y. Bai, *PyBullet, a python module for physics simulation for games, robotics and machine learning*, `http://pybullet.org`, 2016–2018.

[130] A. Kapusta, W. Yu, T. Bhattacharjee, C. K. Liu, G. Turk, and C. C. Kemp, “Data-driven haptic perception for robot-assisted dressing,” in *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, IEEE, 2016, pp. 451–458.

[131] Z. Erickson, M. Collier, A. Kapusta, and C. C. Kemp, “Tracking human pose during robot-assisted dressing using single-axis capacitive proximity sensing,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2245–2252, 2018.