

Framework For Robot-Assisted Doffing of Personal Protective Equipment

by

Antonio Rafael V. Umali

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

by

August 2016

APPROVED:

Professor Dmitry Berenson (Mike Gennert), Major Thesis Advisor

Professor Craig Wills, Head of Department

Abstract

When treating highly-infectious diseases such as Ebola, health workers are at high risk of infection during the doffing of Personal Protective Equipment (PPE). This is due to factors such as fatigue, hastiness, and inconsistency in training. The introduction of a semi-autonomous robot doffing assistant has the potential to increase the safety of the doffing procedure by assisting the human during high-risk sub-tasks. The addition of a robot into the procedure introduces the need to transform a purely human task into a sequence of safe and effective human-robot collaborative actions. We take advantage of the fact that the human can do the more intricate motions during the procedure. Since diseases like Ebola can spread through the mucous membranes of the eyes, ears, nose, and mouth our goal is to keep the human's hands away from his or her face as much as possible. Thus our framework focuses on using the robot to help avoid such human risky motion. As secondary goals, we seek to also minimize the human's effort and make the robot's motion intuitive for the human. To address different versions and variants of PPE, we propose a way of segmenting the doffing procedure into a sequence of human and robot actions such that the robot only assists when necessary. Our framework then synthesizes assistive motions for the robot that perform parts of the tasks according to the metrics above. Our experiments on five doffing tasks suggest that the introduction of a robot assistant improves the safety of the procedure in three out of four of the high-risk doffing tasks while reducing effort in all five tasks.

Acknowledgements

I would like to express my gratitude to my advisor, for ensuring that this project and thesis were done properly, my labmates, for helping in keeping this project going in the right direction, and all the participants of my user studies, for putting up with the heat inside the PPE suits.

Contents

1	Introduction	1
2	Related Work	4
3	Definitions	7
4	CLASSIFYING AND QUANTIFYING HUMAN MOTIONS	8
4.0.1	Classification	8
4.0.2	Assessing Human Motion Quality	9
5	FRAMEWORK OVERVIEW	12
6	TRAJECTORY OPTIMIZATION	21
6.0.1	PPE Simulation	22
6.0.2	Changepoint identification	23
6.0.3	Robot planning costs	25
6.0.4	Robot two arm planning constraint	27
6.0.5	Human planning cost functions	29
7	Results	32
7.0.1	Metrics	33
7.0.2	Experiment results	34

8	Discussion	40
9	Conclusion	42

List of Figures

1.1	Baxter helping in the removal of the apron.	3
5.1	Inputs and outputs shown in this diagram are encapsulations or abstractions of the actual ones.	13
6.1	Rigid representation of the goggles. The black rectangular prism behind the larger, bright red object is the strap of the simulated goggles retracted after being freed from contact. The red points represent its removal trajectory, while the arrows intersecting the human model head represent the rays from the ray casting done in simulation. This ray casting is used to determine when the straps will snap back to the goggles body.	22
6.2	Planning for the goggles transfer doffing in OpenRAVE with TrajOpt. The green points represent the planned trajectory.	28

6.3	2-D example of how the REBA metric partitions the human configuration space. The orange region has the highest effort score, followed by the yellow regions, and then the green region. P1 and P2 show that two points inside the cost field will have different values and thus different gradients. P4 shows how the cost function assigns a cost should there be two adjacent regions with better scores, with the dark blue arrow indicating a better score compared to the light blue arrow. The x's in the score calculations of P1, P2, and P3 show how the cost function does not consider dimensions of the points which are within the bounds of the next best neighboring region.	31
7.1	Robot-Assisted doffing using transfer motions.	37
7.2	Box plots showing the assisted vs. unassisted hands-to-face distance scores for the PPE doffing tasks. The odd-numbered columns show the data for unassisted doffing, while the even-numbered columns show the data for the robot-assisted doffing. The red horizontal lines are the means for each score, the blue represent the quadrants, and the green diamonds are outliers.	38
7.3	Box plots showing the assisted vs. unassisted effort scores for the PPE doffing tasks. The odd-numbered columns show the data for unassisted doffing, while the even-numbered columns show the data for the robot-assisted doffing. The red horizontal lines are the means for each score, the blue represent the quadrants, and the green diamonds are outliers.	39

List of Tables

7.1	Human and robot DOF for the tasks)	34
7.2	Hands-to-Face Distance Score Statistics (lower is better)	34

Chapter 1

Introduction

One of the most dangerous steps in the process of treating highly-infectious diseases like Ebola is the doffing of Personal Protective Equipment (PPE) worn by health-care workers who interact with infected patients and contaminated facilities. According to the Center for Disease Control (CDC), the removal of used PPE is “a high-risk process that requires a structured procedure, a trained observer, and a designated area for removal to ensure protection. PPE must be removed slowly and deliberately in the correct sequence to reduce the possibility of self-contamination or other exposure to Ebola virus” [1]. Because the doffing process involves numerous steps (24 steps in the most recent guidelines [1]) and must be repeated often under stressful conditions, it poses a significant risk of infection for health-care workers.

This paper presents a method that seeks to enable faster, less-risky doffing by using a human-supervised dual-arm manipulator (Baxter) to assist with the process. We propose a framework to synthesize motions for helping in the removal of certain pieces of PPE while reducing the risk of infection to the human and reducing the effort required to perform the task. The robot performs the doffing task in collaboration with the human, either by holding the PPE in a key position while the human

removes a body part from inside it (which we call *support* motion) or by removing the PPE from the human's body itself (which we call *transfer* motion). Since many infectious diseases (including Ebola) can be spread through the mucous membranes of the eyes, ears, nose, and mouth, we synthesize support and transfer motion so as to maximize the distance between the worker's hands and his or her face during the doffing process. A secondary goal is to assist with doffing in a way that is not strenuous for the human (we measure strain using the RULA [2] and REBA [3] metrics). Finally, we aim to make robot's motion as intuitive and comfortable for the human as possible while considering the above metrics. We implemented our framework and compared the performance of assisted doffing against unassisted doffing in five tasks. Our experiments showed that assisted doffing lessens the risk of infection in four out of the five tasks. Furthermore, assisted doffing reduced the effort exerted by subjects in all of the tasks.



Figure 1.1: Baxter helping in the removal of the apron.

Chapter 2

Related Work

Recent compliant robots such as Baxter have allowed robots to interact with humans in close proximity. Several researchers have explored using robots as household companions and assistants [4, 5]. However, these works did not perform collaborative manipulation with the human. [6] presented a framework based on imitation and reinforcement learning for learning to generate robot motions for collaborative manipulation tasks with humans. However, their work focused on the manipulation of rigid objects and did not have the robot and the human in very close proximity to one another, as we do in this work.

Recent work has explored the use of a robot to assist in upper-body dressing for humans with movement limitations [7]. Gaussian Mixture Models were used to model the movement of the user's upper body enabling the user to move into the clothing while the robot holds it still. [8] also addressed the problem of dressing by creating primitive actions which constitute complex motions in human dressing. Using these, they were able to create a simulation and animation of how humans put on pieces of clothing. Furthermore, they were able to identify robotics-oriented properties such as grasp points, end-effector motion, and release motions. Shino-

hara et al. proposed using reinforcement learning to learn motor skills necessary to interact with non-rigid materials [9][10]. They focused on the topological relationship between the robot’s configuration and material, simplifying the problem by assuming that fine details about the material (e.g. wrinkles) are irrelevant to perform most tasks. K. Yamazaki et al.[11] focused on developing a vision-based state-estimator for the materials and learning the mappings between robot motions and material states and were able to perform lower-body dressing on a mannequin. While these works present ideas which might prove to be beneficial to our problem, they are both highly dependent on obtaining accurate perception of the object using computer vision, which is very difficult in real-world scenarios with cloth where there are many occlusions. In contrast, our framework relies on the human’s ability to perceive the clothing thus we do not require vision data. Thus our approach must rely less on real-time feedback information from complicated automated sources. Furthermore, they address the task opposite that of our problem; dressing. Certain assumptions and ideas which maybe applicable to dressing might not translate well when applied to doffing.

Learning from demonstration[12] is an attractive method for synthesizing robot motions from human demonstrations. LfD algorithms do not require an extensive knowledge of the domain dynamics and avoids the pitfalls of model simplification. LfD is a form of supervised learning which learns control policies by either approximating the mapping from actions to states using demonstration, data, developing system models to determine a model for the world dynamics and reward functions, or learn a set of rules associating pre- and post- conditions along with a sparse dynamics model. LfD has been used in robotics for the generation of control policies in tasks such as biped walking [13] and grasping [14]. However, LfD is difficult to use in human-robot interaction tasks because the policies will change depending on

the person the robot is working with. Furthermore, it is difficult to model human reactionary behavior, thus LfD might result in policies which are too strict and do not allow for different cases of human reaction.

While sampling-based planners and search algorithms are common choices for motion planning methods, optimization-based algorithms are seeing an increase in popularity especially for problems with complex constraints [15][16]. Trajectory optimization can be used to either smooth trajectories or to plan for scratch. The latter is attractive in high degree-of-freedom problems because the planning time scales favorably with the number of DOFs [17]. While optimization only optimizes individual points and is prone to converging to local minima, we only seek to find near-optimal, feasible trajectories. We use trajectory optimization in our work because of its speed and ability to handle multiple constraints and cost functions.

Chapter 3

Definitions

A configuration of the human is defined as $q_h \in Q_h$, we use different human models with different numbers of degrees of freedom according to which PPE component is being considered. The configuration of the robot is $q_r \in Q_r$, which has rd dimensions. To address the complexity that deformable objects introduce into state definition and estimation as well as to motion planning, specifically for manipulation [18], we make the assumption that the PPE models we receive as input are composed of rigid parts. This way, we can easily define and simulate the state of the individual components of the PPE as changes in its pose and geometry. The PPE component is thus treated as a (articulated) rigid body whose pose is $T_{ppe} \in SE(3)$.

Chapter 4

CLASSIFYING AND QUANTIFYING HUMAN MOTIONS

Before discussing our framework for robot-assisted doffing, we introduce key concepts that will be used to classify and evaluate human motion in the framework. We present a classification of human motions which will reduce their inherent complexity and simplify our approach and the generation of robot motions. We also present metrics which evaluate these motion classes with respect to their overall safety. These metrics will later be used in generating a predicted human motion and in comparing the final human-robot motions to the original demonstration.

4.0.1 Classification

We classify demonstrated human motion according to the changes in the state of both the PPE and the human. Motions are divided into three classes:

1. *Transfer Motion*—where the body part, which the PPE component being

doffed is attached to, remains in a nearly-static pose throughout the motion while the PPE component undergoes changes that lead to it being detached from the person. E.g., removing the goggles from the head.

2. *Support Motion*—where the PPE component remains in a nearly-static state while the person moves the attached body part away from it. E.g., taking the foot out of the boot.
3. *Adjustment Motion*—where the PPE component, the attached body part, or both undergo a series of small rapid movements. E.g. removing the inner gloves.

We observed that doffing motions performed by humans are usually neither pure transfer or support motions; they are sequences of adjustment motions with either dominant transfer or support characteristics. This means that to preserve the human’s behavior the robot would need to move simultaneously with the human. This makes generating robot motion very difficult because we do not know how the human will react and the robot must sense and move quickly to compensate. Instead, in our framework we convert adjustment motions to transfer and support motions so that the robot and human do not need to move simultaneously.

4.0.2 Assessing Human Motion Quality

To ensure that the new sequence of actions has less risk and effort than the original one, we propose the use of several metrics which represent how much risk an action carries. The first metric is the minimum distance of the worker’s hands to the worker’s face, meant to assess the risk of contamination during a doffing task. We specifically chose the hands as the objects of interest because they perform tasks that expose them to harmful agents.

$$FD(c, f) = \min_{i,j} (d(face_{c,f}[i], hands_{c,f}[j])) \quad (4.1)$$

where FD is the function which computes the minimum hand-to-face distance for a PPE component c and frame f , d is the Euclidean distance function, $face_{c,f}$ and $hands_{c,f}$ are arrays representing the point clouds for their respective body parts at an observed frame f for PPE component c . We then compute the sum of all the positive differences between these distances above a threshold ϵ_d across all frames. We then filter out distances from actions which can be deemed low-risk and increase the penalty for actions which can be considered high-risk (with respect to infection probability) using a quantity we call the *Hands-to-Face Distance Score*:

$$\lambda(c) = \sum_f \max(\epsilon_d - FD(c, f), 0) \quad (4.2)$$

where λ is the function which calculates the *Hands-to-Face Distance Score*. Lower values of this score signify better infection safety. To measure the effort exerted by a subject, we function E in 4.0.2 as our scoring function.

We also aim to reduce the overall effort exerted by the worker during the doffing procedure to reduce the risk of accidental exposure, due to rushing or lack of strength, and dangers due to heat and dehydration. For this we use the Rapid Upper Limb Assessment (RULA)[2] and Rapid Entire Body Assessment (REBA)[3], which are methods used to quickly evaluate the effort of a given human configuration. The problem with these two metrics is that simply summing posture scores will bias strain toward the longer action when the difference in execution times is large enough. To address this we use the following effort scoring function:

$$E(c, p) = \sum_p \max(R(p_i) - \epsilon_e, 0) * t_i, \quad (4.3)$$

where p_i is a posture being held by the subject, ϵ_e is an effort score which represents the minimum effort we want to be considered as significant, and t_i is the amount of time p was held. This scoring function allows us to filter out postures which can be considered negligible effort using ϵ_e . Lower values of this score signify that less significant effort was exerted during the task.

Chapter 5

FRAMEWORK OVERVIEW

It is difficult to plan and execute robot motions for human-robot collaboration tasks where the robot and human are moving simultaneously because human motion is difficult to predict. Furthermore, in the doffing tasks, the robot and human are also manipulating a shared object with complex constraints due to its deformability and its relationship with the human. Due to the difficulty of having the robot and human move simultaneously, we have created a framework that decomposes a doffing task into a sequence of human or robot actions and then plan robot motions which best assist the human. The framework (see Figure 5.1) accepts as input the task demonstration represented by the human trajectory $\tau_h \subset Q_h$ and PPE component trajectories in $\tau_{ppe} \subset SE(3)$. Let n be the length of the demonstration.

$$input_{human} = [q_{h_0}, q_{h_1}, \dots, q_{h_n}] \quad (5.1)$$

$$input_{PPE} = [T_{ppe_0}, T_{ppe_1}, \dots, T_{ppe_n}] \quad (5.2)$$

Where n is the length of the demonstration and T_{ppe_i} is the transformation matrix representing the pose of the PPE with respect to the world frame. The output of

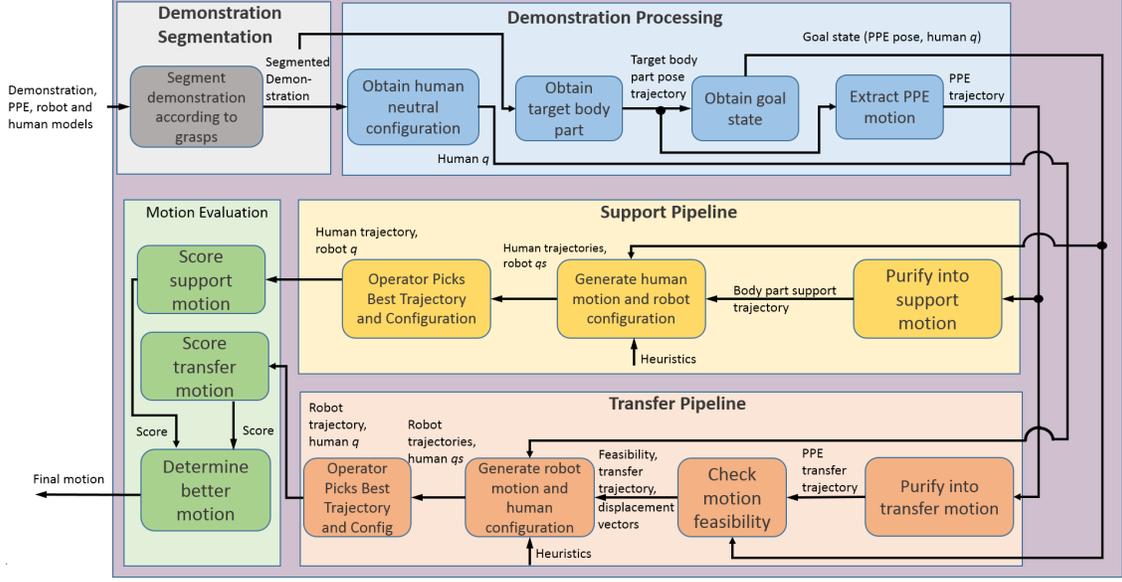


Figure 5.1: Inputs and outputs shown in this diagram are encapsulations or abstractions of the actual ones.

the framework is a sequence of tuples, each containing two trajectories; one for the expected human motion and the other for the robot.

$$output_i = (output_{human_i}, output_{robot_i}) \quad (5.3)$$

$$output_{human_i} = [q_{h_0}, q_{h_1}, \dots, q_{h_n}] \quad (5.4)$$

$$output_{robot_i} = [q_{r_0}, q_{r_1}, \dots, q_{r_n}] \quad (5.5)$$

Where q_{r_i} is a $rd \times 1$ robot configuration vector, with rd being the number of degrees of freedom the robot possesses and q_{h_i} is a $hd \times 1$ human configuration vector, with hd being the number of degrees of freedom the human possesses.

The framework has four major components:

1. **Demonstration Segmentation:** Segments the demonstration into a sequence of sub-tasks which involve the manipulation of the different PPE components. The segmented demonstration is the input of the Demonstration Processing

component.

2. **Demonstration Processing:** Extracts the necessary information that the rest of the framework needs in order to generate the robot and expected human motions. This information is then fed to both the Support and Transfer Pipelines.
3. **Support Pipeline:** Generates an expected human trajectory and robot configuration corresponding to the doffing motion where the robot holds a PPE component in place and the human moves out of the PPE component.
4. **Transfer Pipeline:** Generates a human configuration and robot trajectory corresponding to the doffing motion where the human holds a static posture and the robot removes the PPE component from the human.
5. **Motion Evaluation:** Scores and compares the generated transfer and support motions to determine which of them should be used for the task.

Each of these four major components have sub-components which process different parts of the given input and come up with different portions of that major component's output.

We describe each of these components in detail below.

Segment demonstration according to grasps The first step is to extract the important parts of the demonstration from the input data, i.e. where the human is manipulating the PPE. We define *manipulation phases* as portions of the demonstration where the human has a stable grasp on a PPE component. We use changes in the grasp of the PPE to partition a complex task into a series of sub-tasks. Detecting these transitions automatically in the demonstration requires accurate detection of human grasps, which is not within the scope of this work, thus we perform this segmentation manually.

Obtain target body part This component takes as input the trajectories of the human and PPE component for each segment. Let the human’s body be discretized into a set of rigid bodies H . The target body part b is the one with the greatest cumulative displacement from the PPE component:

$$b = \operatorname{argmin}_{j \in H} \left(\sum_{i=0}^{|\tau_h|} |t_{ppe,i} - t_{body,i,j}| \right) \quad (5.6)$$

Where $t_{body,i,j}$, the translation of body part j at trajectory index i , is computed by forward kinematics on $\tau_h(i)$. The output of this component is a trajectory composed of the transforms describing the changes of b ’s pose throughout the demonstration segment.

Obtain goal state The goal state is the pose of the PPE in the body frame at the end of the demonstration: $T_{body,n}^{ppe}$. This goal state will be used as input for the generation of the robot’s motion.

Obtain human neutral configuration This component outputs the neutral human configuration, q_{h_0} , which is the human’s configuration at the start of the demonstration.

Extracting the motion of PPE relative to target body part The pose space motion of the PPE component relative to b serves as the initial trajectory which the framework will use in generating transfer and support trajectories. This component outputs the PPE component’s trajectory in the frame of b (note that b can be moving as well).

Given the i th PPE component and b transforms $T_{0,i}^{ppe}$ and $T_{0,i}^b$, extracted from the demonstration, we have:

$$T_{body,i}^{ppe} = T_{0,i}^{ppe} (T_{b,i}^0) \quad (5.7)$$

The output of this component is a trajectory composed of transforms representing the pose of the PPE with respect to the target body part’s transform.

Purify into transfer and support motions To address the problem of having adapting to the human’s reactions, the motion of the PPE is purified into support and transfer motions. These motions are composed of a sequence of transforms describing the change in pose of the PPE component (transfer) or b (support) with respect to a static frame—either T_0^b for transfer motions or T_0^{ppe} for support motions. A pure transfer motion base frame is obtained by multiplying the relative motion trajectory transforms by the transform of b at the start of the demonstration, as the base frame:

$$T_{transfer,i} = T_{b,i}^{ppe} (T_{0,0}^b) \quad (5.8)$$

Likewise, a pure support motion can be obtained by multiplying the inverse of the trajectory transforms to the target body part with the PPE component transform, at the start of the demonstration, as the base frame.

$$T_{support,i} = (T_{b,i}^{ppe})^{-1} (T_{0,0}^{ppe}) \quad (5.9)$$

Where i is the current time step, $T_{transfer,i}$ is a transformation matrix defining the

pose of the PPE in the world frame, and $T_{support,i}$ is also a transformation matrix defining the pose of the target body part in the world frame.

The intuition behind the use of these kinds of motions is that by either moving the PPE component along a smooth trajectory or keeping its pose static, the human will be able to perform the adjustment motions themselves. This component will output two trajectories; one for the PPE component and the other for b .

Transfer Pipeline: Checking feasibility of transfer motions Recall that we have the freedom to specify where the human stands with respect to the robot, which determines where the human’s trajectory will be in the robot’s workspace. This component takes as input the obtained transfer trajectory, and attempts to find a transformation of the entire trajectory such that it is reachable by the robot. We do this by first creating a 2-dimensional xy grid and discretize angles about the z -axis. We then apply the transform of each grid cell to the entire trajectory and check if all the points have valid IK solutions for one or both (depending on the task) of the robot’s manipulators. All such feasible trajectories are stored in a set of feasible transfer motions.

, and displace the entire transfer trajectory using these discretizations.

$$T_{new_transfer,i} = T_{transfer,i}(T_{disp,i}) \quad (5.10)$$

Where $T_{disp,i}$ is the displacement transformation matrix.

We check the feasibility of this new transfer trajectory using inverse kinematics. If all of the points have IK solutions i.e. is feasible, then this component adds that trajectory to the set of feasible transfer motions. If there is at least one element in this set, then this component outputs that set of trajectories, a flag indicating that

the transfer motion is executable by the robot along with the corresponding robot trajectory $\tau_r \subset Q_r$ for the transfer motion and pose for where the human should stand. If no feasible (x, y, θ) displacement is found, then for every (x, y, θ) for which the start and end transforms of the adjusted transfer motion are reachable, this component computes a trajectory composed of the closest possible q_r configurations obtained using a Jacobian-based, gradient-descent IK method. The output is then the set of closest-fit τ_r trajectories, a flag indicating that the transfer motion is not feasible, and the set of (x, y, θ) standing poses for which the start and end transforms of the adjusted transfer motion are reachable.

Transfer Pipeline: Generate robot motion and human configuration Generating transfer motions is done using motion planning by trajectory optimization (Section 6), with the output of the feasibility checking component as input. These motions are τ_r which will allow it to remove the PPE from a nearly-motionless human. This component also generates the configuration which the human must hold while the robot removes the PPE from his or her body.

$$transfer_i = (transfer_{h,i}, transfer_{r,i}) \quad (5.11)$$

$$transfer_{h,i} = [q_{h,0}, q_{h,1}, \dots, q_{h,n}] \quad (5.12)$$

$$transfer_{r,i} = [\tau_{r,0}, \tau_{r,1}, \dots, \tau_{r,n}] \quad (5.13)$$

$$\tau_{r,i} = [q_{r,0}, q_{r,1}, \dots, q_{r,n}] \quad (5.14)$$

The output of this component is set of robot and human trajectories (the human being static).

Support Pipeline: Generate human motion and robot configuration This component takes as input the support trajectory and performs trajectory optimiza-

tion (Section 6) to create a new support motion for the human. It also uses the first transform of the trajectory to generate a static robot configuration, using inverse kinematics, for the robot. This configuration will be used to hold the PPE in place. The robot will grasp the PPE upon the human’s command, once the human places the appropriate parts of the PPE in the robots gripper(s).

$$support_i = (support_{h,i}, support_{r,i}) \quad (5.15)$$

$$support_{r,i} = [q_{r,0}, c_{r,1}, \dots, c_{r,n}] \quad (5.16)$$

$$support_{h,i} = [\tau_{h,0}, \tau_{h,1}, \dots, \tau_{h,n}] \quad (5.17)$$

$$\tau_{h,i} = [q_{h,0}, q_{h,1}, \dots, q_{h,n}] \quad (5.18)$$

The output of this component is a set of vectors representing the static robot configurations and a set of human trajectories corresponding to the support motions. We again use trajectory optimization to plan human motions.

Operator picks best trajectory and configuration Our motion generation step generates multiple robot trajectories with different start and end configurations. Since we do not model the human and PPE with high accuracy (indeed such modeling is prohibitively difficult), we require an operator-in-the-loop to select which trajectory will be best suited for the current doffing task. The operator is able to see all generated trajectories and decides which one the robot/human should execute. The input of this component will be all the trajectories from the transfer/support motion generation component, and the output is a single robot/human trajectory and human/robot configuration.

Score human and robot motions This component takes as input a human and robot trajectory then uses metrics to assign a scalar-valued score to the motion. To score the human configuration, a human trajectory is generated by matching each timestep in the robot trajectory to a given human configuration. The human's trajectory is evaluated using the metrics described in Section 4.0.2. The robot trajectory is evaluated using metrics from Sections 6. The overall score of a motion is then:

$$S(\tau_r, \tau_h) = E(\tau_h) + \lambda(\tau_h) + \frac{1}{MS(\tau_r)} \quad (5.19)$$

where MS is the manipulability metric defined in Section 6. Out of the motions considered by the framework, the motion with the lowest S value is the one used to doff the PPE component.

Chapter 6

TRAJECTORY OPTIMIZATION

In this section, we outline how we generate transfer and support motions for both the human and the robot. TrajOpt [17] is used because of its ability to handle non-convex problems using Sequential Convex Optimization [19]. This flexibility allows us to use our motion assessment metrics as well as giving us the freedom to insert other metrics which we enable us to generate appropriate plans in the presence of a human. Robot motions are planned for transfer motions while human motions are planned to create an expected motion for the human for support motions.

To address the problem of mapping unreachable human motions to valid robot motions, we plan with constraints and costs on the resulting poses of the robot's end-effectors. To do so, we needed the following:

1. PPE Simulation
2. Changepoint Identification
3. Cost Functions

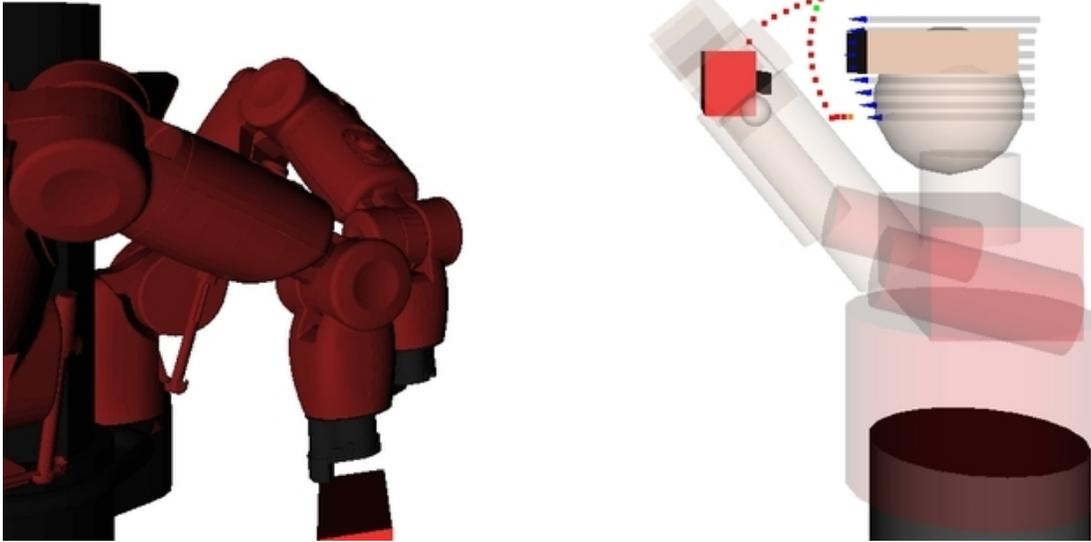


Figure 6.1: Rigid representation of the goggles. The black rectangular prism behind the larger, bright red object is the strap of the simulated goggles retracted after being freed from contact. The red points represent its removal trajectory, while the arrows intersecting the human model head represent the rays from the ray casting done in simulation. This ray casting is used to determine when the straps will snap back to the goggles body.

6.0.1 PPE Simulation

PPE simulation gives us more accuracy when determining segmentation points as well as provide collision information to our planner. The difficulty in simulation lies in properly recreating the deformation [20] [21] [22] that the PPE components undergo during doffing. To address this problem, we use a conservative geometric representation of the PPE that allows us to plan to within some acceptable margin of the actual PPE deformation. We then create robot models which include the PPE as being rigidly attached to the robot as fixed extensions of the end-effector/s. We do these for two reasons:

1. This allows us simulate the moving parts of the PPE by adding them as joints on the robot. For example, the strap on the goggles was added as a prismatic joint which collapses when it is not in contact with the human.

2. This allows us perform collision checking using distance computation [23] in TrajOpt

Providing a more accurate and robust simulator, such as properly modeling the deformable PPE using methods similar to [20] [21] [22], will allow for more accurate motion segmenting and more precise motion planning.

6.0.2 Changepoint identification

We observed that for portions of the demonstration where the PPE component is detached from the human, there should be more freedom in planning the robot motions. Thus we segment the input trajectory into two phases: the removal phase and the placement phase. These phases are separated at the point where the PPE component is considered to be removed from the person. This point is determined by running the demonstration in our simulator and finding the point in the trajectory where there are changes to the PPE component’s physical state/qualities, such as shape and geometry:

$$P_{point}(i) = \begin{cases} \text{true} & \text{change in PPE component state} \\ \text{false} & \text{otherwise} \end{cases} \quad (6.1)$$

where P determines if the i th trajectory index point is a phase separation point. In the event of there being multiple phase separation points, we take the point with the lowest value. Trajectory optimization is then performed at each of these phases, with all of the costs applied to the first(removal) phase and only Legibility (23) for the second(placement) phase. The resulting trajectories from each phase are combined to produce the final trajectory.

Within each phase, we perform another segmentation to determine critical points in the motion. These are points where the Cartesian trajectory of the PPE drastically changes in direction.

$$x_{prev,i} = t_{ppe,i} - t_{ppe,i-1} \quad (6.2)$$

$$x_{next,i} = t_{ppe,i+1} - t_{ppe,i} \quad (6.3)$$

$$\rho_{x_i} = \frac{\arccos(x_{prev,i} \cdot x_{next,i})}{(\|x_{prev,i}\| * \|x_{next,i}\|)} \quad (6.4)$$

$$\psi_{x_i} = \arctan 2(\sin(\rho_{x_i}), \cos(\rho_{x_i})) \quad (6.5)$$

Where ρ is the angle between the vectors $x_{prev,i}$ and $x_{next,i}$ about $x_{prev,i} \times x_{next,i}$ and τ is that angle constrained to the set $[-\pi, \pi]$. We define changes to be changes the motion at point x_i if τ_{x_i} exceeds a threshold ϵ_a : We also check for points where the linear or angular velocity of the motion undergoes a significant increase or decrease:

$$v_{prev,x_i} = \frac{t_{ppe,i} - t_{ppe,i-1}}{\Delta t_{i-1}} \quad (6.6)$$

$$v_{next,x_i} = \frac{t_{ppe,i+1} - t_{ppe,i}}{\Delta t_i} \quad (6.7)$$

$$\|v_{next,x_i} - v_{prev,x_i}\| < \epsilon_v \quad (6.8)$$

$$\epsilon_{v,i}, \Delta t_{i-1}, \Delta t_i \in \mathbb{R}$$

Where ϵ_v is a constant determined by the previous $i - 1$ points. The output of the motion segmentation are the indices of the critical points in the trajectory,

determined by the function:

$$C_{point}(i) = \begin{cases} \text{true} & \|v_{next} - v_{prev}\| < \epsilon_v \text{ or} \\ & |\rho_i| > \epsilon_a \\ \text{false} & \text{otherwise} \end{cases} \quad (6.9)$$

The poses of the PPE during these critical points, along with the goal state pose $T_{world,n}^{ppe}$, are then considered to be pose constraints on the PPE component and are easily converted to pose constraints on the end-effector/s.

6.0.3 Robot planning costs

We assign a pose-space deviation cost to the generated robot motions to minimize deviation from the demonstration motion. This cost, denoted Ω , is the difference in the of the i th trajectory point's relative pose between the target body part and the PPE from the i th demonstration point's relative pose between the same objects:

$$DP_{T_b, \tau_{pr,k}}^{T_a} = \begin{bmatrix} t_{T_b,k}^{T_a} \\ \arctan 2(R_{T_b,32,k}^{T_a}, R_{T_b,33,i}^{T_a}) \\ - \arcsin(R_{T_b,31,i}^{T_a}) \\ \arctan 2(R_{T_b,21,i}^{T_a}, R_{T_b,11,i}^{T_a}) \end{bmatrix} \quad (6.10)$$

$$\Omega(PPE, body_{\tau_{pr,i}}) = ||| DP_{body, \tau_{dr,i}}^{PPE} ||| - ||| DP_{body, \tau_{pr,i}}^{PPE} ||| \quad (6.11)$$

Where τ_p denotes the planned trajectory, τ_d denotes the demonstration trajectory, and DP denotes pose difference between two transforms T_a and T_b at the k th point in the trajectory. We also add a cost which penalizes deviation from the curvature

of the demonstration. We calculated ψ_{x_i} and $\psi_{x_{d_i}}$, using (10), for each point from the planned and demonstration trajectories, respectively. We then take the absolute difference as the cost ζ .

$$\zeta(x_i) = |\psi_{x_i} - \psi_{x_{d_i}}| \quad (6.12)$$

For two-arm planning, we calculate $\zeta(x_i)$ for the end-effector points of each arm and take the sum as the cost.

We use the manipulability measure [24] to maximize the dexterity and resistance to forces of the trajectory as well as maintaining a good distance from joint limits. Given a trajectory τ_r the manipulability MS is:

$$M(q_r) = \sqrt{\det(J(q_r)J(q_r)^T)} \quad (6.13)$$

$$MS(\tau_r) = \sum_{i=0}^{n-1} M(\tau_r(i)) \quad (6.14)$$

M calculates the manipulability of a configuration, $J(\tau_r(i))$ is the Jacobian of the robot's end-effector for i th trajectory point. We implement manipulability as a per-configuration cost (M). For two-arm planning, we calculate the manipulability of each arm and take the sum as the score.

It is also easier for the human to cooperate with the robot if they know what to expect from the robot's motion. Dragan et al. formalized notions of motion predictability and legibility [25]. They proposed mathematical models for these based on the principle of rational actions and used these models to generate legible motion for a robot using trajectory optimization [26]. Given a trajectory τ , motion legibility L can be quantified as:

$$C[\tau] = \frac{1}{2} \int \tau'(t)^2 dt \quad (6.15)$$

$$P(G|\tau_{S \rightarrow q}) = \frac{1}{Z} \frac{\exp(-C[\tau_{S \rightarrow q}] - V_{G_R}(q))}{\exp(-V_{G_R}(S))} P(G_R) \quad (6.16)$$

$$V_G(q) = \min_{\tau \in \Xi_{S \rightarrow q}} C[\tau] \quad (6.17)$$

$$L[\tau] = \frac{\int P(G_R|\tau_{S \rightarrow \tau(t)}) f(t) dt}{\int f(t) dt}, \quad (6.18)$$

where $C[\tau]$ is an arbitrary cost functional, G_R and G are the expected and known goal configurations, respectively. Z is a normalizer across the space of trajectories and $P(G_R)$ is a prior which can be assumed to be uniform in the absence of any information. $P(G|\tau_{S \rightarrow q})$ is the probability that the user infers the right goal given trajectory snippet $\tau_{S \rightarrow q}$ at configuration q . We use the legibility measure as a cost function during our placement phase to keep our trajectories smooth as well as communicate to the human that the PPE is being placed away from them. We implement legibility as a partial-path cost.

6.0.4 Robot two arm planning constraint

Two-arm planning tasks involve maintaining a kinematic closure constraint which preserves the pose difference between the robot's two end-effectors. To impose this constraint, we use Cortés' and Siméon's idea of active and passive subchains[27]. We first take the pose difference between the two end-effectors at the start of the motion:

$$\Gamma(i) = DP_{R, \tau_{pr, i}}^L \quad (6.19)$$

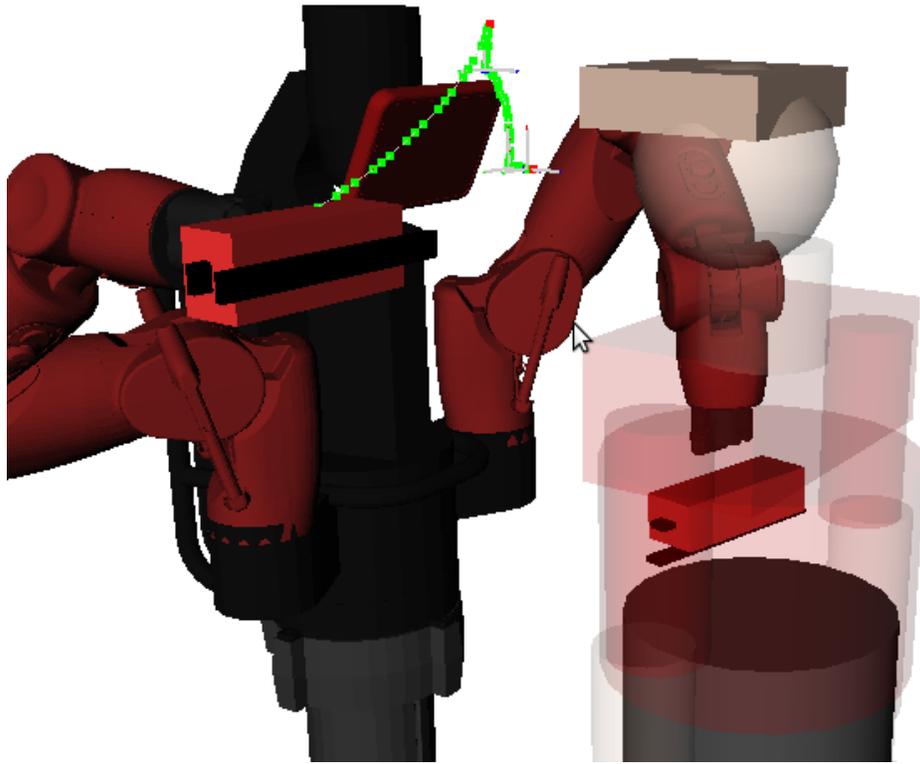


Figure 6.2: Planning for the goggles transfer doffing in OpenRAVE with TrajOpt. The green points represent the planned trajectory.

Where L and R represent the transforms of the left and right end-effectors, respectively and Γ is a function which takes the pose difference between the left and right end-effectors during the i th point of the motion. The closure constraint is then:

$$\|\Gamma(0)\| - \|\Gamma(i)\| = 0 \quad (6.20)$$

This constraint is then imposed on one of the arms, giving the opposite (active) arm the freedom to move while the constrained (passive) arm follows it to preserve the constraint.

6.0.5 Human planning cost functions

To generate expected human motion we used hands-to-face distance score as a cost to prevent excessive arm movement, biasing most of the movement to the torso kinematic chain and the REBA cost to plan for the minimal effort motion. However, the problem with REBA is that there is no well-defined gradient; scores are defined on a series of if-else blocks with various ranges for the joint values. Furthermore, the combination of the separate linkage scores to determine the final overall body score has no explicit formula. Thus, to obtain a gradient, we modify our computation of the REBA cost for planning purposes.

We observed that the REBA metric divides a human's configuration space into subspaces. Each subspace is a scalar field whose value is the REBA score assigned to the joint angles falling inside the subspace's bounds. Since the scores are uniform throughout their subspace, there are no gradients which will enable the optimizer to move within and out of a subspace. Thus we change the cost such that it varies throughout the subspace should the subspace not be the optimal (not having the lowest REBA score possible). We assign the score of a point in the subspace to be

the smallest difference in each of the dimensions to the boundary of the optimal adjacent neighboring subspace:

$$BS(q_{h_i}) = \min_j (BD(q_{h_i}, r_{b_j})) \quad (6.21)$$

$$BD(q_{h_i}, r_{b_j}) = \begin{cases} & (q_h[k] < r_{b_j}[k] \text{ or} \\ BM(q_{h_i}, r_{b_j}) & q_h[k] > r_{b_j}[k + hd]), \\ & E(r_{b_j}) < E(q_{h_i}) \\ 0 & \text{otherwise} \end{cases} \quad (6.22)$$

$$BM(q_{h_i}, r_{b_j}) = \begin{aligned} & \min_k (\min(\|q_h[k] - r_{b_j}[k]\|), \\ & \min(\|q_h[k] - r_{b_j}[k + hd]\|)) \end{aligned} \quad (6.23)$$

$$r_{b_j} = [\theta_{0,min}, \theta_{1,min}, \dots, \theta_{hd-1,min}, \theta_{hd,min}, \theta_{hd+1,min}, \dots, \theta_{2hd,min}] \quad (6.24)$$

where BS is the region boundary distance score, BM is the distance between joint angle q_{h_i} and r_{b_j} , the vector representing the bounds of the j th REBA region. See Figure 6.3 for an illustration. These cost fields were precomputed to further speed up planning. The optimizer then uses the gradient

$$\nabla q_{h_i} = \min_l (BS(q_{h_i}) - BS(q_{h_i,l}')), \quad (6.25)$$

where ∇q_{h_i} is the gradient and $q_{h_i,l}'$ is a vector obtained by perturbing the l th element of q_{h_i} by some small value ϵ_c .

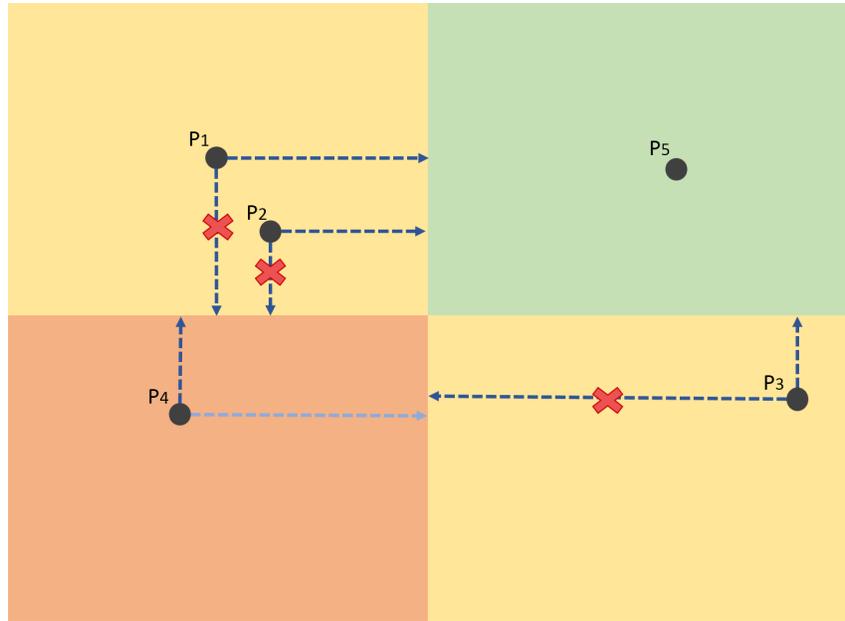


Figure 6.3: 2-D example of how the REBA metric partitions the human configuration space. The orange region has the highest effort score, followed by the yellow regions, and then the green region. P1 and P2 show that two points inside the cost field will have different values and thus different gradients. P4 shows how the cost function assigns a cost should there be two adjacent regions with better scores, with the dark blue arrow indicating a better score compared to the light blue arrow. The x's in the score calculations of P1, P2, and P3 show how the cost function does not consider dimensions of the points which are within the bounds of the next best neighboring region.

Chapter 7

Results

To verify our hypothesis that a robot assistant makes the doffing procedure safer for the human, we conducted an experiment where several volunteers were to perform the doffing procedure with and without the robot. We focus on the doffing of five separate PPE components: the apron, goggles, faceshield, hood, and coverall. Each subject was required to don the complete PPE to simulate an actual field deployment scenario. Subjects were observed by 4 different cameras: two webcams, an audio-visual camera, and a Microsoft Kinect2. We use data from ten human subject trials in our experiment data analysis with $\epsilon_d = 0.1\text{m}$ and $\epsilon_e = 7$ REBA units. The former is what we believe to be a distance where slight miscalculations in movement may lead to a person touching his or her face, while the latter is the maximum score a posture can get before being deemed high-effort [3].

For the unassisted doffing, subjects were required to remove the components of the PPE in a specific order, taking great care to follow the CDC guidelines [1]. If a component of the PPE was mishandled the entire process was repeated.

For the assisted doffing, we used Rethink Robotics' Baxter robot equipped with two parallel grippers and a single vacuum gripper. Subjects are instructed to avoid

touching their face during the entire doffing procedure. A kill switch was used to stop the robot in case of an emergency.

To ensure that the proper actions are taken during both assisted and unassisted doffing procedures, each subject is required to watch a series of videos detailing the a) proper donning of the PPE b) the CDC guidelines for PPE doffing and c) the steps and removal phases in the robot-assisted doffing.

7.0.1 Metrics

To assess risk, we used point cloud data collected from the Kinect sensor to track the hands and face of each subject during the doffing of the five specified PPE components, at 30hz, and give them a score using λ . To assess effort, we used the recordings from the AVC camera and manually scored postures at 60hz using E .

We setup the human model for each task as follows: we first take various measurements of the subject's body, such as height, shoulder width, and arm length, and create an anatomically similar human model. For the goggles and faceshield removal tasks, we put an extra rigid block on top of the model's head. This block serves as an extra obstacle TrajOpt has to avoid and get some clearance over the subject's head. Finally, we set the active DoFs of the human model to be the entire body except the coverall. We also include an XYZ translation joint on the feet of the human model to move it along the ground. This is to avoid having to plan foot-steps and balance for the support motions. We did not plan support motions for the coverall because the only way to remove it was a sequence of transfer and support motions. Removing the upper part of coverall required the robot to pull the coverall off, by the lapel, the subject's shoulders, hold it in place as the subjects moves his or her shoulders out of the coverall, followed by the robot pulling the coverall down and holding it in place for the user to pull his or her arms out. Removing the upper

Task	$ Q_h $	# Robot Manipulators	Manipulator Type	$ Q_r $
Apron	18	2	Gripper	14
Faceshield	18	1	Suction	7
Goggles	18	1	Suction	7
Hood	18	2	Gripper	7
Coverall	0	2	Gripper	7

Table 7.1: Human and robot DOF for the tasks)

	Unassisted			Robot-Assisted		
	Mean	StdDev	Outliers	Mean	StdDev	Outliers
Apron	0.6737	0.4546	0	0.0958	0.2306	1
Faceshield	6.028	1.159	2	0.0025	0.0049	1
Goggles	6.604	2.096	3	4.4824	3.534	0
Hood	1.365	1.359	3	5.946	1.979	1
Coverall	5.447	2.144	0	5.207	3.431	1

Table 7.2: Hands-to-Face Distance Score Statistics (lower is better)

part of the coverall was done by the robot pulling the coverall down the user’s legs and holding it in place as the human lifts his or her legs out of the coverall. Since we cannot remove the coverall any other way, we only planned transfer motions for the robot hence the human DoF is 0. See Table 7.1 for details about the robot setup.

7.0.2 Experiment results

For each subject we tested both transfer and support variants of the assisted tasks. For the analysis we took the variant with the better (lower) hands-to-face distance score from the trials and used its corresponding effort score.

Figure ??(a) and Table 7.2 show that the robot-assisted doffing outperformed the unassisted doffing in three out of the four infection-risk tasks (i.e. reducing the hands-to-face distance score). We do not consider coverall doffing as an infection-

risk task because it is removed using the inner gloves, which are assumed to be uncontaminated. It is clear that the unassisted hood doffing is safer than its robot-assisted counterpart. This is because in the unassisted doffing, the subject grasps the hood from the top of his or her head, which naturally minimizes hands-to-face distance score. In contrast, the robot-assisted doffing of the hood requires the subject to grasp the lower fringes of the hood, which is closer to the face than the top part. It can also be seen that there is more variation in the hands-to-face-distance scores in the faceshield and hood assisted doffing. Both tasks required the subject to properly adjust the respective PPE components for the robot to achieve a secure grasp. For the faceshield, the subject had to hold it in place using the bottom lip of the shield so that the vacuum gripper has a flat surface for successful suction. For the hood, subjects found it difficult to place the fringes onto the grippers as adjusting the hood blocked their line of sight. Thus a lot of the placement was done by touch. Assisted doffing produced significantly less outliers than the unassisted doffing, showing that the assisted doffing produced more consistent trials than the unassisted one. This can be attributed to the more strict procedure and less freedom of motion in the assisted tasks.

Figure ??(b) shows that assisted doffing required less effort from the subjects compared to unassisted doffing for all the tasks. Although assisted doffing takes significantly more time, it required less movement of the subjects. Assisted doffing tends to bias the subject's movements toward either the trunk or arm regions. This leaves either the trunk region exerting little to no effort during arm motions and vice-versa, which places most of the assisted doffing effort scores below ϵ_e (hence the zero scores for the apron, goggles, faceshield, and hood tasks), as high-risk postures generally have both the arms and the trunk exerting effort. Four out of five of the unassisted doffing tasks required the subject to remove something by passing it

over their head using their arms. Thus these tasks produced postures wherein the torso and neck were bent and the arms were at full extension. Such postures were responsible for producing spikes of high effort scores from the assisted tasks. For the coverall, the bend-down motion subjects had to execute in order to remove the PPE during unassisted doffing produced very large increases in the effort score. In contrast, with Baxter holding down the coverall, the subjects did not need to bend down and grab the PPE.



Figure 7.1: Robot-Assisted doffing using transfer motions.

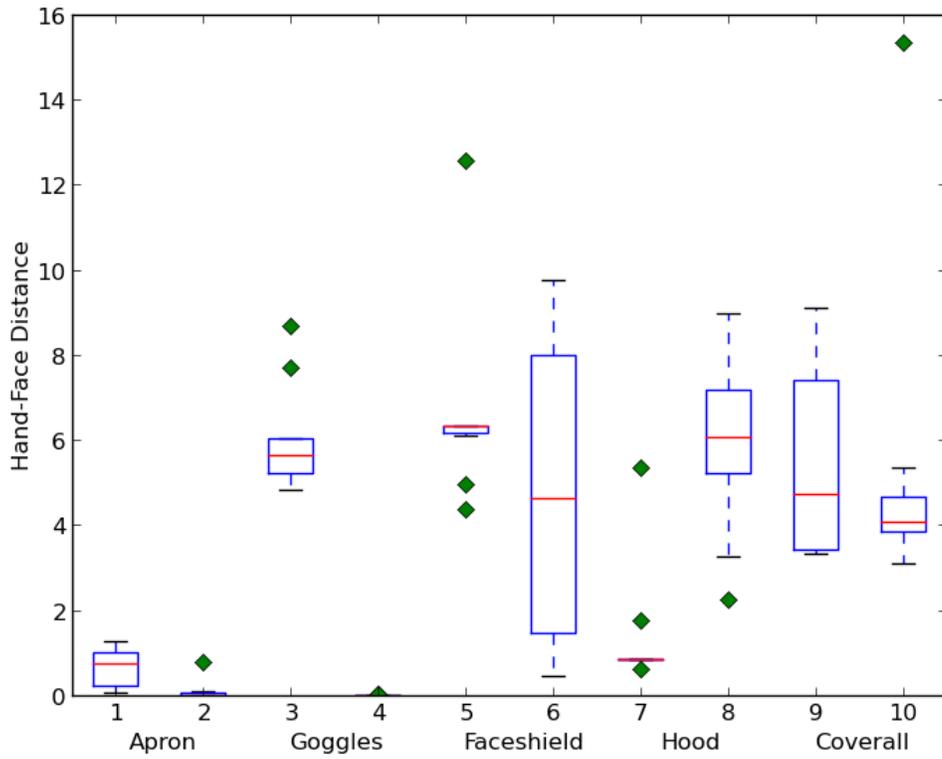


Figure 7.2: Box plots showing the assisted vs. unassisted hands-to-face distance scores for the PPE doffing tasks. The odd-numbered columns show the data for unassisted doffing, while the even-numbered columns show the data for the robot-assisted doffing. The red horizontal lines are the means for each score, the blue represent the quadrants, and the green diamonds are outliers.

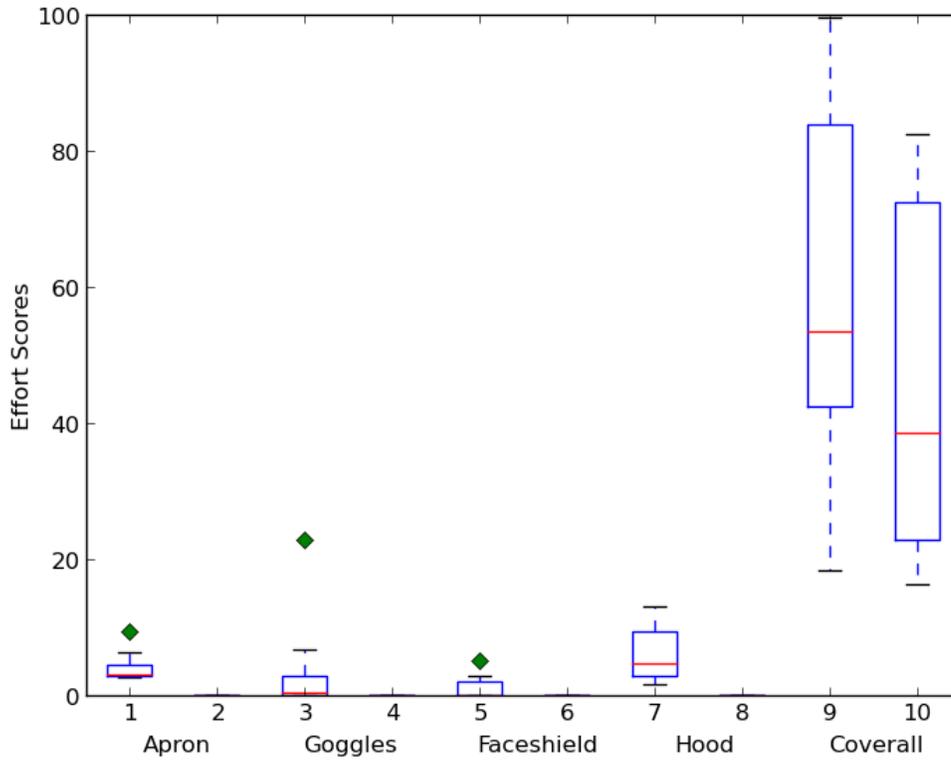


Figure 7.3: Box plots showing the assisted vs. unassisted effort scores for the PPE doffing tasks. The odd-numbered columns show the data for unassisted doffing, while the even-numbered columns show the data for the robot-assisted doffing. The red horizontal lines are the means for each score, the blue represent the quadrants, and the green diamonds are outliers.

Chapter 8

Discussion

While the results of the experiments do not overwhelmingly favor assisted doffing, it must be taken into consideration that the individual components of the framework implementation can be further improved and thus achieve better results. Better state simulation will allow for more accurate segmentation and motion planning, as trajectory optimization is largely dependent on forward simulating the perturbations to the current trajectory in order to obtain the gradient. Cost functions are also considered inputs to the motion generation component, thus cost functions which more accurately depict the reward of performing a motion or a different combination of cost functions can largely improve the performance of the framework implementation.

Future work can be done in creating a feedback controller for the robot assistant which modifies the generated transfer or support trajectory according to sensory information. Because of the component based architecture of the framework, the addition of a feedback controller can be added on as another component or process block to the framework. An idea for how this controller can be implemented is using optimization to modify the next m points of the trajectory while the last $n - m$ points

are adjusted according to the modifications on the first m ones. However, modeling human behavior and creating reward functions which will be able to capture the "goodness" of a robot reactionary motion is difficult.

Chapter 9

Conclusion

We hypothesized that the use of a semi-autonomous robot assistant during the doffing of PPE would increase the overall safety of the procedure. We proposed a framework which takes an existing, human-only doffing demonstration and synthesizes human and robot motions to produce a robot-assisted version of the demonstration. We also introduced a way of classifying doffing motions based on their relative motion with respect to both the PPE and the human, in order to reduce the complexity involved in creating human and robot motions from the demonstrated motion. We then tested our framework using human subject trials with subjects performing both unassisted and assisted doffing of PPE. Our results show that for the majority of PPE components which were removed with the help of a robot assistant, one or both of either infection risk and/or effort exerted were reduced in the robot assisted variant of the procedure. Furthermore, the introduction of the robot assistant introduced less variance in the execution of each doffing task. The component-based architecture of the framework allows portions of the procedure to be upgraded in order to improve the overall performance of the system or achieve specific goals.

Bibliography

- [1] “Center for disease control and prevention. guidance on personal protective equipment (ppe) to be used by healthcare workers during management of patients with confirmed ebola or persons under investigation (puis) for ebola who are clinically unstable or have bleeding, vomiting, or diarrhea in u.s. hospitals, including procedures for donning and doffing ppe.” [Online]. Available: <http://www.cdc.gov/vhf/ebola/healthcare-us/ppe/guidance.html>
- [2] L. McAtamney and E. Corlett, “Rula: A survey method for the investigation of work-related upper limb disorders,” *Applied Ergonomics*, vol. 24, pp. 91–99, 1993.
- [3] S. Hignett and L. McAtamney, “Rapid entire body assessment: Reba, applied ergonomics,” *Applied Ergonomics*, vol. 31, pp. 201–205, 2000.
- [4] R. Bemelmans, G. Gelderblom, P. Jonker, and L. Witte, “Socially assistive robots in elderly care: A systematic review into effects and effectiveness,” in *JAMDA*, 2012.
- [5] K. Yamazaki, R. Ueda, S. Nozawa, Y. Mori, T. Maki, N. Hatao, K. Okada, and M. Inaba, “A demonstrative research for daily assistive robots on tasks of cleaning and tidying up rooms,” in *Proceedings of the 14th Robotics Symposia*, 2009.
- [6] W. Sheng, A. Thobbi, and Y. Gu, “An integrated framework for human–robot collaborative manipulation,” in *IEEE Transactions on Cybernetics*, vol. 45, 2014, pp. 2030–2041.
- [7] Y. Gao, H. Chang, and Y. Demiris, “User modelling for personalised dressing assistance by humanoid robots,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 45, 2015, pp. 1840–1845.
- [8] A. Clegg, J. Tan, G. Turk, and K. Liu, “Animating human dressing,” in *Transactions on Graphics (SIGGRAPH)*, 2015.
- [9] D. Shinohara, T. Matsubara, and M. Kidode, “Learning motor skills with non-rigid materials by reinforcement learning,” in *IEEE International Conference on Robotics and Biomimetics*, 2011.

- [10] D. Shinohara, M. Kidode, and T. Matsubara, “Reinforcement learning of a motor skill for wearing a t-shirt using topology coordinates,” *Advanced Robotics*, vol. 27, 2013.
- [11] K. Yamazaki, R. Oya, K. Nagahama, K. Okada, and M. Inaba, “Bottom dressing by a dual-arm robot using a clothing state estimation based on dynamic shape changes,” *Advanced Robotic Systems*, vol. 13, 2016.
- [12] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robotics and Autonomous Systems*, vol. 57, 2009.
- [13] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, “Learning from demonstration and adaptation of biped locomotion,” *Robotics and Autonomous Systems*, vol. 47, pp. 79–91, 2004.
- [14] C. Fernandez, M. A. Vicente, R. P. Neco, and R. Puerto, “Robot grasp learning by demonstration without predefined rules,” *International Journal of Advanced Robotic Systems*, vol. 8, 2011.
- [15] J. J. Kim and J. J. Lee, “Trajectory optimization with particle swarm optimization for manipulator motion planning,” *IEEE Transactions on Industrial Informatics*, vol. 11, pp. 620–631, 2015.
- [16] N. A. Vien and M. Toussaint, “Pomdp manipulation via trajectory optimization,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.
- [17] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, “Finding locally optimal, collision-free trajectories with sequential convex optimization,” in *Robotics: Science and Systems (RSS)*, 2013.
- [18] D. Berenson, “Manipulation of deformable objects without modeling and simulating deformation,” in *Intelligent Robots and Systems (IROS)*, November 2013.
- [19] R. G. Strongin and Y. D. Sergeyev, “Global optimization with non-convex constraints: Sequential and parallel algorithms,” *Mathematics and Statistics (Archive)*, vol. 45, 2000.
- [20] M. Liao, Q. Zhang, W. H., R. Yang, and M. Gong, “Modeling deformable objects from a single depth camera,” in *2009 IEEE 12th International Conference on Computer Vision*, 2009.
- [21] F. Faure, G. B., and D. Bousquet, G. Pai, “Sparse meshless models of complex deformable solids,” in *ACM SIGGRAPH 2011 papers*, 2011.

- [22] K. Sundaraj, C. Mendoza, and C. Laugier, “A fast method to simulate virtual deformable objects with force feedback,” in *7th International Conference on Control, Automation, Robotics and Vision*, 2002.
- [23] E. G. Gilberg, D. W. Johnson, and S. S. Keerthi, “A fast procedure for computing the distance between complex objects in three-dimensional space,” *IEEE Journal of Robotics and Automation*, 1988.
- [24] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and D. R., “Manipulability analysis,” in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2012.
- [25] A. Dragan, K. Lee, and S. Srinivasa, “Legibility and predictability of robot motion,” in *Human-Robot Interaction*, 2013.
- [26] A. Dragan and S. Srinivasa, “Generating legible motion,” in *Robotics: Science and Systems (RSS)*, 2013.
- [27] J. Cortés and T. Siméon, “Sampling-based motion planning under kinematic loop-closure constraints,” *Algorithmic Foundations of Robotics VI*, vol. 17, pp. 75–90, 2005.