Dissertation

Algorithms, Implementation, and Studies on Eating with a Shared Control Robot Arm

Laura V. Herlant

March 20, 2018

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania

Thesis Committee:

Siddhartha S. Srinivasa, *advisor*, University of Washington
Christopher G. Atkeson, CMU RI
Jodi Forlizzi, CMU HCII
Leila A. Takayama, UC Santa Cruz

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

People with upper extremity disabilities are gaining increased independence through the use of assisted devices such as wheelchair-mounted robotic arms. However, the increased capability and dexterity of these robotic arms also makes them challenging to control through accessible interfaces like joysticks, sip-and-puff, and buttons that are lower-dimensional than the control space of the robot. The potential for robotics autonomy to ease control burden within assistive domains has been recognized for decades. While full autonomy is an option, it removes all control from the user. When this is not desired by the human, the assistive technology in fact has made them *less* able and discards useful input the human might provide, leveraging for example their superior situational awareness, that would add to system robustness.

This thesis takes an in-depth dive into how to add autonomy to an assistive robot arm in the specific application of eating, to make it faster and more enjoyable for people with disabilities to feed themselves. While we are focused on this specific application, the tools and insights we gain can generalize to the fields of deformable object manipulation, selection from behavior libraries, intent prediction, robot teleoperation, and human-robot interaction. The nature of the physical proximity and heavy dependence on the robot arm for doing daily tasks creates a very high-stakes human-robot interaction.

We build a system that is capable of fully autonomous feeding by (1) predicting bite timing based on social cues, (2) detecting relevant features of the food using RGBD sensor data, and (3) automatically selecting a goal for a food-collection motion primitive to bring a bite from the plate to the operator's mouth. We investigated the desired level of autonomy through user studies with an assistive robot where users have varying degrees of control over the bite timing, control mode-switching, and direct teleoperation of the robot to determine the effect on cognitive load, acceptance, trust, and task performance.

I would like to dedicate this thesis to my two beautiful sons who were born during its writing,
and to my husband for his tireless support of my education and career.

Acknowledgements

I want to thank my advisor, Siddhartha Srinivasa, for always imparting his relentless enthusiasm. He has served my interests as a student and researcher, and provided nothing less than boundless support for my endeavors and future. I would like to thank my thesis committee members for their guidance through this process, feedback, discussions, and ideas.

I am very grateful for the supporting staff and collaboration of the Personal Robotics Lab. Clint Liddick, Michael Koval, and Jen King were always there to help when I inevitably broke some inner workings of prpy or openrave. I want to thank Rachel Holladay, who cheerfully helped me run human subject experiments. Brenna Argall, and her students were of considerable help when we first got the robot and collaborated to create a joint code base.

I want to thank several interns and student collaborators who I have had the opportunity to mentor and who have contributed to the implementation of many of the algorithms contained in this thesis. In particular: Ben Weinstein-Raun for his work on the MICO robot's driver implementation, Gabriel Quéré for his contribution to mode-switching on the robot, Puneet Puri for his work on meticulously gathering 3D food scans, and Emmanual Eppinger for his part in gathering relevant social features to predict bite timing.

I am grateful to Kinova Robotics, for their fantastic customer service and partnership. Their desire to implement a feeding mode in the currently available robots has been a driving factor for me to realize this work.

This work was supported by the DARPA SIMPLEX program through ARO contract number 67904LS-DRP, National Institute of Health R01 (award R01EB019335), National Science Foundation CPS (award 1544797), the Office of Naval Research, the Richard K. Mellon Foundation, the Google Hertz Foundation Fellowship Award, and the National Science Foundation Graduate Fellowship Program.

Contents

List of Figures

List of Tables

1
Introduction

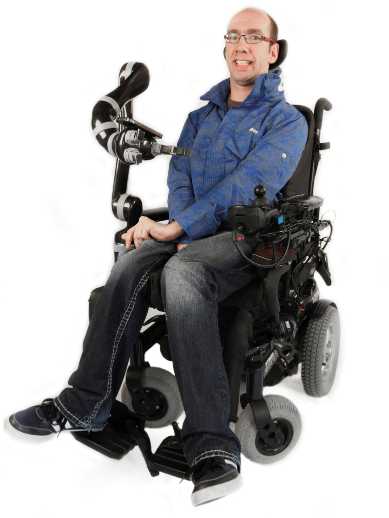
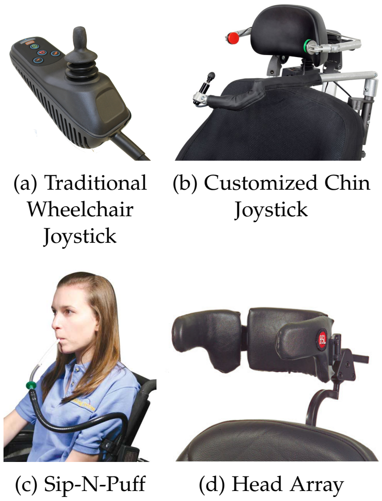


Figure 1.1: The JACO robot, a testing platform that will be used in this work.

Assistive machines—like powered wheelchairs, myoelectric prostheses and robotic arms—promote independence and ability in those with severe motor impairments [Hillman et al., 2002b, Prior, 1990, Sijs et al., 2007, Huete et al., 2012, Yanco, 1998]. As the state-of-the-art advances, more dexterous and capable machines hold the promise to revolutionize ways in which those with motor impairments can interact within society and with their loved ones, and to care for themselves with independence.

However, as these machines become more capable, they often also become more complex. Which raises the question: how to control this added complexity? A confounding factor is that the more severe a person's motor impairment, the more limited are the control interfaces available to them to operate. The control signals issued by these interfaces are lower in dimensionality and bandwidth. Thus, paradoxically, a greater need for sophisticated assistive devices is paired with a diminishing ability to control their additional complexity. Assistive robot arms on the market are mostly using direct control which interprets signals from a traditional interface as commands to change the robot's joint or Cartesian configuration (see section 2.1.2 for details).

Traditional interfaces often cover only a portion of the control space of more complex devices like robotic arms [Tsui et al., 2008a]. For example, while a 2-axis joystick does fully cover the 2-D control space (heading, speed) of a powered wheelchair, to control the end-effector of a robotic arm is nominally a 6-D control problem. This already is a challenge with a 2-D control interface, which is only exasperated if limited to a 1-D interface like a Sip-N-Puff or switch-based head array [Nuttin et al., 2002, Valbuena et al., 2007, Mandel et al., 2009, Prenzel et al., 2007, Luth et al., 2007, Simpson et al., 2008, Firoozabadi et al., 2008, Vaidyanathan et al., 2006, Galán et al., 2008].



(a) Traditional Wheelchair Joystick

(b) Customized Chin Joystick

(c) Sip-N-Puff

(d) Head Array

Figure 1.2: Traditional assistive control interfaces.

To ease the burden of control, some robotic arms are now being provided with special control modes for specific tasks. For assistive robot arms, an example is the "drinking mode" in which the operator's control input is remapped to moving the robot's end effector along an arc of a predetermined radius. We are seeing these task-specific assisted modes in other applications as well, such as using high-level grasp commands for a hand-shaped gripper [Michelman and Allen, 1994].

Similarly, we propose creating a "feeding mode" which would enable assistive arm users to be able to eat independently. Independent feeding is a high-impact task on both improved self-image and reduction of care-giving hours [Chiò et al., 2006, Jacobsson et al., 2000, Prior, 1990, Stanger et al., 1994]. Unlike the drinking mode, we will study sharing control between with robot and the user for different parts of the feeding task to see in which parts robot assistance has the most positive impact. There are also social challenges with feeding that are not as pronounced with drinking or other manipulation tasks. We draw from insights in occupational therapy and human caregivers as well as robotics.

> *This thesis takes an in-depth dive into how to add autonomy to an assistive robot arm in the specific application of eating, to make it faster and more enjoyable for people with disabilities to feed themselves.*

While we are focused on this specific application, the tools and insights we gain can generalize to the fields of deformable object manipulation, selection from behavior libraries, intent prediction, robot teleoperation, and human-robot interaction. The nature of the physical proximity and heavy dependence on the robot arm for doing daily tasks creates a very high-stakes HRI interaction.

Next, we summarize the challenges that we will face creating an assistive feeding mode with the robot arm.

Challenge 1: Controling a high-dimensional robot with a low-dimensional input

A common technique to control a high-dimensional system like an arm with a low-dimensional input like a joystick is through switching between multiple control modes, such as those shown in fig. 1.3. The operation of an assistive device via different control modes is reminscient of upper-limb prosthesis control [Ajiboye and ff. Weir, 2005, Chu et al., 2006, Nishikawa et al., 1999, Scheme and Englehart, 2011, Simon et al., 2011, Tenore et al., 2008, 2009].In the case of prosthetics, control is diverted between different "functions" (e.g. elbow, wrist). The parallel for a robotic arm is to divert control between different subsets of the joint-control space. (Modes that operate subsets of the end-effector control space of course are equally viable.) Within the

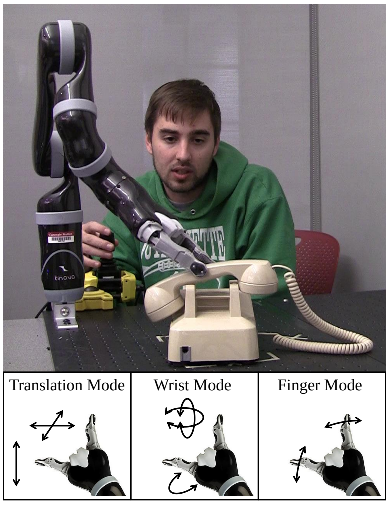


Figure 1.3: In this example, the input is a 3-axis joystick, and the three control modes are translation, wrist, and gripper mode.

field of prosthetics, function switching is known to be cumbersome, and the opportunity for autonomous switching to ease this burden has been identified [Pilarski et al., 2012] (though it is not yet feasible to implement on today's prosthetic hardware). Our interviews with daily users of the Kinova JACO arm identified mode switching as a key problem with robotic arms as well, both in terms of time and cognitive load. We further confirmed objectively that mode switching consumes about 17.4% of execution time even for able-bodied users controlling the MICO robot (a smaller, lighter version of the JACO robot).

Once we identified the high cost associated with mode switching, the question then becomes how to alleviate that burden. While full autonomy is an option, it removes all control from the user. When this is not desired by the human, the assistive technology in fact has made them *less* able. It also discards useful input the human might provide, leveraging for example their superior situational awareness, that would add to system robustness.

Control sharing is a way to offload some control burden, without removing all control authority, from the human [Dragan and Srinivasa, 2013, 2012, Yanco, 2000, Philips et al., 2007, Vanhooydonck et al., 2003, Bourhis and Sahnoun, 2007]. The most common paradigms augment or adjust control signals from the human (e.g. to bridge the gap in control signal dimensionality), or partition the control problem (e.g. high-level decisions like which task to execute lie with the human, and low-level execution decisions lie with the robot).

Here, we propose one alternative role for the autonomy: to assist the user in transitioning between different subsets of the control space—that is, to autonomously remap signals from the user interface to different control *modes* (e.g. subsets of the control dimensions) using a time-optimal model (section 3.2.1). We also hypothesize that automatically switching control modes achieves a sweet-spot of assistance: helping with the tedious parts of the task while still giving full control over continuous motion.

Challenge 2: Food is non-rigid, highly variable, and deformable.

The manipulation of rigid objects has been the dominant subject of robotic manipulation research. To manipulate deformable objects, past work has either evaluated deformation characteristics (such as elasticity or viscosity) [Jacobsson et al., 2000] or performed planning using a topological state representation [Saha and Isto, 2006]. If we want to use a physics model-based technique, we would have to estimate physical properties of each type of food on the fly (potentially causing robustness and trust issues), since a plate full of food does not have a set topology.

Feeding devices that are on the market enact a single preprogrammed motion to get food (with one exception; more details in **??**). As a result, the food must be placed in one of several bowls that are attached to the robot base. The users are given the high-level choice over which bowl is used. However, there is no low-level ability to select bites of food, and certain types of food are not compatible with the single programmed motion.

Instead, we propose having the robot learn from experience a strategy for collecting food morsels. Depending on the properties of food, people perform different techniques for acquiring bites. If trying to eat soup, a scooping motion is called for. If trying to eat a piece of cut fruit, a skewering motion is more appropriate. We propose generating a database of such actions from human demonstration, and then learning which action is appropriate and where on the plate it is most likely to be successfully applied based on color and depth features of the food. The advantage of using this formulation is that it is intuitive to give the user control over which action and where it is applied when we are testing different levels of autonomy. If the robot learned the parameters for the action representation directly from the visual features of the food, then it would not necessarily be possible for the user to define these parameters manually in a meaningful way. We present a taxonomy of the action primitives and implement the skewering action.

Challenge 3: Eating is an inherently social act with its own social rules and norms

Dining is a social act which provides a personal link to the wider community [Marshall, 2005]. People use mealtime as a time to have discussion and companionship. Assisted dining is a careful creation of a new schema with the goal to replicate self-reliant eating as closely as possible Martinsen et al. [2008]. It takes human caregivers months to learn the non-verbal communication schema for a smooth interaction. The robot will also need to be able to tune into the subtle signals for communicating desired actions.

We trained a model to learn bite timing based on social cues captured via a microphone and a video camera pointed at the user's face. We will start by learning this model with able-bodied users since past work has laid the premise that replicating the way self-reliant people eat is the gold standard. We propose trying to generalize the learned model across users, and to users with disabilities. It might be that an individualized model needs to be learned per person, or it might be that a model learned across several users performs comparably or better than a user-triggered bite timing baseline. We will evaluate the performance through a user study with participants both with and without disabilities.

Challenge 4: Balancing assistance with independence; customizing shared control

In assistive robotics, abandonment is a big concern [Phillips and Zhao, 1993]. People want to feel like they are in control, but also want a system that will enable them to do more tasks without needing the assistance of another person. Different people also have different levels of control ability, based on their physical capabilities and the type of interface they use. In order to build a feeding system that will cater to all needs, both physically and psychologically, we propose having varying levels of autonomy over subparts of the feeding task.

We validate through user studies which aspects of dining should be controlled by the robot, by the operator, or shared (via mode-switching). To develop a useful assistive feeding system, a user-centric design is critical.

2
Background

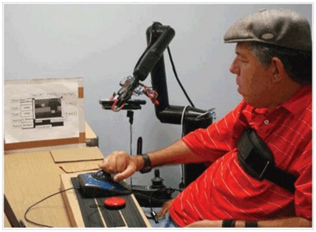
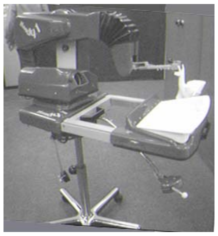
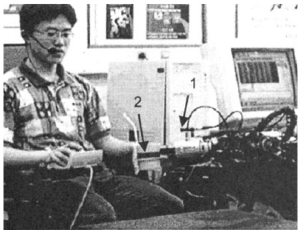
In this chapter we cover relevant background information to familiarize the reader with the domain of assistive robot arms. First, we introduce assistive robot arms that are commercially available or developed in research labs, summarize common control interfaces and their comparative strengths and weaknesses, and address how assistive robot arms are assessed and evaluated – particularly in the context of using shared control systems (section 2.1). Next, we present prior work that has been done in the context of food manipulation with robots (section 2.2) which is typically in the domain of food quality control testing. Finally, we discuss factors to consider in the context of assisted feeding with human caregivers (section 2.3.1), and summarize commercially available assisted dining tools (section 2.3.2).

2.1 Teleoperation of Assistive Robot Arms

We will present a brief summary of assistive robotic arms that are either available commercially or have been developed as research platforms at universities, and the interfaces that have been used to control them. A summary of some of the most influential assistive robot arm designs can be found in table 2.1.

2.1.1 Summary of Available Assistive Robot Arms

The majority of assistive arms use direct control, where the operator uses an interface to activate the actuators of the robot. Direct joint control is when the operator's interface directly controls the robot's joint position or velocity. Direct Cartesian control is when the operator's interface directly controls the position and orientation of the end effector of the robot, but does not explicitly choose how each joint of the robot will move to affect that Cartesian motion. Early assistive arms were controlled through direct joint control, but the

Manus ARM/iARM	Handy-1	KARES I	KARES II
			
• 6 DOFs • Commercial product by Exact Dynamics • Wheelchair-mounted • Controlled through wheelchair's input [Driessen et al., 2001, Gräser, 1998]	• Modes for feeding, cosmetics, and facial hygiene • Controlled by a single switch • Stationary base [Topping, 2002]	• 6 DOFs • Research prototype • Stereo camera in the hand • Controlled directly or given a visual target • Wheelchair-mounted [Bien et al., 2003]	• 6 DOFs • Research prototype • Stereo camera in the hand • Controlled by eye-mouse, haptic suit, or EMG • Wheelchair-mounted [Bien et al., 2003]

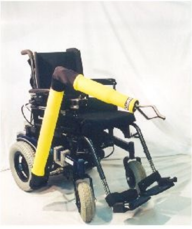
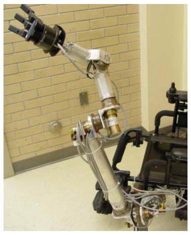
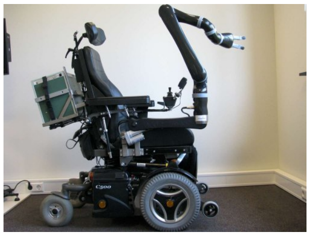
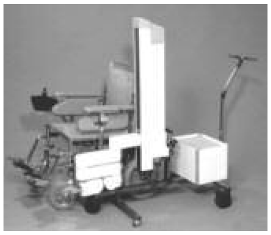
Raptor	WMRA	JACO	The Weston
			
• 4 DOFs • Commercial product by Applied Resources • Joint velocity commands • Controlled by joystick, 10 button keypad, or sip and puff • Wheelchair-mounted [Mahoney, 2001]	• 7 DOFs • Research prototype • Cartesian velocity commands • Controlled by joystick or BCI2000 [Alqasemi and Dubey, 2007, Edwards et al., 2006, Palankar et al., 2009a]	• 6 DOFs • Commercial product by Kinova Robotics • Controlled through wheelchair's input or separate joystick • Wheelchair-mounted [Campeau-Lecours et al., 2016]	• 4 DOFs • Research prototype • Controlled through cursor on screen with button menus • Wheelchair-mounted [Hillman et al., 2002a]

Table 2.1: Summary of influential assistive robot arms and their characteristics.

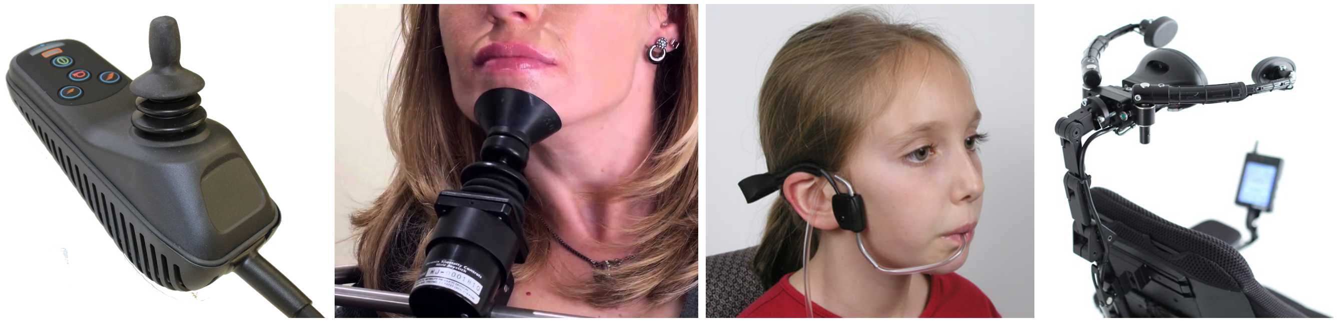
intuitiveness of Cartesian control was quickly recognized and added [Alqasemi et al., 2005]. Kumar et al. [1997] give an overview of the earliest assistive robot arms.

Since 2000, a number of wheelchair-mounted robot arms have become commercially available. The iARM (derivative of the well-known MANUS robot created by Exact Dynamics) is a 6 degree of freedom (DOF) robot arm with a parallel gripper. The JACO (produced by Kinova Robotics) is a 6 DOF robot with a parallel gripper that has either 2 or 3 fingers. Both robots can be controlled by a variety of assistive interfaces, can move automatically to prerecorded positions, and have a special mode for drinking.

Across available assistive robot arms, 6 degrees of freedom is becoming standard. The lower the degrees of freedom, the fewer actuators are needed, and the lower the cost. The higher the degrees of freedom, the greater the number of configurations for a given actuator position. The more configurations possible, the less likely it is that the arm will approach a singularity while using Cartesian control. Designers are converging towards 6 DOF designs because it balances cost and maneuverability.

Assistive robot arms can be designed as stationary workstations or to be mounted permanently or temporarily to an electric wheelchair. Typically due to weight and electricity requirements, wheelchair-mounted robot arms are limited to electric wheelchairs and are not generally designed to be attached to manual wheelchairs. Using a stationary workstation makes calibration of sensors easier and removes constraints on size and portability that would be required for a wheelchair-mounted arm. Using a wheelchair-mounted design increases the range of tasks that can be performed and greatly increases the robot's workspace by introducing the addition of the electric wheelchair's 2 degrees of freedom. As the hardware and electronics are getting smaller through technological advances, there is a shift towards using wheelchair-mounted robot arms.

Choosing a wheelchair-mounted design tightens the physical constraints placed on the robot. The robot-wheelchair combination must be small enough to fit through doorways. Additionally, people who use a wheelchair have a restricted ability to change their viewpoint. People with cervical SCI or multiple sclerosis (MS) may experience neck weakness, leading to a significantly decreased range of motion [LoPresti et al., 2000], and therefore a significant decreased range of viewpoint. With a stationary workstation, new vantage points can be achieved by moving the wheelchair base, but with an assitive robot arm attached to the wheelchair itself, new viewpoints w.r.t. the robot are only achieved limited to small torso and head movements to change their field of view.



(a) Joystick (b) Chin Joystick (c) Sip and puff (d) Head Array

Figure 2.1: A sip and puff interface, in which the operator controls a powered wheelchair through mouth-generated changes in air pressure. A head array interface by Permobil, in which the operator controls a powered wheelchair through switches mounted in the headrest.

2.1.2 *Summary of Assitive Control Interfaces*

We will present a summary of interfaces that have been used to control assistive robot devices. First we will examine interfaces for smart wheelchairs, as there is a large overlap between smart wheelchair interfaces and assistive robot interfaces since they face many of the same challenges. Then we will discuss how the interfaces have been modified or extended to include assistive robot arms.

The vast majority of electric wheelchair users (95%) use a joystick (or chin joystick), sip-and-puff, or head interface to control their wheelchair [Fehr et al., 2000]. The standard wheelchair joystick has two axes that can be controlled simultaneously, one for forward and backward speed, and the other for turning angle. Many attachments may be added to the joystick handle, such as a high friction ball or custom printed mold, to customize the joystick to each individual's comfort and physical abilities. A typical powered wheelchair joystick is shown in fig. 2.1a. A sip and puff interface is a straw-like device that can be activated by changing the air pressure with the operator's mouth. A sequence of puffs (where the operator blows into the straw) and sips (where the operator sucks air from the straw) generates a digital signal that is in turn interpreted by the wheelchair controller [Mougharbel et al., 2013]. A sip and puff interface is shown in fig. 2.1c. Head control is typically achieved through a series of switches mounted in a headrest that are activated by head movement [Kuno et al., 2003, Matsumotot et al., 2001]. An example of a head array used to control the wheelchair's motion is shown in fig. 2.1d.

More exotic interfaces exist, such as voice control [Cagigas and Abascal, 2004], eyegaze direction [Yanco, 1998], and EMG [Han et al., 2003], but have yet to be widely adopted. Less reliable control input techniques are being combined with obstacle-avoidance and other assistive algorithms to enable robust control simpson2005smart.

The wide variety of physical capabilities of power wheelchair users is what leads to such a large variety of input devices. The choice of which interface to use will vary on an individual basis. One thing that all the control interfaces have in common is two degrees of freedom. A wheelchair is effectively a mobile robot, one in which rotation and forward/backward motion are the two independent control signals. In the case of a head button array or a sip and puff interface, the signals are discrete. In the case of a joystick or eyegaze direction, the signals are continuous. While such interfaces are sufficient for controlling a 2 DOF system, they do not all directly scale to controlling a robot arm which can require 4-7 DOF control.

The commercially available robot arms (MANUS, Raptor and JACO) are controlled via a joystick by cycling through which axes are being controlled at a time – either joint axes or Cartesian axes with respect to the gripper. Some robot arms can save positions which the operator can access quickly via switches. A common example for wheelchair-mounted robot arms would be to save a position in which the robot is retracted to allow the operator's wheelchairs to fit through doorways.

Researchers have experimented with goal-oriented control, in which the operator indicates a goal object they would like to pick up, and the robot moves towards the goal object by itself. Providing goal-oriented control through a graphical interface has had mixed results [Tsui and Yanco, 2007, Laffont et al., 2009]. Task performance may show an increase over traditional interfaces, but it was not necessarily preferred.

Using a neural interface system to control a high DOF robot arm is a popular research domain, and great strides are being made in the brain-computer interface (BCI) technology. Speed, accuracy and reliability continue to be challenges in using BCIs in real-world applications [McFarland and Wolpaw, 2008]. BCIs typically classify the brain signals into a finite number of categories which are then translated to predefined robot arm motions or binned velocity commands (Examples: Hochberg et al. [2012], Palankar et al. [2009b], Onose et al. [2012]). The best parallel in conventional interfaces would be a head array, with potentially many more binary switches that require less physical effort to press. BCIs face the same challenge as traditional interfaces in controlling a high dimensional robot with a limited number of inputs.

2.1.3 *Evaluation Techniques*

Next, we will discuss evaluation techniques that have been used by other researchers and clinicians to evaluate control systems for assistive robot arms.

Task completion times with respect to an able-bodied user are often used for robotic manipulation applications, when there is no other baseline to use [Tsui et al., 2008b]. However, there is not a standard set of tasks used to assess assistive arm technology, so comparison across devices can be challenging. For example, Tijsma et al. [2005a] uses the following evaluation tasks: (1) Pick up an upside-down cup and place it right-side up on the table, (2) put two blocks into a box, (3) pick up two pens from out of sight and place them on the table. Maheu et al. [2011] uses: (1) grasping a bottle located at different locations, (2) pushing buttons of a calculator, (3) Taking a tissue from a tissue box, (4) grasping a straw, and (5) pouring a glass of water from a bottle. Chung et al. [2016] created a device with buttons and levers to mimic the activities of button presses and opening doors. One thing that is consistent across all the tasks used is that the evaluators are trying to get a cross-section of motions that are needed to perform activities of daily living. We chose to follow the evaluation process outlined in Tsui et al. [2008b], in which we met with an occupational therapist, discussed our goals, and found the best parallel to a preexisting clinical test. More details are provided in section 3.1.2.

In addition to objective functional evaluations of the performance of the assistive robot arm, other factors such as user friendliness, ease of operation, and effectives of input device must be taken into account [Romer and Stuyt, 2007]. In a survey of performance metrics for assistive robotic technology, researchers found that the performance metric should focus more on the end-user evaluations than on the performance of the robot [Tsui et al., 2008b]. The end-user evaluation has been interpreted as the mental effort [Tijsma et al., 2005a], independence [Chaves et al., 2003], trust and preference [Tsui et al., 2008b].

The choice of end user evaluation metrics comes from the unique requirements that assistive technology must have. Physical and mental fatigue can cause participants to stop using the robotic arm [Tijsma et al., 2005b], so it is important that the control interface and algorithms produce as little strain on the operator as possible. People with disabilities rely on helpers to perform physical actions, making it even more important to maintain situational autonomy, that is, the ability of a person to assign a goal when faced with a particular situation [Hexmoor, 2000]. Feeling in control over one's body and life, independence, and feeling of personal autonomy are among the most important attributes of assistive technology regardless of disability [Lupton and Seymour, 2000].

2.2 *Food Manipulation with Robots*

In this section, we will present a brief survey of robotics research done in the context of food manipulation: detecting and identifying food, grasping food, and cutting food.

Considerable work using computer vision with food detection and classification has been done in the context of quality assurance of a particular item. Algorithms have been designed for nuts, apples, oranges, strawberries, meats, cheese, and even pizza [Brosnan and Sun, 2002]. Each algorithm is tailored to the specific domain. A summary of the type of features that have been successfully used in these targeted applications is given in [Du and Sun, 2006] and summarized in table 2.2. While these features were successful in targeted applications, they may not be sufficient to distinguish *between* types of food. Take for example a grain detector [Ding and Gunasekaran, 1994] which uses shape, center, and orientation of individual kernels of grain. These features would not successfully distinguish between a round nut and a round pea of similar shape and size.

Characterization	Products
Area	Apple
Hinge	Oyster
Color	Apple, Citrus, Lemon, Mandarin, Barley, Oat, Rye, Wheat, Bell pepper, Muffin
Morphological Features	Apple, Corn, Edible bean, Rye, Barley, Oat, Wheat
Textural features	Barley, Oat, Rye, Wheat, Edible bean
Spectral Images	Tomato, Poultry carcass
Hue histograms	Apple, Potato
Gradient magnitude	Raisin, Asparagus
Curvature	Carrot
Edges	Asparagus

Table 2.2: Summary of features used for quality assurance of different food products.

Some recent work has tried to classify images of food into types of dishes, usually for the purpose of determining calorie counts [Bossard et al., 2014, Zheng et al., 2017, Sudo et al., 2014, Oliveira et al., 2014]. What follows are potential features or feature-extraction techniques that have been used in this domain. While we want to manipulate food, food classification is a complementary task, since the type of food is likely correlated with the way in which to manipulate it.

Just as there are several options for food features, there are many learning algorithms that have been applied to the food quality assurance problem [Du and Sun, 2006]. We will formulate this as a supervised learning problem, where the labels are obtained from robot executions and results. More details on our approach can be found in chapter 4.

In addition to quality control, some recent work has been focused on robot manipulation for cooking or preparing food for consumption. Cutting for example, has been achieved through through optical and tactile feedback for foods with different hardnesses ranging from apples to bananas [Yamaguchi and Atkeson, 2016]. Robot that can cook are usually specialized specifically for that purpose and require ingredients to be placed in preselected bins or containers [Ma et al., 2011, Bollini et al., 2011, Sugiura et al., 2010]. While it's not directly relevant to food manipulation, physical properties of food have been examined through a mastication robot which emulates the motion of the human jawbone during chewing [Xu et al., 2008].

2.3 *Assisted Feeding*

In this section, we describe the impact of dining with a caregiver on a person with disabilities. We discuss some social considerations that are relevant to this domain and summarize commercially available eating aides to be used in place of or in tandem with a human caregiver.

2.3.1 *Assisted Feeding with a Human Caregiver*

Dining together is a cornerstone of society and provides a personal link to the wider community that attests to the shared understanding that underpins much of our routine food consumption [Marshall, 2005]. Dually, social phenomena impact population eating patterns [Delormier et al., 2009]. The ritual can be a very complex process, with remarkably similar table manners both historically and the world over [Visser, 2015]. Dining habits have a particularly high impact on the morale of those with disabilities. In the example of stroke patients, the loss of meal-related autonomy may threaten their hope for the future, whereas hope returns when meals become easier [Jacobsson et al., 2000]. Prior work has identified self-feeding as a highly sought-after ability for people with disabilities [Prior, 1990, Stanger et al., 1994].

Figure 2.2: A caregiver assisting with feeding.

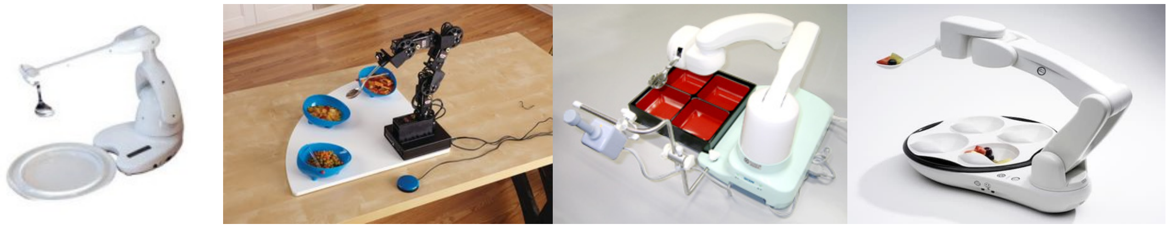
In addition to raising the self-worth of people with disabilities, independent dining might have a considerable impact on caregiver hours. Feeding is one of the most time consuming tasks for caregivers [Chiò et al., 2006]. Training of new caregivers is particularly important and difficult with respect to feeding. Initially, the disabled person would have to explicate all that they wanted the helper to do and the technical aspects of execution, and they could not enjoy the food while constantly focusing on the meal procedure [Martinsen et al., 2008].

Eating is a complex process requiring sensitive coordination of a number of motor and sensory functions. When a person has to rely on assisted feeding, meals require that patient and caregiver coordinate their behavior [Athlin et al., 1990]. In order to achieve this subtle cooperation, the people involved must be able to initiate, perceive, and interpret each other's verbal and non-verbal behavior. The main responsibility for this cooperation lies with caregivers, whose experiences, educational background and personal beliefs may influence the course of the mealtime [Athlin and Norberg, 1987]. However, personal wishes must be communicated to such an extent that the caregiver can consider the diner's needs [Martinsen et al., 2008].

Figure 2.3: Complex social cues and interactions occur during meals, which extend to assisted feeding situations.

A qualitative study by Martinsen et al. [2008] on the topic of eating with a caregiver found that while assisted feeding requires the construction of a new eating pattern, it is a careful creation of a new schema which uses conventions among self-reliant people as a frame of reference. "The goal of the interaction is to as closely as possible replicate the meal experience from before the disability, when the eating pattern was independent of conscious reflection and not possible to articulate." This supports the premise of using insights learned from able-bodied people's eating patterns to inform the way an assistive feeding device should behave. Even the physical properties of the device should deviate as little as possible from the way self-reliant people eat, as the study found that people with disabilities do not wish to draw attention to their dependency on help from others. This supports using a silverware-like utensil and human-like robot motion to avoid drawing undo attention to the device.

The same study found that privacy becomes a concern when having a caregiver present during dinner with friends or relatives. People have to consider whether they prefer assistance from the caregiver or from a relative. One study participant stated: "So far, it is a great strain not to be able to sit and talk with your friends as you used to do." Using a robotic feeding device would allow for increased privacy and intimacy during mealtime. However, personalization to each individual is crucial to the feeder-receiver relationship [Martinsen et al., 2008], and should be incorporated into the control of the assistive feeding device.



(a) Meal-Mate$^{\text{TM}}$ (b) Meal Buddy (c) My Spoon (d) Obi

Figure 2.4: Four assistive feeding devices currently on the market.

2.3.2 *Commercially Available Eating Aides*

Several specialized feeding devices for people with disabilities have come onto the market in the past decade, some of which are shown in fig. 2.4, and they all work in a similar manner.

The Meal Buddy Assistive Feeder is made by Patterson Medical and features three bowls of food that are rigidly attached to the base of a robot arm. A button is used to select which bowl, and to initiate bite collection.

More information on the Meal Buddy Assistive Feeder at `http://www.pattersonmedical.com/`

My Spoon is made by Secom, and features a robot arm with four bowls for food and an actuated silverware end effector, that will open and close to grasp food before presenting it to the user. Since being introduced in 2002, researchers have expended the system's original interface to vary the level of control [Soyama et al., 2004]. In manual mode, the operator uses a joystick to position the gripper above the piece of food, then presses a button to have the fork/spoon collect the food. In semi-automatic mode, the operator uses the joystick to select one of the four containers and then the robot picks up pieces of food in a predefined order that have been laid out in a grid in each of the bowls. In automatic mode, the operator presses a button and is given a piece of food in a predetermined sequence. In all modes with autonomy, the food is placed in predefined locations and not detected visually or otherwise.

More information on My Spoon at `http://www.secom.co.jp/english/myspoon/`

The Obi is made by Desin and features four bowls of food that are rigidly attached to the base of a robot arm. The Obi also has a "teach" mode where a caregiver can place the robot at a position near the mouth that will be remembered when the user triggers a bite via button press.

More information on the Obi at `https://meetobi.com/`

Meal-Mate$^{\text{TM}}$ Eating Device is made by RBF Healthcare uses a spoon attached to a robot arm, which will move down to a plate and back up with the press of a button, but while the spoon is on the plate, a joystick or arrow buttons can be used to move the spoon along the plate.

More information on Meal-Mate$^{\text{TM}}$ at `http://rbfindustries.co.uk/healthcare/`

All these feeding devices are designed only for food manipulation, and require specialized food containers to function effectively. With the exception of Meal-Mate$^{\text{TM}}$, all the feeding devices are controlled with buttons to select which food bowl and then to trigger a bite action, with control over the robot's motion. Similarly, none of these devices have any way to sense success or failure of taking bites, nor a way to automatically time when to provide bites to the operator.

3
Teleoperation of Assistive Robotic Manipulators

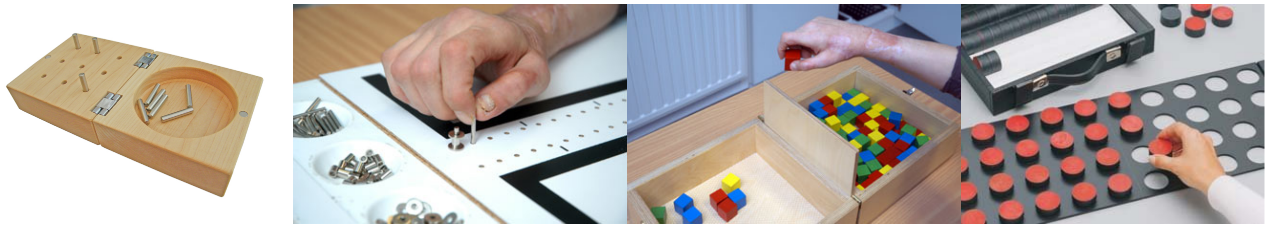
In this chapter, we discuss challenges and pitfalls common to all teleoperated assistive robot manipulators. We introduce modal control and mode-switching strategies. We generate a model to predict mode switching occurrences, and evaluate automatic mode switching as an assistive control strategy. Finally, we discuss generalization of strategies to mitigate mode switching to other tasks and assistive devices such as prosthetics.

3.1 Exploration of Modal Control

In this section, we introduce modal control and describe a study in which we identify and quantify the challenges of teleoperating an assistive robot with a modal control interface.

3.1.1 Kinova Interviews

Our own interviews with current users of the Kinova arm pinpointed that the struggles with modal control relate back to the need to constantly change modes. Our users found switching between the various control modes, seen in fig. 1.3, to be slow and burdensome, noting that there were "a lot of modes, actions, combination of buttons". Each of these mode changes requires the user to divert their attention away from accomplishing their task to consider the necessary mode change [Tijsma et al., 2005a,b]. The cognitive action of shifting attention from one task to another is referred to as *task switching*. Task switching slows down users and can lead to increased errors regardless of the interface they are using to make this switch [Monsell, 2003, Wylie and Allport, 2000, Meiran et al., 2000, Strobach et al., 2012, Arrington and Logan, 2004]. Simply the need to change modes is a harmful distraction that impedes efficient control.



(a) 9 Hole Peg Test (b) Purdue Pegboard Test (c) Box and Block Test (d) Minnesota Test

Figure 3.1: Functional tests that repeat specific motion primitives.

3.1.2 *Inventory of OT Manipulation Metrics*

There are a wide variety of evaluation techniques used for wheelchair-mounted assistive robotic manipulators. A literature review of end-user evaluations [Chung and Cooper, 2012] revealed that task completion time and task completion rate were the most used evaluation metric, but the tasks being used were different across researchers and platforms leading to difficult comparisons. In one Manus arm evaluation, there were no predefined set of tasks, but rather the user was allowed to use the arm naturally about their environment. Afterward, the common tasks across users were inventoried and evaluated [Eftring and Boschian, 1999]. Another study with the Manus was evaluated by having users grasp the following objects: remote controller, a cereal box, two jars, a soda can, and a water bottle [Kim et al., 2012]. A different study with the Manus was tested with the task of grasping a foam ball that was hanging on a string [Tsui and Yanco, 2007]. While each task is reasonable, it is very difficult to draw comparisons between the studies, even though all are done on the same system.

There are advantages to evaluating our assistive robot arm system with the same standardized tests that Occupational Therapists (OT) use with human patients. Standard OT tests are already validated, have years of application, provide a metric that can be used across robots and people, and makes our results more transparent to a larger community. Additionally, this would allow for a larger interaction with insurance companies or healthcare economists [Mahoney, 1997].

Manipulation tests can be divided into strength tests, which measure the muscular ability of the human hand and upper limbs, and functional tests, which measure the ability to perform particular tasks or motions. Because the robot's hardware will exclusively determine its performance on strength tests and will not change drastically over time, evaluations using functional tests make more sense for comparing across robot and control platforms.

A survey of functional tests shows that functional tests fall into two categories: repeated tasks that test specific motion primitives, and tasks that are based on daily living and cover a several motion primitives per task. The Purdue Pegboard test [Tiffin and Asher, 1948], 9 Hole Peg Test [Sunderland et al., 1989], Box and Block Test [Mathiowetz et al., 1985], and Minnesota rate of manipulation test, shown in fig. 3.1, ask the user to provide a number of short repeated tasks and use the overall task time as the primary evaluation metric. Tests based on daily activities include the Motor Activity Log (MAL) [Taub et al., 2011, Uswatte et al., 2005], the Jebsen Taylor Hand Function Test [Tipton-Burton, 2011], the Action Research Arm (ARA) test [McDonnell, 2008], Sollerman hand function test [Sollerman and Ejeskär, 1995], and the Chedoke Arm and Hand Activity Inventory (CAHAI) [Barreca et al., 2004].

Our prior work has used tasks from the CAHAI test, which has the advantage of including both qualitative and quantitative metrics for evaluation and having tasks that can be easily modified to be performed with one hand. It is optimized for evaluating stroke recovery, which is not the target population for this work. We argue that spinal cord injury patients *with an assistive robot manipulator* is more similar to the situation of a stroke patient, who has varying levels of control over their limb.

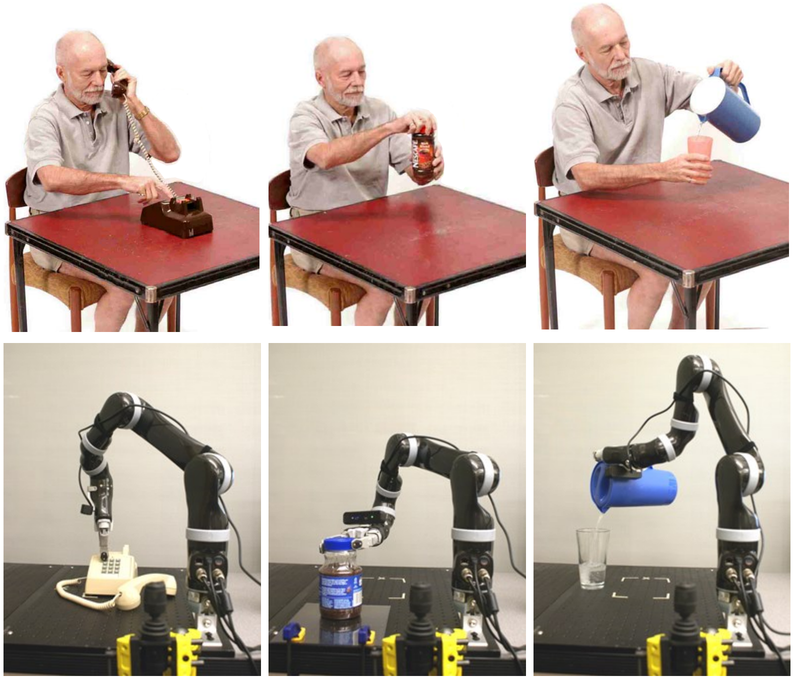
The CAHAI test is performed in a controlled environment with a specific set of tasks and items. This makes it easy to compare across users and practical to administer, but it may suffer from not being the most representative set of tasks to express how the robot would be used by each person. If we were performing a field study during typical operational use, we would see a much greater variety of tasks and could measure effects that take a longer time to develop such as self-efficacy, changes in caregiver hours, and muscle fatigue. However, in longitudinal studies it can be difficult to compensate for diminishing mental or physical capacities.

3.1.3 Chedoke Tasks with the Robot with Able-bodied Users

To objectively measure the impact of mode switching, we ran a study with able-bodied users performing household tasks with the Kinova MICO arm using a joystick interface.

Experimental Setup Users sat behind a table on which the MICO arm was rigidly mounted. They used the standard Kinova joystick to control the arm.

Tasks The tasks we chose are modified from the Chedoke Arm and Hand Activity Inventory (CAHAI), a validated, upper-limb measure to assess functional recovery of the arm and hand after a stroke [Barreca et al., 2004]. The CAHAI has the advantage of drawing tasks from instrumental activities of daily living, which are representative of our desired use case, rather than occupational therapy assessment tools such as the 9-Hole Peg Test [Kellor et al., 1971] or the Minnesota Manipulation Test [Cromwell, 1960] that also evaluate upper extremity function, but do not place the results into context.



(a) Telephone (b) Coffee (c) Pouring water

Figure 3.2: Three modified tasks from the Chedoke Arm and Hand Activity Inventory, which able-bodied users performed through teleoperating the MICO robot.

Manipulated Factors We manipulated which task the user performed. The three tasks we used were: calling 911, pouring a glass of water from a pitcher, and unscrewing the lid from a jar of coffee. These three tasks were chosen from the CAHAI test because they could easily be modified from a bimanual task to being performed with one arm. The three tasks are shown in fig. 3.2.

Procedure After a five minute training period, each user was given a maximum of ten minutes per task. The order of tasks was counterbalanced across the users. The joystick inputs and the robot's position and orientation were recorded throughout all trials. After all the tasks were attempted, we asked the users to rate the difficulty of each task on a 7-point Likert scale and what aspects make performing tasks difficult with this robot.

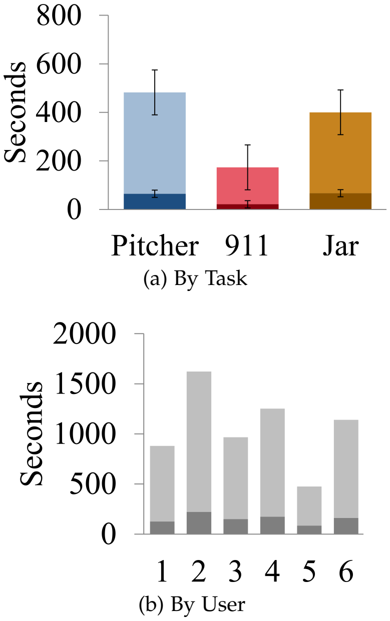
Participants and Allocation We recruited 6 able-bodied participants from the local community (4 male, 2 female, aged 21-34). This was a within subject design, and each participant performed all three tasks with a counterbalanced ordering.

Analysis On average, 17.4 ± 0.8% of task execution time is spent changing control modes and not actually moving the robot. The mode changing times were calculated as the total time the user did not move the joystick before and after changing control mode. The fraction of total execution time that was spent changing modes was fairly consistent both across users and tasks as seen in fig. 3.3. If time spent changing mode could be removed, users would gain over a sixth of the total operating time.

The tasks the users performed were reported to be of unequal difficulty. Users responded that the pitcher pouring was the most difficult task (M=5.5,SD=0.7), followed by unscrewing the jar (M=5.2,SD=0.7), and the easiest task was dialing 911 (M=4,SD=0.6). The total execution time shown in fig. 3.3 mirrors the difficulty ratings, with harder tasks taking longer to complete. Difficulty could also be linked to the number of mode switches, mode switching time, or ratio of time spent mode switching, as shown in fig. 3.4. The hardest and easiest tasks are most easily identified when using switching time as a discriminating factor. The pitcher and jar tasks both rated as significantly more difficult than the telephone task, which may be due to the large number of mode changes and small adjustments needed to move the robot's hand along an arc — as one user pointed out: "Circular motion is hard."

One might argue that we are basing our findings on novice users, and their discomfort and hesitation switching modes will diminish over time. However, over the course of half an hour using the arm, and an average of more than 100 mode switches, users did not show any significant decrease in the time it takes to change mode (fig. 3.5). The continued cost of mode switching is further supported by our interviews, in which a person using the JACO for more than three years stated "it's really hard with the JACO because there are too many mobilizations and too many movements required."

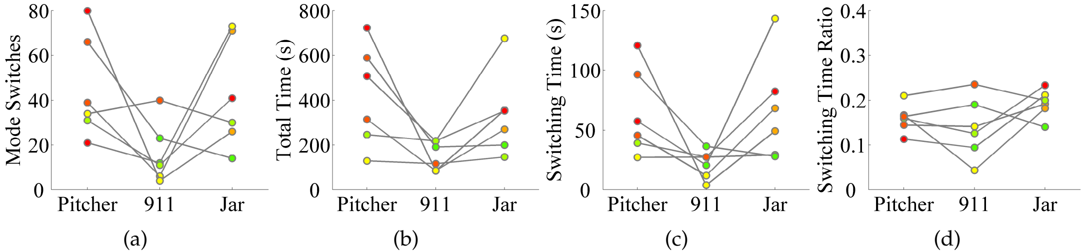
The users had three possible modes and used two buttons on the top of the joystick to change between them. The left button started translation mode, the right button started orientation mode, and pressing both buttons simultaneously started gripper mode. Changing into gripper mode was particularly since the timing between the two buttons had to be very precise lest the user accidentally press



(a) By Task

(b) By User

Figure 3.3: Execution time, with time spent mode switching shown in the darker shades.



(a)

(b)

(c)

(d)

Figure 3.4: The connected components are a single user, and the colors represent the difficulty that user rated each task with red being most difficult, yellow being second most difficult, and green being least

	Translation	Wrist	Finger
Translation	-	1.98 ± 0.15s	1.94 ± 0.16s
Wrist	2.04 ± 0.51s	-	3.20 ± 1.85s
Finger	1.30 ± 0.13s	0.98 ± 0.24s	-

Table 3.1: Mean mode switching times in seconds.

the left or right button when releasing and switch to translation or wrist mode. The cost to change from one mode to another was not constant across the modes; table 3.1 shows the average time it took to change from the mode in the row to the mode in the column. While in this case the difference can be explained by the chosen interface, it could be important to consider if switching from one particular control mode to another causes a larger mental shift in context. Such differences would require the cost of mode switches to be directional, which we leave for future work.

3.2 *Time-Optimal Mode Switching Model*

The users of the JACO arm identified that frequently changing modes was difficult. We objectively confirmed the difficulty of mode changing by having able-bodied users perform everyday tasks with

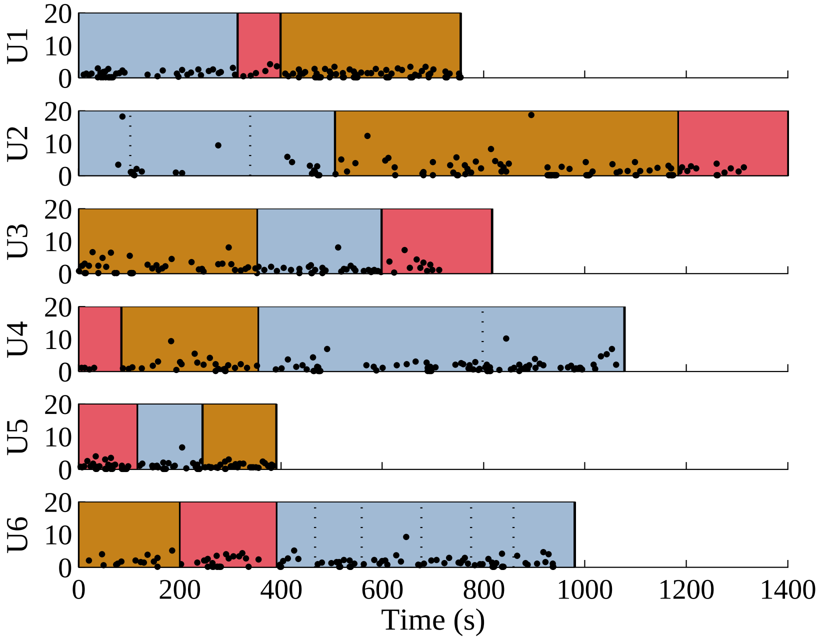


Figure 3.5: Each point is a mode switch, with the y-value indicating the mode switching time, and the x-value indicating when the mode switch occurred. The colors correspond to the task (blue: pouring pitcher, brown: unscrew jar of coffee, red: dial 911), and the order of tasks can be seen for each user as arranged from left to right. Dashed lines in the pitcher task identify locations where the user dropped the pitcher and the task had to be reset.

the MICO arm. Having identified mode switching as a problem in this complex scenario, we tried to model the problem in a much simpler scenario and provide the foundations for scaling the solution back up to the full space of the MICO arm.

3.2.0.1 Study 2: 2D Mode Switching Task

Study 1 demonstrated that people using modal control spend a significant amount of their time changing modes and not moving the robot. The next step is to model when people change modes so that the robot can provide assistance at the right time. We identified certain behaviors from Study 1 that could confound our ability to fit an accurate model. We observed that different people used very different strategies for each of the tasks, which we postulated is because they were performing multi-step tasks that required several intermediate placements of the robot's gripper. In some trials, users changed their mind about where they wanted to grab an item in the middle of a motion, which we could detect by the verbal comments they made. To gather a more controlled set of trajectories under modal control, we ran a second study in which we more rigidly defined the goal and used only two modes. To fully constrain the goal, we used a simulated robot navigating in two dimensions and a point goal location. We kept all the aspects of modal control as similar to that of the robot arm as possible. Using a 2D simulated robot made it simpler to train novice users and removed confounds, allowing us to more clearly see the impacts of our manipulated factors as described below.

Experimental Setup In this study, the users were given the task of navigating to a goal location in a planar world with polygonal obstacles. We had each user teleoperate a simulated point robot in a 2D world. There were two control modes: one to move the robot vertically, and one to move it horizontally. In each mode, the users pressed the up and down arrow keys on the computer keyboard to move the robot along the axis being controlled. By using the same input keys in both modes, the user is forced to re-map the key's functionality by pressing the spacebar. This is a more realistic analogy to the robot arm scenario, where the same joystick is being used in all of the different control modes to control different things.

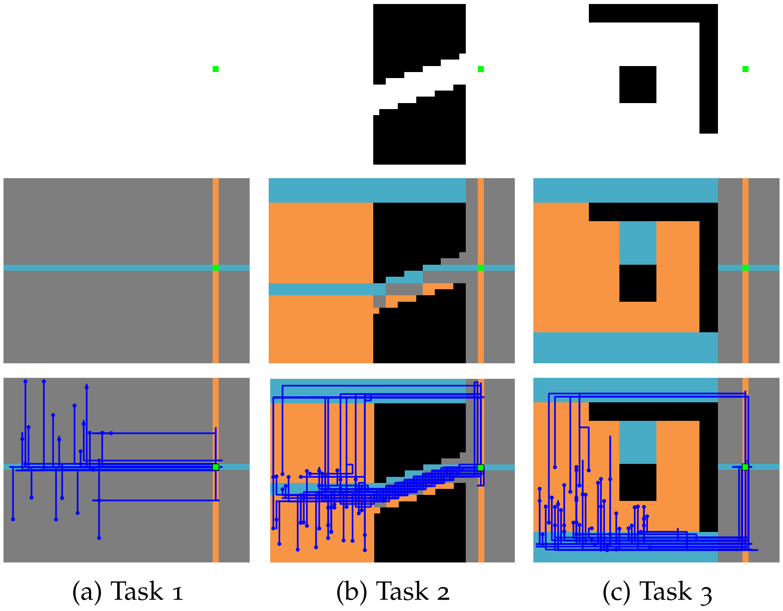


Figure 3.6: Top row: Three tasks that the users performed with a 2D robot. The green square is the goal state and the black polygons are obstacles. Middle row: regions are colored with respect to the optimal control mode; in blue regions it is better to be in x mode, in orange regions it is better to be in y mode, and in gray regions x and y mode yield the same results. Bottom row: user trajectories are overlaid on top of the decision regions, illustrating significant agreement.

Manipulated Factors We manipulated two factors: the delay when changing modes (with 3 levels) and the obstacles in the robot's world (with 3 levels). To simulate the cost of changing modes, we introduced either no delay, a one second delay, or a two second delay whenever the user changed modes. Different time delays are analogous to taking more or less time to change mode due to the interface, the cognitive effort necessary, or the physical effort. We also varied the world the robot had to navigate in order to gather a variety of different examples. The three tasks are as follows: (1) an empty world, (2) a world with concave and convex polygons obstacles, and (3) a world with a diagonal tunnel through an obstacle, and are shown in the top row of fig. 3.6. The nine randomized conditions are summarized in table 3.2.

	Task 1	Task 2	Task 3
No delay	Task 1, No delay	Task 2, No delay	Task 3 No delay
1 sec. delay	Task 1, 1 sec. delay	Task 2, 1 sec. delay	Task 3, 1 sec. delay
2 sec. delay	Task 1, 2 sec. delay	Task 2, 2 sec. delay	Task 3 2 sec. delay

Table 3.2: Within Subjects Conditions for 2D Mode Switching Study.

Procedure This was a within subjects study design. Each user saw only one task, but they saw all three delay conditions. Each user had a two trial training period with no delay to learn the keypad controls, and then performed each of the three delay conditions twice. Five users performed each task. The goal remained constant across all the conditions, but the starting position was randomly chosen within the bottom left quadrant of the world. We collected the timing of each key press, the robot's trajectory, and the control mode throughout each of the trials.

Measures To measure task efficiency, we used three metrics: the total execution time, the number of mode switches, and the total amount of time switching modes. We also recorded the path the user

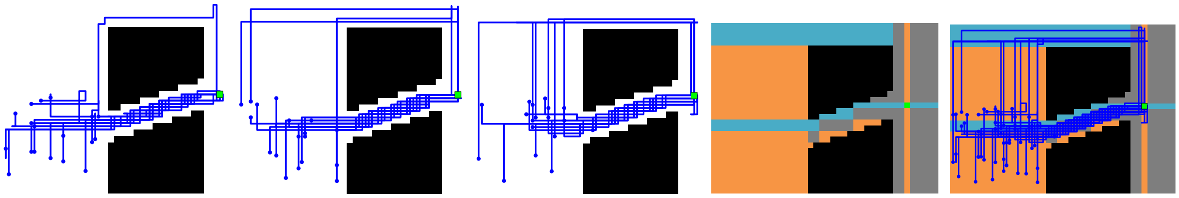


Figure 3.7: User strategies for Task 2 are shown via their paths colored in blue. As the delay increases, some users choose to go around the obstacles rather than through the tunnel, to avoid switching mode. There is still significant agreement with the time-optimal strategy.

moved the robot and which control mode the robot was in at each time step.

Participants We recruited 15 participants aged 18-60. While this is a fairly small sample, all subjects saw all conditions, and this was primarily an exploratory study to examine how increasing the cost of mode switching qualitatively impacts user strategies.

Analysis When the cost of changing modes increases, people choose different strategies in particular situations. This is best seen in Task 2, where there were two different routes to the goal, whereas in Task 1 and Task 3 the map is symmetrical. When there was no mode delay, nearly all users in Task 2 navigated through the tunnel to get to the goal, fig. 3.7. When the delay was one second, some users began to navigate around the obstacles completely, and not through the tunnel. While navigating the tunnel was a shorter Euclidean distance, it required more mode changes than navigating around the tunnel entirely. Therefore we saw that when the cost of mode changes increased, people were taking paths that reduced the number of mode switches.

We noticed that the user trajectories could be very well modeled by making the assumption that the next action they took was the one that would take them to the goal in the least amount of time. Since switching modes is one of the possible actions, it becomes possible to use this simple model to predict mode switches. The next section discusses the time-optimality model in more detail.

3.2.1 *Time-Optimal Mode Switching*

The time-optimal policy was found by assigning a cost to changing mode and a cost to pressing a key. These costs were found by empirically averaging across the time it took the users from Study 2 to perform these actions. Using a graph search algorithm, in our case Dijkstra's algorithm [Dijkstra, 1959], we can then determine how much time the optimal path would take. By looking at each (x,y) location, we can see if the optimal path is faster if the robot is in x-mode or y-mode. The time-optimal mode for each particular (x,y) location is the mode which has a faster optimal path to the goal. A visualization of the optimal mode can be seen in fig. 3.6 for each of the tasks. Time-optimal paths change into the optimal mode as soon as the robot enters one of the x-regions or one of the y-regions. By plotting the user trajectories over a map of the regions, we can see where users were suboptimal. If they were moving vertically in the x-region or horizontally in the y-region, they were performing suboptimally with respect to time.

In Task 1, users were in the time-optimal mode 93.11% of the time. In Task 2, users were in the time-optimal mode 73.47% of the time. In Task 3, users were in the time-optimal mode 90.52% of the time. Task 2 and Task 3 require more frequent mode switching due to the presence of obstacles.

3.3 *Evaluation of Automatic Mode Switching*

Once we determined that people often switch to be time-optimal, we tested how people would react if the robot autonomously switched modes for them. Using the same tasks from Study 2, we used the time-optimal region maps (fig. 3.6), to govern the robot's behavior.

	No Delay	Delay
Manual	Manual, No delay	Manual, Delay
Automatic	Automatic, No delay	Automatic, Delay
Forced	Forced, No delay	Forced, Delay

Table 3.3: Within Subjects Conditions for Time-Optimal Mode Switching Study.

3.3.0.1 Study 3 : 2D Automatic Mode Switching

Manipulated Factors We manipulated two factors: the strategy of the robot's mode switching (with 3 levels) and the delay from the mode switch (with 2 levels) as summarized in table 3.3. The mode switching strategy was either manual, automatic or forced. In the manual case, changing the robot's mode was only controlled by the user. In the automatic case, when the robot entered a new region based on our optimality map, the robot would automatically switch into the time-optimal mode. This change would occur only when the robot first entered the zone, but then the user was free to change the mode back at any time. Within each region in the forced case, by contrast, after every action the user took, the robot would switch into the time-optimal mode. This meant that if the user wanted to change to a suboptimal mode, they could only move the robot one step before the mode was automatically changed into the time-optimal mode. Hence the robot effectively forces the user to be in the time-optimal mode.

Similar to Study 2, we had a delay condition, however we considered the following three cases: (1) no delay across all assistance types, (2) a two second delay across all assistance types, and (3) a two second delay for manual switching but no delay for auto and forced switching. The purpose of varying the delay was to see if the users' preference was impacted equally by removing the imposed cost of changing mode (delay type 3), and by only removing the burden on the user to decide about changing mode (delay type 1 and 2).

Hypotheses

H1: People will prefer when the robot provides assistance.

H2: Forced assistance will frustrate users because they will not be able to change the mode for more than a single move if they do not accept the robot's mode switch.

H3: People will perform the task faster when the robot provides assistance.

Procedure After giving each user two practice trials, we conducted pairs of trials in which the user completed the task with the manual mode and either the forced or automatic mode in a counterbalanced randomized order. Testing the automatic assistance and forced assistance across the three delay conditions led to six pairs. For each pair, users were asked to compare the two trials on a forced choice 7-point Likert scale with respect to the user's preference, perceived task difficulty, perceived speed, and comfort level. At the conclusion of the study, users answered how the felt about the robot's mode switching behavior overall and to rate on a 7-point Likert scale if the robot changed modes at the correct times and locations.

Participants We recruited 13 able-bodied participants from the local community (7 male, 6 female, aged 21-58).

Analysis People responded that they preferred using the two types of assistance significantly more than the manual control, $t(154) = 2.96$, $p = .004$, supporting **H1**. The users' preference correlated strongly with which control type they perceived to be faster and easier (R=0.89 and R=0.81 respectively).

At the conclusion of the study, users responded that they felt comfortable with the robot's mode switching (M=5.9, SD=1.0), and thought it did so at the correct time and location (M=5.7, SD=1.8). Both responses were above the neutral response of 4, with t(24)=4.72, $p < .001$ and t(24)=2.34, $p = .028$ respectively. This supports our finding that mode switching can be predicted by following a strategy that always places the robot in the time-optimal mode.

Since this was an experiment, we did not tell participants which trials the robot would autonomously changing modes in. As a result, the first time the robot switched modes automatically, many users were noticeably taken aback. Some users immediately adjusted, with one saying "even though it caught me off guard that the mode automatically switched, it switched at the exact time I would have switched it myself, which was helpful". While others were initially hesitant, all but two of the participants quickly began to strongly prefer when the robot autonomously changed for them, remarking that it saved time and key presses. They appreciated that the robot made the task easier and even that "the program recognized my intention".

Over time people learned where and when the robot would help them and seemed to adjust their path to maximize robot assistance. People rarely, if ever, fought against the mode change that the robot performed. They trusted the robot's behavior enough to take the robots suggestions [Mead and Matarić, 2009, Baker and Yanco, 2004, Kubo et al., 2009]. We found no significant difference between the forced and automatic mode switching in terms of user preference $t(76) = 0.37, p = 0.71$. Some users even stated that there was no difference at all between the two trials. Therefore we did not find evidence to support **H2**.

Task efficiency, measured by total execution time and total time spent changing modes (as opposed to moving the robot), was not significantly different between the manual control, auto switching, and forced switching conditions. Therefore we were not able to support **H3**. However, this is not surprising as the assistance techniques are choosing when to switch modes based on a model that humans already closely follow. It follows that the resulting trajectories do not differ greatly in terms of path length nor execution time.

4
Robotic Food Manipulation

In this chapter we construct a robotic system for the purpose of food manipulation through the use of a commercially-available assistive robot arm and selected sensors. We describe and compare different methods of scanning plates of food and develop a taxonomy of food collection strategies. We train a neural network to determine and rank potential bite locations for the robot to autonomously acquire food morsels. Finally, we evaluate the model and discuss how to generalize to other foods and acquisition strategies.

4.1 Preparing the Robot for Food Manipulation

The Kinova MICO robot has a two-fingered, underactuated, parallel gripper. While the tips of the two fingers can touch to hold objects smaller than a centimeter across, it is designed to hold larger objects or surfaces such as a glass or a doorknob. Directly manipulating food is not recommended due to concerns of hygiene, since it would be difficult to adequately clean the fingers after use (the robot is not waterproof). Holding a fork in the robot's grippers would also be a problem since the tips of the fingers are fairly slippery and do not have a large contact area with the fork, thus risking dropping the fork when it comes into contact with food or the plate. We specifically avoid making the decision to rigidly mount the fork to the robot's end effector because the robot is an assistive tool providing a wide array of manipulation abilities, such as pressing buttons and opening doors, that having a permanently mounted fork would interfere with. Instead, we use a 3D printed fixture, supplied by Kinova Robotics and shown in fig. 4.1, which the robot can grasp and release as desired by the robot operator. Inside the fixture, a fork (or other utensil) can be affixed via a hand-tightened bolt.

We considered whether to use two robot arms during food manipulation or constrain our design to a single arm. Using two robots would enable bimanual manipulations such as cutting while holding a piece of food, or pushing food with one utensil onto another utensil. Two robots would however increase the overall cost of the system two-fold, decrease maneuverability of the overall wheelchair-robot system, and at least double the control complexity for direct teleoperation. While bimanual assistive robot systems have been attempted for the purpose of specialization [Song et al., 2010], we determined a single robot arm to be capable for an acceptable percentage of food acquisition and feeding needs.

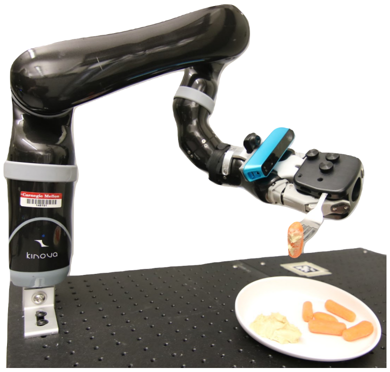


Figure 4.1: Adaptations of the MICO robot to enable food manipulation.

To add sensing to the arm, we mounted an Occipital Structure Sensor to the robot's wrist to gather depth data. When determining mounting location, we decided that mounting it on the robot's proximal link would enable a greater range of viewpoints, which would be of particular value to the robot's intended operators who have limited ability to change their own vantage point. We mounted the sensor slightly above the gripper to avoid occlusions from small objects in the hand, such as the silverware fixture. We oriented the sensor to be aligned with the robot's gripper as it makes an approach to an object. In retrospect, it could have been useful to point the sensor along the fork's axis, which is perpendicular to the gripper axis, in order to get real time feedback while acquiring a bite. With the mounting location we chose, the robot can examine a plate of food, then perform an acquiring action, but has no visual feedback while the robot is interacting with the plate via the fork.

When selecting a depth sensor, we cared a great deal about the minimum range, as the MICO robot arm only has a reach of 0.7 meters, and would often be mounted at a height equal or below the tabletop. Even with the Occipital Structure Sensor, we were reaching its minimum range specifications, so eventually switched it for an Orbbec Astra S, which has a minimum range of 0.4 meters.

4.2 *Food Acquisition Strategies*

We began our investigation of food acquisition by observing how humans eat. We limited our observations to able-bodied subjects using a fork. We don't consider methods to acquire food that use hands, chopsticks, spoons, knives, or other utensils. One reason for this is that most foods can be eaten with a fork, even if they are primarily eaten in another way (with the exception of dilute soups or broths). Another is that in North America, use of the fork is more common than, say, use of chopsticks, and therefore more accessible for us to study.

We recorded video of participants eating a variety of foods with only a fork. Foods we provided include beef, broccoli, rice, salad, chicken pieces, cheese, cut fruit, and noodles. During our initial

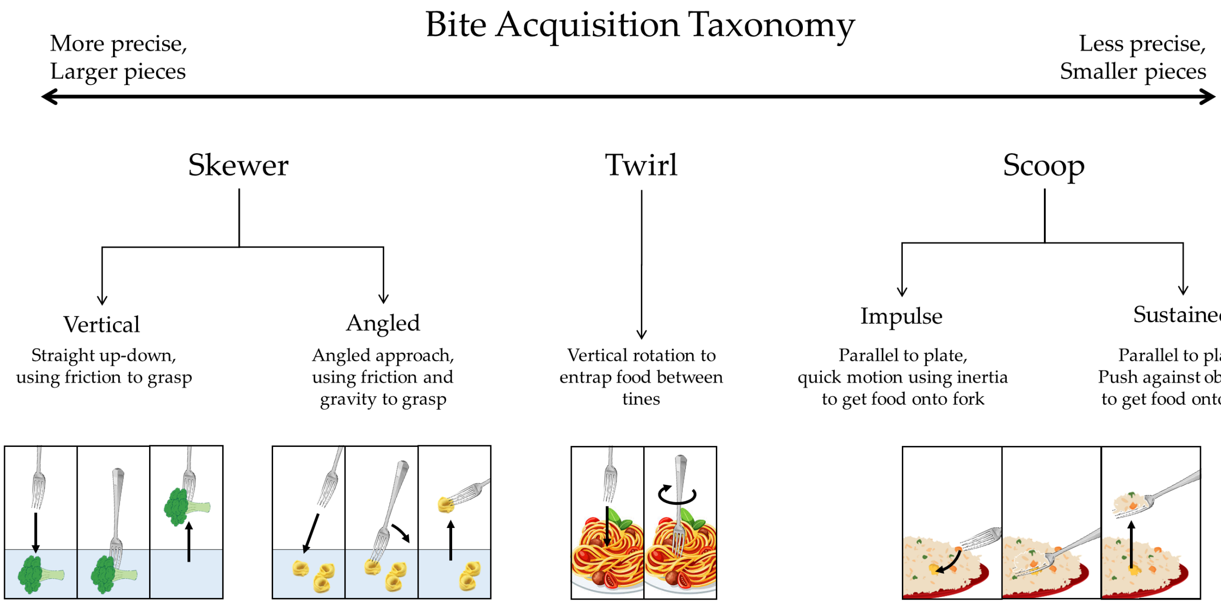

Bite Acquisition Taxonomy
More precise,
Larger pieces
Less precise,
Smaller pieces
Skewer
Twirl
Scoop
Vertical
Straight up-down, using friction to grasp
Angled
Angled approach, using friction and gravity to grasp
Vertical rotation to entrap food between tines
Impulse
Parallel to plate, quick motion using inertia to get food onto fork
Sustained
Parallel to plate, Push against obstacle to get food onto fork

Figure 4.2: Taxonomy of food acquisition primitives.

observations, participants were feeding themselves just as during a normal meal, and in later observations, we asked participants to pick up food via the fork and mime feeding a mannequin the food.

Watching the videos provided a qualitative source of information for the techniques people use to eat, but we were also interested in considering the fork's motion so that it could be recreated with the robot. To that end, we added a cube of fiducial markers to the end of the fork to track its location and orientation in the videos. We chose the marker location to maintain maximum visibility while manipulated, but even so, there were frames where no markers could be detected and noise in the tag detectors made it impossible to get a smooth trajectory. Instead, we used the OptiTrack motion capture system with infrared markers mounted on the fork to collect natural fork trajectories as shown in fig. 4.3.

The fudicials we used are AprilTags [Olson, 2011]

We examined the video recordings an identified that there are discrete strategies for food acquisition, and further that the strategy used was dependent on the food being eaten. We organized each of these strategies as food acquisition primitives in a taxonomy similar to the grasping taxonomy developed by Cutkosky [1989] as shown in fig. 4.2. Next, we describe each of the primitives.

Skewer In a skewering motion, the fork tines are brought downward to impale a morsel of food, and then lifted back upwards and

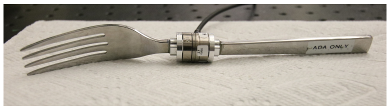


Figure 4.3: Fork with infrared markers mounted on the opposite end from the fork tines.

into the mouth. In a *Vertical skewer*, the fork is pressed into the food at a near vertical, and lifted back upwards vertically. The piece of food is held on entirely by friction and is not in danger of slipping off the end of the fork when held vertically. In an *Angled skewer*, the fork is pressed into the food at an angle, and then pivoted to use a combination of friction and gravity to hold the morsel of food on the fork. In an angled skewer, the morsel of food would slip off the fork if held vertically.

Scoop In a scooping motion, the fork is held with its primary axis nearly parallel to the plate, and the flat part of the tines are pushed underneath the food so that at the end of the motion, a morsel of food is resting atop the fork tines. The fork must remain horizontal after food collection to maintain control of the food. In an *impulse scoop*, the fork is scraped quickly along the plate to use the inertia of the food staying in place to position food atop the tines. In a *sustained scoop*, the fork is scraped along the plate until the food is forced atop the tines when it hits the edge of the plate or enough other food to keep it from moving with the fork.

Twirl In a twirling motion, the fork is brought downward in a manner similar to skewering, entrapping some part of food between the tines, and is then rotated around the vertical axis to twist food around the fork tines before being angled and lifted to a horizontal position.

We have arranged the primitives in fig. 4.2 along trends that were consistent with our observations. When people used vertical skewering, it was with harder foods in larger pieces such as carrots, broccoli, and apples. When people used angled skewering, it was still with large pieces but with softer foods, such as pasta or chicken pieces. Scooping could be used with larger pieces, but was primarily observed with granular food such as rice or salad. Twirling seems to be a special case reserved only for long noodles such as spaghetti. It is an interesting manipulation strategy because it often combines scooping and skewering.

In addition to the actual acquisition motions, the fork was used to prepare food for acquisition. People used the edge of the fork to cut larger pieces of soft food into manageable-sized bites. Preparatory actions were performed, such as pushing food to the edge of the plate, or gathering it into a larger pile before executing one of the acquisition strategies.

To recreate these primitives on the robot, we need a parameterized representation that can be instantiated for a given bite location. Because skewering can be used on the widest variety of foods, we chose to use a vertical skewering motion as a starting point for food acquisition with the robot.

We represented a skewering motion as a vertical downward motion of the fork tines through a piece of food to the plate. Once the plate is reached, the fork is lifted vertically, then rotates 90 degrees to bring the fork flat and level. Once a bite is acquired, the robot can bring the skewered food to the operator's mouth. We used this simple representation of skewering so that the only parameter needed is the food's location, without respect to food orientation, shape, or type. A more complicated definition of skewering could include also an approach direction, application force, lifting angle, and any other number of motion descriptors.

While this skewering motion is inspired by watching humans eating, we could make it more truly human-like by fitting a function to the fork trajectories of human examples. Using the simple skewering strategy outlined here has been effective in our tests, and we did not see a need to change its motion. Scooping, twirling, and even angled skewering will be harder to manually implement because they rely on adapting to the state of the food as the motion is being performed. Even while performing vertical skewering, people may adapt the forces they apply, but we have not seen evidence that they adjust the fork's trajectory.

4.3 Food Detection

We have identified discrete strategies that can be used to acquire bites of food. In order to identify where on a plate of food the strategies should be applied, the robot needs to be able to detect prospective bite locations. In this section we describe a hand-tuned detector, a SVM-based model that uses locally computed features, and a neural-network model that uses raw RGBD sensor values.

4.3.1 Hand-tuned Bite Detector

To detect potential bite locations, we started with classical techniques from computer vision. Using the point cloud obtained from the RGBD sensor, we first used RANSAC [Fischler and Bolles, 1987] to identify the plane of the table. Then, we used a round mask to filter out data that was outside the plate of food on the table. This was necessary to avoid including a glass of water or other objects on the table as potential bite locations. The remaining point cloud was segmented into connected components. At this point, if all the food items were discretely located on the plate, the centroids of each connected component would be enough to serve as a robust bite location. However, in the case of overlapping food, the centroid might not be the ideal place to skewer. In the case that a connected component is above a threshold bite size, bite candidates are chosen along the edge of the connected component, at a distance equal to half the threshold bite size. This allows the robot to select bites along the edge of a pile, working its way to the center, and reducing its risk of skewering a section of food too large to reasonable acquire or eat.

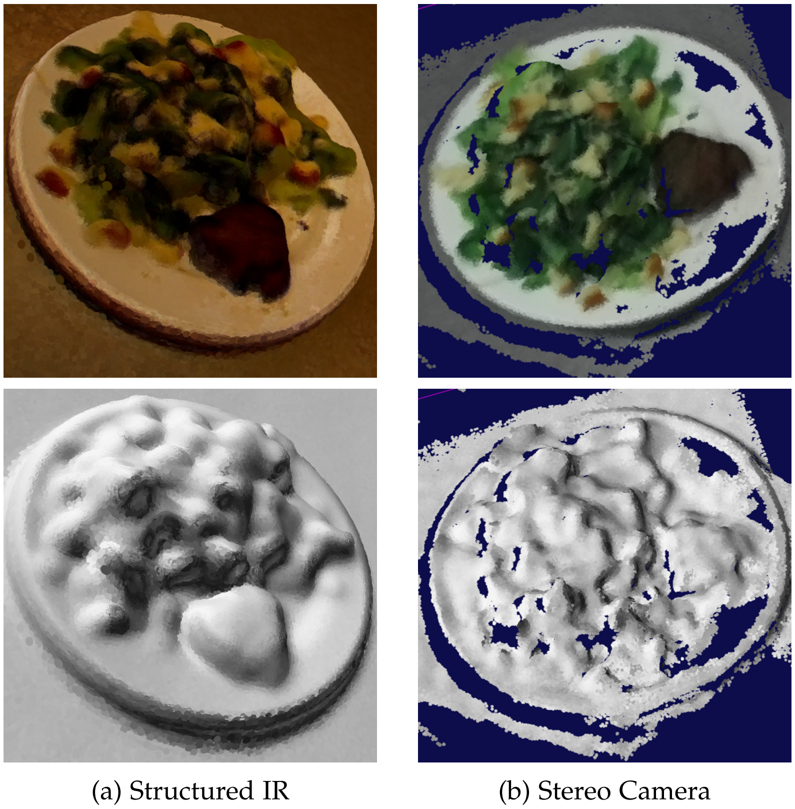
4.3.2 Local Features from Dense Scans

The hand-tuned bite detector is very effective for discrete pieces of food, but it had a hard time detecting foods that were smaller than usual, or making very good guesses about bite locations for piles of food. Instead of tuning parameters in our hand-built bite detection model for each new type of food, we opted to train a model to learn bite locations to help account for the large variation in bite size and food appearance.

We chose to leverage the available point cloud data by using Point Feature Histograms (PFH) [Rusu et al., 2009], Fast Point Feature Histograms [Rusu, 2010], and Point Feature Histograms with Color (PFHRGB) [Rusu et al., 2009] to describe the local geometry around each point in the dataset. PFHs have been successfully used for object classification [Hetzel et al., 2001, Himmelsbach et al., 2009], segmenting surfaces in living environments [Rusu, 2010], and point cloud registration [Rusu et al., 2008]. We believe that the local geometry will be very important for determining good bite locations, as the plate and different types of food have very distinctive shapes and textures.

The goal of the point feature histogram (PFH) formulation is to encode a point's k-neighborhood geometrical properties by generalizing the curvature around the point using a 125 dimensional histogram of values. It is capable of capturing all possible surface variations by taking into account interactions between the normals of k neighbors. The resultant vector is dependent on quality of normals. This vector is binned into 5 bins for each angular, which creates total of 125 bins and defines the length of the vector. This high dimension hyperspace provides a distinct signature for the feature representation of geometry. It is invariant to the 6D pose of the underlying surface and copes very well with different sampling densities and noise within its neighborhood.

The difference between PFH and fast point feature histograms (FPFH), is that FPFH does not fully interconnect all neighbors of the point of interest p_q. It thus misses some value pairs which might have contributed to capturing the geometry around the query point, making it less accurate than PFH, but faster to compute. The FPFH includes additional point pairs outside the r radius sphere and has



(a) Structured IR

(b) Stereo Camera

Figure 4.4: Comparison between a dense scan made with structured IR light (a) and a stereo camera (b).

a larger span of up to $2r$ away. The overall complexity of FPFH is greatly reduced due to the decrease in neighbors, making it possible to use it in real-time applications. The histogram that is produced is simplified by de-correlating each feature dimension and concatenating them together.

The point feature histograms with color (PFHRGB) is calculated in similar fashion, but in addition to the information about the relationship of the normals, it also includes color channels. The PFHRGB has three additional parameters: one for the ratio between each color channel of source and the corresponding color channel of destination point. These histograms are binned in similar way to the PFH, and generate another 125 floating point values. This brings the total size of vector to 250.

The descriptiveness of PFHs is related to the resolution of the point cloud, and the size of the influence region. Some food features are very small in comparison to the sensor resolution (e.g. grains of rice or pieces of salad), so fairly dense point clouds of the food are needed. To train a classifier based on the feature histograms, we also needed to procure point-wise labeling of the point clouds into “skewerable” and “non-skewerable” regions.

Colored point cloud datasets have been published to benchmark

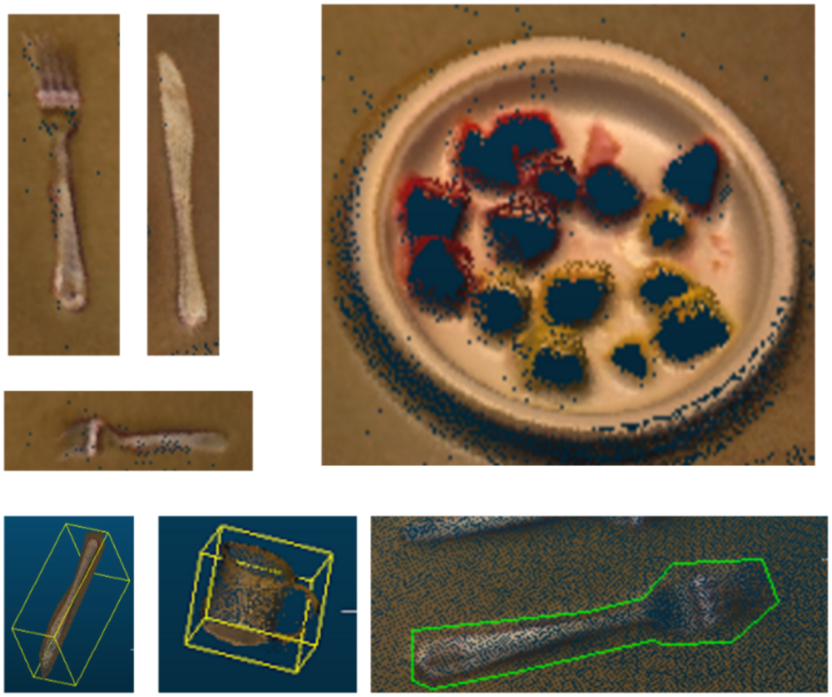


Figure 4.5: Examples of objects which must be given non-bite labels to effectively train a classifier that does not use a mask for objects not on the user's plate.

computer vision and machine learning algorithms. Potential source datasets that include food items are the TST Intake Monitoring database [Gasparrini et al., 2015], the Cornell Activity Datasets CAD-120 [Koppula et al., 2013], the RGBD-HuDaAct: A Color-Depth Video Database for Human Daily Activity Recognition [Ni et al., 2011], and the Leeds Activity Dataset [Aldoma et al., 2014]. The number of point clouds in each dataset that includes food on a plate is very limited, and is always of a full plate, never of any partially eaten examples. We therefore supplemented existing datasets with our own collection of RGBD images of plates of partially eaten food.

To generate dense colored point clouds using the wrist-mounted RGBD sensor, the robot is programmed to move to a pose where the RGBD camera can look down at the known plate location. The robot moves along a scanning trajectory, in which the plate is always in view of the camera, to collect several seconds worth of data that are then merged via Elastic Fusion [Whelan et al., 2015] to create a denser, more complete point cloud. We also explored using a stereo camera to generate a dense scan of the food, but found that using a stereo camera to generate the dense scans was more prone to holes as visualized in fig. 4.4.

To label the generated scans, we manually segmented the point clouds into potential bite and non-bite locations. Because we are

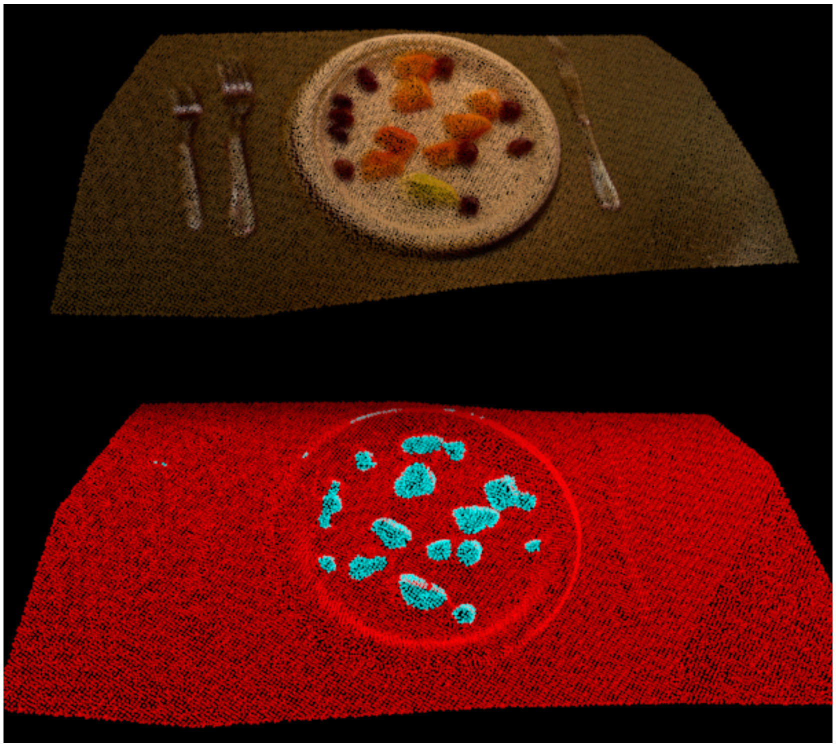


Figure 4.6: A dense scan of a plate of food after applying Elastic Fusion. The bottom point cloud is colored by hand-labeling of potential bite and non-bite locations.

no longer using a mask for the plate to remove other objects in the scene, it was important to include negative examples such as the glass of water, other silverware, and napkins in the scene during training such as those in fig. 4.5. With the features and labels for each point in the scans, we trained a non-linear Support Vector Machine to classify each point as a potential bite location or a non-bite location.

Applying the trained SVM results in a labeling for every point in the point could. A repesentative example is shown in fig. 4.6 in which the bite locations have been colored in blue and the non-bite locations in red. Unlike the hand-tuned bite detector, this method presents many bite locations for each piece of food, instead of a single centroid. The edges of pieces of food are labeled as non-bite locations, which is desirable since skewering on the edge of a piece of food is more likely to be unsuccessful.

There are two major shortcomings to this technique for choosing bite locations. The first is that generating the dense scan and computing local features is time consuming, on the order of 5-10 seconds per bite. This kind of delay is frustrating as a user and can be disruptive to dining socially since the robot has to move in very imposing ways to produce the food scan. The second shortcoming is that the model is very sensitive to the food examples it was trained with and that adding data for a new food is expensive due to manual labeling. We trained only with pieces of fruit, but there is a wide variety of food shapes, sizes, and colors that were not yet explored.

To fix the first shortcoming, we can draw from the success of applying neural networks to raw sensor data for classification and segmentation [Dong and Xie, 2005]. We will use only single images instead of dense scans, and adjust the depth of the neural network as needed to discover important local information.

To fix the second shortcoming, we can gather supervised data from robot trials instead of hand-labeling. While this does not make the labeling necessarily less expensive in terms of time or effort, it does enable online learning to take place, where the robot can get better at manipulating food the longer it is in use. To that end, we use the force-torque sensor embedded within the fork as shown in fig. 4.7 to determine how much food was captured after an attempt by the robot to collect food. With this measure of success, we can define our problem as follows.

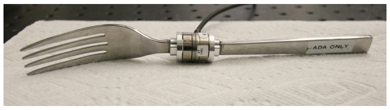


Figure 4.7: Force-torque sensor embedded within a fork to measure forces and torques applied at the tinetips.

4.4 Learning Bite Locations

The objective is to select an (x, y) plate coordinate at which to perform a skewering action given an input RGBD image of the whole plate.

4.4.1 Data Collection

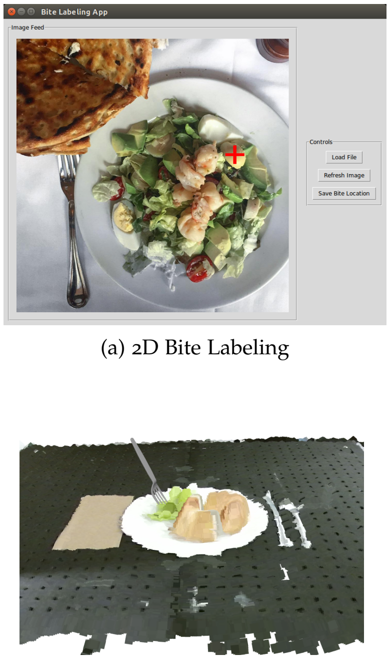
During each trial, the robot makes an observation of the plate of food, selects the policy parameters needed to execute a food acquisition policy, and measures the success of the food acquisition attempt.

The initial observation is a single RGBD image of a plate of food with a resolution 256x256. Due to the time restrictions of scanning and computing a combined image/point cloud from multiple vantage points, the size and detail of the observation is limited by the RGBD sensor's capability for a single image. To compare performance of a wrist-mounted camera versus an overhead camera, we collected both sources and trained with them separately.

For this experiment, we used the skewering primitive because it can be specified with the fewest number of parameters, but the same setup can be used with different parameterizations.s The policy parameters θ needed to execute the robot's action are simply the (x, y) location on the plate of the skewering target.

Success is measured at the end of the trial by rotating the fork to a horizontal and using the force-torque sensor to calculate the mass in grams of any food matter attached to the fork. For unsuccessful bites, no food mass will be acquired. For successful bites, we can choose to use the mass as a metric of success – the more food, the more successful.

We considered jump-starting the learning process by using human-labeled bite locations via either a 2D or 3D interface (as shown in fig. 4.8), but determined that bite selection is a very subconscious decision and examples provided in such a manner would be artificial and potentially even detrimental examples. Instead, we ran several hundred trials in which the robot would pick a random (x, y) location on a plate of food, attempt a skewer, and record its level of success.



(a) 2D Bite Labeling

(b) 6D Bite Labeling

Figure 4.8: GUI for performing standardized hand labeling of bite locations in (a) 2D and (b) full 6D.

Prior to each trial, the robot moved to a position where the plate was in view of the overhead Kinect sensor and where the wrist-mounted Astra-S sensor could see the whole plate without hitting the minimum range limits of the sensor. The RGBD image from each sensor was recorded as a "pre-bite" image. Next, an (x, y) target was chosen from a uniform distribution over the surface of the plate: $x \in \mathcal{U}(-r, r)$, $y \in \mathcal{U}(-r, r)$, for plate radius r, rejecting samples outside the circle of the plate. The robot performed a skewering action as described in section 4.2. After the skewering motion, the robot's wrist rotated 90° so that the fork is held at a horizontal to the table. This happens to mimic human behavior, but is done for the purpose of measuring the weight of the attached food via the force-torque sensor embedded in the fork. We tested the accuracy of measuring weight this way with standardized metal weights and determined the accuracy to be within 0.5 grams. After this measurement is made, the robot scrapes the fork against a soft sponge which enables the robot to empty the fork in preparation of the next trial.

We collected 500 robot trials to use as training data, using mixed fruit as the type of food. We chose mixed fruit because it is a type of food that is acquired almost exclusively using a skewering motion, and it provides a variety of colors, textures, shapes, sizes, and physical properties. Of the trials, 21.4% resulted in a successful acquisition of a bite. The average successful bite collected 4 grams of food.

4.4.2 *Model 1: Local Bite Classification*

Given the pre-acquisition images and the skewering location, we cropped the images to 32 x 32 pixels centered around the skewering location. For each of the cropped images, we used the force readings on the fork at the end of the execution to produce a binary label of success or failure if the fork collected food mass weighing above the threshold of 1 gram. Examples for each of the classes is shown in fig. 4.10. Intuitively, the positioning of food on the whole plate will

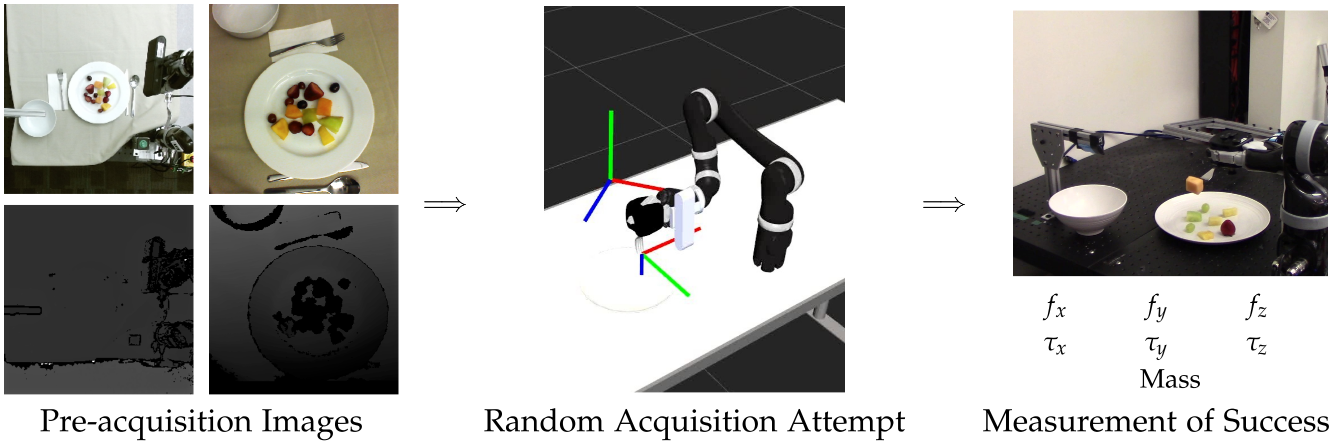
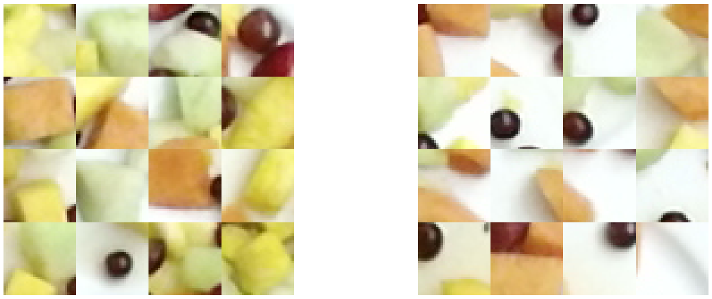


Figure 4.9: The learning setup in which the robot captures an image of the plate, performs a skewering action, and receives a reward based on the amount of food on the fork.

not impact the success or failure of a given bite since the physical interaction will be governed by local dynamics between the fork and the food during the skewering motion.

In the training examples of fig. 4.10, we can identify some sources of failure for the negative examples. In the failed bite locations, food is either not present or there are multiple pieces of food together, but the skewering motion is targeting the space where to pieces of food are touching one another.



(a) Success (b) Failure

Figure 4.10: Examples of local images of bites where the robot succeeded or failed to skewer a bite.

Deep convolutional neural networks have been used extensively for image classification and segmentation [LeCun et al., 2015]. We are performing image classification as well, but the classes instead of being related to visual properties of the object itself, are related to the performance of a manipulation strategy enacted on the object.

We used two layers of a Residual Network for our network architecture [He et al., 2016], which is composed of multiple ResNet blocks. In the original LeNet forumlation, there are alternating layers of convolution, non linear combinations (ReLU), and pooling, followed by a fully connected layer with a softmax activation function classification [LeCun et al., 1998]. The ResNet blocks add "shortcut connections", skipping one layer as shown in fig. 4.11 and leading to

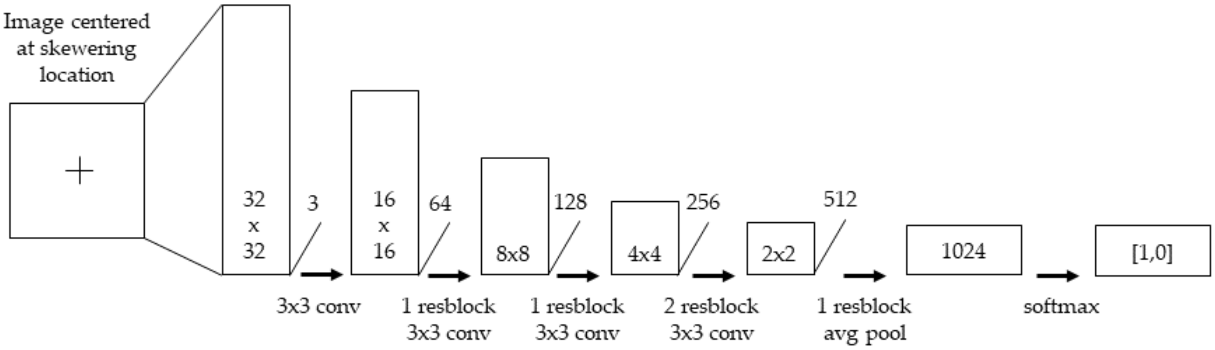


Figure 4.12: Model1: ResNet which takes an image centered around a skewering location and outputs a 2x1 vector of [0,1] values. The input image shown is for color, but we use the same network layout for depth and color+depth images.

deeper networks with higher accuracy.

We used TensorFlow [Abadi et al., 2015] in Python in our implementation, and trained the network with a machine that has a NVIDIA GeForce GTX 950 GPU. The layout of the network is shown in fig. 4.12.

To apply the output of this network to the problem of determining the actual (x, y) bite locations given an image of a whole plate of food, we use a sliding window to generate a pixel-wise labeling for the image of the whole plate. From the pixel-wise labeling, we can choose to prioritize bite locations in the order of labeling probability, or use the centroids of the largest connected components in an attempt to increase robustness of the skewering.

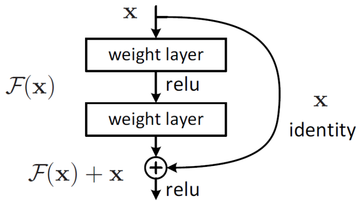

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4.11: The building block for residual learning networks from [He et al., 2016]

4.4.3 *Model 2: Local Bite Weight Regression*

As in section 4.4.2, for local bite weight regression, we cropped the pre-acquisition images to 32 x 32 pixels around the skewering location. Instead of thresholding the force readings into success or failure, we instead use the food mass measurement as the output of a regression model. The objective is given a cropped image, to produce the amount of food mass that will result after a skewering action performed at the image's center.

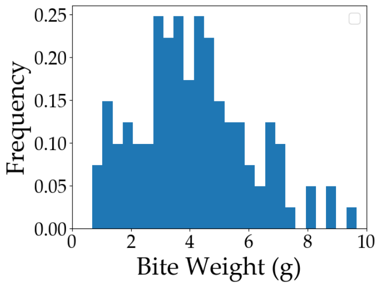


Figure 4.13: Distribution of bite weight readings for training.

The distribution of food masses is shown in fig. 4.13. Because all of the skewering attempts that did not result in a bite have a mass of zero, the distribution of food masses is heavily skewed in that direction.

The structure of this network is similar to that in section 4.4.2, except for the last layer in which we use a fully-connected layer without a softmax operation. We used the $l1$ cost metric between the true mass and the network's output. The layout of the network is shown in fig. 4.14.

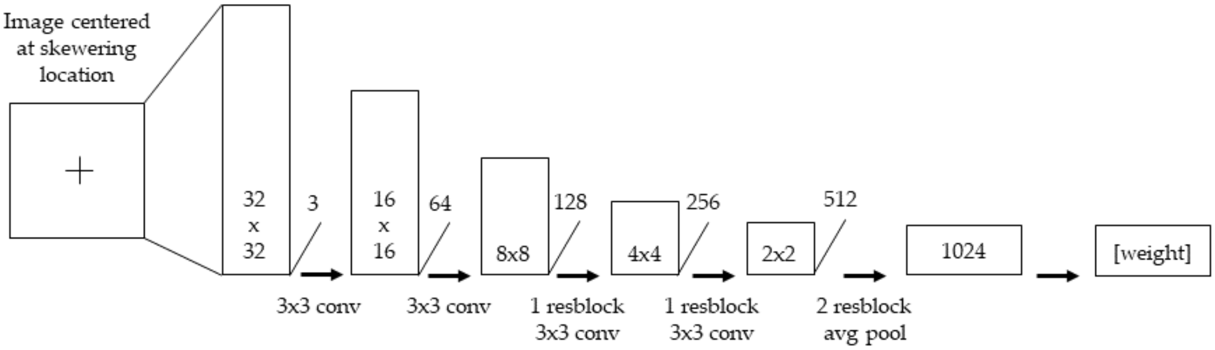


Figure 4.14: Model2: ResNet which takes an image centered around a skewering location and outputs a single value for the food weight. The input image shown is for color, but we use the same network layout for depth and color+depth images.

To apply the output of this network to the problem of determining the actual (x, y) bite locations given an image of a whole plate of food, we use a sliding window to generate a pixel-wise mass prediction for the image of the whole plate. From the pixel-wise labeling, we can choose to maximize the mass prediction to produce a behavior of picking up the largest piece of food, use the labeling probability to maximize the expected mass that is collected, or choose locations that will produce bite masses in a specified range (perhaps based on user preferences).

4.4.4 Model 3: Bite Location Regression

The goal of this model is to directly produce the (x, y) location of a bite-sized food given an un-cropped image of a plate of food. The training data is compiled by only using successful trials – defined as skewering attempts in which at least 1 gram of mass was acquired. The full pre-acquisition image is used, and the objective of the network is to produce an (x, y) location that is as close as possible to the actual (x, y) that was executed by the robot to produce a successful bite.

The advantages of this model over the two local models described in section 4.4.2 and section 4.4.3 are that it uses the entire plate as input, which could be necessary in meals such as spaghetti where interconnected pieces of food may span the whole plate. By directly implementing the problem formulation, this network makes no assumptions about the relevance of the area local to the skewer target, nor the impact of the mass of collected bites.

The major disadvantage to this technique is that due to the nature of how the data must be collected, we never have more than a single (x, y) location for a given image of a plate of food, and there is no guarantee that it is the best bite location. Additionally, only suc-

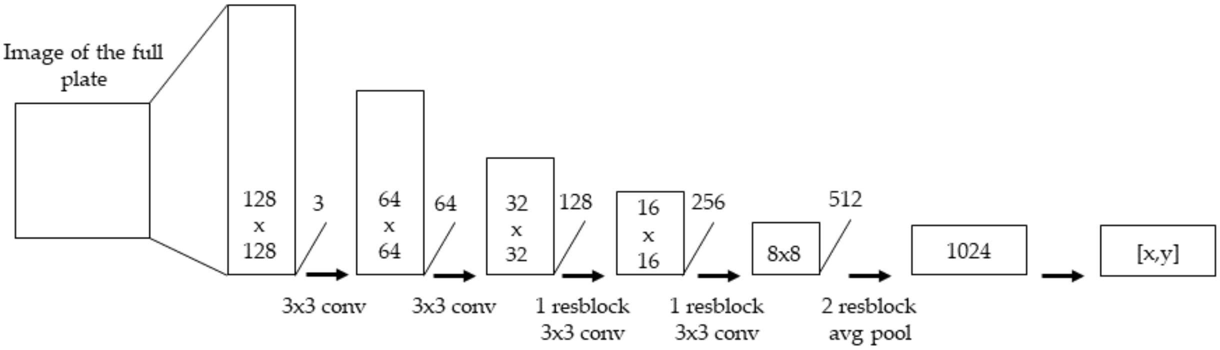


Figure 4.15: Model3: ResNet which takes an image of the full plate and outputs a 2x1 vector corresponding to a skewering location. The input image shown is for color, but we use the same network layout for depth and color+depth images.

cessful trials can be used in training, which vastly cuts down on the amount of training data available. This could be alleviated by adding hand-labeled bite locations via an interface such as that shown in fig. 4.8, but we would lose the valuable real-world feedback of the robot's actual performance given that skewering location, and have to rely on the judgment of the human labeler.

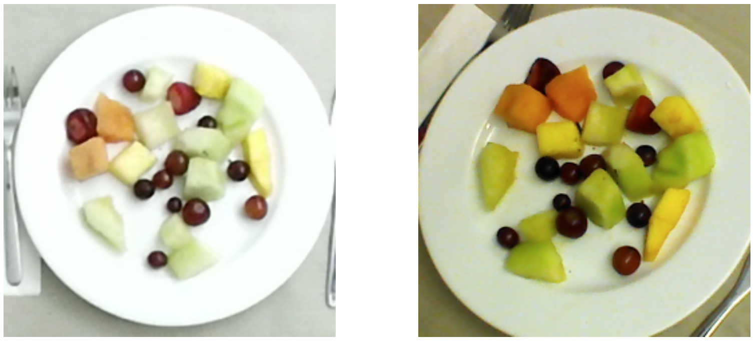
The overall structure of network is similar to that described in depth in section 4.4.2, but the input images are now 256 x 256 pixels. The last layer is a fully-connected layer without a softmax operation, with two output nodes, for the x and y locations. The layout of the network is shown in fig. 4.15.

The output of this network is a single (x, y) location, which limits the ability to add additional constraints or heuristics to bite selection.

4.4.5 *Comparison and Performance*

Wrist or Overhead Camera When this work is ready for distribution in the home, the ideal setup would only use a wrist-mounted camera instead of an overhead camera to allow for maximum portability. To that end, we train each model with only the wrist camera and only the overhead camera. The wrist camera is closer to the plate of the food and therefore has a better viewing angle, but the resolution is lower, so the output images are comparable. We did our best to maximize the angle between the wrist camera and the table to minimize occlusions from the food, but were constrained by the maximum range of the robot arm. Two example images are shown in fig. 4.16. The plate for both images was 256 x 256 pixels, and the wrist had minimal occlussions, so to maintain consistency in comparing results, the overhead images were used for training the models for the results in this section.

Depth Channel We collected an RGBD dataset partially on the


(a) Overhead

(b) Wrist

Figure 4.16: A comparison of the two sensor viewing angles of the plate. Occlusions from the wrist due to food are minimal even from the wrist-mounted sensor and the resolution is comparable between both sensors.

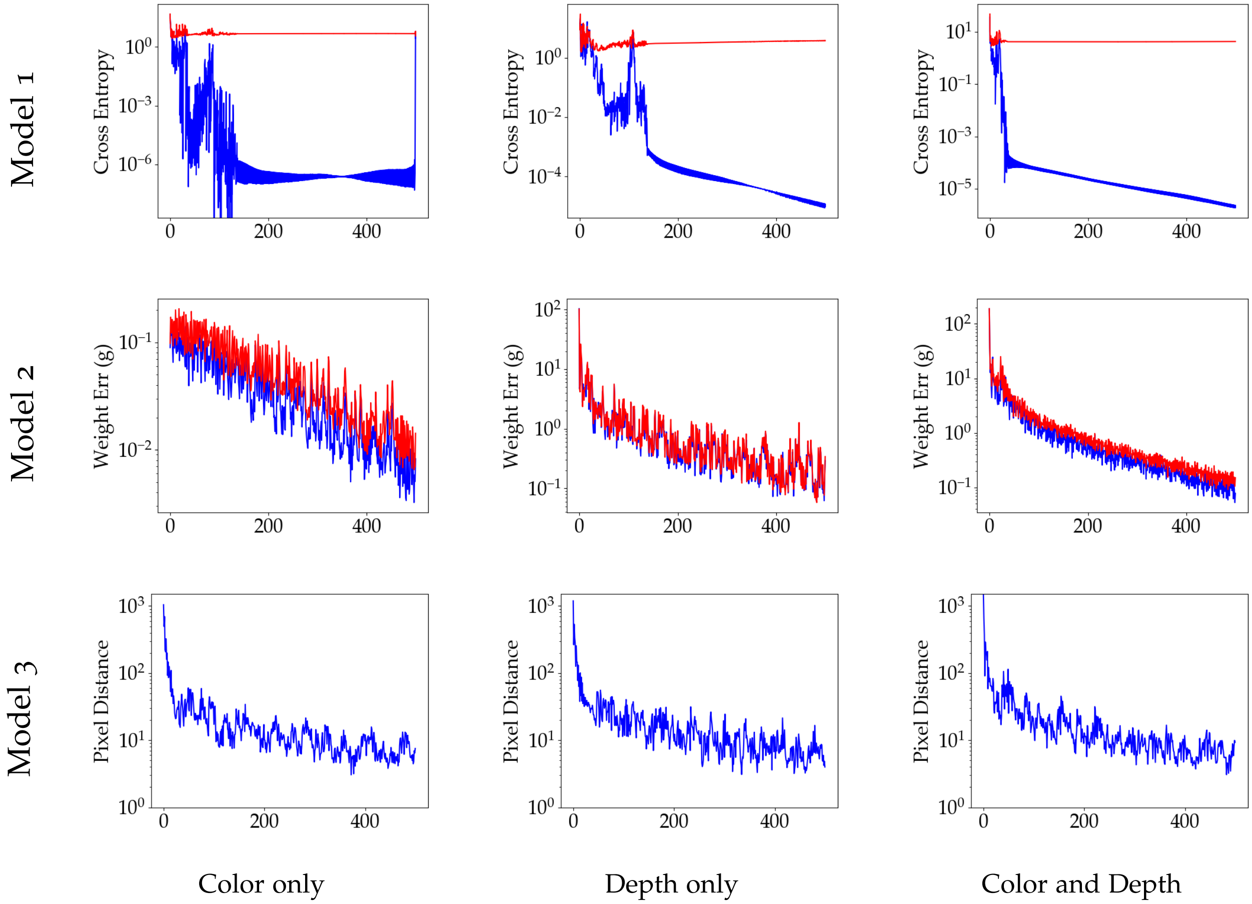


Figure 4.17: Metrics during the training and testing phases for each of the networks. The training is shown in blue, and testing in red.

premise that the depth channel would be important for learning appropriate bite locations. To test this, we trained each model using only the color channels, using only the depth channel, and using the color channels and the depth channel combined.

Metrics for training and testing for each of the models are shown in fig. 4.17. For Model 1, the cross-entropy values are plotted for the binary labeling of skewering location or non-skewering location. For

Model 2, the absolute difference in weight is plotted. For Model 3, the mean squared pixel distance between the predicted (x, y) location and the actual location.

In Model 1, which generates a binary labeling for each image around a skewering location, the performance on the test set is poor over all types of training input. For Model 2, the test error follows the training error very closely. We see slightly better results when combining color and depth images for training. The combined image model predicts the food weight to within 1.5 grams. For Model 3, there was very limited data to perform extensive tests since it was only trained with successful bites. Nevertheless, an initial training with Model 3 converged to a mean pixel distance of about 10 pixels.

To make use of the output of Models 1 and 2, a sliding window across a full image was used to evaluate potential skewer locations in a pixelwise manner. We then ranked the potential locations by either the cross entropy and positive label (for Model 1) or by predicted bite weight (for Model 2). The output of Model 3 is an (x, y) location and can be directly used as the only potential skewer location. The results of Models 1 and 2 are shown on an example image in fig. 4.18.

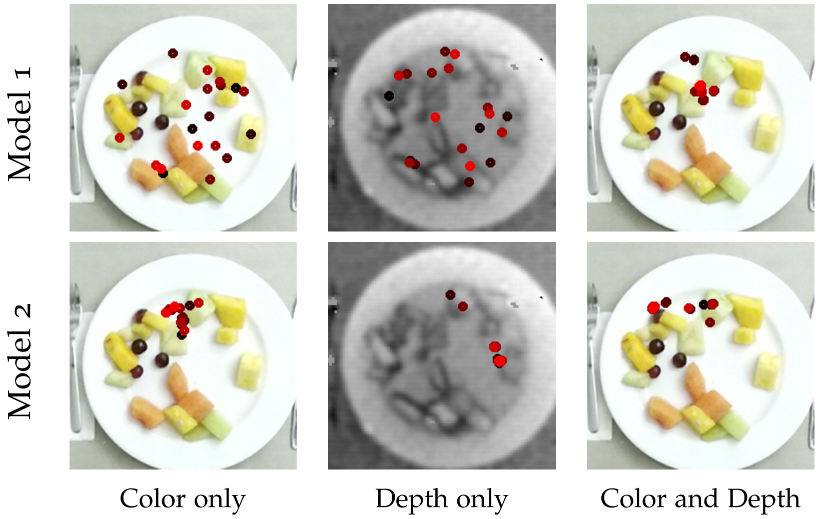


Figure 4.18: The top 20 choices for bite locations as predicted by Model 1 and Model 2, as trained with different source data. The lighter and more red labels are higher ranked bite locations - in the case of Model 1 a higher probability of correct classification - in the case of Model 2 a higher predicted food weight.

5
Human-Robot Interaction Considerations

All assistive devices face a trade-off between performance and acceptance. The rate of abandonment in assistive technology (shown in fig. 5.1), particularly technology infused with a higher degree of intelligence – i.e. predictive and/or corrective algorithms – is a large roadblock to their adaptation [Phillips and Zhao, 1993]. Carefully considering the social and physical ramifications of a new assistive technology is therefore critical in predicting its adaptation and feasibility of long-term integration into the target population's lives.

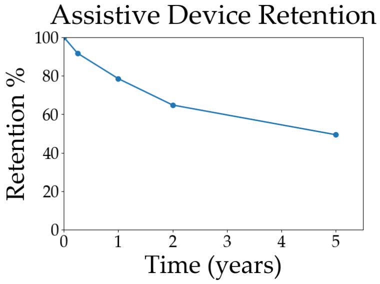
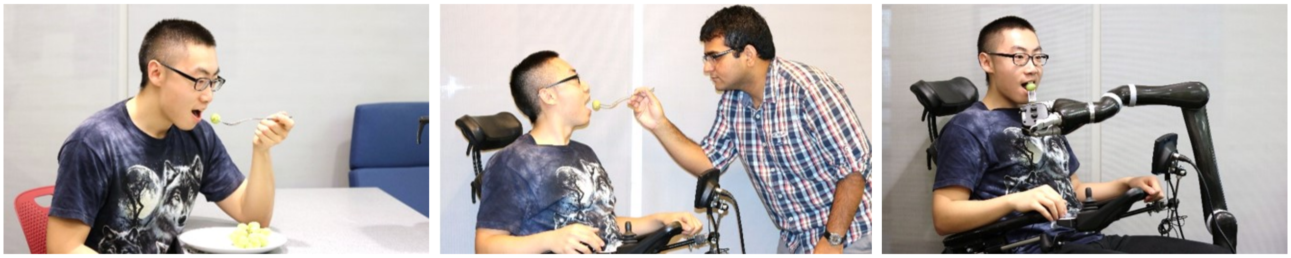


Figure 5.1: Abandonment of assistive mobility devices as collected in [Phillips and Zhao, 1993]

In chapter 3 we considered the level of control preferred in switching control modes. Now we consider a similar question about the level of the control that is needed and preferred in the context of eating with an intelligent robot. Starting with the endpoints of the spectrum of shared control – fully manual control and fully autonomous control – we can ground ourselves and anticipate the range of performance and reactions to systems within that spectrum. Through interviews with users of the JACO assistive robot arm we have established that manual control in the context of eating is not feasible due to practical constraints on the time it takes to manually acquire a bite of food and bring it to the operator's mouth (For more details see **??**). For an automated feeding mode therefore, the speed and timing are critical to success.

In chapter 4 we described a method for the robot to autonomously acquire pieces of food. To complete an autonomous feeding system, the robot must now be given the ability to deliver the forkful of food to the operator's mouth. Further, the robot must do so in a way which is acceptable to the operator in the hopes of mitigating the potential for abandonment of the technology. In this chapter, we will present our approach to timing when to give the operator bites of food, an analysis of the performance of that approach, and finally a discussion on other HRI factors not included in this study but that would be relevant to a long-term automated feeding implementation.



(a) Self-self (b) Caregiver-self (c) Robot-self

Figure 5.2: Feeding as a handover for able-bodied users (self-self), for disabled users and their caregivers (caregiver-self) and for disabled users and their assistive robots (robot-self).

5.1 *Modeling Bite Timing Triggered by Social Cues*

Timing is a subtle and vastly important thing. While eating a meal, particularly in the context of others, when a person takes bites is dependent on many factors such as whether they are already chewing a bite, whether they want to say something, whether they already have food on their fork, etc. We build and test a model for predicting the timing of taking bites based on social cues of the person that is eating.

5.1.1 *Bite Timing Model*

To generate a model for predicting bite timing, we borrow from work done in the context of human-robot handovers. We consider eating as a handover where food is being brought to the mouth from one of 3 possible origins: the person's own hand ("self-self"), a human caregiver ("caregiver-self"), or a robotic feeding device ("robot-self") as shown in fig. 5.2. We expect each source to make a difference on the handover interaction. For example, when being fed by a human caregiver, the person will need to use a gesture or audio indication of what food they want or when to take a bite [Martinsen et al., 2008]. When being fed by a robotic feeding device, such as those in fig. 2.4, a physical button press is required to initiate the handover. When the person is using their own hands to make the handover, there will be coordinated movement between the head, mouth, and arm.

In the handover literature, the handover action can be defined as a state machine with a finite number of states and non-zero transition probabilities. For handovers, one such state representation would be `hold`, `transfer`, and `not hold` [Grigore et al., 2013]. Another model with more granularity would be `hold`, `approach`, `signal`, `transfer`, and `not hold` [Strabala et al., 2013]. We can break down feeding into a similarly simple state representation informed by observing how able-bodied people eat. From the collected data, we will extract features that can potentially indicate the hidden state. We will then estimate the transition probabilities and use a Hidden Markov Model (HMM) to predict when the next bite will occur.

5.1.1.1 Data Collection

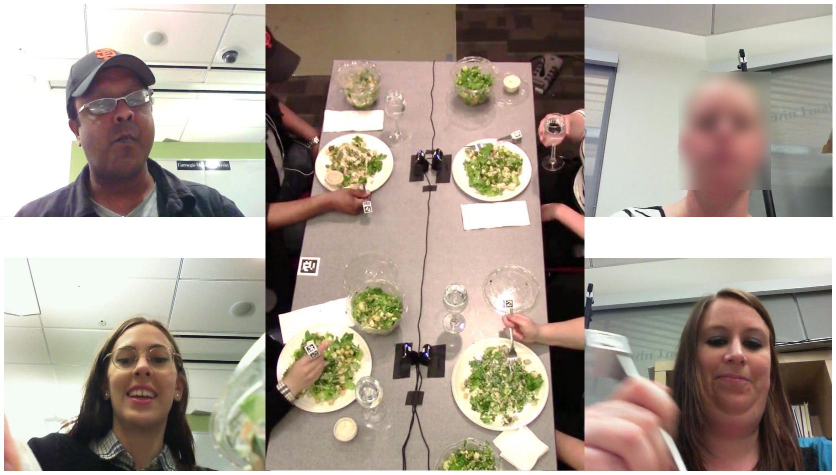


Figure 5.3: Experimental setup for gathering data to build a model for bite prediction. A table-top camera is pointed at each participant to gather gaze direction, bite timing, and detect who is speaking.

To determine which states are needed to represent eating and collect observational data that might be useful to distinguish between states, we performed a study in which 1-4 able-bodied participants were asked to sit around a table and eat lunch while having a conversation. In the case of a single participant, they were asked not to read, answer the phone, or otherwise distract themselves from eating. For groups, we added the conversational component to encourage the social cues of eating to manifest themselves. To reduce the number of confounding variables, we chose to use food that was bite-sized and only required a fork to eat and restricted participants to those who could eat unassisted. During the meal, a camera was placed in front of each participant pointing at their face as shown in fig. 5.3. The conversations were undirected and the experimenters were not present, to allow for natural topic choice and conversation flow among participants. We administered 18 sessions, with a total of 40 participants (12 male, 19 female, ages 20-61), that generated 16.5 hours of video and audio recordings.

We eventually want to apply the findings of this work to creating a system for people with disabilities to dine independently. As such, the choice of sensors was deliberate. We used a small web camera pointed at the participant from the table perspective. This maintains the privacy of other people the operator may be dining with. Similarly, there could be an option to either place a small wireless camera on the table before each meal (we are using low enough resolution to make wireless video practical), or to attach the camera to

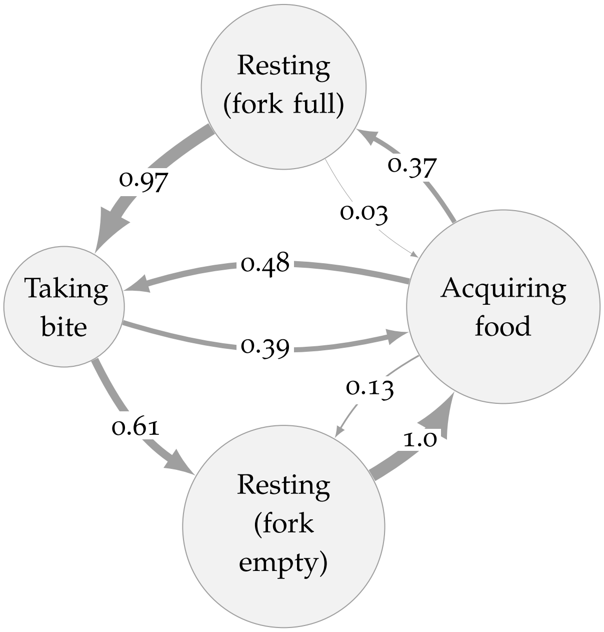


Figure 5.4: Eating state machine with transition probabilities. The area of each state is proportional to the average amount of time spent in each state across users.

the wheelchair pointing upward at the operator from a discreet angle. We recorded video from overhead for visualization purposes, but only the cameras facing the participants were used to learn an eating model.

5.1.1.2 Eating States

The simplest state representation that could capture bite timing would be to split the eating activity into `biting` and `gathering` states. In the biting state, the user moves the fork full of food to their mouths and takes a bite. In the acquiring state, the user is manipulating the food on the plate in order to load the fork with food in preparation for taking the next bite. Upon reviewing the participant videos, it was clear that this was not the whole story. In between taking bites and gathering food, we observed an abundance of fork queuing behaviors.

Rather than directly taking a bite after gathering a full bite of food, users would hold the loaded fork near the plate 37% of the time, waiting for an average of 13.49 seconds before bringing the fork to their mouth for the next bite. Similarly, after taking a bite, users would only gather food immediately 39% of the time. The remaining 61% of the time, they would hold the empty fork near the plate, waiting for an average of 10.34 seconds before gathering the next forkful. The time spent in these two waiting states comprised an average of 56% of the total time eating, which seems to be a very inefficient way to consume a meal if the only goal is to consume nutrients.

Qualitatively, one source of waiting with a full fork is due to the fact that the participant is still chewing the previous bite and it would be impractical to take the next bite until finished. Another more subtle cause is when the participant is speaking and waits until they have concluded their thought before taking their next bite to avoid interrupting themselves mid-sentence. People also wait with an empty fork while listening intently to another speaker, or if they themselves are speaking with too much focus to simultaneously look down at the plate and gather food. The times spent resting the fork are important for the social aspects of dining.

Therefore, we decided that it was more appropriate to separate eating into four states: `gathering`, `waiting with fork full`, `biting`, and `waiting with fork empty`. By annotating the videos with these four states, we estimated the transition probabilities and how much time was spent in each state, as shown in fig. 5.4. There are a few transition probabilities of note. There is a probability of 1.0 that waiting with a fork empty transitions to gathering food. This makes sense if we consider that no one would take a bite from an empty fork and that an empty fork cannot spontaneously be filled without performing the gathering step. Surprisingly, the transition from waiting with a full fork to gathering food is non-zero. This transition occurs when the person decides after filling their fork that they would like to eat something else in their next bite. Similarly, gathering actions that transition to a resting empty fork are used to redistribute food on the plate but without actually loading the fork.

From the labeled states, we examined the timing information between bites to see how evenly-spaced bites taken are. On average, there are 14.23 seconds (SD=13.44) between bites across all users. The full distribution is shown in fig. 5.5. By calculating the cumulative distribution function for having taken a bite, we can choose a timing threshold for each probability of having taken a bite. We fit a Burr distribution (for continuous non-negative random variables) to the bite timing data to make the choice analytically.

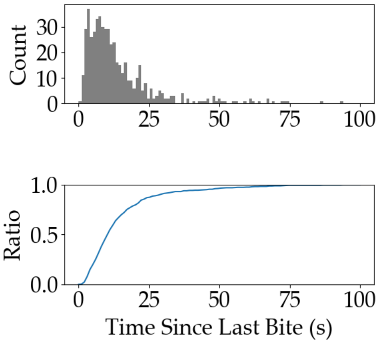


Figure 5.5: The amount of time since taking the last bite is plotted as a histogram (top) and as a cumulative density function (bottom).

5.1.1.3 Eating Features

While the state of the biting process is not directly observable without placing a sensor on the fork or using an overhead camera to monitor food movement, we can observe several features that are related to the current state. By monitoring these emission features, we can estimate the hidden state. Because of the high-dimensionality and limited descriptiveness of the raw video and audio signals, we have identified video and audio features that may be relevant to the eating state.

From the video, we used facial recognition and pose estimation [De la Torre et al., 2015] to calculate the gaze direction (up, down, left, right, or straight ahead), whether the mouth was open or closed, and whether a face was detected or not as shown in fig. 5.6. Prior work shows that gaze cues impact the strategies for item handovers [Moon et al., 2014, Admoni et al., 2014, Strabala et al., 2013], so we would expect it to be a relevant feature in this domain as well. The mouth being open or closed can be a useful feature for detecting an initial bite, but could also be confounding when a person is talking. When the face detection failed, it was usually due to the participant drastically changing their position by shifting in their seat, or by extreme head movement to the left or right. For people with upper-limb disabilities, the neck range of motion is less than an able-bodied person [LoPresti et al., 2000], so we do not believe this problem will exist in the target application.

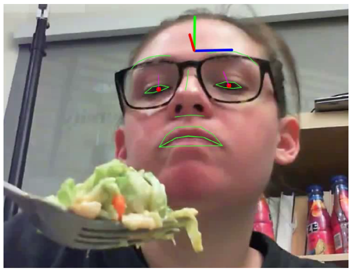


Figure 5.6: Social cues are automatically extracted from videos on each participant as they are eating and conversing.

Each person-facing camera recorded its own audio, so by using the volume, we could identify if the person being recorded was speaking or not. Carrying on a conversation requires turn-taking between speakers, and that implicitly requires eye-contact and other indicators of paying attention. Etiquette also implies that people will not be able to take bites while talking, since chewing while talking is generally considered bad manners. Anecdotally, we can see this theory in action in fig. 5.3, where the participant in the upper right is talking, the participant in the lower right is listening, and only the participant on the left who is not engaged in the conversation is taking a bite. Therefore, we expect talking to be a relevant feature for predicting the eating state.

All the features are automatically detected in order to preserve the ability to use this model with an assistive robotic feeding device. A summary of the social features used is in table 5.1.

Feature	Dimensionality	Source
Gaze direction	6	Video
Head pose	6	Video
Mouth area in pixels	1	Video
Time since last bite	1	Video
Face detected	Binary	Video
Talking	Binary	Audio

Table 5.1: Social cues used to predict bite timing.

5.1.1.4 HMM

HMMs are used to model sequential, statistical processes in a variety of fields, including handovers and speech and gesture recognition[Rabiner and Juang, 1986]. Formally, an HMM can be written as a tuple $\lambda = (Q, V, A, B, \pi)$, where:

- $Q = \{q_1, q_2, \ldots, q_N\}$, finite number of N states
- $V = \{v_1, v_2, \ldots, v_M\}$, discrete set of possible M symbol observations
- $A = \{a_{ij}\}$, $a_{ij} = \Pr(q_j \text{ at } t | q_i \text{ at } t)$, state transition probability distribution
- $B = \{b_j(k)\}$, $b_j(k) = \Pr(v_k \text{ at } t | q_j \text{ at } t)$, observation symbol probability distribution in state q_j
- $\pi = \{\pi_i\}$, $\pi_i = \Pr(q_i \text{ at } t = 1)$, initial state distribution

For eating, $N = 4$ and Q = {`gathering`, `waiting with fork full`, `biting`, `waiting with fork empty`}. The state transition matrix A was calculated from the hand-labeled data as shown in fig. 5.4. Since the data we collected contains full meals, the initial state distribution π is zero everywhere except in the `waiting with fork empty` state.

Initialization of the observation symbol probability distribution matrix B was performed by using the features from videos whose states were hand-labeled. These samples show a representative number of observations for each state and transition.

The HMM parameters were then reestimated using a normalized Baum-Welch Algorithm described in [Grigore et al., 2013] as follows:

Forward Algorithm:

$$\hat{\alpha}_t(i) = \frac{\pi_i b_i(O_1)}{\sum_{k=1}^{N} \pi_k b_i(O_1)} \tag{5.1}$$

$$\hat{\alpha}_{t+1}(i) = \frac{b_i(O_{t+1}) \sum_{j=1}^{N} \hat{\alpha}_t(j) a_{ji}}{\sum_{k=1}^{N} b_k(O_{t+1}) \sum_{j=1}^{N} \hat{\alpha}_t(j) a_{jk}}, 1 \leq i \leq T \tag{5.2}$$

where the forward variable $\alpha_t(i) = \Pr(O_1, O_2, \ldots, O_t, q_t = q_i | \lambda)$ represents the probability of the partial observable sequence $O_1, O_2, \ldots, O_t$ until time t and state q_i at time t, considering model λ.

Backward Algorithm:

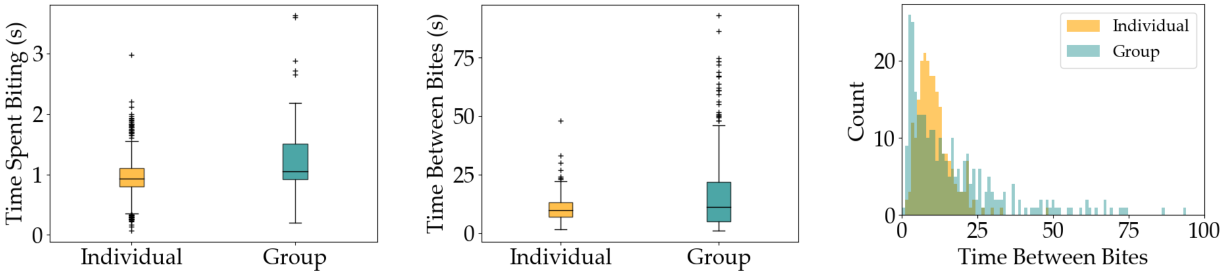
$$\hat{\beta}(i) = \beta_t(i) \prod_{k=t+1}^{T} \mathbf{J}_k \tag{5.3}$$

where $\mathbf{J}_k$ is the normalizer

$$\hat{\beta}_t(i) = \beta_t(i) = 1 \tag{5.4}$$

$$\hat{\beta}_t(i) = \mathbf{J}_{t+1} \sum_{j=1}^{N} \hat{\beta}_t(j) a_{ij} b_j(O_{t+1}), 1 \leq t \leq T \tag{5.5}$$

where the backward variable $\beta_t(i) = \Pr(O_{t+1}, O_{t+2}, \ldots, O_T | q_t = q_i, \lambda)$ represents the probability of the partial observation sequence from $t + 1$ to the end, given state q_i at time t, and the model λ.



(a)

(b)

(c)

Figure 5.7: Comparison between group and individual bite timing. The amount of time spent transferring food from the fork to the mouth is consistent between group and individual settings (a). The time between bites has the same mean, but differing distributions (b,c).

Once the model parameters have been reestimated, the most likely state sequence is determined by applying the Viterbi Algorithm [Rabiner and Juang, 1986]. Finally, to evaluate the model's effectiveness at predicting the correct hidden state, we compared hand-labeled states to the model's prediction on the corresponding observation sequence.

The correct state was predicted 61.22% of the time. One reason that this we did not see better performance is that the observations for `waiting with fork full` and `waiting with fork empty` are very similar, so the HMM has to rely more heavily on the transition probabilities. However, a more meaningful metric would be how accurately the HMM predicted the correct bite timing, since it is this state transition we are the most interested in. To do so, we used the hand-labeled states to identify when bites were initiated, and then calculated the offset between the true bite times and those that were predicted using the HMM. The average bite timing error was 1.57 seconds, and 90% of the bite timing predictions had an error of less than 2 seconds.

We tested to see if performance would increase by using a simpler model with only two hidden states, $Q =$ {`biting`, `not biting`}. We computed the new parameter initialization in the same way as with four states. With the 2-state HMM, the correct state was predicted 61.9% of the time, which is comparable to the 4-state HMM. However, when we look at the bite timing error, using only 2 states had an average of 3.53 seconds of error and only 46% of the bite timing predictions had an error of less than 2 seconds. For state estimation, using a simple 2 state representation (biting or not biting) yields results comparable to the more complex 4 state representation. However, when it comes to predicting bite timing, we can conclude that including the intermediate waiting states in the HMM provides valuable information.

5.1.2 *Differences Based On Group Size*

We considered that individuals eating alone may have a different bite frequency than those eating in groups because they are not simultaneously carrying on a conversation. To test this premise, we compare both the time between bites and the time spent actually taking a bite (transferring food from the fork to the mouth) in fig. 5.7. We would expect that the transfer of food to the mouth would be consistent since it is a quick physical act that is performing qualitatively the same way individually or in a group setting, and quantitatively we found this to be the case.

The time spent between taking bites does vary based on the group size. Individuals have a much more normally distributed bite separation and a lower average time between bites ($M = 10.83$ seconds for individuals and $M = 16.81$ seconds for groups). People eating in groups had a much larger variance in bite separation ($SD = 16.60$ for groups versus $SD = 5.92$ for individuals), with a heavy left-skew.

The difference in variance can be explained by the individual having fewer distractions and demands on their mouth and attention. The individual diners' biting rate is constrained only by the rate that they chew and the rate that they acquire the next piece of food. While it may be influenced by other factors such as the type of food being chewed/acquired and the size of the bite, people were fairly consistent when eating alone. People eating in groups alternate speaking. If they are the speaker, bites become further spaced out. If they are the listener, bites are taken more quickly to make up for the slower eating times in an attempt to target a similar meal completion time as the other diners. Because we examined group sizes of 2-4, most people were listeners at any given time, which may explain the left-skewed distribution.

To test whether the individual case versus the group case need to modeled separately, we trained an HMM as described in section 5.1.1.4 on only individual diners and a separate HMM on only groups of diners. Once trained, we tested each to see how well they generalized to the other condition. The results are displayed in table 5.2.

		Tested on		
		Individual	Group	Both
Trained on	Individual	2.56	3.33	2.65
	Group	2.26	0.72	1.41
	Both	1.83	1.40	1.57

Table 5.2: Average bite timing error measured in seconds for a HMM with different training sets. 70% of the data was used for training and 30% for validation.

The model trained only on individuals had the poorest performance because many of the social cues are no longer useful - no one

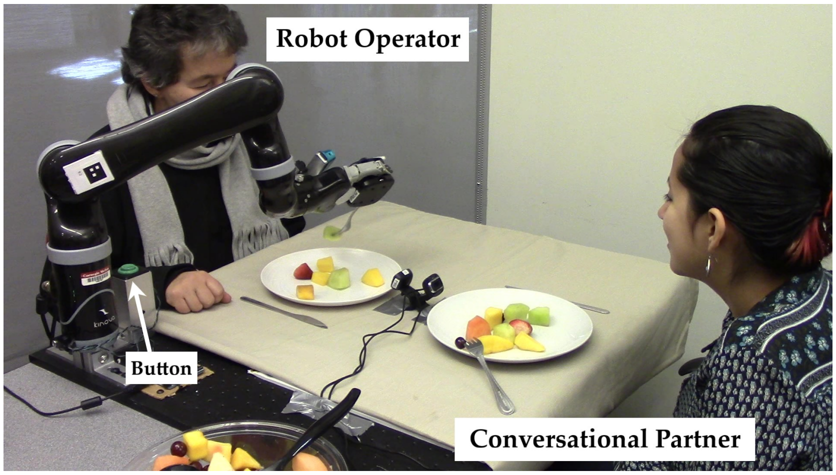


Figure 5.8: Experimental setup for the evaluation of the bite timing model.

is talking so there is no change in audio, gaze is almost entirely directed towards the plate throughout the meal, etc. We again chose to evaluate model performance by the average bite timing area instead of the percentage of correct hidden state estimation since the purpose of this model is to predict when to give a bite, not necessarily to predict the other phases of the eating process.

5.2 *Evaluation of Bite Timing Control Techniques*

To evaluate the bite timing model in practice, we ran a study to determine how using different methods of controlling the bite timing impacted the subject's ability to effectively eat and interact.

Experimental Setup In this study, two participants eat a meal together. We chose to use pairs of participants instead of groups of 3 or 4 as in the prior study in order to reduce the complexity of the interaction and because the model learned on pairs did not yield significantly different results than the model learned on 3 or 4 people. One of the participants is selected to be the robot operator and uses the MICO robot arm to eat with. The other participant sits across the table and eats their food under their own power. Throughout the meal, both participants are encouraged to engage in conversation with each other while eating.

Manipulated Factors We manipulated one factor: the method used to initiate bite delivery to the robot operator. In the manual condition, the robot operator is responsible for triggering the delivery of a bite of food from the robot via pressing a large physical button located near the base of the robot. This condition is designed to mimic the operation of commercial stationary feeding aides, such as those described in section 2.3.2. In the regular intervals condition, the robot brings a bite to the operator's mouth at regular time intervals, without any input from the operator. To choose the amount of time to use between bites, we analyzed the collected bite timing data from groups and individuals and chose to use the average across all users of 14.23 seconds between bites. In the socially aware condition, the robot brings a bite to the operator's mouth only when the bite timing model indicates that the robot operator would like to take a bite according to the bite timing model. Each subject saw only one condition, making this a between subjects study.

Procedure Each participant was asked to rate their level of hunger prior to the trial. For each pair of participants, one was randomly selected via coin toss to operate the robot. In the 3 instances where only one participant was present for their trial, they were given the role of robot operator. The participants were seated across from one another and food was then distributed on each of their plates. We explained that the robot would bring the food to the same location in front of the mouth for each bite, and that the robot operator would need to lean forward to manually take the piece of food from the fork – i.e. the robot would not be detecting and trying to approach the robot operator's mouth. We calibrated the stationary set point for the operator's height and comfort.

We ran the robot through several bite sequences, demonstrating the movement of the robot, and if applicable how it responded to the button pressing. Both participants were part of this training process. The conversational partner was given a list of topics and starter questions to help get the conversation going. The content of these questions pertained mostly to general background information such as "What is your career or what are you studying" rather than controversial questions such as politics or religious beliefs. We added the question prompts after a pilot in which participants were asked merely to maintain a conversation with no direction to the discussion because we found in the pilot that the conversation centered around what the robot was doing, and therefore was not an accurate representation of normal conversations.

Measures To evaluate the efficacy of food consumption, the total time it takes for the meal to be completed was recorded. To allow for possible robot failures in automatically acquiring pieces of food from the plate, any time the robot wasted due to unsuccessfully attempting to acquire food was subtracted from the total time.

To evaluate the social fluency, a survey was given to both participants to capture their impressions of the interaction. We asked robot operators to rate the timing of the bites of the robot and whether the robot was a distraction to their conversation. We asked both partici-

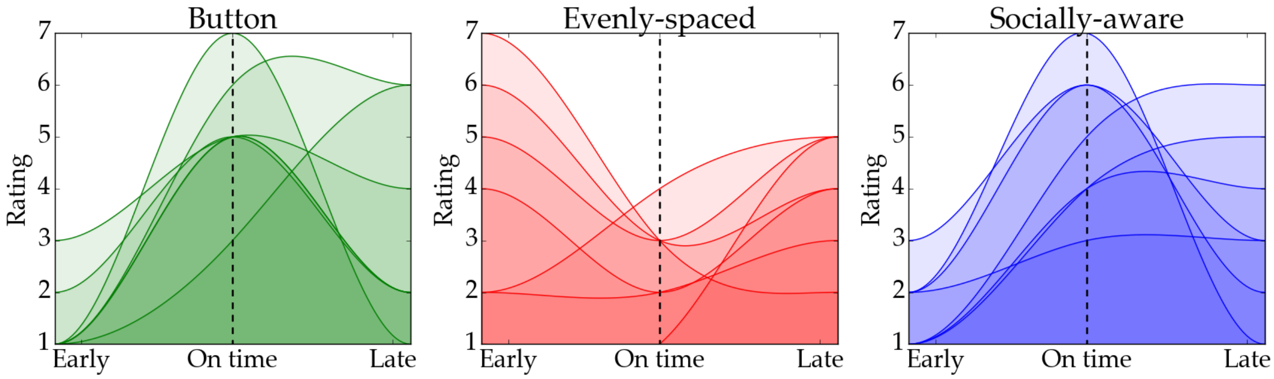


Figure 5.9: Raw scores fit with curves for each participant in the 3 bite-triggering conditions of how much they agree that the robot delivered bites early, on time, or late, on a 7-point Likert scale.

pants to complete the Muir Trust Questionnaire and the NASA Task Load Index.

Participants We performed 21 sessions with 2 people each, and randomized which of the 3 conditions was used. We also performed an additional 3 sessions in which one participant was absent, so the role of conversational partner was left empty and were therefore not included in the following statistical analyses. One of these sessions was in the button-pressing condition, and 2 were in the evenly-spaced condition.

Analysis We asked each robot operator in different parts of the survey to rate statements about whether the robot delivered a bite too early, at the desired time, or too late. The raw scores for each user and a curve connecting them to roughly approximate the distribution as perceived by each user is shown in fig. 5.9. We combined these responses over each condition and normalized so that each user's scores would sum to the same ammount in order to compare across conditions as shown in fig. 5.10. We found that the button-pressing condition was the most closely aligned with the bite timing that people desired, which confirms that operator-triggered bites are the gold standard to which we should be comparing our model's results. The evenly-spaced condition had the most variability, and we saw a dichotomy of ratings in which some users found the bites to be too early and others too late. This is consistent with using the average bite spacing, some users are sure to eat more slowly, and some are sure to eat more quickly. It underlies the need either for customization to each user in the even spaced condition or a model that can handle differences among operators. The socially-aware condition received ratings consistent with the button-pushing gold standard.

For both the evenly-spaced condition and the socially-aware con-

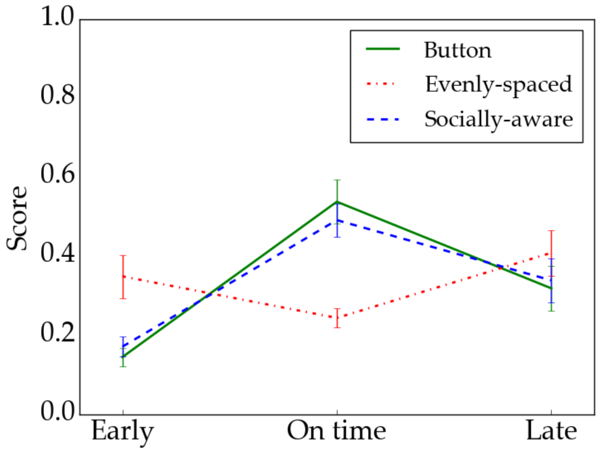


Figure 5.10: Comparison between normalized scores for each of the 3 bite timing conditions.

dition, there were occasions where the robot was not fast enough at acquiring the next bite of food to deliver it when the model would dictate the next bite should be delivered. However, this was only true in 3.4% of bites in the evenly-spaced condition and 4.9% in the socially-aware condition, and produced a 1.27 second delay on average. The delays occurred in part due to occasional failure of the acquisition process to actually gather a bite. In the event that the robot did not successful skewer a piece of food – which was observed and recorded by the experimenter – the robot would attempt to skewer another piece of food, continuing until a piece was successfully acquired. This way every time the robot brought the fork to the user's mouth, it had a morsel of food on it. The downside of this technique is that on rare occasion gathering a morsel of food took longer than the interval between bites. Watching the robot fail to collect a piece of food on its first try probably impacted the trust participants placed in the robot, but as it was consistent across all conditions, it does not impact conclusions drawn between the conditions.

In addition to questions about the bite timing, we asked participants two questions about whether they were able to carry on a normal conversation and whether they felt distracted by the robot. We asked two questions to both the robot operator and the conversational partner. We found that both participants in each trial did not find the robot to be a distraction to their conversation ($M = 2.90$, $STD = 1.22$ overall on a 1-7 scale, below the neutral response of 4, with $t(42) = 6.61$, $p < .0001$) and that these ratings were not significantly different for the social condition.

In general, we found that the robot operator and the conversational partner did (who was not being fed with the robot) had approximately equal levels of trust in the robot system. This was somewhat surprising as the robot was physically much closer to the operator than to the conversational partner, which we would have expected to lead to higher anxiety in the robot operator. However, in the button-pressing case, the robot operator had significantly *more* trust in the robot than their conversational partner ($t(7) = 4.86$, $p = .0046$ for a paired t-test). We speculate this is due to the operator selecting when the robot is to present a bite, thus increasing their perceived level of control and predictability of the robot. The button was positioned in such a way that it was visible to the conversational partner when it was pressed, but between maintaining a conversation and acquiring their own food to eat, the conversational partner was not always aware of when the button had been pressed. In the evenly-spaced and socially-aware conditions we saw no significant difference between the trust level of the robot operator and the conversational partner.

The results of the NASA TLX for measuring cognitive load revealed that controlling the robot was not a draining task for participants across all conditions (average score of 3.12 out of a maximum of 20 for high load). We asked the conversational partners to also fill out the NASA TLX for the task of manually feeding themselves with a normal fork to use as a comparison point. The difference between the robot operator and the conversational partner is significant, but not between bite timing conditions.

We found that how hungry the participants were was not strongly correlated with their responses on the robot's timing, trust, or cognitive load.

5.3 *Discussion*

From this work, we have established that timing plays an important role in robotic feeding, just as it does in many other applications involving robots such as handovers [Strabala et al., 2013], theatrical performances [Zeglin et al., 2014], and dialogue [Fong et al., 2003]. The consequences of presenting a bite to the diner earlier than expected can include an interruption to conversation or to finishing chewing the prior bite and is poorly tolerated. The consequences of presenting a bite later than desired can include frustration towards the robot and disruption of the natural flow of conversation during the meal.

Through a user study to evaluate different levels of control over how to trigger a bite, we identified that individual people have differing preferences. This indicates that the level of control is not a "one size fits all" solution. These studies were performed with able-bodied users, and if there is already variation in opinions within this group of subjects, varied physical and cognitive abilities will surely expand the variance in opinion. This highlights the need for individualization not only among the physical interfaces for people with disabilities, but also among the level of control that is entrusted to the robot and the operator respectively.

6
Future Work

6.1 Teleoperation and Modal Control

6.1.1 Generalizing the Time-Optimal Mode Switching Model

In this work, we used a provided goal location for users. This made the prediction of mode switches simpler because the robot did not also have to reason over a probability distribution of goals. If there were several equally good goals in a particular scenario, our time-optimal mode switching strategy would still be effective, but if some goals were preferred more than others, the robot would need to maintain a distribution over goals and rewards respectively.

Along the same lines, we considered a single-shot action with a single goal at the end. In reality, tasks often require sequences of actions, each with their own sub-goals. Take the example of dialing a telephone. First the receiver must be grasped. Second, it must be placed on the table. Third, individual keys must be pressed on the telephone's base. Breaking down a task into logical subtasks is a prerequisite for applying the mode-switching model. For well-studied interactions, such as grasping, studies of human hand movement have been used to identify phases of the task [Kang and Ikeuchi, 1995].

While our studies involved exclusively able-bodied subjects, we want to see how these results, particularly those relating to acceptance, generalize to people with disabilities. Study 3, as described in section 3.3.0.1, was restricted to a 2D point robot and we are looking to reconduct this experiment on the MICO robot arm. The optimal mode regions will become optimal mode volumes, and the assisted mode switching will occur when the robot enters a new volume.

In summary, next steps for generalizing the time-optimal mode switching model are:

1. Predict mode-switching over a distribution of goals instead of a single known goal.

2. Segment tasks into discrete sub-tasks to apply the mode-switching model.

3. Expand studies to include people with disabilities.

6.1.2 Alternatives to Automatically Changing Mode

How best to improve modal control is an open question. Here we presented one technique: having the robot perform mode switches automatically.

An alternative form of assistance would be to have a priority ordered list of modes. With a single button press the user could transition to the most likely next mode, as estimated by the robot. In the case of an incorrect ordering the user would cycle through the list. This kind of assistance would complement Kinova's LCD screen as described in section 3.1.1. There are also prosthetics in which the user must select which axis to control with their shoulder motion. By reordering the list of modes, it would reduce the physical strain needed to change modes, while still giving the operator full control over mode selection.

We have suggested a method for more easily switching modes. Another approach is to remove mode switching entirely by having the user only control one mode. The remaining modes would be controlled by the robot, in a type of assistance we have defined as *extramodal assistance* [Herlant et al., 2016]. When using extramodal assistance, avoiding mistakes in goal prediction becomes increasingly important. In the case of the robot moving in an undesirable way in the non-controlled modes, the user would have to change modes manually, correct the mistake and then revert back to their original task, causing a costly interruption.

If we can redefine the mapping between operator inputs and the robot's position, we can also remove the need for modal control. In normal teleoperation with a joystick for example, each axis of the joystick corresponds to either moving along a Cartesian plane or rotating about an axis through the origin of the robot's gripper. One exception is when the robot is controlled in "drinking mode" where pushing one axis on the joystick controls the gripper's position along an arc with a predefined diameter and center [Campeau-Lecours et al., 2016] as shown in fig. 6.1. We could take this idea further and identify a sub-manifold needed for the task of eating. The two axes on the joystick would then map to coordinates on the sub-manifold. This strategy may be successful for a number of repeated tasks (like drinking and eating) but if it becomes a general solution, it could result in long lists of specialized modes which will need to be traversed as an additional step before task execution.

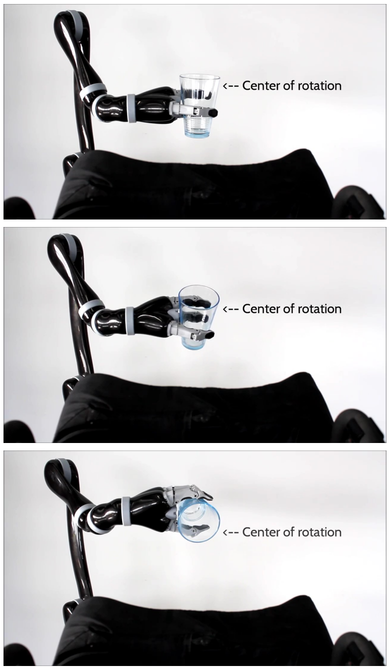


Figure 6.1: Drinking mode, in which tilting the joystick moves the gripper to rotate a specified radius around a given point - configured to be the rim of a particular glass.

By investigating modal control and helpful interventions we strive to close the gap between what users want to do with their assistive arms and what they can achieve with ease. In summary, next steps for exploring assistance strategies besides automatically changing modes are:

1. Reorder the list of modes based on probability of switching to that mode given the goal.

2. Have the robot control modes that are not being actively controlled by the operator.

3. Map the control input to a 2D sub-manifold within the 6D robot workspace for specific tasks.

6.2 *Food Manipulation*

Generalizing to other types of foods may be tricky if the physical properties are very different from those with which the robot has been trained. A possible way to reduce training and increase the model accuracy would be to perform a pre-classification where the robot first identifies types of food on the plate and then learns the mapping to acquisition parameters for each of them separately. Food identification for a full plate of food has been looked at in the context of estimating dietary value for a plate of food, but not much work has been done in identifying specific regions which contain one food or another. A few frameworks exist for automatically detecting and creating specialized skills [Stulp et al., 2014] which could also be applied.

This work could be extended beyond predicting skewering success to predicting the next state of the plate after a skewering action has been taken. In effect, predicting o_{t+1} from o_t and θ_t. Learning this transition between states could be used in lieu of a physics model for interacting with food to create a data-driven food simulator. Having access to such a simulation would enable longer-horizon food manipulation planning, which could include preparatory motions such as pushing food into a pile so it is easier to skewer or pushing food to the edge of the plate where it is easier to scoop.

The skewering motion used in this section is completely open loop – the robot takes sensor readings of the plate before skewering and after skewering, but is effectively blind during the actual execution. This limitation means that actively changing forces applied or using mitigating strategies for food slipping off the fork is not possible. Positioning a camera along the fork axis would enable live feedback for the robot to make such adjustments and probably increase the rate of success in gathering bites. A sensor at that angle could also be used to visually servo the fork into the operator's mouth in a safe way.

Finally, we could expand our taxonomy of food acquisition primitives by collecting fork trajectories from people as they eat and performing a clustering analysis on the trajectories. It may be that there are in fact other primitives or sub-primitives that are difficult to identify visually, but that can be quantified through trajectory analysis. We did not catalog the variety of preparatory motions or non-acquisition contact that was made with the food since that was not our focus on this work, but it could open an interesting planning problem akin to repositioning or orienting an object before grasping. Methods for performing the automatic segmentation are usually based on using a Hidden Markov Model to identify key points to generalize between trajectories and then Dynamic Movement Primitives to parametrize the trajectories [Vakanski et al., 2012, Niekum et al., 2012] but methods are being developed which can segment and parametrize the trajectories simultaneously via Gaussian Mixture Models [Lee et al., 2015] or perform probabilistic segmentation [Kulic et al., 2009].

In summary, next steps for generalizing and expanding our food manipulation approach are:

1. Identify regions of food on the plate first, and then apply the learned acquisition strategy for each region.

2. Predict how the plate of food will look after an action is performed to create a data-driven food simulator for longer-horizon planning.

3. Add sensors to enable reactive acquisition strategies that can respond to food slippage and use visual servoing.

4. Quantify other acquisition and non-acquisition primitives for planning.

6.3 HRI Implications

One limitation of the socially-aware bite timing model we developed is that it does not account for changes in bite timing due to the type of food being consumed. For example, some fibrous foods take longer to chew and would slow down the frequency of bites. Some foods are harder to acquire, such as spaghetti which may take a long sequence of actions to neatly bring to the mouth, also slowing down the frequency of bites. Other foods, by their very nature require a different eating rhythm. Raclette and fondue both rely on melting cheese during the meal, enforcing constraints on when diners can eat.

We can foresee a more advanced system which can use information gained during the food acquisition process as well as the observed social cues to predict bite timing.

In this work, we looked at bite timing and comparing different triggering strategies which give more or less control to the robot operator. The question of how to balance control between the human operator and the robot's algorithms is tricky and depends significantly on context, the human and robot's capabilities, and the human's preference. Even within the context of feeding, there are additional control parameters that would be compelling to explore; in our studies, the operator could not control which food was being acquired, or the path the robot took to get the food.

To develop a useful assistive feeding system, a user-centric design is critical. One aspect of having a target population with limited physical mobility means that the user will have limited ability to change their viewpoint. The further we move towards an autonomous system, the more important it may be for the robot to avoid blocking the user's view of important objects in the scene. For example in feeding, it might be important to have the end of the fork always be visible for the user to feel comfortable during autonomous operation. A next step would be to collect eye gaze data from people performing manipulation and feeding tasks in order to optimize the robot motion to maintain visibility of relevant parts of the scene.

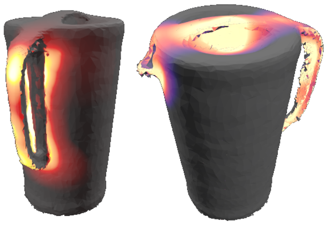


Figure 6.2: The pitcher textured with the gaze saliency calculated using simulated gaze data.

We can think of the operator's eyes as a spotlight that is pointed at relevant parts of the scene (fig. 6.3). We accumulate the amount of light that is absorbed by each object's surface over the course of task execution to tell how salient that part of the object is. But the eye gaze is not a laser beam penetrating one object at a single point, it diffuses with distance forming a cone. As it spreads, we will decrease the amount being added to the salience score as a function of the distance to the center of the gaze direction. Since the gaze saliency is now a property of each object surface, the gaze saliency can be computed off-line from collected gaze data as a texture for the object mesh (section 6.3). Finally, the saliency could be used to inform the robot's motion planning to optimize visibility of salient parts of the scene.

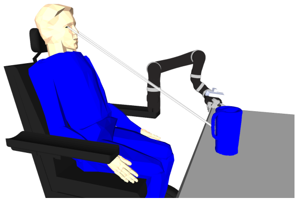


Figure 6.3: Visualization of a single moment in time of gaze and gaze target.

In summary, next steps to explore the HRI implications of eating with an assistive robot arm are:

1. Add the type of food as an input to the learned bite-timing model.
2. Compare different levels of user vs. robot control using the models created in this thesis for mode switching, bite timing, bite selection, etc.
3. Incorporate visibility metrics into autonomous motions of the

robot.

6.4 Conclusion

We identified that switching between control modes is a critical stumbling block to smooth teleoperation for assistive robot arms. We modified a standard occupational therapist test to quantify the impact of mode switching. We then used insights from a lower-dimensional teleoperated robot to build a time-optimal mode switching model to predict when robot operators would change modes given certain assumptions about their goals. We evaluated the time-optimal mode switching model and found it did not decrease task performance and was preferred by most participants. By predicting when users switch control modes, we can gain insights to the design of teleoperation systems and open the door for new types of assistance.

Due to the wide and varying physical properties of food, we chose to use a data-driven approach to food collection. We classified observed food gathering strategies and created a taxonomy of food collection primitives. We implemented the skewering primitive for the robot to perform autonomously. The parameters for the food gathering strategy were learned through a series of robot trials, in which it attempted to pick up pieces of fruit from a plate. We limited the design to be feasible in a realistic home or restaurant setting by adding different perspectives, and randomizing food placement on the plate. By directly learning the robot's policy parameters, we can easily expand on this work to use a wide array of motion primitives or start to include force control.

We have shown the importance of social cues while eating with a robot, and trained a model to use them to successfully predict when the operator wants to take a bite of food. We used this model to perform autonomous feeding with a robot arm and evaluated several levels of operator control. We have gained insights into the relationship between a person and their human/robot dining assistant engaged in the intimate activity of feeding.

It will be challenging for our work to date to keep up with the new control interfaces and exotic food dishes that are being frequently introduced. As with any data-driven approach, our food acquisition technique is only as good as the data it is trained with, and as the parameterized acquisition policies the robot may execute. Learning by albeit slower but more accurate user demonstrations would be a promising way to jump-start the learning process with new foods. The social culture around the act of eating is different across societies and may also need be reevaluated for other countries and communities.

Overall, we believe that we have clearly presented the shortcomings and potential ways to mitigate teleoperation systems that use modal control and that we presented techniques that can be used to successfully manipulate food material and collaborate with the operator to create a seamless dining experience. We look forward to the application of these techniques in commercially available products that will positively impact people's lives.

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Henny Admoni, Anca Dragan, Siddhartha S Srinivasa, and Brian Scassellati. Deliberate delays during robot-to-human handovers improve compliance with gaze communication. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 49–56. ACM, 2014.

Abidemi Bolu Ajiboye and Richard F. ff. Weir. A heuristic fuzzy logic approach to EMG pattern recognition for multifunctional prosthesis control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(3), 2005.

Aitor Aldoma, Thomas Fäulhammer, and Markus Vincze. Automation of "ground truth" annotation for multi-view RGB-D object instance recognition datasets. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5016–5023. IEEE, 2014.

Redwan Alqasemi and Rajiv Dubey. Maximizing manipulation capabilities for people with disabilities using a 9-dof wheelchair-mounted robotic arm system. In *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*, pages 212–221. IEEE, 2007.

Redwan M. Alqasemi, Edward J. McCaffrey, Kevin D. Edwards, and Rajiv V. Dubey. Analysis, evaluation and development of wheelchair-mounted robotic arms. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 469–472. IEEE, 2005.

Catherine M. Arrington and Gordon D. Logan. The cost of a voluntary task switch. *Psychological Science*, 15(9):610–615, 2004.

Elsy Athlin and Astrid Norberg. Interaction between the severely demented patient and his caregiver during feeding. *Scandinavian Journal of Caring Sciences*, 1(3-4):117–123, 1987.

Elsy Athlin, Astrid Norberg, and Kenneth Asplund. Caregivers' perceptions and interpretations of severely demented patients during feeding in a task assignment system. *Scandinavian Journal of Caring Sciences*, 4(4):147–156, 1990.

Michael Baker and Holly A. Yanco. Autonomy mode suggestions for improving human-robot interaction. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2948–2953, 2004.

Susan Barreca, Carolyn Gowland, Paul Stratford, Maria Huijbregts, Jeremy Griffiths, Wendy Torresin, Magen Dunkley, Patricia Miller, and Lisa Masters. Development of the chedoke arm and hand activity inventory: theoretical constructs, item generation, and selection. *Topics in stroke rehabilitation*, 11(4): 31–42, 2004.

Zeungnam Bien, Dae-Jin Kim, Myung-Jin Chung, Dong-Soo Kwon, and Pyung-Hun Chang. Development of a wheelchair-based rehabilitation robotic system (kares ii) with various human-robot interaction interfaces for the disabled. In *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, volume 2, pages 902–907 vol.2, July 2003. DOI: 10.1109/AIM.2003.1225462.

Mario Bollini, Jennifer Barry, and Daniela Rus. Bakebot: Baking cookies with the pr2. In *The PR2 Workshop: Results, Challenges and Lessons Learned in Advancing Robots with a Common Platform, IROS*, 2011.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461, 2014.

G. Bourhis and M. Sahnoun. Assisted control mode for a smart wheelchair. In *IEEE International Conference on Rehabilitation Robotics*, pages 158–163. IEEE, 2007.

Tadhg Brosnan and Da-Wen Sun. Inspection and grading of agricultural and food products by computer vision systems–a review. *Computers and Electronics in Agriculture*, 36(2):193–213, 2002.

Daniel Cagigas and Julio Abascal. Hierarchical path search with partial materialization of costs for a smart wheelchair. *Journal of Intelligent and Robotic Systems*, 39(4):409–431, 2004.

Alexandre Campeau-Lecours, Véronique Maheu, Sébastien Lepage, Hugo Lamontagne, Simon Latour, Laurie Paquet, and Neil Hardie. Jaco assistive robotic device: Empowering people with disabilities through innovative algorithms. In *Rehabilitation Engineering and Assistive Technology Society of North America (RESNA) Annual Conference*, 2016.

E. Chaves, A. Koontz, S. Garber, R. Cooper, and A. Williams. Clinical evaluation of a wheelchair mounted robotic arm. *RESNA Presentation given by E. Chaves*, 2003.

A. Chiò, A. Gauthier, A. Vignola, A. Calvo, P. Ghiglione, E. Cavallo, A. A. Terreni, and R. Mutani. Caregiver time use in ALS. *Neurology*, 67(5):902–904, 2006.

Jun-Uk Chu, Inhyuk Moon, and Mu-Seong Mun. A real-time EMG pattern recognition system based on linear-nonlinear feature projection for a multifunction myoelectric hand. *IEEE Transactions on Biomedical Engineering*, 53(11):2232–2239, 2006.

C. Chung, H. Wang, M. J. Hannan, A. R. Kelleher, and R. A. Cooper. Daily task-oriented performance evaluation for commercially available assistive robot manipulators. *International Journal of Robotics and Automation Technology*, 2016.

C. S. Chung and R. A. Cooper. Literature review of wheelchair-mounted robotic manipulation: user interface and end-user evaluation. In *RESNA Annual Conference*, 2012.

Florence S. Cromwell. *Occupational Therapist's Manual for Basic Skills Assessment Or Primary Pre-vocational Evaluation*. Fair Oaks Print. Company, 1960.

Mark R Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on robotics and automation*, 5(3):269–279, 1989.

Fernando De la Torre, Wen-Sheng Chu, Xuehan Xiong, Francisco Vicente, Xiaoyu Ding, and Jeffrey Cohn. Intraface. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, volume 1, pages 1–8. IEEE, 2015.

Treena Delormier, Katherine L. Frohlich, and Louise Potvin. Food and eating as social practice–understanding eating patterns as social phenomena and implications for public health. *Sociology of Health & Illness*, 31(2):215–228, 2009.

Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1): 269–271, 1959.

K Ding and S Gunasekaran. Shape feature extraction and classification of food material using computer vision. *Transactions of the ASAE*, 37(5):1537–1545, 1994.

Guo Dong and Ming Xie. Color clustering and learning for image segmentation based on neural networks. *IEEE transactions on neural networks*, 16(4):925–936, 2005.

A. Dragan and S. Srinivasa. Formalizing assistive teleoperation. In *Robotics: Science and Systems*, July 2012.

Anca D. Dragan and Siddhartha S. Srinivasa. A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805, 2013.

B J F Driessen, H G Evers, and J A v Woerden. Manus–a wheelchair-mounted rehabilitation robot. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 215(3):285–290, 2001. DOI: 10.1243/0954411011535876. URL `https://doi.org/10.1243/0954411011535876`. PMID: 11436271.

Cheng-Jin Du and Da-Wen Sun. Learning techniques used in computer vision for food quality evaluation: a review. *Journal of food engineering*, 72(1):39–55, 2006.

K. Edwards, R. Alqasemi, and R. Dubey. Design, construction and testing of a wheelchair-mounted robotic arm. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3165–3170, May 2006. DOI: 10.1109/ROBOT.2006.1642183.

Håkan Eftring and Kerstin Boschian. Technical results from Manus user trials. *International Conference On Rehabilitation Robotics*, pages 136–141, 1999.

Linda Fehr, W Edwin Langbein, and Steven B Skaar. Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey. *Journal of rehabilitation research and development*, 37 (3):353, 2000.

S. Mohammad P. Firoozabadi, Mohammad Reza Asghari Oskoei, and Huosheng Hu. A human-computer interface based on forehead multi-channel bio-signals to control a virtual wheelchair. In *Proceedings of the 14th Iranian conference on biomedical engineering (ICBME)*, pages 272–277, 2008.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*, pages 726–740. Elsevier, 1987.

Terrence Fong, Charles Thorpe, and Charles Baur. Collaboration, dialogue, human-robot interaction. In *Robotics Research*, pages 255–266. Springer, 2003.

Ferran Galán, Marnix Nuttin, Eileen Lew, Pierre W. Ferrez, Gerolf Vanacker, Johan Philips, and J. del R. Millán. A brain-actuated wheelchair: asynchronous and non-invasive brain–computer interfaces for continuous control of robots. *Clinical Neurophysiology*, 119(9):2159–2169, 2008.

Samuele Gasparrini, Enea Cippitelli, Ennio Gambi, Susanna Spinsante, and Francisco Flórez-Revuelta. Performance analysis of self-organising neural networks tracking algorithms for intake monitoring using kinect. In *IET International Conference on Technologies for Active and Assisted Living (TechAAL)*, pages 1–6. IET, 2015.

Axel Gräser. Technological solutions to autonomous robot control. *Improving the Quality of Life for the European Citizen: Technology for Inclusive Design and Equality*, 4:234, 1998.

Elena Corina Grigore, Kerstin Eder, Anthony G Pipe, Chris Melhuish, and Ute Leonards. Joint action understanding improves robot-to-human object handover. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4622–4629. IEEE, 2013.

Jeong-Su Han, Z Zenn Bien, Dae-Jin Kim, Hyong-Euk Lee, and Jong-Sung Kim. Human-machine interface for wheelchair control with emg and its evaluation. In *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, volume 2, pages 1602–1605. IEEE, 2003.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Laura Herlant, Rachel Holladay, and Siddhartha Srinivasa. Assistive teleoperation of robot arms via automatic time-optimal mode switching. In *Human-Robot Interaction*, March 2016.

Günter Hetzel, Bastian Leibe, Paul Levi, and Bernt Schiele. 3d object recognition from range images using local feature histograms. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2001.

Henry Hexmoor. A cognitive model of situated autonomy. In *Pacific Rim International Conference on Artificial Intelligence*, pages 325–334. Springer, 2000.

Michael Hillman, Karen Hagan, Sean Hagan, Jill Jepson, and Roger Orpwood. The weston wheelchair mounted assistive robot-the design story. *Robotica*, 20(2):125–132, 2002a.

Michael Hillman, Karen Hagan, Sean Hagen, Jill Jepson, and Roger Orpwood. The Weston wheelchair mounted assistive robot – the design story. *Robotica*, 20:125–132, 2002b.

Michael Himmelsbach, Thorsten Luettel, and H-J Wuensche. Real-time object classification in 3d point clouds using point feature histograms. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 994–1000. IEEE, 2009.

Leigh R Hochberg, Daniel Bacher, Beata Jarosiewicz, Nicolas Y Masse, John D Simeral, Joern Vogel, Sami Haddadin, Jie Liu, Sydney S Cash, Patrick van der Smagt, et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372, 2012.

Alberto Jardón Huete, Juan G. Victores, Santiago Martinez, Antonio Giménez, and Carlos Balaguer. Personal autonomy rehabilitation in home environments by a portable assistive robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(4):561–570, 2012.

Catrine Jacobsson, Karin Axelsson, Per Olov Österlind, and Astrid Norberg. How people with stroke and healthy older people experience the eating process. *Journal of Clinical Nursing*, 9(2):255–264, 2000.

Sing Bing Kang and Katsushi Ikeuchi. Toward automatic robot instruction from perception-temporal segmentation of tasks from human hand motion. *IEEE Transactions on Robotics and Automation*, 11(5): 670–681, 1995.

Marjorie Kellor, J. Frost, N. Silberberg, I. Iversen, and R. Cummings. Hand strength and dexterity. *The American Journal of Occupational Therapy*, 25(2):77–83, 1971.

Dae-Jin Kim, Zhao Wang, and Aman Behal. Motion Segmentation and Control Design for UCF-MANUS–An Intelligent Assistive Robotic Manipulator. *IEEE/ASME Transactions on Mechatronics*, 17 (5):936–948, October 2012. ISSN 1083-4435. DOI: 10.1109/TMECH.2011.2149730.

Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from RGB-D videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

Kazuya Kubo, Takanori Miyoshi, and Kazuhiko Terashima. Influence of lift walker for human walk and suggestion of walker device with power assistance. In *Micro-NanoMechatronics and Human Science, 2009. MHS 2009. International Symposium on*, pages 525–528. IEEE, 2009.

Dana Kulic, Wataru Takano, and Yoshihiko Nakamura. Online segmentation and clustering from continuous observation of whole body motions. *IEEE Transactions on Robotics*, 25(5):1158–1166, 2009.

Vijay Kumar, Tariq Rahman, and Venkat Krovi. Assistive devices for motor disabilities. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 1997.

Yoshinori Kuno, Nobutaka Shimada, and Yoshiaki Shirai. Look where you're going [robotic wheelchair]. *IEEE Robotics & Automation Magazine*, 10(1):26–34, 2003.

Isabelle Laffont, Nicolas Biard, Gérard Chalubert, Laurent Delahoche, Bruno Marhic, François C Boyer, and Christophe Leroux. Evaluation of a graphic interface to control a robotic grasping arm: a multicenter study. *Archives of physical medicine and rehabilitation*, 90(10):1740–1748, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

Sang Hyoung Lee, Il Hong Suh, Sylvain Calinon, and Rolf Johansson. Autonomous framework for segmenting robot trajectories of manipulation task. *Autonomous robots*, 38(2):107–141, 2015.

Edmund LoPresti, David M. Brienza, Jennifer Angelo, Lars Gilbertson, and Jonathan Sakai. Neck range of motion and use of computer head controls. In *Proceedings of the fourth international ACM conference on Assistive technologies*, pages 121–128. ACM, 2000.

Deborah Lupton and Wendy Seymour. Technology, selfhood and physical disability. *Social science & medicine*, 50(12):1851–1862, 2000.

T Luth, Darko Ojdanic, Ola Friman, Oliver Prenzel, and Axel Graser. Low level control in a semi-autonomous rehabilitation robotic system via a brain-computer interface. In *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*, pages 721–728. IEEE, 2007.

Wen-Tao Ma, Wei-Xin Yan, Zhuang Fu, and Yan-Zheng Zhao. A chinese cooking robot for elderly and disabled people. *Robotica*, 29(6):843–852, 2011.

Veronique Maheu, Julie Frappier, Philippe S Archambault, and François Routhier. Evaluation of the jaco robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

Richard Mahoney. Robotic products for rehabilitation: Status and strategy. In *Proceedings of ICORR*, volume 97, pages 12–22, 1997.

Richard M Mahoney. The raptor wheelchair robot system. *Integration of assistive technology in the information age*, pages 135–141, 2001.

Christian Mandel, Thorsten Luth, Tim Laue, Thomas Rofer, Axel Graser, and Bernd Krieg-Bruckner. Navigating a smart wheelchair with a brain-computer interface interpreting steady-state visual evoked potentials. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1118–1125. IEEE, 2009.

David Marshall. Food as ritual, routine or convention. *Consumption Markets & Culture*, 8(1):69–85, 2005.

Bente Martinsen, Ingegerd Harder, and Fin Biering-Sorensen. The meaning of assisted feeding for people living with spinal cord injury: a phenomenological study. *Journal of Advanced Nursing*, 62(5):533–540, 2008.

Virgil Mathiowetz, Gloria Volland, Nancy Kashman, and Karen Weber. Adult norms for the box and block test of manual dexterity. *American Journal of Occupational Therapy*, 39(6):386–391, 1985.

Y Matsumotot, Tomoyuki Ino, and T Ogsawara. Development of intelligent wheelchair system with face and gaze based interface. In *Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on*, pages 262–267. IEEE, 2001.

Michelle McDonnell. Action research arm test. *Australian journal of physiotherapy*, 54(3):220, 2008.

Dennis J McFarland and Jonathan R Wolpaw. Brain-computer interface operation of robotic and prosthetic devices. *Computer*, 41(10), 2008.

Ross Mead and Maja J Matarić. The power of suggestion: teaching sequences through assistive robot motions. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 317–318. ACM, 2009.

Nachshon Meiran, Ziv Chorev, and Ayelet Sapir. Component processes in task switching. *Cognitive psychology*, 41(3):211–253, 2000.

Paul Michelman and Peter Allen. Shared autonomy in a robot hand teleoperation system. In *Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on*, volume 1, pages 253–259. IEEE, 1994.

Stephen Monsell. Task switching. *Trends in cognitive sciences*, 7(3):134–140, 2003.

AJung Moon, Daniel M Troniak, Brian Gleeson, Matthew KXJ Pan, Minhua Zheng, Benjamin A Blumer, Karon MacLean, and Elizabeth A Croft. Meet me where i'm gazing: how shared attention gaze affects human-robot handover timing. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 334–341. ACM, 2014.

Imad Mougharbel, Racha El-Hajj, Houda Ghamlouch, and Eric Monacelli. Comparative study on different adaptation approaches concerning a sip and puff controller for a powered wheelchair. In *Science and Information Conference (SAI), 2013*, pages 597–603. IEEE, 2013.

Bingbing Ni, Gang Wang, and Pierre Moulin. Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *IEEE Workshop on Consumer Depth Cameras for Computer Vision in conjunction with ICCV*, 2011. URL `http://www.adsc.illinois.edu/demos.html`.

Scott Niekum, Sarah Osentoski, George Konidaris, and Andrew G Barto. Learning and generalization of complex tasks from unstructured demonstrations. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5239–5246. IEEE, 2012.

Daisuke Nishikawa, Wenwei Yu, Hiroshi Yokoi, and Yukinori Kakazu. EMG prosthetic hand controller discriminating ten motions using real-time learning method. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '99)*, 1999.

Marnix Nuttin, Dirk Vanhooydonck, Eric Demeester, and Hendrik Van Brussel. Selection of suitable human-robot interaction techniques for intelligent wheelchairs. In *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*, pages 146–151. IEEE, 2002.

Luciano Oliveira, Victor Costa, Gustavo Neves, Talmai Oliveira, Eduardo Jorge, and Miguel Lizarraga. A mobile, lightweight, poll-based food identification system. *Pattern Recognition*, 47(5):1941–1952, 2014.

Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3400–3407. IEEE, 2011.

G Onose, C Grozea, A Anghelescu, C Daia, CJ Sinescu, AV Ciurea, T Spircu, A Mirea, I Andone, A Spânu, et al. On the feasibility of using motor imagery eeg-based brain–computer interface in chronic tetraplegics for assistive robotic arm control: a clinical test and long-term post-trial follow-up. *Spinal cord*, 50(8):599, 2012.

M. Palankar, K. J. De Laurentis, R. Alqasemi, E. Veras, R. Dubey, Y. Arbel, and E. Donchin. Control of a 9-dof wheelchair-mounted robotic arm system using a p300 brain computer interface: Initial experiments. In *2008 IEEE International Conference on Robotics and Biomimetics*, pages 348–353, Feb 2009a. DOI: 10.1109/ROBIO.2009.4913028.

Mayur Palankar, Kathryn J De Laurentis, Redwan Alqasemi, Eduardo Veras, Rajiv Dubey, Yael Arbel, and Emanuel Donchin. Control of a 9-dof wheelchair-mounted robotic arm system using a p300 brain computer interface: Initial experiments. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 348–353. IEEE, 2009b.

Johan Philips, José del R. Millán, Gerolf Vanacker, Eileen Lew, Ferran Galán, Pierre W. Ferrez, Hendrik Van Brussel, and Marnix Nuttin. Adaptive shared control of a brain-actuated simulated wheelchair. In *IEEE International Conference On Rehabilitation Robotics*, pages 408–414. IEEE, 2007.

Betsy Phillips and Hongxin Zhao. Predictors of assistive technology abandonment. *Assistive technology*, 5 (1):36–45, 1993.

Patrick M Pilarski, Michael R Dawson, Thomas Degris, Jason P Carey, and Richard S Sutton. Dynamic switching and real-time machine learning for improved human control of assistive biomedical robots. In *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, pages 296–302. IEEE, 2012.

Oliver Prenzel, Christian Martens, Marco Cyriacks, Chao Wang, and Axel Gräser. System-controlled user interaction within the service robotic control architecture massive. *Robotica*, 25(02):237–244, 2007.

S. D. Prior. An electric wheelchair mounted robotic arm-a survey of potential users. *Journal of medical engineering & technology*, 14(4):143–154, 1990.

Lawrence Rabiner and B Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

GertWillem RBE Romer and Harry JA Stuyt. Compiling a medical device file and a proposal for an international standard for rehabilitation robots. In *2007 IEEE 10th International Conference on Rehabilitation Robotics*, pages 489–496. IEEE, 2007.

Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348, 2010.

Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3384–3391. IEEE, 2008.

Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

Mitul Saha and Pekka Isto. Motion planning for robotic manipulation of deformable linear objects. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2478–2484. IEEE, 2006.

Erik Scheme and Kevin Englehart. Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6):643–660, 2011.

Joris Sijs, Freek Liefhebber, and Gert Willem RBE Romer. Combined position & force control for a robotic manipulator. In *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*, pages 106–111. IEEE, 2007.

Ann M. Simon, Levi J. Hargrove, Blair A. Lock, and Todd A. Kuiken. Target achievement control test: Evaluating real-time myoelectric pattern-recognition control of multifunctional upper-limb prostheses. *Journal of Rehabilitation Research & Development*, 48(6):619–628, 2011.

Tyler Simpson, Colin Broughton, Michel JA Gauthier, and Arthur Prochazka. Tooth-click control of a hands-free computer interface. *Biomedical Engineering, IEEE Transactions on*, 55(8):2050–2056, 2008.

Christer Sollerman and Arvid Ejeskär. Sollerman hand function test: a standardised method and its use in tetraplegic patients. *Scandinavian Journal of Plastic and Reconstructive Surgery and Hand Surgery*, 29(2): 167–176, 1995.

Won-Kyung Song, Jongbae Kim, Kwang-Ok An, In-Ho Lee, Won-Jin Song, Bum-Suk Lee, Sung-Il Hwang, Mi-Ok Son, and Eun-Chang Lee. Design of novel feeding robot for korean food. In *International Conference on Smart Homes and Health Telematics*, pages 152–159. Springer, 2010.

Ryoji Soyama, Sumio Ishii, and Azuma Fukase. 8 selectable operating interfaces of the meal-assistance device âĂIJmy spoonâĂİ. In *Advances in Rehabilitation Robotics*, pages 155–163. Springer, 2004.

Carol A Stanger, Carolyn Anglin, William S Harwin, and Douglas P Romilly. Devices for assisting manipulation: a summary of user task priorities. *IEEE Transactions on rehabilitation Engineering*, 2(4): 256–265, 1994.

Kyle Wayne Strabala, Min Kyung Lee, Anca Diana Dragan, Jodi Lee Forlizzi, Siddhartha Srinivasa, Maya Cakmak, and Vincenzo Micelli. Towards seamless human-robot handovers. *Journal of Human-Robot Interaction*, 2(1):112–132, 2013.

Tilo Strobach, Roman Liepelt, Torsten Schubert, and Andrea Kiesel. Task switching: effects of practice on switch and mixing costs. *Psychological Research*, 76(1):74–83, 2012.

Freek Stulp, Laura Herlant, Antoine Hoarau, and Gennaro Raiola. Simultaneous on-line discovery and improvement of robotic skill options. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 1408–1413. IEEE, 2014.

Kyoko Sudo, Kazuhiko Murasaki, Jun Shimamura, and Yukinobu Taniguchi. Estimating nutritional value from food images based on semantic segmentation. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 571–576. ACM, 2014.

Yuta Sugiura, Daisuke Sakamoto, Anusha Withana, Masahiko Inami, and Takeo Igarashi. Cooking with robots: designing a household system working in open environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2427–2430. ACM, 2010.

Alan Sunderland, Deborah Tinson, Lesley Bradley, and R Langton Hewer. Arm function after stroke. an evaluation of grip strength as a measure of recovery and a prognostic indicator. *Journal of Neurology, Neurosurgery & Psychiatry*, 52(11):1267–1272, 1989.

Edward Taub, Karen McCulloch, Gitendra Uswatte, David M Morris, Mary Bowman, Jean Crago, Danna Kay King, Staci Bishop, Francilla Allen, and Sherry Yakley. Motor activity log (mal) manual. 2011.

Francesco Tenore, Robert S. Armiger, R. Jacob Vogelstein, Douglas S. Wenstrand, Stuart D. Harshbarger, and Kevin Englehart. An embedded controller for a 7-degree of freedom prosthetic arm. In *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008.

Francesco V. G. Tenore, Ander Ramos, Amir Fahmy, Soumyadipta Acharya, Ralph Etienne-Cummings, and Nitish V. Thakor. Decoding of individuated finger movements using surface electromyography. *IEEE Transactions on Biomedical Engineering*, 56(5):1427–1434, 2009.

Joseph Tiffin and ESTON J Asher. The purdue pegboard: norms and studies of reliability and validity. *Journal of applied psychology*, 32(3):234, 1948.

Hylke A Tijsma, Freek Liefhebber, and J Herder. Evaluation of new user interface features for the manus robot arm. In *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pages 258–263. IEEE, 2005a.

Hylke A Tijsma, Freek Liefhebber, and J Herder. A framework of interface improvements for designing new user interfaces for the manus robot arm. In *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pages 235–240. IEEE, 2005b.

Michelle Tipton-Burton. Jebsen–taylor hand function test. *Encyclopedia of Clinical Neuropsychology*, pages 1365–1365, 2011.

Mike Topping. An overview of the development of handy 1, a rehabilitation robot to assist the severely disabled. *Journal of Intelligent and Robotic Systems*, 34(3):253–263, 07 2002. URL `http://login.ezproxy.lib.vt.edu/login?url=https://search.proquest.com/docview/881674364?accountid=14826`.

Katherine Tsui, Holly Yanco, David Kontak, and Linda Beliveau. Development and evaluation of a flexible interface for a wheelchair mounted robotic arm. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 105–112. ACM, 2008a.

Katherine M Tsui and Holly A Yanco. Simplifying wheelchair mounted robotic arm control with a visual interface. In *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*, pages 97–102, 2007.

Katherine M Tsui, Holly A Yanco, David J Feil-Seifer, and Maja J Matarić. Survey of domain-specific performance measures in assistive robotic technology. In *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, pages 116–123. ACM, 2008b.

Gitendra Uswatte, Edward Taub, David Morris, Mary Vignolo, and Karen McCulloch. Reliability and validity of the upper-extremity motor activity log-14 for measuring real-world arm use. *Stroke*, 36(11): 2493–2496, 2005.

Ravi Vaidyanathan, Monique Fargues, Lalit Gupta, Srinivas Kota, Dong Lin, and James West. A dual-mode human-machine interface for robotic control based on acoustic sensitivity of the aural cavity. In *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pages 927–932. IEEE, 2006.

Aleksandar Vakanski, Iraj Mantegh, Andrew Irish, and Farrokh Janabi-Sharifi. Trajectory learning for robot programming by demonstration using hidden markov model and dynamic time warping. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):1039–1052, 2012.

Diana Valbuena, Marco Cyriacks, Ola Friman, Ivan Volosyak, and Axel Graser. Brain-computer interface for high-level control of rehabilitation robotic systems. In *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*, pages 619–625. IEEE, 2007.

Dirk Vanhooydonck, Eric Demeester, Marnix Nuttin, and Hendrik Van Brussel. Shared control for intelligent wheelchairs: an implicit estimation of the user intention. In *Proceedings of the 1st international workshop on advances in service robotics (ASER'03)*, pages 176–182. Citeseer, 2003.

Margaret Visser. *The rituals of dinner: The origins, evolution, eccentricities, and meaning of table manners*. Open Road Media, 2015.

Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: science and systems*, volume 11, 2015.

Glenn Wylie and Alan Allport. Task switching and the measurement of âĂIJswitch costsâĂİ. *Psychological research*, 63(3-4):212–233, 2000.

Wei Liang Xu, JD Torrance, BQ Chen, Johan Potgieter, John E Bronlund, and J-S Pap. Kinematics and experiments of a life-sized masticatory robot for characterizing food texture. *IEEE Transactions on Industrial Electronics*, 55(5):2121–2132, 2008.

Akihiko Yamaguchi and Christopher G Atkeson. Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, pages 1045–1051. IEEE, 2016.

H. A. Yanco. *Shared user-computer control of a robotic wheelchair system*. PhD thesis, Citeseer, 2000.

Holly A Yanco. Wheelesley: A robotic wheelchair system: Indoor navigation and user interface. In *Assistive technology and artificial intelligence*, pages 256–268. Springer, 1998.

Garth Zeglin, Aaron Walsman, Laura Herlant, Zhaodong Zheng, Yuyang Guo, Michael C Koval, Kevin Lenzo, Hui Jun Tay, Prasanna Velagapudi, Katie Correll, et al. Herb's sure thing: A rapid drama system for rehearsing and performing live robot theater. In *Advanced robotics and its social impacts (ARSO), 2014 IEEE Workshop on*, pages 129–136. IEEE, 2014.

Jiannan Zheng, Z. Jane Wang, and Chunsheng Zhu. Food image recognition via superpixel based low-level and mid-level distance coding for smart home applications. *Sustainability*, 9(5):856, 2017.