University of "Roma Tre"

Doctoral Thesis

Design of a Robotic Rehabilitation Hand for Optimal Grasping

Author:
Giulia Franchi

Supervisor:
Stefano Panzieri

A dissertation submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science and Automation

at the

MCIP Lab
Computer Science and Automation Department

February 2016

“ai sogni inattesi...”

University of "Roma Tre"

Abstract

Compuer Science Engineering
Computer Science and Automation Department

Doctor of Philosophy in Computer Science and Automation

Design of a Robotic Rehabilitation Hand for Optimal Grasping

by Giulia Franchi

Building robotic devices able to replicate the human behavior to obtain motor recovery, functional substitution or human-robot interaction as human-like as possible, is becoming increasingly popular. In the last past years several robotic rehabilitation devices demonstrate how they can improve the performance of the rehabilitation therapy performed by a human therapist in terms of action repetition and accurate tracking of the desired trajectory. A proliferation of robotic devices for gait rehabilitation applications have been manufactured. Most of them may be intimidating to patients, others may require long and involved patient setups. Using a human-like robotic device, such as a robotic arm and a robotic hand, may be fruitful. Starting with the use of a multipurpose robotic arm to be attach to the human leg to automate gait training and relieve the physical burdens placed upon therapists, we design a mechanism able to attach the robot arm to a human limb, in a fast ans safety way. Afters studying grasping in industrial robot we developed a new robotic hand to be used with the robotic rehabilitation device: the Easyhand, a small. light, easy to build and to control robotic hand to be adapt to use with the multipurpose robotic arm to be attach to the human leg. This dissertation present a the design of a new robotic hand for optimal grasping in rehabilitation field.

Contents

List of Figures

List of Tables

Introduction

Biology is really usefull especially when applied to engineering challenges. Infact robotic systems, that are biologically inspired, recently become a good substitute to the human operator particularly in the field of medical robotics where human labor is required for rehabilitation therapies.

As Rod Grupen said:

> *[...]At bottom, robotics is about us. It is the discipline of emulating our lives, of wondering how we work.[...]*

Building robotic devices able to replicate the human behavior to obtain motor recovery, functional substitution or human-robot interaction as human-like as possible, is becoming increasingly popular. In the last past year severals robotic rehabilitation devices demonstrate how they can improve the performance of the rehabilitation therapy performed by a human therapist in terms of action repetition and accurate tracking of the desired trajectory. It is widely recognized that a human-inspired robotic rehabilitation therapy helps patients to re-learn movements, taking advantages from the plasticity of the neuromuscular system. Research has found that by actively engaging stroke patients in repetitive tasks, the brain is able to rewire neurological pathways to motor functions to relearn movement. The awareness and movement of hemi-paretic limbs can occur and functional recovery can continue even years after the brain injury. Restoration of walking ability is an important goal of rehabilitation but conventional gait training programs are often labor intensive. For example, in body-weight-supported treadmill training, physical therapists provide manual assistance to move a patient's leg and/or pelvis in a desired trajectory, which can demand high therapist effort.In this kind of walking recovery rehabilitaion programs there is a growing interest in using robot to automate gait training and relieve the physical burdens placed upon therapists. A proliferation of robotic devices for gait rehabilitation applications have been manufactured. Most of them may be intimidating to patients (e.g. a large exoskeleton or several wires and push-rods); others may require long and involved patient setups. Using a human-like

robotic device, such as a robotic arm and a robotic hand, may be fruitful. Specifically, adapting a commercially available robotic arm and a new light and Easyhand for gait rehabilitation may provide advantages over traditional approaches.

In fact, a robotic arm could perform many functions and could be more cost-effective than purchasing several specialized robots; robotic arms also have a human-like morphology, which may make the robot less alien to patients. Finally, using a made on purpose robotic hand should made the robotic arm able to alter gait by pushing and/or pulling on a person, made the rehabilitation process more human-like and not intimidating to patients, having better results than the human therapy because of the precision of tracking desired trajectories and easier for the therapist and for the patient. Using a robotic hand would allow researchers to focus on improving robot control algorithms for greater success in gait rehabilitation.

Grasping is one of the human skills that robotic researchers mostly attempt to imitating and robotic hands are one of the more designed and builded robotic parts. In robotic hands construction the inspiration of the human behavior is increasing. Analyzing grasping action performed by human beings and studying the anatomy of the human hand and of its behavior during grasping, it is possible to obtain necessary information to design new human-like robotic hands and new rehabilitation devices. The definition of the kinematic structure of the hand and of the fingers is, in fact, the basis for designing new dexterous robotic hands and devices devoted to interact with humans (such as rehabilitation devices). Therefore the first part of this work is focused on the study of an industrial robotic hand, providing the basis for a further study regarding the hand dynamics. Several experiments have been done for understanding and compare a human hand with a robotic one. In assistive robotics, such as restoration of walking or others rehabilitation therapies, the ability of tracking the desired trajectories with smooth movements and stable grasp is essential.

Therefore, one of the aims of this thesis is to develop a new robotic hand to be used with the robotic rehabilitation device: the Easyhand, a small. light, easy to build and to control robotic hand. The Easyhand is a 3D printed underactuated hand designed specifically to be mounted on the Baxter robot from Rethink robotics. Because this hand is designed specifically for Baxter, we are able to make some important simplifications in the design relative to other 3D printed hands. In particular, the Easyhand is smaller than most other 3D printed hands and it is powered by the native Baxter gripper actuator. As a result, our hand is cheaper, lighter, and easier to interface with than other robot hands available for Baxter. The Easyhand is an open-source project, which means all of the plans to make a robotic hand are published online with no patents, anyone has the right to make their own and even sell it themselves. The robotic hand I'm going

to present in this work is currently the most inexpensive "research-grade" robot hand available.

> *[...]In the future, I'm sure there will be a lot more robots in every aspect of life. If you told people in 1985 that in 25 years they would have computers in their kitchen, it would have made no sense to them.[...]* Rodney Brooks

Chapter 1

Overview on How to Use Grasping for Rehabilitation

Nowdays the interest for the realization of rehabilitation devices able to replicate the human behavior has had a strong increase [1], [2]. An accurate analysis of the robotics field applied to rehabilitation has shown as the bio-inspiration enables:

- designing rehabilitation robotic devices assuring a therapy for motor recovery more similar to that effected by a human therapist;
- realizing a human-robot interaction as safe and natural as possible;
- to realize robotic hands with special dexterity features.

The work illustrated in this thesis describe different projects made to design a device and a hand for rehabilitation. In order to apply the studies that will be described in the following chapter to the rehabilitatio field, a few hypotheses widely confirmed in the literature have been accepted. Rehabilitation robotic devices [3], [4], [5], [6] aim to help the patient regain learning on how to move exploiting the plasticity of the neuromuscular system, that is its ability to learn again the motor patterns thanks to the repeated execution of pre-established movements. Medical studies [7], [8] have evidenced that, as a consequence of a neuromuscular damage, the human motor system has to learn again the right spatio-temporal scheme of muscle activation. With respect to the human therapist, the robot for rehabilitation guarantees greater accuracy as regards action iteration and follow-up of the desired trajectories [9]. Upper limb robotic prostheses should be as much as possible similar to human limbs (as for size, weight and shape) and at the same time they should guarantee dexterity, stability, grasping adaptability and movement naturalness [10], [12], [13], [14].

Restoration of walking ability is an important goal of rehabilitation following neurological disorders such as stroke or spinal cord injury. Conventional gait training programs are often labor intensive and non-standardized. For example, in body weight support treadmill training, one or more therapists need to provide manual assistance to move the patient's leg and/or pelvis into a "kinematically correct" trajectory. The manual assistance needs to be provided repetitively and intensively throughout a training session, which creates enormous physical burden to the therapists. In order to relieve the strenuous effort of the therapists, there is a growing interest of using robots to automate gait training. There are currently a multitude of robotic devices for gait rehabilitation applications. These robots are able to alter gait by pushing and/or pulling on the human body, and have met with varying degrees of success in terms of rehabilitation outcomes [50–55]. These gait rehabilitation robots share three characteristics:

- they are highly-specialized devices that only perform gait rehabilitation, 2) they may seem intimating or alien to patients (e.g. a huge exoskeleton or a tangle or wires and push-rods;
- they may seem intimating or alien to patients (e.g. a huge exoskeleton or a tangle or wires and push-rods);
- it takes a long time to set up robots on patients

One approach to address these limitations is to use commercially available robotic arms as the training platform. Such robots are typically multipurpose, and an "arm" configuration resembles human morphology. The advantage of using a multipurpose robotic arm for gait rehabilitation is it can perform many functions, e.g. it could be attached to any point on the body for lower- and/or upper-body rehabilitation relatively fast and easy and apply forces in any direction. For universities and clinics this may be more cost-effective than purchasing several specialized robots. The advantage of using a human-like morphology is that it may make the robot less intimating or alien to patients. Although using a multipurpose robotic arm for gait rehabilitation holds promise, a number of challenges must be overcome for this to be feasible.

1.1 Grasping in industrial robots

The analysis of the problems encountered in the realization of robotic devices able to faithfully reproduce the operations performed by humans, such as just grasping the leg, has led us to thoroughly study the grasping problem for industrial hand. In fact, from the study of robotic hand anatomy and from the analysis of robotic hand behavior

during grasping, it is possible to obtain useful information for developing grasping device and for improving knowledge about hand kinematics in order to design innovative and human-like rehabilitation devices.

In particular, an analitic study of the Robotiq 3-Finger Adaptive Gripper, a commercially available underactuated gripper with 4 actuators and 10 degrees of freedom, is going to be presented in Chapter 3. For a complete analysis of the gripper behaviour, the motion sensing system Kinect has been used. In this way, it has been possible to perform the triangulation of the visual features necessary for estimating the gripper posture. A set of markers has been placed on the gripper to detect and track the gripper motion by using a blob detection algorithm.

The gripper has a high grip strength, a resilient rigid linkage transmission system, and simple control protocol. It is currently being used in several research laboratories and by several teams in the DARPA Robotics Challenge. Although several similar research prototypes have been studied for over a decade [15] we are not aware of existing mathematical models of the robot's behavior suitable for simulation and optimization purposes. As a result of this study a quasistatic hybrid systems model of the robot is presented. Extensive experiments were performed in order to compare the proposed model to the measured behavior of the physical gripper, showing accuracy within the range of measurement error. The model is also implemented in an open-source robot simulator and made available for public use.

1.2 Grasping: a new method

From the idea of using grasping for rehabilitation, and while studying the industrial robot, a novel method for doing grasping come in our mind and we developed a new technique for building low-cost, kinesthetic robot control devices that duplicate the geometry and kinematics of a robot, but at smaller scale suitable for desktop use: the ROBOPuppet.

The puppet is a 3Dprinted miniature of the target robot with encoders embedded in the joints that translates the user's physical actions with the model directly to the robot's actions. The kinesthetic mode of operation is familiar to those who have played with action figures as a child, and we hypothesize that it lets users control complex motions in a more intuitive way than using joysticks and joint-level control. A key characteristic of the approach is its generality; it can be applied to a variety of different robots via the use of standardized hardware modules and geometry processing steps. The method is also highly accessible: we make use of inexpensive, off-the-shelf electronics and new

rapid prototyping technologies, specifically the wide availability of 3D printers to non-specialized users [35], [47], to create models that accurately mimic the proportions, look, and feel of the target robot. Because the puppet is lower-fidelity, imprecise replica, we apply a sanitizing procedure to ensure the robot's safety. As a preliminary demonstration of the ROBOPuppet method, we built a prototype puppet of the Stäubli TX90L 6DOF robot arm. A 30% scale device was created and used to control the arm both in real-time simulation as well as on the physical robot.

1.3 Robotic rehabilitation mechanism

Because one of the aim of the present work has been the design of a rehabilitation device able to grasp the human leg, all the previous projects have been done for studying the grasping problem. But a parallel project to solve the problem on how to attach the robotic hand to a robotic arm developed a patent for a rehabilitation device. Starting with the use of a multipurpose robotic arm to be attach to the human leg to automate gait training and relieve the physical burdens placed upon therapists, we design a mechanism able to attach the robot arm to a human limb, in a fast ans safety way.

In fact using a robotic arm for gait rehabilitation holds promise, but several challenges must be overcome. First, there must be a way to attach the robot arm to a human limb, preferably one that takes a minimum amount of time. Second, there should also be a mechanism to automatically and instantaneously detach the robotic arm if a patient stumbles. Finally, the robotic arm should be transparent, i.e. it should be able to follow a moving subject and not interfere with their nominal gait. In Chapter 4 we presents a mechanical interface that allows for a safe connection between a robotic arm and a human limb.

1.4 Robotic rehabilitation hand

Since the human hand is an example of high dexterity system above all others, after studying the grasping problem with robots and behind the construction of the robotic rehabilitation device, it is clear that we need a new hand able to replicate the human hand ability in the rehabilitation process. Because underactuated robot hands produced using 3D printing have recently become a viable alternative to conventional commercially available robot hands, we decide to use the 3d printing process to make a new robotic hand. One of the key motivations for using a 3D printed hand rather than a commercial alternative is cost: 3D printed hands can typically be manufactured in a

research lab for less than $1000 US dollars. In particular, the Yale OpenHand project makes available drawings, parts lists, and assembly instructions for a series of 3D printed hands [71, 77]. This is important because it gives many more researchers access to relatively sophisticated robot hands. The alternative is to spend tens of thousands of dollars to purchase a robot hand from a vendor such as Schunk, Barrett Technology, Robotiq, *etc.*.

Moreover, the compliant and underactuated nature of compliant 3D printed hands makes them suitable for a variety of robust grasping scenarios. In fact underactuated hands are capable of grasping a wide variety of object sizes and shapes by passive adaptation of the finger bending to the object geometry. An underactuated power grasp consists of two steps: in the first step,the **sweeping phase**, the proximal links of the fingers are in contact with the object, and in the second step, the **caging phase**, the distal links of the fingers close around the object, achieving a power grasp. Since grasp contact is maximized, this enables underactuated hands to perform form closure grasps. Both the existence of many commercially available underactuated hands [61–66] and a solid theoretical analysis of underactuated robotic hands [57, 73] emhasize the advantages of underactuation in robotic hands.

In Chapter 5 we presents the Baxter Easyhand a 3D-printed, underactuated, open-source, economic, two-finger robotic hand that overcomes the limitations of the Baxter Electric Parallel Gripper. The design and build process of our robot hand is based on the iRobot-Harvard Hand [74] and the Yale Open Hand [70], which both originated from the SDM hand [72]. While all of these hands come with a different amount of fingers and actuators, they are all underactuated. The Easyhand is most similar to the SDM and the Open Hand in that we also only use a single actuator to control the motion of the fingers.

The key feature of the Baxter Easyhand is that it the fingers are actuated using the Baxter gripper actuator rather than a Dynamixel servo. Essentially, the tendons of the Easyhand are routed so that they can be attached directly to the actuator of the native Baxter gripper. This has several advantages. First, it reduces the weight of the hand since it is no longer necessary to lift the additional servos. Second, it enlarges the resulting workspace of the combined arm/hand system because it enables us to move the hand closer to the wrist actuator and thereby reach a larger space of hand positions/orientations. Third, it makes it easier to program the hand because the Easyhand uses the native Baxter gripper ROS driver. This driver already incorporates all the relevant hand speed/force settings. In addition, the Easyhand is smaller than other 3D printed hands that are part of the Yale OpenHand project. This is important because it makes the gripper sizing more comparable with the Baxter arm. With a maximum

payload of five pounds, the Baxter arm is essentially designed to lift small light objects. The Easyhand is more compatible with this payload. Compared to the native Baxter parallel jaw gripper, the Easyhand has one additional critical advantage: it can grasp any object that fits within the 9 cm aperture between the fingers. Each finger on the native Baxter parallel jaw gripper can only translate a maximum of 2 cm. As a result, it is only possible to grasp objects larger than 4 cm in diameter by manually removing the fingers and attaching them to the hand in a configuration where they are further apart. However, if one does this, then the fingers will no longer touch when they are closed fully.

In contrast, the Easyhand fingers will close from a 9 cm widest aperture all the way to a fully closed configuration. We have designed the tendon arrangement so that the maximum travel of the underlying natrive Baxter gripper corresponds roughly with the maximum travel of the Easyhand fingers. We use the same motor as the Baxter gripper and provide both open-source design instructions and the models required to create our hand. All the material can be found at

http://www.ccs.neu.edu/research/helpinghands/easyhand/easyhand_assy_instructions.html. This allows to easily reproduce our manipulator using affordable of-the-shelf materials and to mount it on the Baxter robot in a quick and easy way.

1.5 Conclusions

Summarizing, the purposes of this thesis are:

- to find, from the analysis of industrial robot grasping behavior, a general rule for performing a stable, human-like grasp with a robotic hand;
- to provide a different method of doing grasping with industrial robot, that help to find optimal grasps configuration with big robotic arms;
- to release a new mechanism to connect the human leg with a romotic arm for rehabilitation;
- to release a new 3d printed robotic hand easy and cheap to build.

The construction of the mechanism and the robotic hand have the goal to produce a new rehabilitation device for optimal grasping.

Chapter 2

Studying the Kinematic of an Industrial Robotic Hand

Despite decades of active research, reliable grasping in unstructured environments like the home remains a major challenge in robotics. Small amounts (millimeters) of uncertainty in object shape, relative position, and robot control errors can cause grasping failure or damage to the object. One approach is to develop active adaptive behaviors that incorporate visual and tactile feedback, which requires precise sensing. Moreover, designing appropriate feedback behaviors is often challenging due to the high dimensionality of hands, which can have dozens of degrees of freedom (DOFs). An alternate approach relies on clever hardware design of passive elements, like springs, that automatically conform a hand to the shape of objects without the need for sensor feedback [15]. This approach can also greatly reduce the number of actuator elements, simplifying the control process [18–20]. Such hands are referred to as *underactuated* because there are fewer actuators than the number of degrees of freedom; each actuator drives multiple degrees of freedom in a coupled manner by complex transmission systems (tendons, gear trains, soft elements, adaptive synergies [17]. Although underactuated hands can conform to a wide variety of objects, their behavior is harder to model using traditional techniques due to nonlinear and nonsmooth coupling between degrees of freedom. Existing tools for grasp analysis and optimization (e.g., GraspIt!) fail to account for such nonlinearities, which may lead to incorrect predictions. This Technical Report presents a model of the Robotiq 3-Finger Adaptive Gripper, a commercially available underactuated gripper with 4 actuators and 10 degrees of freedom. The gripper has a high grip strength, a resilient rigid linkage transmission system, and simple control protocol. It is currently being used in several research laboratories and by several teams in the DARPA Robotics Challenge. Although several similar research prototypes have been studied for

over a decade [15] we are not aware of existing mathematical models of the robot's behavior suitable for simulation and optimization purposes. This report describes a quasistatic hybrid systems model of the robot suitable for such purposes. Hybrid systems model the coupled dynamics of continuous and discrete state variables, and are appropriate for dynamic discontinuities, such as those involved in contact [21]. They have been used in dexterous manipulation, see, e.g., [22] , [23], [24], [25] to model changes in contact state from no robot-object contact, to stable contact, to sliding contact. In this paper we use hybrid systems not only model changes in contact state but also the mechanical coupling and breakaway mechanism of the Robotiq gripper. For any given motor control, the model outputs the configuration of the robot in quasistatic equilibrium with a fixed object. Extensive experiments were performed in order to compare the proposed model to the measured behavior of the physical gripper, showing accuracy within the range of measurement error. The model is also implemented in an open-source robot simulator and made available for public use.

2.1 The Adaptive Gripper

The 3 finger-Robotiq Adaptive Gripper is a robotic peripheral that is designed for industrial applications [26]. It is designed to adapt to parts of varying sizes and shapes. It has three articulated fingers, i.e. finger A in front of finger B and finger C, that each have three joints (three phalanges per finger), as shown in Figure 2.1 (Source: Robotiq, reprinted with permission). The palm integrates four servomotors and the control electronics. Three finger phalanges are driven by one of the servomotors while the bases of fingers A and B rotate around the palm in a spread movement which is generated by the fourth servomotor. When an inner phalange contacts an object with a certain force, a breakaway system based on a clutch decouples the outer phalange from the inner phalange so that it can perform an enveloping grasp of the object. The force value at which the breakaway system is activated is determined by torsional springs inside the phalange joints. The breakaway system is also engaged when an inner phalange reaches a joint limit (Figure 2.2). This configuration allows the underactuated fingers to automatically adapt to the shape of object they grip as the servomotor is driven to the closed position. When the outermost phalange makes contact or reaches its joint limit, the finger cannot be driven any further. The gripper controller detects this situation when the motor current rises above a certain user-defined threshold. At this point, the motor no longer driven forward and the finger remains fixed in place until a more open motor position is commanded.

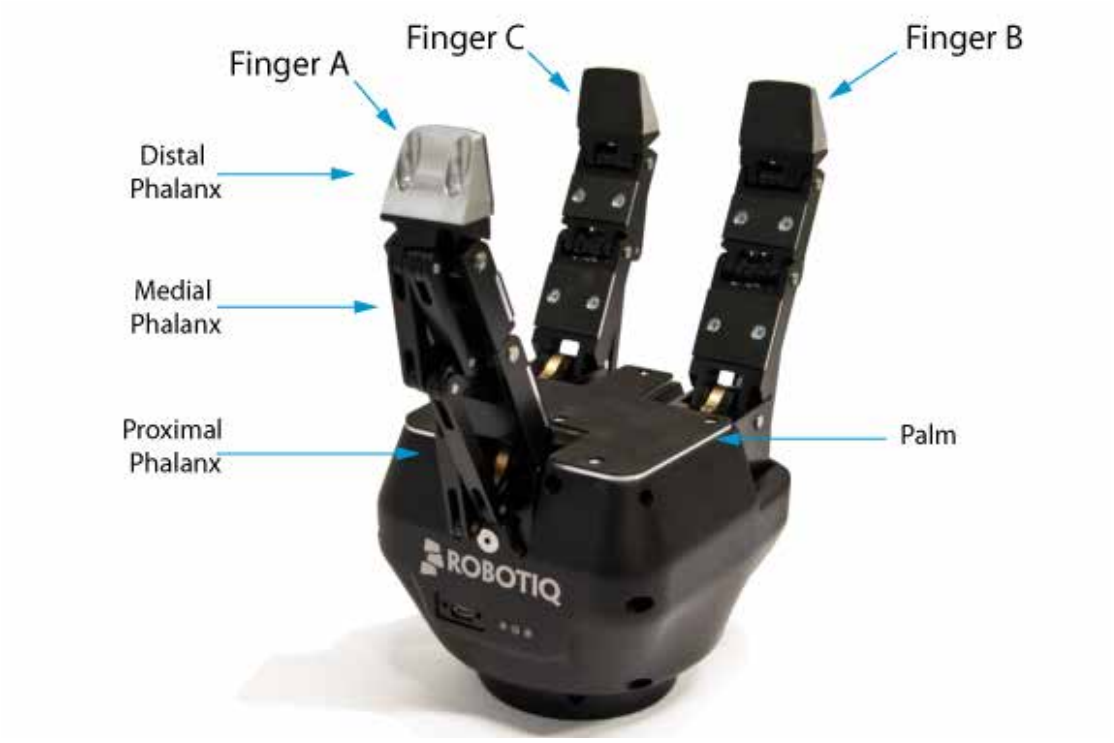


FIGURE 2.1: Robotiq Adaptive Gripper, S-Model.

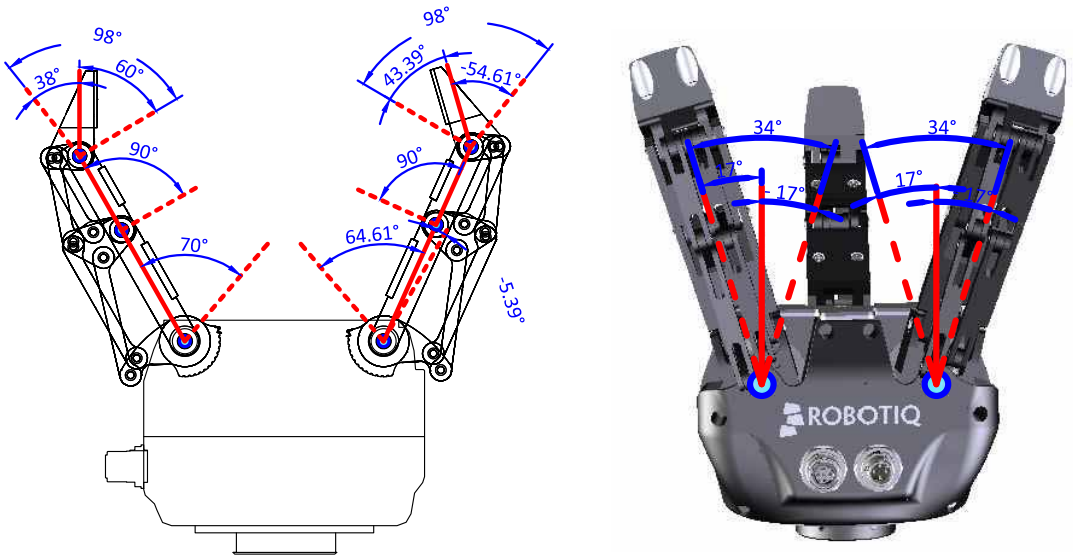


FIGURE 2.2: Joint limits of the gripper.

2.2 Single finger hybrid dynamics

Hybrid systems are general mathematical models that describe the behavior of continuous dynamics and discrete events under control inputs. We first present the dynamical model of a single finger with and without an obstacle; each finger is identical so their interactions need not be modeled unless two fingers touch. In such a case the opposing

finger acts as an obstacle to the first, and vice versa. The quasistatic movement of the finger through a closing motion is described as follows.

- A gear attached to the servomotor moves a linkage in the back of the finger which presses the first phalange toward the palm through a torsional spring mechanism.
- When the movement of this phalange is blocked (either by an object or when hitting a joint stop) then a second transmission linkage presses the second phalange forward.
- When the second phalange is blocked, the third phalange moves from a similar transmission linkage.
- The third phalange eventually is blocked and closing halts.

A key assumption of our approach is that the obstacle remains in fixed position once its motion blocks any phalange (i.e., does not roll or slip). This assumption is typically justified because a gripped object only applies sufficient force to block a phalange when it is massive or it is pinned between two opposing faces of the gripper. Moreover, each finger is covered by a high friction rubber that rarely exhibits slip. We note that our model avoids an explicit representation of the transmission linkage components in the rear of each finger. Although it may be possible to model it, the transmission is irrelevant to grasp analysis and resolving the several closed loop constraints would add significantly to the complexity of the model. Hence, we focus on the behavior of the reduced coordinates of the phalanges only. The finger model describes a continuous

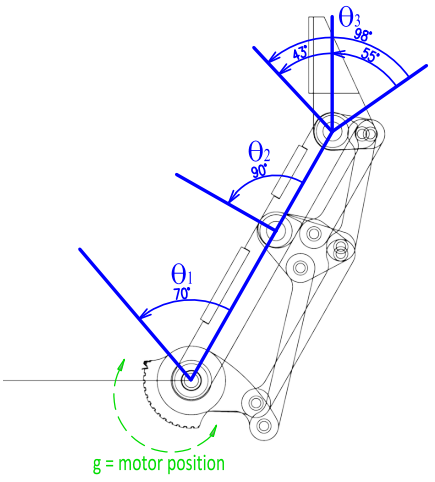


FIGURE 2.3: Finger's continuous state parameters and joint limits

state by the tuple:

$$x = (\Theta_1, \Theta_2, \Theta_3, g) \in [\Theta_{1,min}, \Theta_{1,max}] \times [\Theta_{2,min}, \Theta_{2,max}] \times [\Theta_{3,min}, \Theta_{3,max}] \times [0, 255] \tag{2.1}$$

where Θ_1, Θ_2, and Θ_3 are the relative joint angles of each phalanges with respect to the proximal parent, and g is the reference position of the servomotor. The control input

$$u \in [-1, 1] \tag{2.2}$$

describes the change in g from one time step to another during normal operation. In the controller's fixed-precision implementation g and u are integer values, but we model them as continuous values for simplicity. We also define four discrete *movement phases* of the finger:

1. The first phalange moves freely, the second stays extended, and the third moves opposite the first to maintain a vertical orientation.
2. The first phalange is blocked, the second phalange moves, and the third moves opposite the second to maintain a vertical orientation.
3. The second phalange is blocked, and the third link moves.
4. The third phalange is blocked, and the finger freezes.

We denote a similar phases $1'$ and $2'$, wherein the third phalange hits its lower joint limit of $-55°$ and cannot maintain a vertical orientation. Considering that a blockage can be caused by both an object (which occurs at variable position) as well as a joint limit, we consider these as different states in the hybrid system. A labeling for all possible discrete states of one finger is shown in Figure 2.4.
A discrete state is a tuple $(c_1, c_2, c_3, l_1, l_2, l_3)$ of binary variables, where

$c_i = 1$ indicates movement of link i is blocked by contact with an object.

$l_i = 1$ indicates movement of link i is blocked by its upper joint limit, i.e., $\Theta_i = \Theta_{i,max}$.

$l_3 = -1$ indicates reverse movement of link 3 is blocked by its lower joint limit, i.e., $\Theta_3 = \Theta_{3,min}$.

We remark that not all discrete states are reachable, for example, any state with $c_i = l_i = 1$ for any $i = 1, 2, 3$ are not possible. For each state $(c_1, c_2, c_3, l_1, l_2, l_3)$ we can describe $\Delta\Theta$ and Δg as a function of x and u.
where:

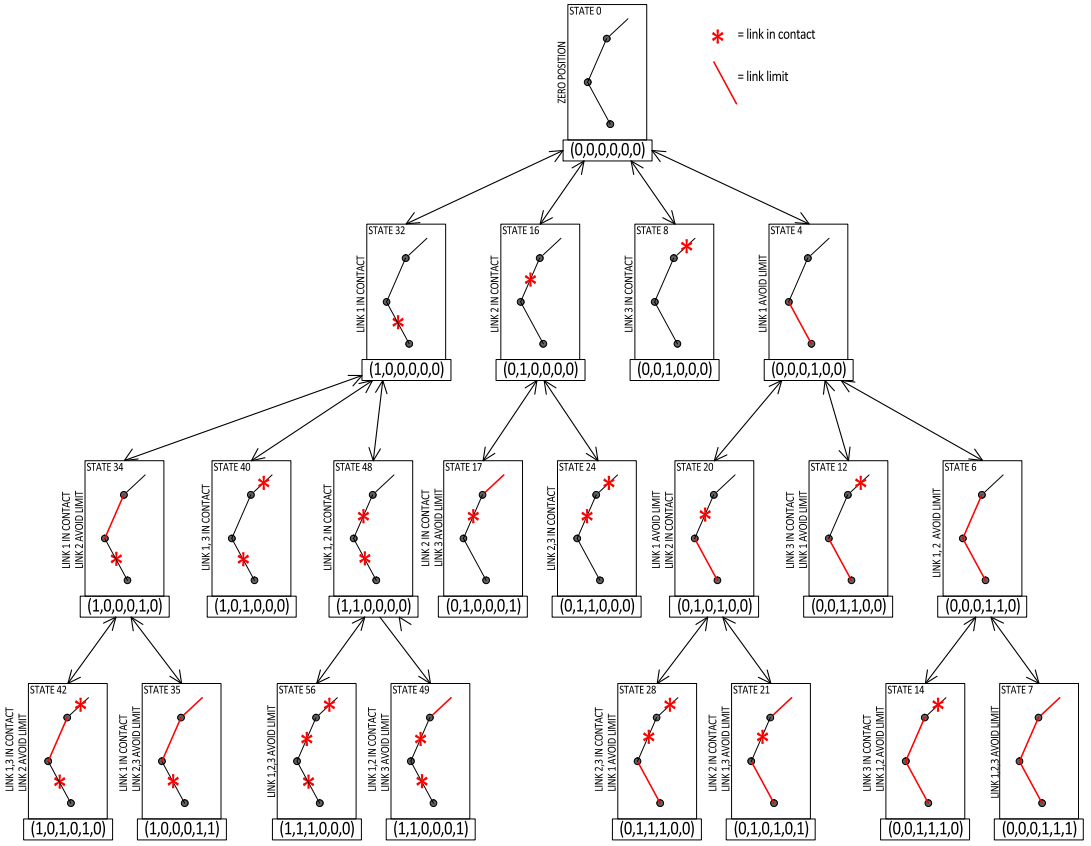


FIGURE 2.4: Finger's possible discrete states

Phase	State tuples	$\Delta\Theta_1$	$\Delta\Theta_2$	$\Delta\Theta_3$	Δg
1	$(0,0,0,0,0,0)$	$f_1(x,u)$	0	$-f_1(x,u)$	u
$1'$	$(0,0,0,0,0,-1)$	$f_1(x,u)$	0	0	u
2	$(1,0,0,0,0,0)$, $(0,0,0,1,0,0)$	0	$f_2(x,u)$	$-f_2(x,u)$	u
$2'$	$(1,0,0,0,0,-1)$, $(0,0,0,1,0,-1)$	0	$f_2(x,u)$	0	u
3	$(\cdot,1,0,\cdot,0,0)$, $(\cdot,0,0,\cdot,1,0)$	0	0	$f_3(x,u)$	u
4	$(\cdot,\cdot,1,\cdot,\cdot,0)$, $(\cdot,\cdot,0,\cdot,\cdot,1)$	0	0	0	0

TABLE 2.1: Possible discrete states

$f_1(x,u) = m_1 u$, with $m_1 = \Theta_{1,max}/140$

$f_2(x,u) = m_2 u$, with $m_2 = \Theta_{2,max}/100$

$f_3(x,u) = m_3(g)u$, with $m_3(g) = \Theta_{3,min} + (\Theta_{3,max} - \Theta_{3,min})/(255 - g)$

Transition conditions are given as follows.

1. $c_i = 0 \to c_i = 1$: an object is hit by link i.
2. $c_i = 1 \to c_i = 0$: $\Theta_{i+1} = \Theta_{i+1,min}$ and $u < 0$.
3. $l_i = 0 \to l_i = 1$: $\Theta_i + \Delta\Theta_i \geq \Theta_{i,max}$.

4. $l_i = 1 \rightarrow l_i = 0$: $\Theta_i + u < \Theta_{i,max}$.

5. $l_3 = 0 \rightarrow l_3 = -1$: $\Theta_3 + \Delta\Theta_3 \leq \Theta_{3,min}$.

6. $l_3 = -1 \rightarrow l_3 = 0$: $\Theta_3 + u > \Theta_{3,min}$.

A state machine depiction of the finger hybrid system is given in Figure 2.5. To obtain a general hybrid system for the hand with just put together the three state machine scheme for one finger.

2.3 Analytical model of one finger and one-link contact

The hybrid system model gives a dynamic equation that is useful for forward simulation but is not as useful for analysis, where one may want to quickly determine the finger configuration when the servomotor is driven to a particular value. Here we determined analytically the unique state corresponding to a particular value of g in the absence of obstacles. Moreover, if one knew the value of g where the obstacle would first be touched, a unique state for a given g can also be determined quickly. In future work, we are interested in using these equations for grasp analysis and optimization.

2.3.1 Analytical Equations without Object

Here we derive the joint angle equations in the absence of an object causing blockage. Joint limits are incorporated here. In this table, define the constants $m_1 = \Theta_{1,max}/140$ and $m_2 = \Theta_{2,max}/100$.

The stoppage at $g = 240$ is due to self-collision with the finger and the palm. Note

Phase	Motor range	Θ_1	Θ_2	Θ_3
1	$0 \leq g \leq 110$	$m_1 g$	0	$-m_1 g$
$1'$	$110 < g \leq 140$	$m_1 g$	0	$\Theta_{3,min}$
2	$140 < g \leq 240$	$\Theta_{1,max}$	$m_2(g - 140)$	$\Theta_{3,min}$
4	$240 < g \leq 255$	$\Theta_{1,max}$	$\Theta_{2,max}$	$\Theta_{3,min}$

TABLE 2.2: Joint angle equations

that we can also express the behavior of Θ_3 in Phases 1 and $1'$ in the form $\Theta_3 = \max(-m_1 g, \Theta_{3,min})$.

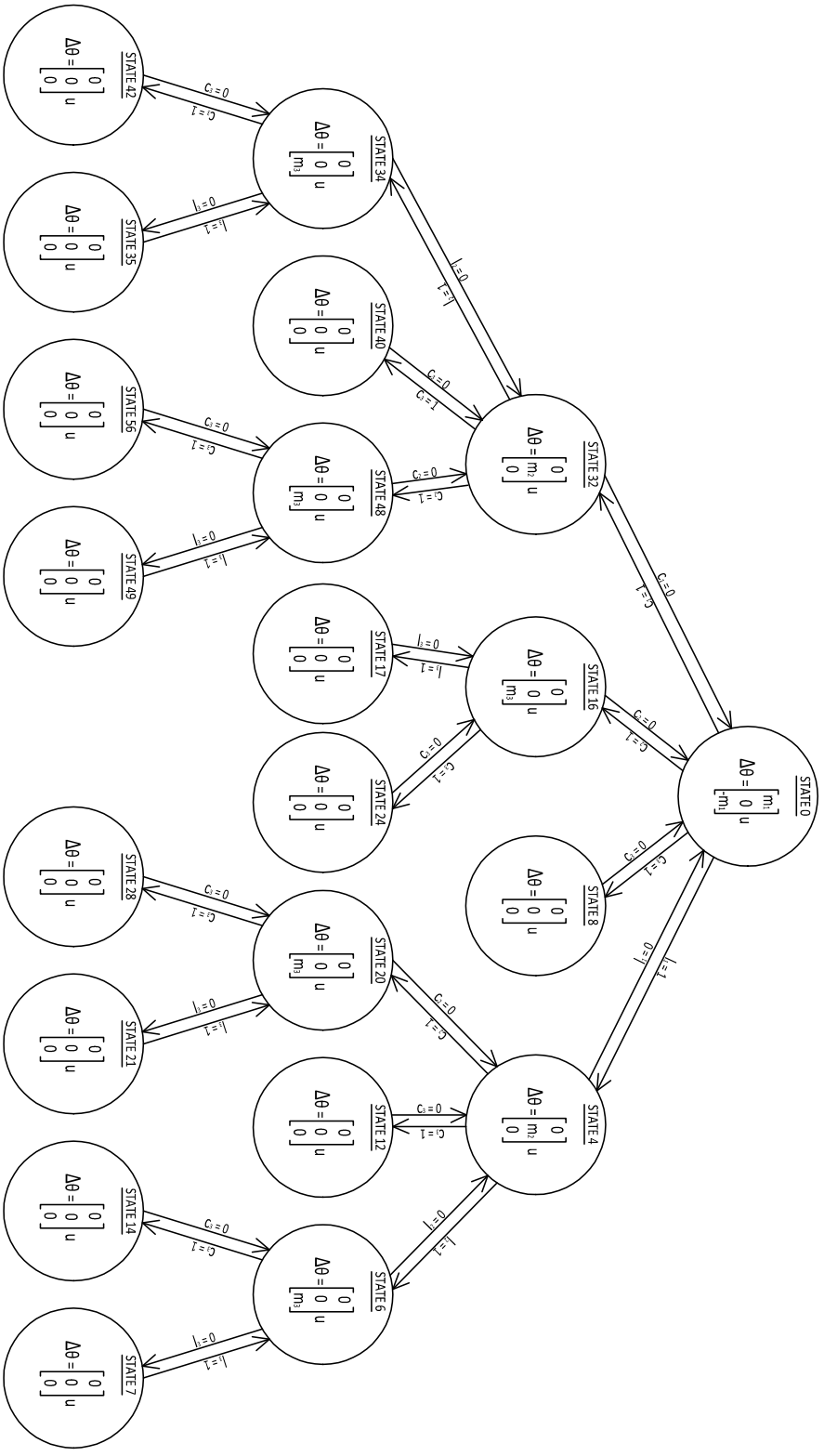


FIGURE 2.5: Finger's State Machine representation

2.3.2 Analytical Equations with Object

Suppose g_{ob} is the motor position in which any link of the finger is first blocked by the object. As the finger closes, we can integrate the equations of motion to determine the link behavior for $g > g_{ob}$. In the below tables, let us define the constant $c = \Theta_{3,max} - \Theta_{3,min}$. The derivation of the following equations is straightforward except for the behavior of Θ_3, which is non-linear in Phase 3. From experimental data, we observed that Θ_3 reaches its limit when g is halfway between 255 and the value of g when Θ_2 stops moving. In Fig 2.6 the Θ_3 stopping condition is represented: experiments demonstrate that it hits its limit when g reaches halfway between the stopping point of Θ_2 and 255. This equation also fits the slight curve observed in the data. The table 2.3 governs behavior for contact made only on the **first link:**

Obs. range	Motor range	Θ_1	Θ_2	Θ_3
$g_{ob} \leq 140$	$0 < g - g_{ob} \leq 100$	$m_1 g_{ob}$	$m_2(g - g_{ob})$	$\max\ (-(\Theta_1 + \Theta_2), \Theta_{3,min})$
$g_{ob} \leq 140$	$g - g_{ob} > 100$	$m_1 g_{ob}$	$\Theta_{2,max}$	$\min\ (\Theta_{3,max}, \Theta_{3,min} + (\frac{c(g-g_{ob}-100)}{255-g}))$

TABLE 2.3: First link behavior

The table 5.5 governs behavior for contact made only on the **second link:**

Obs. range	Motor range	Θ_1	Θ_2	Θ_3
$g_{ob} \leq 140$	$g_{ob} < g$	$m_1 g_{ob}$	0	$\min\ (\Theta_{3,max}, \Theta_{3,min} + (\frac{c(g-g_{ob})}{255-g}))$
$140 < g_{ob}$	$g_{ob} < g$	$\Theta_{1,max}$	$m_2(g_{ob} - 140)$	$\min\ (\Theta_{3,max}, \Theta_{3,min} + (\frac{c(g-g_{ob})}{255-g}))$

TABLE 2.4: Second link behavior

When **both link 1 and link 2 make contact**, only the third link can move. Let g_{ob1} and g_{ob2} be the motor position in which link 1 and link 2 are blocked by the object. They must satisfy $g_{ob1} \leq 140$ and $0 < g_{ob2} - g_{ob1} \leq 100$. In this case, the third link's behavior is governed by the equation $\Theta_3 = \min\ (\Theta_{3,max}, \Theta_{3,min} + (\frac{c(g-g_{ob2}}{255-g}))$. Finally, when the **third link makes contact**, the gripper stops moving.

2.4 Experimental Validation

We verify the motion model of the gripper by moving the gripper through a sequence of configurations and comparing joint angle predicted by the model for each configuration with the joint angle measured empirically. This is difficult because the joints do not have angular encoders. Instead an external visual tracking apparatus was devised.

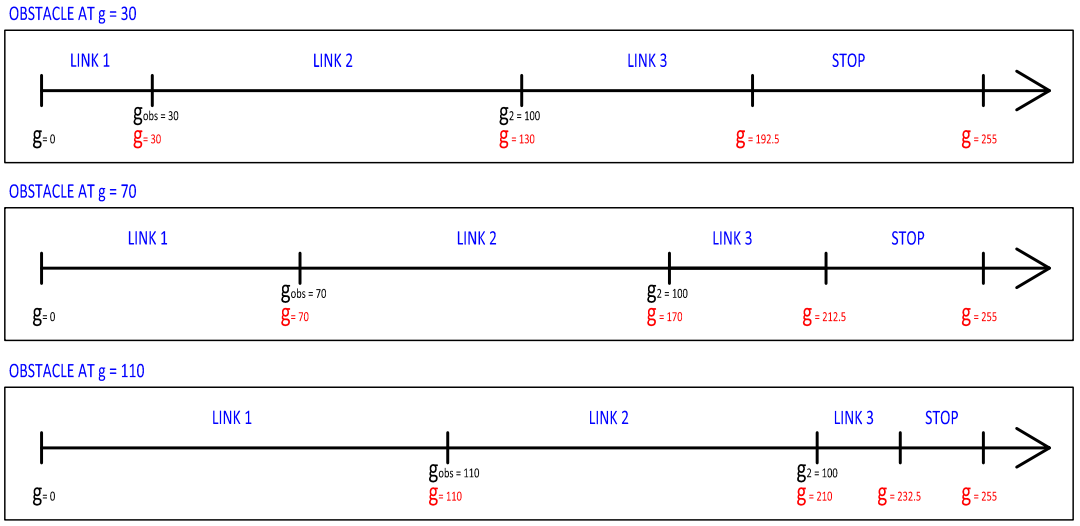


FIGURE 2.6: Θ_3 stopping condition.

2.4.1 Visual Tracking Measurement

We attach fiducial markers to each joint such that the markers are centered on the joint (Figure 2.7). We then recorded the 3D position of each marker in the workspace using data provided by an RGB+D camera. The markers are tracked using color blob

FIGURE 2.7: Visual Tracking with RGB+D

tracking on the 2D image provided by the camera. The blobs are tracked as axis-aligned bounding rectangles on the image, and the blob tracker provides both the center coordinates and bounds of each blob (in pixels), along with its detected color. The blob tracker is based on the CMVision library, which is described in [30] (see [31] too). One method to compute the 3D positions of each joint would be to use the center coordinates of each blob to index into the organized point cloud (see [28]) provided by the camera and retrieve the corresponding point. In practice, however, this often results in invalid points because the depth computation may have failed to result in a valid value for that particular pixel. Instead, the following procedure is used to compute 3D positions:

1. Color blobs corresponding to the fiducial markers are detected in the 2D image.
2. All valid points in the corresponding point cloud that fall within the bounding rectangle of the blob are retrieved.
3. Statistical outlier removal (see [29]) is performed to filter points that are unlikely belong to the marker.
4. The centroid of the remaining points is taken as the center of the fiducial marker.
5. We know from the construction of the gripper that the joints are all co-planar, so a plane is fit to the set of centroids representing joint positions in the workspace, and those centroids are projected onto that plane.
6. Finally, we know also that all joints have parallel axes of rotation, thus, given the absolute positions of the joints in space, we can construct vectors between them and compute the joint angles explicitly using the definition of the dot product.

It was also helpful to extend step 2 to accumulate data over a user-defined number of frames generated by the camera. This reduced the effect of camera noise and resulted in more stable readings; for these experiments, a window of 25 frames was used. We collected angles at each integer step of g ranging from 0 to 255 (full closed to fully open). Noise in the measurement process sometimes produced outliers, particularly for link 3, because the distance from one marker to the next is smaller than that of any other joints. So, we employed an outlier removal process. We considered an outlier to be a change in sensed angle that is more than 5 degrees from one value of g to the next (which certainly does not actually occur on the physical robot). We limited the measurement to be within ± 5 degrees of the previous and next inlier points.

2.4.2 Validation

First, the model was tested without an object in five experiments. Figure 2.8 shows a comparison between the model and thel measured behavior of Θ_1, Θ_2, and Θ_3 in 5 runs without obstacles.
We tested the model accuracy analizing the Mean Square Error (MSE) of the model, the MSE of the data, the Mean Standard Deviation (MSD) of the model and the MSD of the data. Table 2.5 demonstrates that the model has comparable accuracy to the measurement accuracy on link 2 and is well within measurement accuracy for links 1 and 3. Next, we tested the model with objects of varying position. We calculate g_{ob} as the motor position in which the finger first contacts the object. The object is a fixed rigid bar with adjustable position in a box rigidly attached to the robot's base

Data Analysis				
	MSE		MSD	
Angle	Model	Data	Model	Data
Θ1	0.74	1.41	0.50	0.68
Θ2	2.00	1.70	0.70	0.72
Θ3	1.93	6.25	0.75	1.24

TABLE 2.5: MSE and MSD analysis.

(Figure 2.9). Figure 2.10 shows an experiment with an object in positions $g_{ob} = 30$, $g_{ob} = 70$, $g_{ob} = 110$, and $g_{ob} = 200$.

2.5 Simulation Model Implementation

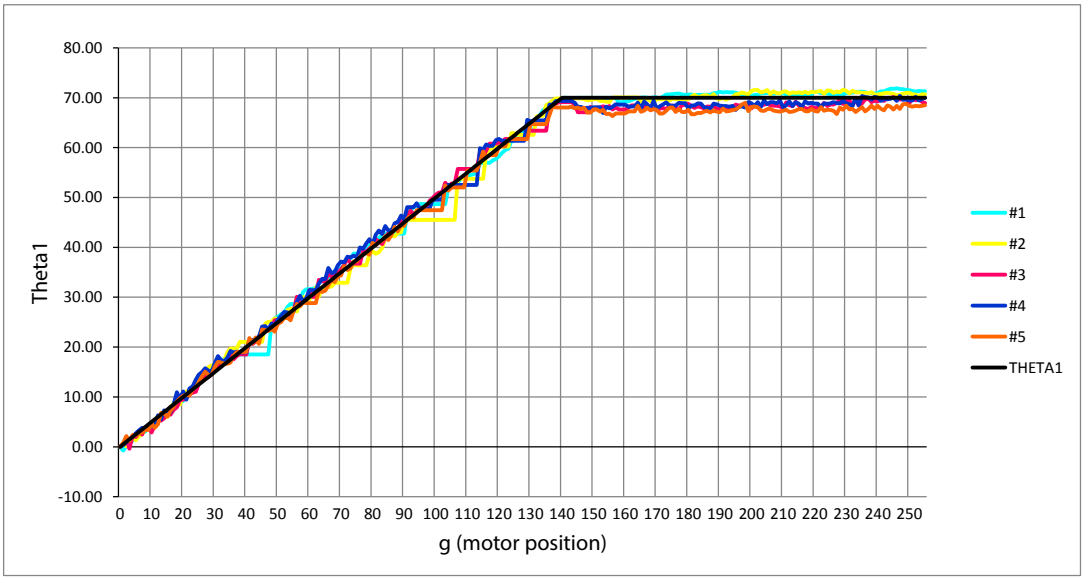
After testing the mathematical model through the observation of the real behavior of the gripper, we implemented it in the Klamp't robot simulator [42] (Figure 2.11). This is a physics simulation that simulates contact forces and joint torques. The joint angles outputted by our model are inputted into the system as reference values for PID controllers in the phalange joints. Fig. 2.12 shows different picture with progressive value of g during a simulation without object.

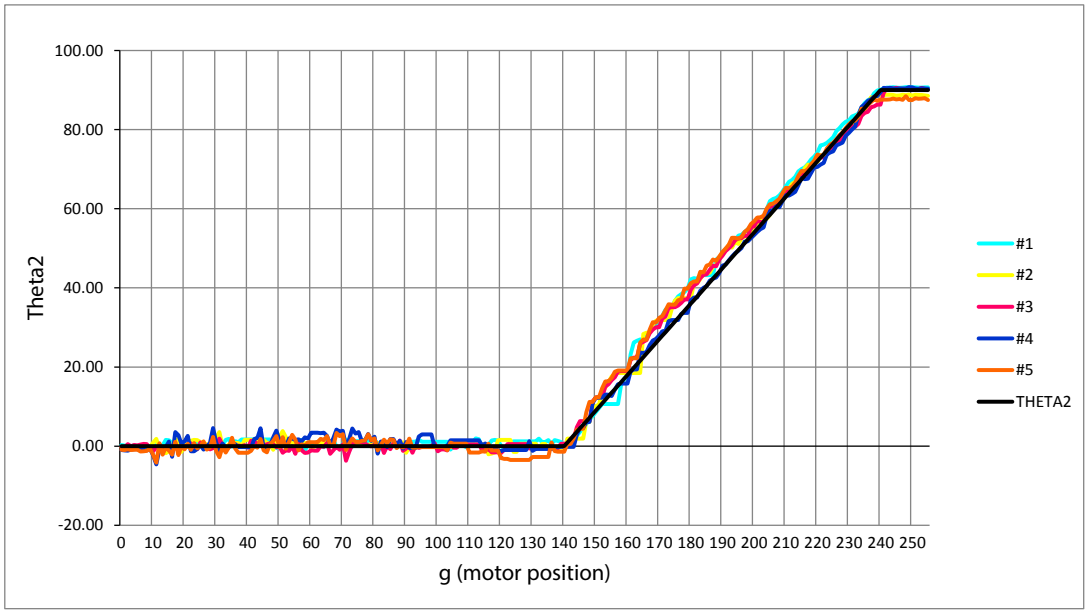
2.5.1 KLAMPT Model Test with object

Hereinafter some significant simulations to test the Klampt model with object are presented. Observe that in the simulation with gob = 160 a picture of the real experiment is shown. The object set in the simulation environment is a long and thin cylinder and it's considered a fix object. This scenario was prepared to compare the simulation with the real experiment we made on the real gripper using the metallic box.

2.6 Conclusions

A hybrid system model of an underactuated robotic hand is presented, both in the form of dynamic equations and their analytical solutions. The model is verified with a vision-based data collection system to match the experimental data well within measurement error. The model is implemented in a demo program in Klamp't, which is freely available from `http://klampt.org`. In future work we wish to perform a more precise empirical validation and to implement the model for grasp optimization and object recognition without additional sensors. We also intend to build a more complete model of the force characteristics of the gripper in contact with an object.





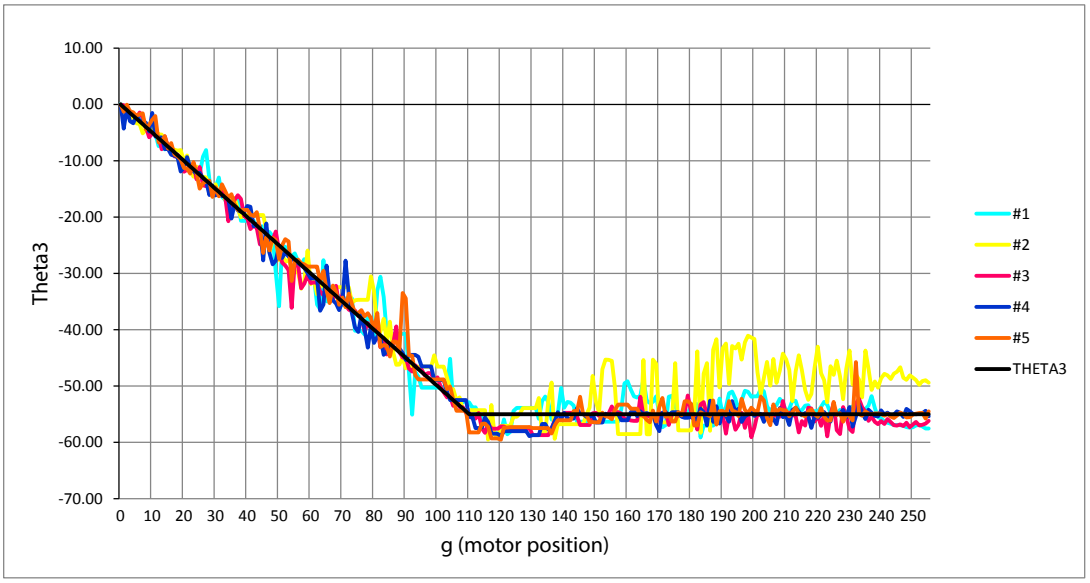


FIGURE 2.8: Model testing without object.

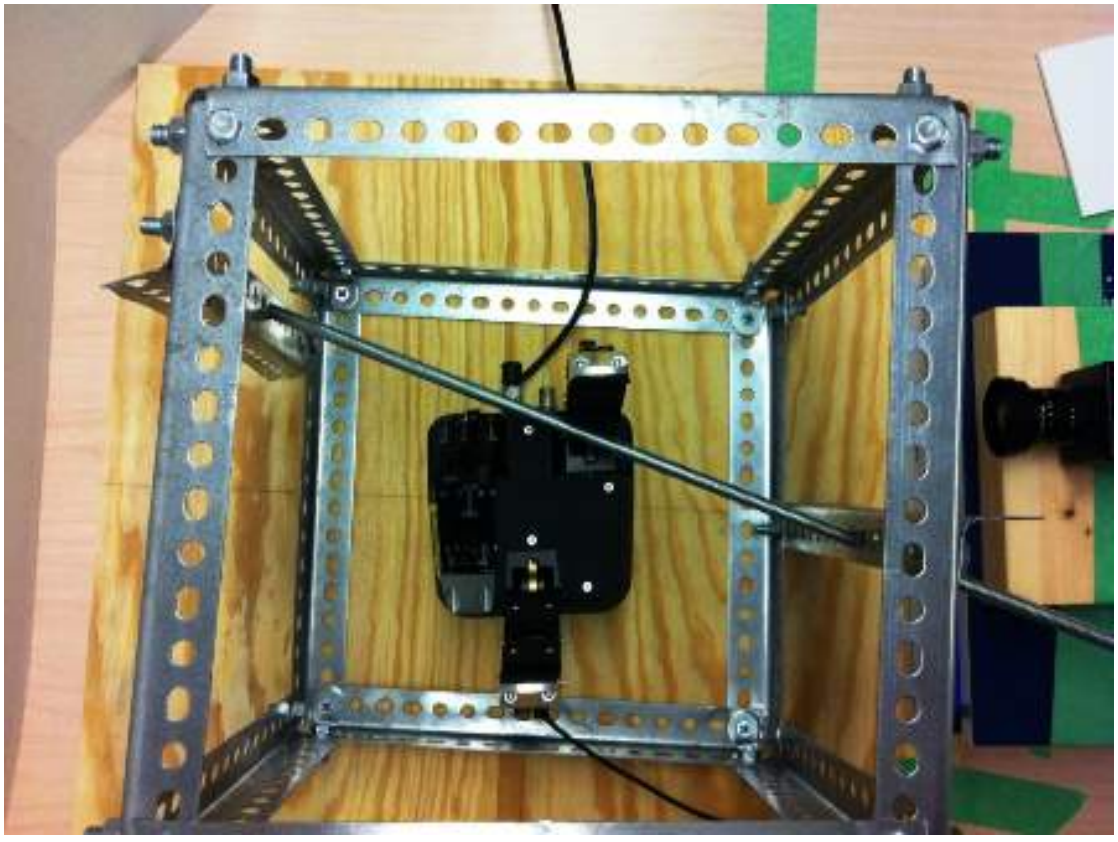


FIGURE 2.9: Experiment environment

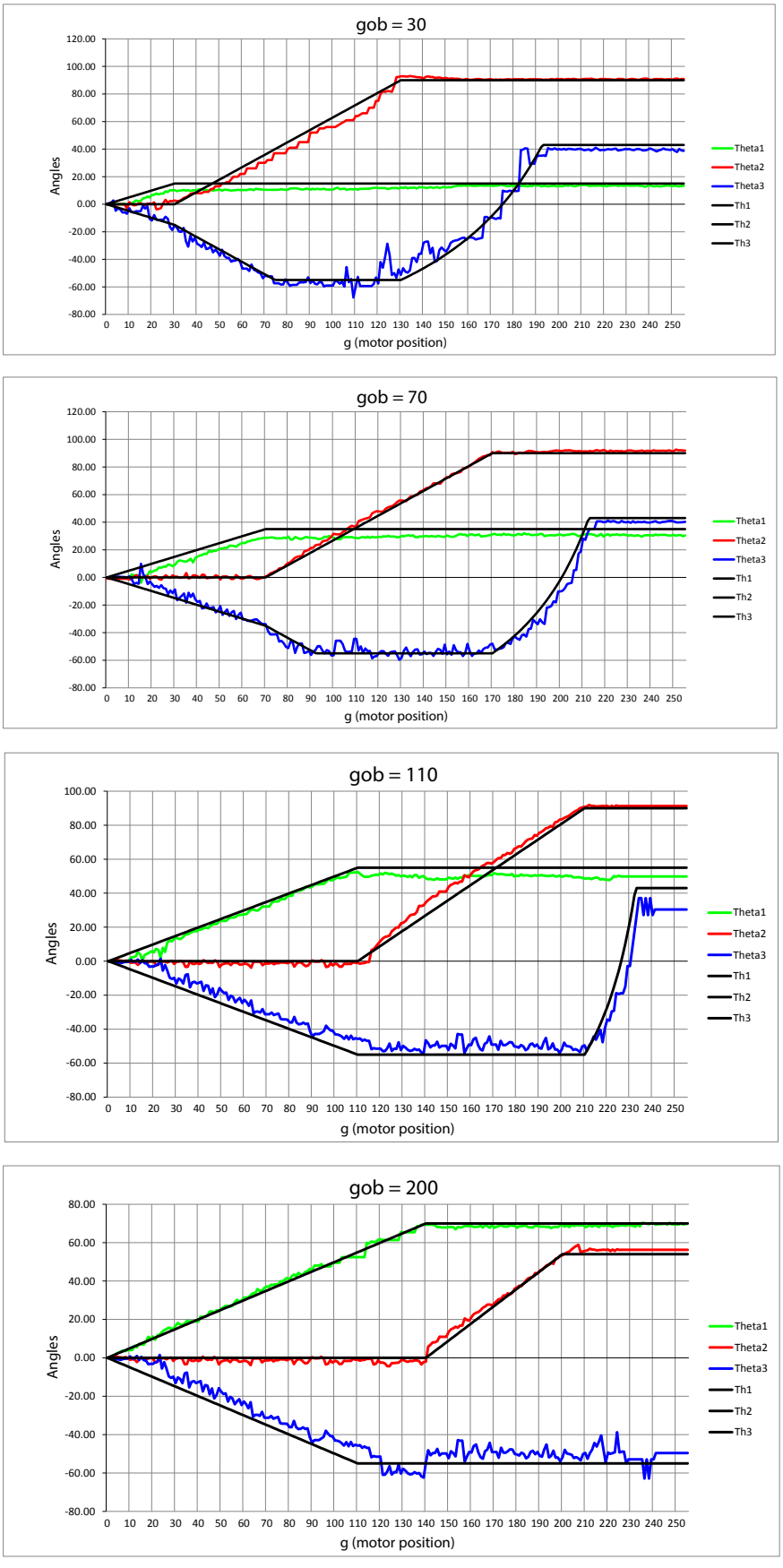


FIGURE 2.10: Experiments with different object positions

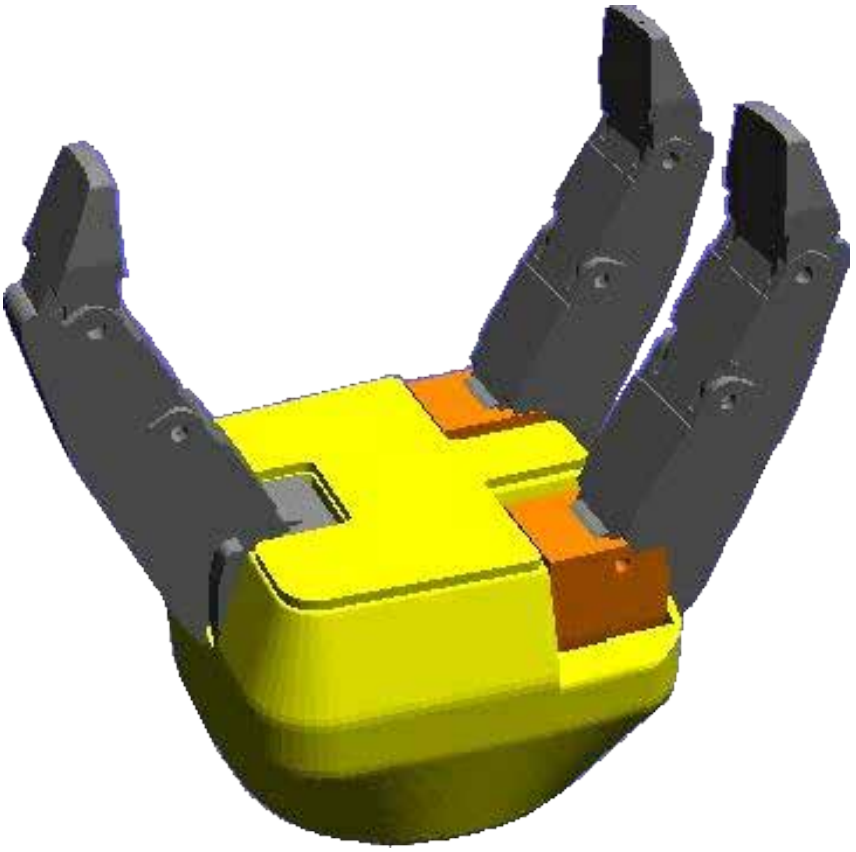


FIGURE 2.11: Geometrical Model in Klamp't environment

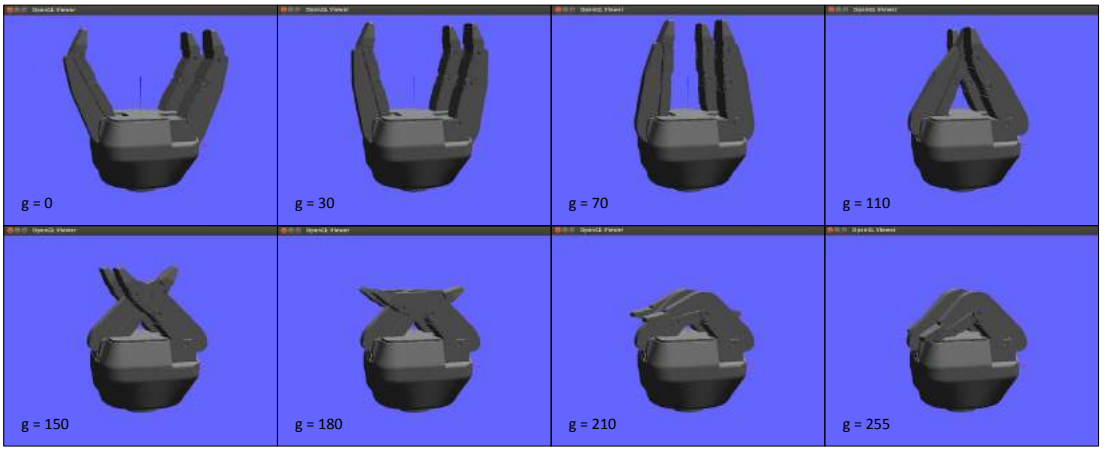


FIGURE 2.12: Simulation without object

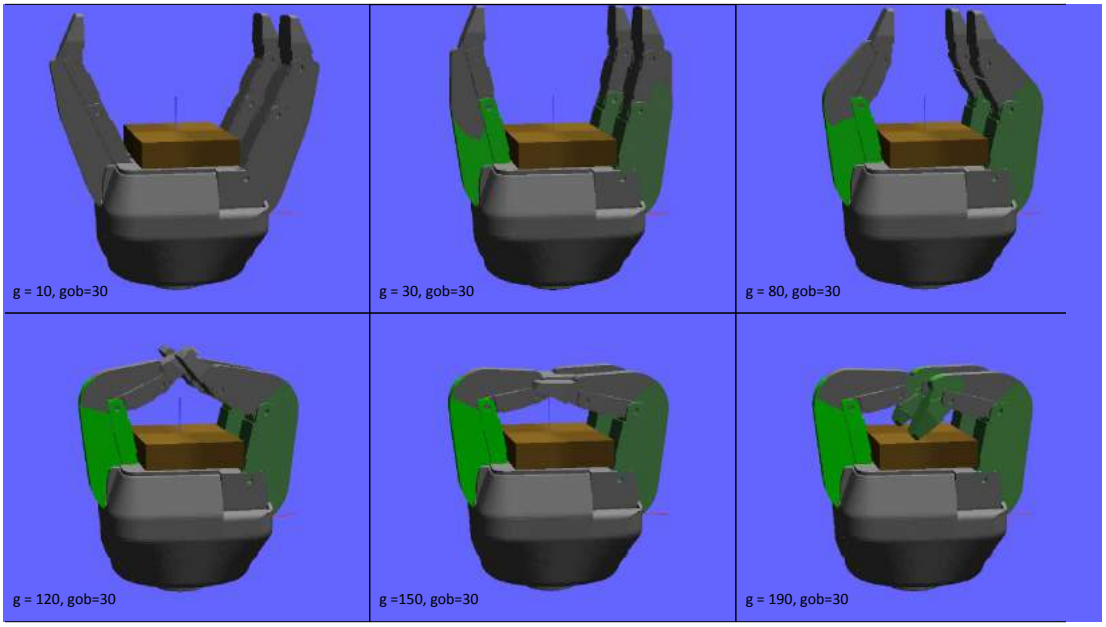


FIGURE 2.13: Simulation with object: gob = 30

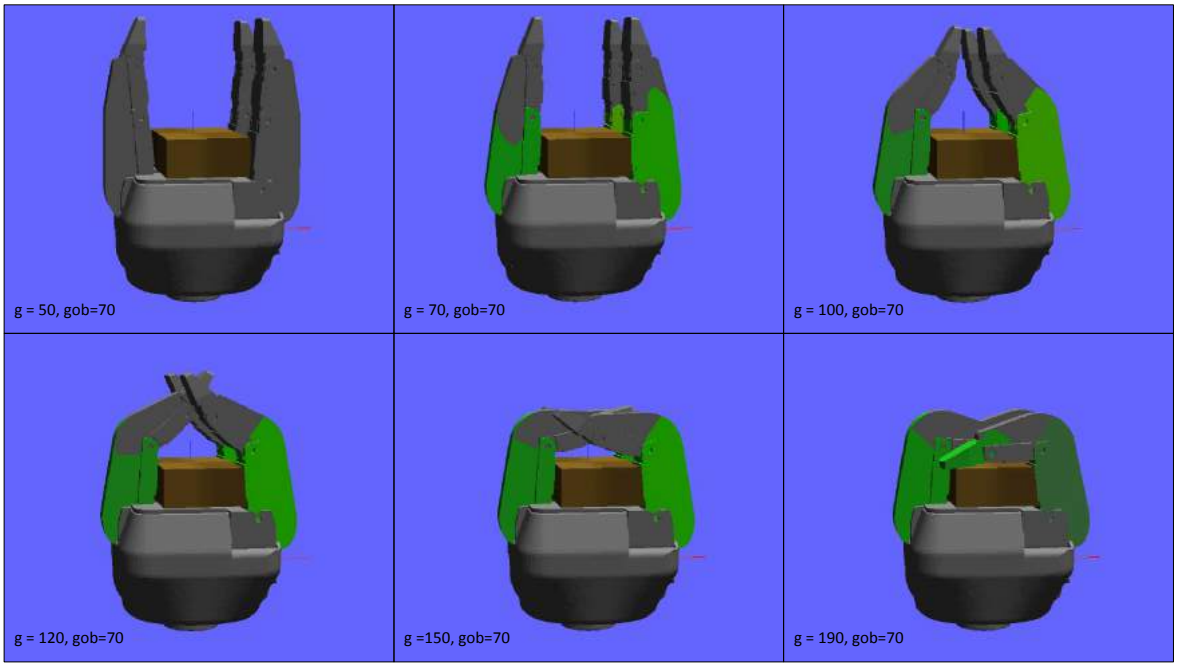


FIGURE 2.14: Simulation with object: gob = 70

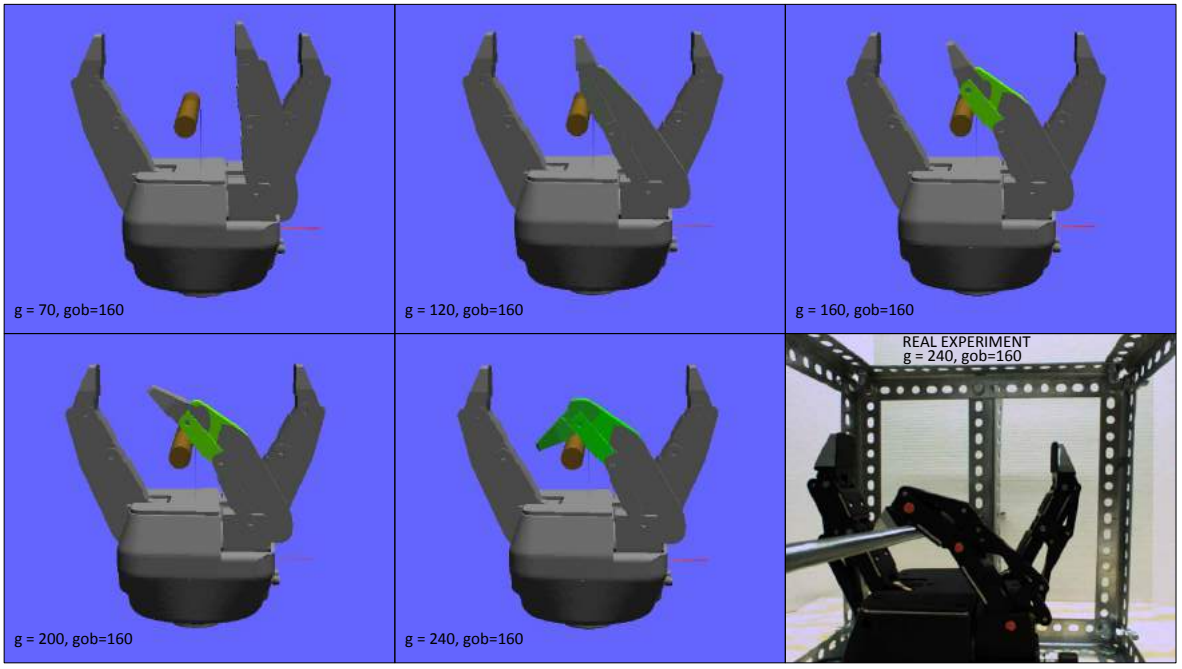


FIGURE 2.15: Simulation with object: gob = 160

Chapter 3

A Novel Method for doing Grasping with an Industrial Robot

ROBOPuppet is a method to create inexpensive, tabletop-sized robot models to provide teleoperation input to full-sized robots. It provides a direct physical correspondence from the device to the robot, which is appealing because users form an immediate "mental mapping" of the input-output behavior. We observe that untrained users can immediately exploit tactile and physical intuition when controlling the puppet to perform complex actions the target robot. The key contribution of this paper is a build procedure that embeds standardized encoder modules into scaled-down CAD models of the robot links, which are then 3D printed and assembled. This procedure is generalizable to variety of robots, and parts cost approximately seventeen dollars per link. We also present a simple software tool for fast calibration of the puppet-robot mapping, and a safety filtering procedure that sanitizes the noisy inputs so that the robot avoids collisions and satisfies dynamic constraints. A prototype ROBOPuppet is built for a 6DOF industrial manipulator and tested in simulation and on the physical robot.

3.1 Teleoperation in robotics

Robots are teleoperated by humans in many applications, including nuclear plants, robotic surgery, explosive ordnance disposal, and search and rescue. But a common challenge for users is to learn a "mental map" of the correspondence between the input device and the target robot. Joint angle control, via buttons or joysticks, requires learning the nonlinear map from joint angles to workspace motion. Cartesian end-effector

FIGURE 3.1: ROBOTPuppet and the Stäubli TX90L.

control is typically more intuitive, but the user must learn disparities between kinematic limits of the input device and those of the target robot. Moreover, end-effector motions may produce unexpected effects on other joints, which makes collision avoidance challenging. An alternate approach is often taken in programming from demonstration (PbD) [33] via *kinesthetic* teaching, in which the user directly pushes and pulls the robot. This is intuitive because the taught poses are in one-to-one correspondence with executed poses, and physical interaction immediately provides a sense of shape, dimension, and weight of the target robot. However, these systems require force sensing hardware and also cannot be used for remote teleoperation because they require physical co-presence. This paper introduces ROBOPuppet, a new technique for building low-cost, kinesthetic robot control devices that duplicate the geometry and kinematics of a robot, but at smaller scale suitable for desktop use. The puppet is a 3D-printed miniature of the target robot with encoders embedded in the joints that translates the user's physical actions with the model directly to the robot's actions. The kinesthetic mode of operation is familiar to those who have played with action figures as a child, and we hypothesize that it lets users control complex motions in a more intuitive way than using joysticks

and joint-level control. A key characteristic of the approach is its generality; it can be applied to a variety of different robots via the use of standardized hardware modules and geometry processing steps. The method is also highly accessible: we make use of inexpensive, off-the-shelf electronics and new rapid prototyping technologies, specifically the wide availability of 3D printers to non-specialized users [35] [47], to create models that accurately mimic the proportions, look, and feel of the target robot. Because the puppet is lower-fidelity, imprecise replica, we apply a sanitizing procedure to ensure the robot's safety. In summary, the contributions of ROBOPuppet are:

- A low-cost joint encoder assembly that is embedded in the printed parts of the ROBOPuppet to provide encoder readings and adjustable friction for maintaining its configuration under gravity.
- A systematic, step-by-step process for building a custom ROBOPuppet for a new robot via geometry preprocessing, 3D printing, and assembly.
- A calibration tool for easily calibrating a mapping from encoder values from the puppet to desired joint angles of the robot.
- A real-time planning method for translating puppet movements into robot movements that respect the robot's dynamic limits and avoid collisions with known obstacles.

As a preliminary demonstration of the ROBOPuppet method, we built a prototype puppet of the Staübli TX90L 6DOF robot arm. A 30% scale device was created and used to control the arm both in real-time simulation as well as on the physical robot. The prototype puppet costs a total of $85. Instructions, requisite 3D models, and software for ROBOPuppet are available at http://robopuppet.org

3.1.1 Related Work

The Staübli TX90L robot is controlled manually using the SP1 control pendant (Fig. **??**), which allows a user to control joint angles or Cartesian motion via twelve buttons (increase and decrease value on each axis) along with two buttons to increase/decrease speed. This is a typical controller for an industrial robot. Other robot control devices range from computer GUIs[36], haptic controllers [38], joysticks, 3D sensors, and game control pads [44, 49]. Input devices can be categorized between joint level vs cartesian control and kinesthetic vs non-kinesthetic approaches. [40] Cartesian control is often more intuitive than joint-level control, but requires careful design of the mapping to joint space to avoid singularities and kinematic limits of the robot and limits of the

FIGURE 3.2: SP1 Control pendant for the Stäubli TX90L.

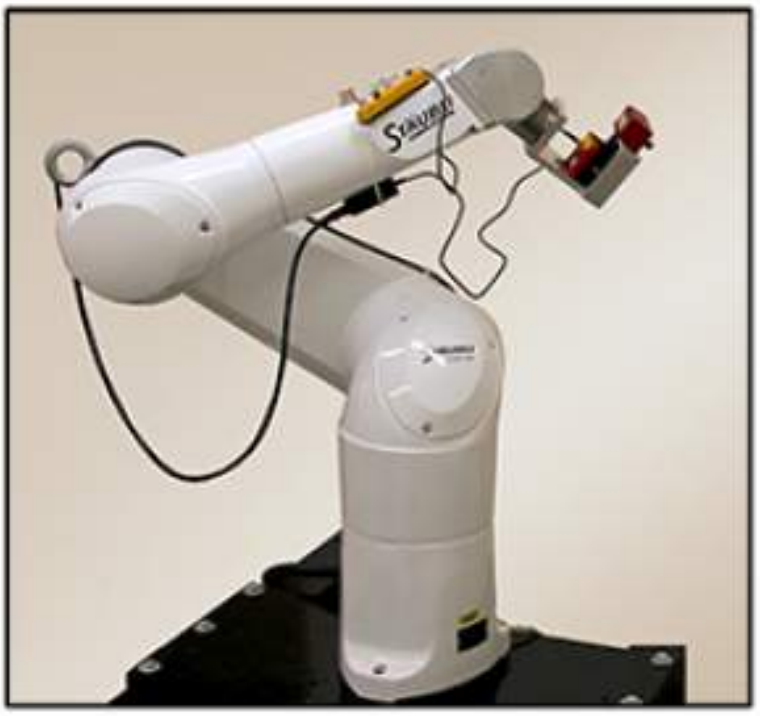


FIGURE 3.3: Stäubli TX90L robot.

input device. Furthermore, it is challenging in Cartesian mode to control non-end effector points on the robot (for example, an elbow). Non-kinesthetic devices like control pendants, GUIs, and joysticks are often more ergonomically suited for long-term operation, e.g. holding a pose for a long time. Kinesthetic control devices like haptic devices translate bodily movements in workspace to robot movements in workspace, and are

FIGURE 3.4: ROBOPuppet controller for the Stäubli TX90L.

often more appealing for novice users to operate due to the ability to feel forces felt by the robot [34, 45]. In contrast, ROBOPuppet proposes a control device in one-to-one correspondence with the target robot, combining the benefits of direct joint control and kinesthetic feedback. Direct kinesthetic control of joints is not a new idea. The most widely used technique is Programming by Demonstration (PbD) for teaching configurations to robots via direct physical manipulation [33]. This has been applied to several commercial industrial robots. However, physical manipulation of the robot is not possible in remote environments. The most closely related work to ours is the development of a 5-axis robotic motion controller to teleoperate an industrial robotic arm [46]. The application of this controller is similar to one considered here, but is more expensive, requires significant design and mechanical engineering expertise, and only matches the link lengths and joint angles of the target robot. By contrast, ROBOPuppet is a generally applicable process for building low-cost control devices, it requires very little experience to build and assemble, and the device matches the geometry of the robot.

3.2 Method

The method consists of five major elements.

1. Standardized, inexpensive *joint encoder assemblies* that contain encoders, can be adjusted according to the joint orientation of the links of the robot, and enable the puppet to support its own weight against gravity using adjustable friction controls.

2. *Pocket geometries*, which are 3D models which are subtracted from the printed parts to provide structures for holding the base of the encoders in the joint surface and the encoder shafts in the opposing joint surface. They also provide holes for bolts used to control the friction between joint surfaces, and access ports for wiring and other innards of the puppet that are necessary for assembly.

3. A straightforward *build process* for building a custom ROBOPuppet for a new robot using a robot CAD model, the provided joint encoder assemblies, and pocket geometries in conjunction with 3D printing. Wiring is also completed in this step.

4. A *calibration tool* that allows the user to easily create a mapping from the input encoder values to appropriate joint angles on the robot. This tool asks the user to place the puppet into several poses and automatically calibrates the mapping.

5. An *input sanitizer* that ensures that requested motions are safe and feasible for the robot to perform. A real-time motion planner translates raw inputs to output trajectories that satisfy kinematic and dynamic limitations of the robot. This element can also avoid self-collisions and collisions with the environment, if an environment model is available.

We claim that **a viable ROBOPuppet can be constructed for a target robot** at any scale such that the scaled links fit within the 3D printer's workspace, and are sufficiently large to embed the joint encoder assemblies.

3.3 Construction

The construction process can be further subdivided into geometry preprocessing, printing, and assembly phases. To build the ROBOPuppet for a new target robot, the robot's CAD model is scaled such that the smallest joint surface to be instrumented can successfully hold the joint encoder assemblies and, once scaled, the void primitives are subtracted out of each part. This section presents a step-by-step process for installation and assembly of the final puppet out of printed parts and joint encoder assemblies. The workflow for this process can be seen in Fig. 3.5.

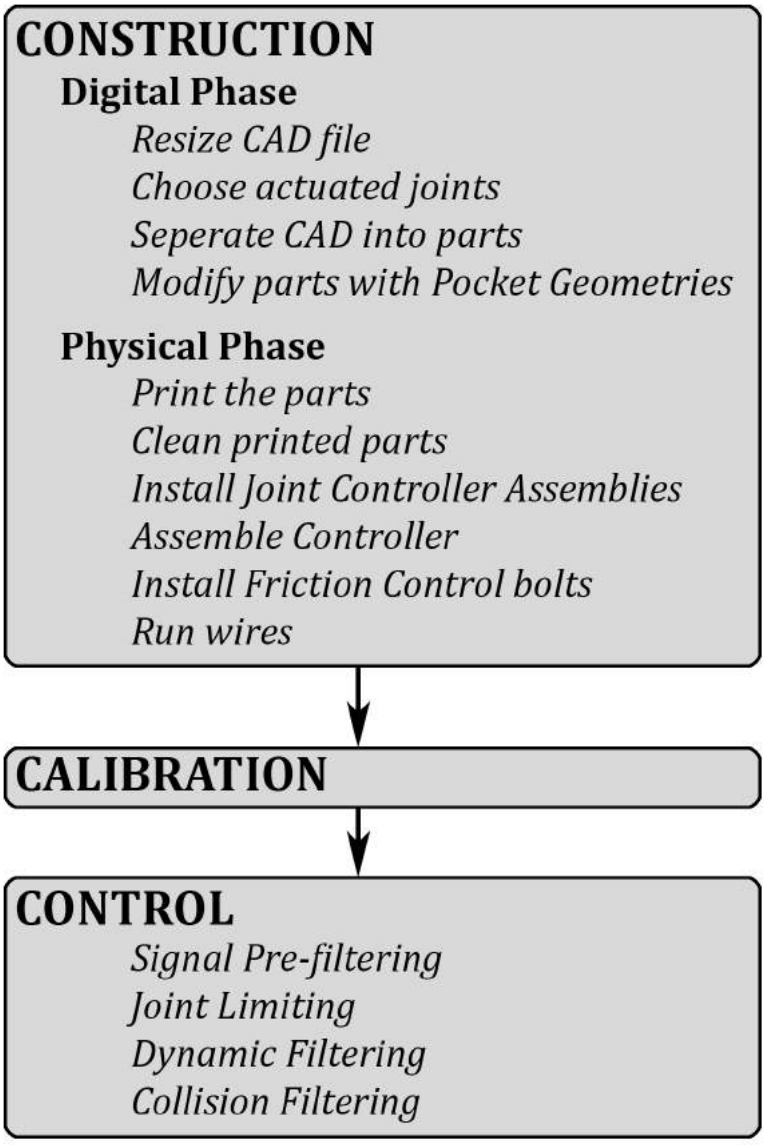


FIGURE 3.5: Workflow for create a ROBOPuppet controller.

3.3.1 Joint Encoder Assemblies

Our joint angle encoders use 5kΩ linear taper potentiometers. They are carved with threading down the length of the encoder shaft for the installation of the friction control bolt and a small hole is drilled to contain the roll pin, a small pin used to translate motion of the part into motion in the encoder shaft. The potentiometer joint encoder assembly has a base dimension of 2.5 cm x 3.25 cm x 1.75 cm with a shaft of diameter of 0.75 cm and a length of 5.25 cm and is installed in a printed bracket that is sized to easily fit in the modified parts. A mounted joint encoder assembly, ready to be installed in the printed parts, is shown in Fig. 3.6(b).

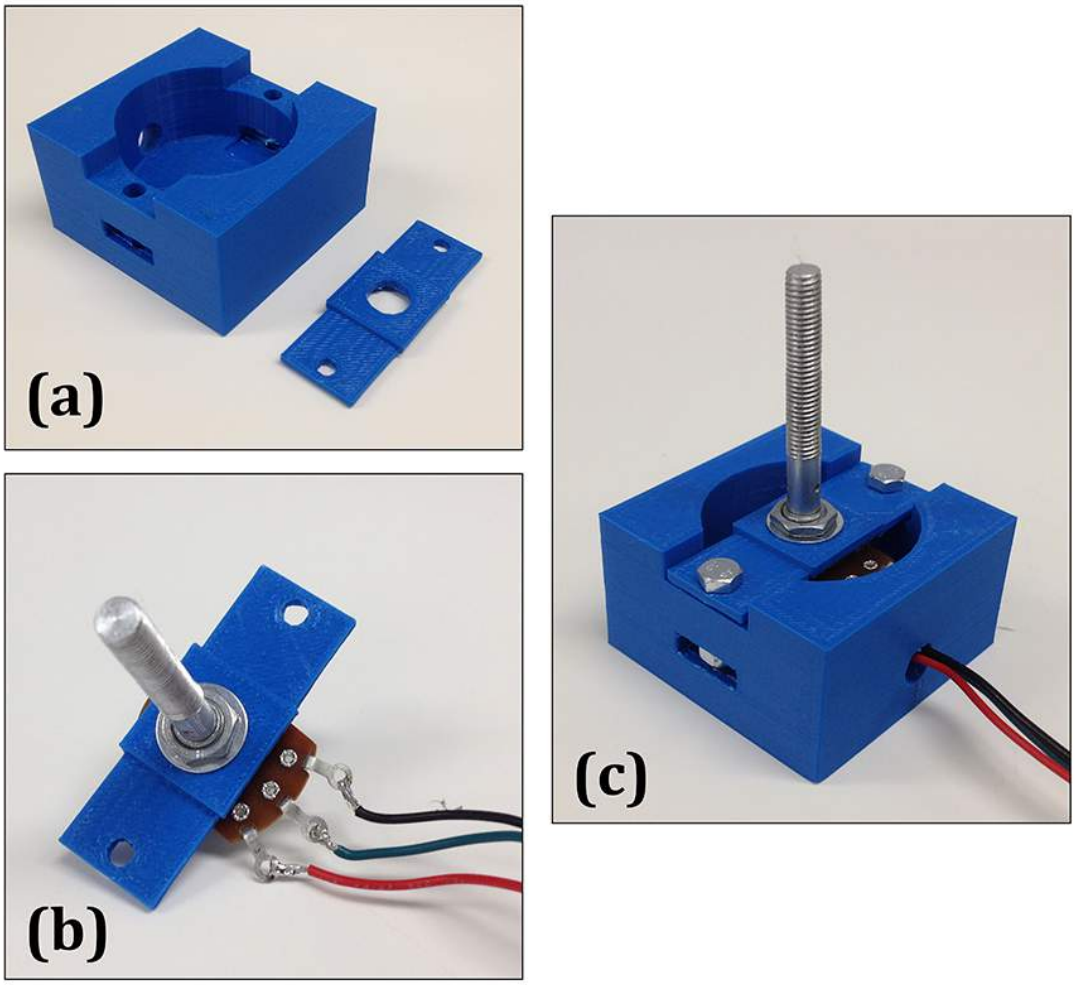


FIGURE 3.6: (a) Printed part with modified joint surface and mounting bracket. (b) Joint encoder assembly is installed into bracket (c) Bracket with mounted joint encoder assembly is installed in joint surface using bolts through the bracket attached with nuts accessed in the installation access tunnel.

3.3.2 Digital Phase

During the digital phase the target robot's CAD model is downloaded and the robot is scaled to the desired size. The minimum scale factor for a ROBOPuppet can be determined by identifying smallest joint surfaces that are to be controlled and ensuring that these surfaces can hold the joint encoder assemblies. The user chooses the joints to instrument and the CAD file is separated along these joints into individual meshes suitable for printing. Each mesh is modified using pocket geometries to provide space to access and install the joint encoder assemblies. (Fig. 3.7) **Pocket Geometries** Our build process uses three *pocket geometries* that have been designed to ease the modification of all model links to fit the joint encoder assemblies.

1. Joint Encoder Pockets - complex geometries that create a void in which the base

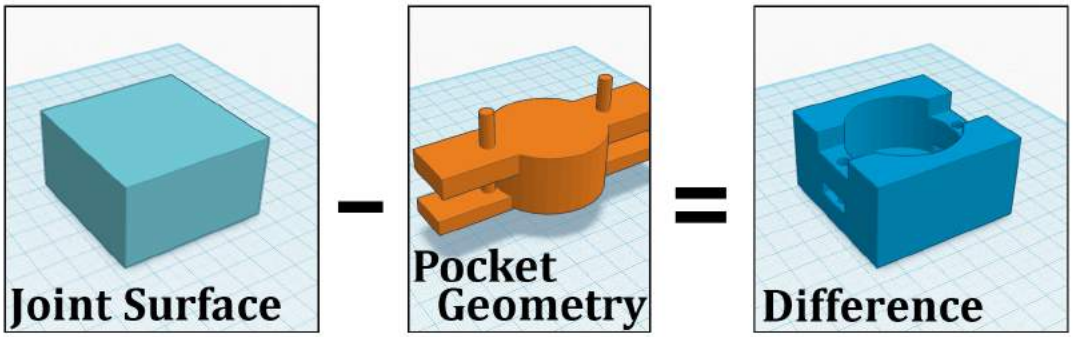


FIGURE 3.7: Pocket geometries.

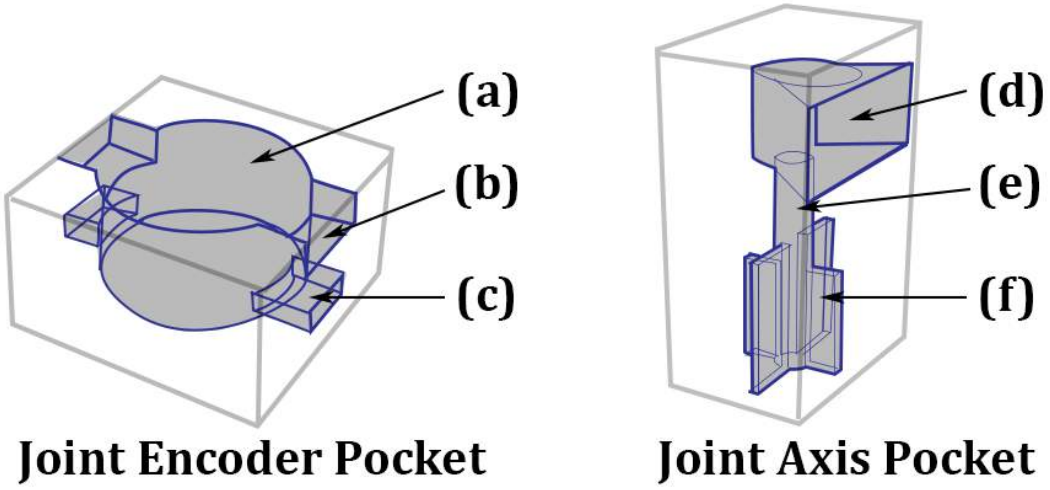


FIGURE 3.8: (a) Encoder void, (b) Encoder bracket mount point, (c) Installation access tunnel, (d) Friction bolt void, (e) Encoder shaft pocket, (f) Roll pin pocket

of the encoder resides, spaces to mount a bracket to affix the encoder to the joint surface and access tunnels to assemble the controller.

2. Joint Axis Pockets - geometries that create a pocket to hold the encoder shaft on opposing joint surfaces with voids for the roll pin and tunnels and voids for the friction control bolt.

3. Convenience voids - simple geometries used to create access tunnels and voids in printed parts to give access during installation and maintenance.

The pocket geometries were created in such a way that, during mesh modification, they are to be centered about the axis of rotation; this makes them simple to use and ensures that the encoder's shaft is centered properly and the part moves correctly when the controller is assembled. Examples of pocket geometries embedded within joint surfaces can be seen in Fig. 3.7 and Fig. 3.8.

3.3.3 Physical Phase

The physical phase of the construction stage involves printing the parts, installing joint encoder assemblies into the joint surfaces, and assembling the parts into the finished controller. **Printing.** The puppet is printed on a 3D printer. Post processing must be done in order to clean any extraneous material from the printing process. **Hardware installation.** At this point polystyrene foam sheeting is installed on joint surfaces to aid in friction control and creating a smooth action. Finally, joint encoders assemblies should be installed using the bracket mounts created in each joint surface. (Fig. 3.6) **ROBOPuppet Assembly.** Joints are assembled by first installing a roll pin into a hole pre-drilled into the encoder shaft and fitting the encoder shaft into the prepared joint axis pocket. Friction bolts are installed during this process to both fix the joint bodies together and adjust friction and action between the joint surfaces. Wiring was run on the outside of the model's surface and enough slack was left in the wiring to allow a full range of motion (Fig. 3.9). The joint encoders are aligned to ensure that the encoder limitations roughly match to the joint limitations on the target robot. Joints are assembled by first installing a roll pin into a hole pre-drilled into the encoder shaft and fitting the encoder shaft into the prepared joint axis pocket. Friction bolts are installed during this process to both fix the joint bodies together and adjust friction and action between the joint surfaces. Wiring was run on the outside of the model's surface and affixed using hot glue and purchased brackets with care taken to ensure there was enough slack between joint allow a full range of motion (Fig. 3.9). Once the puppet is assembled, the joint encoders are aligned to ensure that the encoder limitations roughly match to the joint limitations on the target robot.

3.4 Mapping Puppet Motions to Robot Motions

Once RoboPuppet is built, its joint angle values must be mapped to matching joint angles of the robot and transmitted to the robot controller. Free software solutions are used to keep the barrier of entry low. Reading the joint encoder values was done using the Arduino environment [32] and all calibration and control tasks were completed using the Klamp't software library [42]. We must overcome two challenges: first, the mapping from encoder values to the robot's joint angles must be calibrated; second, direct transmission of commanded joint angles leads to unsafe behavior. This section describes our approach to these issues.

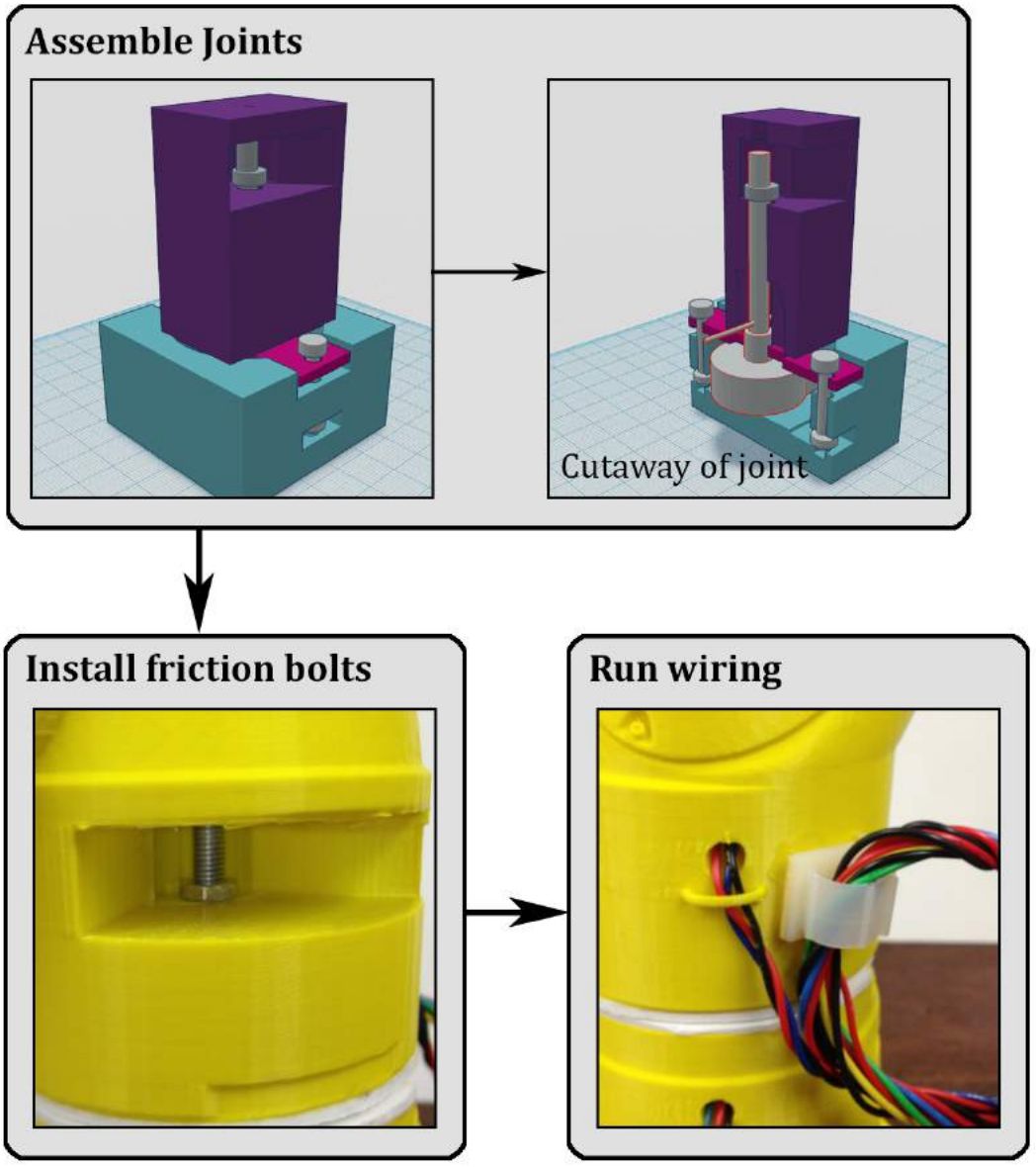


FIGURE 3.9: Installing the joint assemblies and friction control bolts in the printed model.

3.4.1 Calibration

Since the puppet uses linear potentiometers the mapping from encoder values to joint angles is also linear. To calibrate it we use a tool that asks the user to pose the puppet in a number of target robot configurations (two or more) displayed in a 3D GUI. The correspondence between puppet pose and robot configuration is then estimated using linear regression. Because posing the puppet by hand with visual comparison is not precise, we consider several techniques to reduce errors. The first option is to simply use many configurations and hoping errors average out. This is somewhat tedious for

the user, and and users can make systematic errors. A better option is to choose target configurations that *use physical correspondences* to reduce errors. Examples include configurations that lie in local minima / maxima of the gravity potential, configurations that make contact with natural landmarks, such as the puppet's table plane or self-contact, or configurations that line up with natural features on the robot's geometry. Due to the absence of strong lines on the Staübli TX90L we use the gravity- and contact-based approaches, with configurations that are vertically outstretched and configurations that touch the table with the elbow and end effector tip. We also observed that asking users to simultaneously pose many joints is harder than just posing one or two joints. A total of 5 configurations were used in our calibration program, with at most 2 joints moved between configurations (Fig. 3.10).

3.4.2 Safety Filtering

Instead of direct transmission of the puppets motion, we introduce filtering methods for overcoming the following safety issues:

1. The encoders are relatively low-precision and suffer from jitter,
2. Commanded values may violate the robot's joint limits due to mismatches in the encoders' joint stops,
3. The puppet may move too quickly for the robot to catch up,
4. The puppet may cause the robot to self collide or collide with objects in the remote environment.

These are addressed in sequence by the following steps. **Signal Pre-filtering.** Because the puppet's analog encoders are digitized to 10-bit values, the digitized values tend to oscillate between two adjacent values even when the puppet is not moving. To smooth these oscillations a deadband filter of width 1 is applied to the signal, as well as an exponential filter with smoothing factor 0.5, before applying the calibrated mapping. **Joint Limiting.** The smoothed joint angle command is capped to lie within the robot's joint limits. **Dynamic Filtering.** The puppet often moves or accelerates faster than the robot can, so we limit the joint angle command by the robot's velocity and acceleration limits. Simple acceleration and velocity limiting introduces oscillatory effects and can even cause the robot to overshoot its joint limits due to insufficient stopping room. So, we use an online trajectory optimizer [41] to produce time-optimal velocity- and acceleration-bounded trajectories that start at the robot's current configuration and velocity, and end in the target configuration. This optimization has an analytical

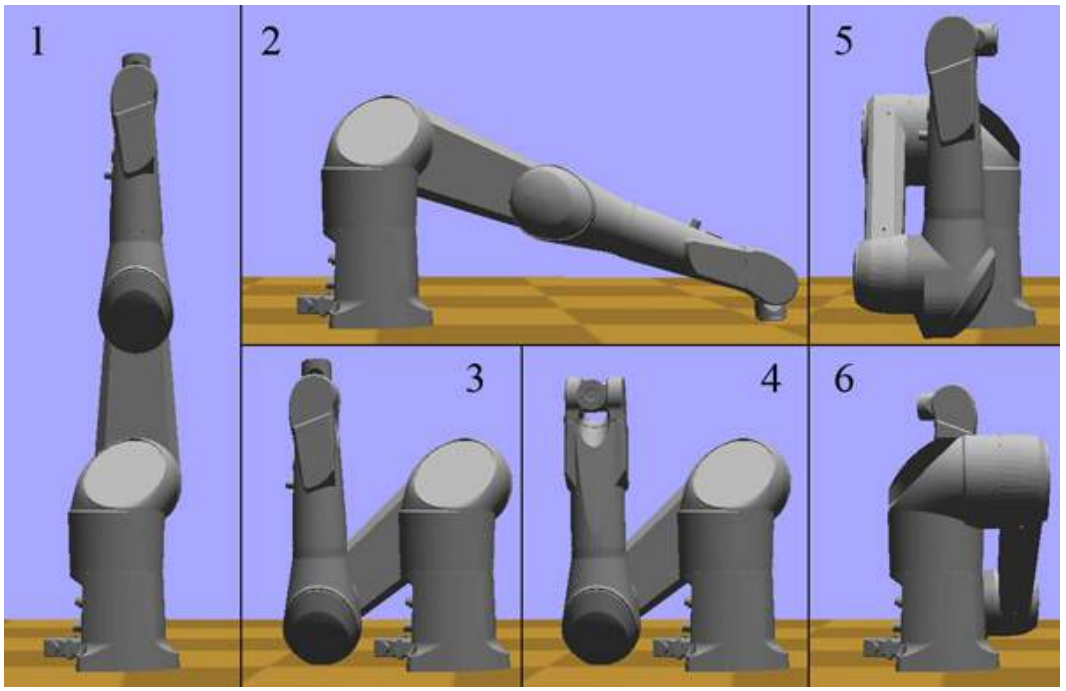


FIGURE 3.10: To calibrate the puppet, the user is asked to pose it in a sequence of configurations.

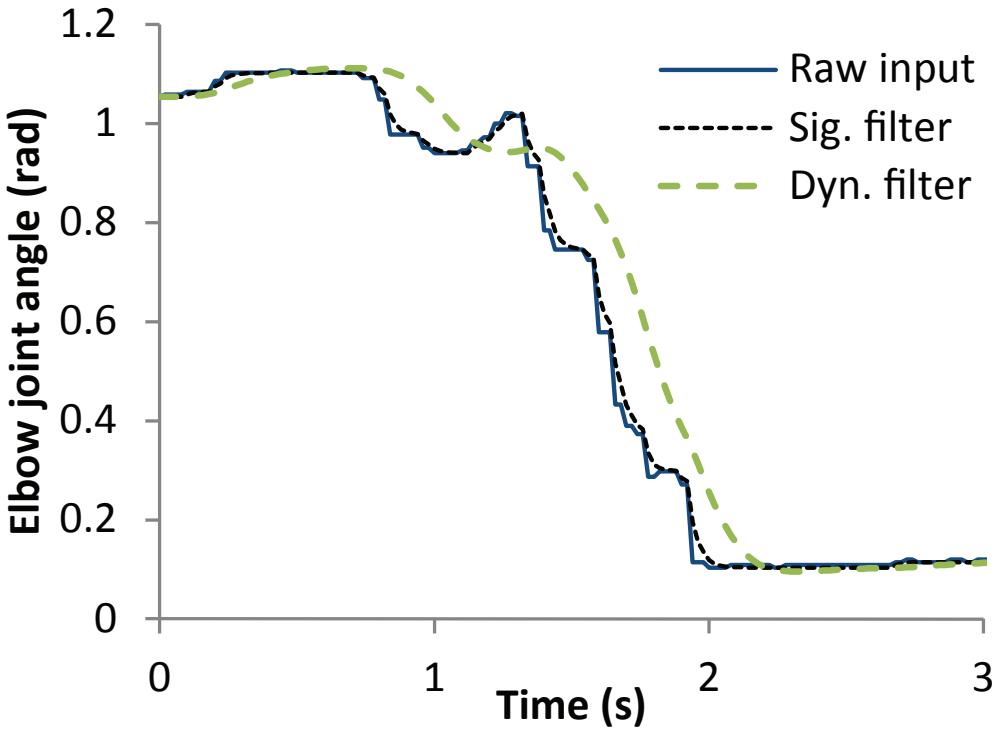


FIGURE 3.11: Raw, noisy joint angle commands (Raw input) are passed through a signal pre-filter (Sig. filter) to eliminate jerks, and then through a dynamic filter (Dyn. filter) to generate smooth trajectories that respect the robot's velocity and acceleration limits.

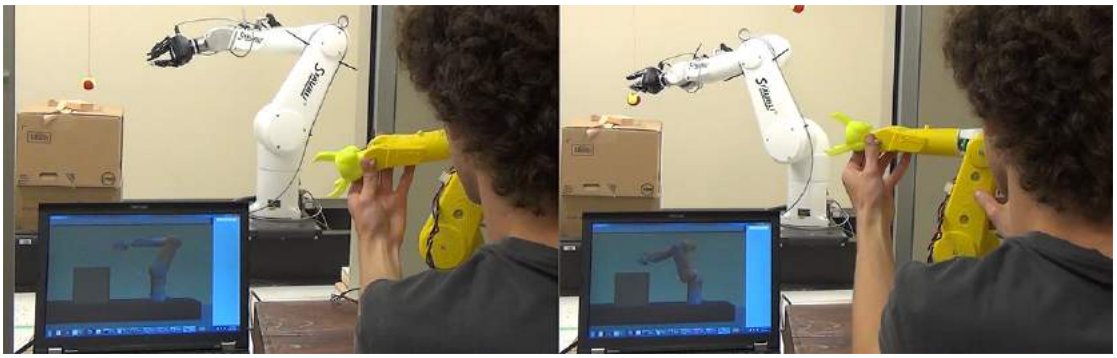


FIGURE 3.12: A first time user controls the Staübli TX90L robotic arm with the ROBOPuppet controller. The user was asked to interact with targets in the environment (hanging tennis balls and blocks on the table).

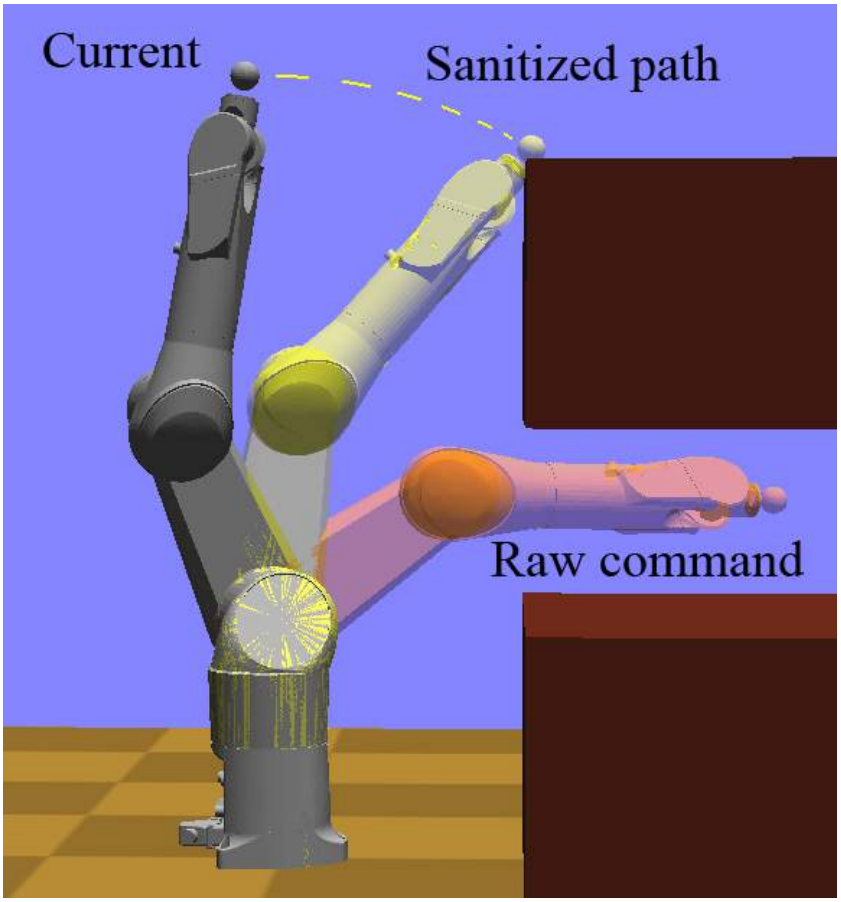


FIGURE 3.13: The input sanitizer includes a collision filtering step that prevents the robot from colliding into known obstacles.

piecewise-quadratic solution and hence can be done quickly enough to be performed at every time step. The effect is similar to the trajectory generation approach of [43]. The effects of signal filtering and dynamic filtering are depicted in Fig. 3.11. **Collision Filtering.** If the robot has a model of obstacles in its environment, then it can decide whether to accept the motion to the target configuration via collision detection (Fig. 3.13). Specifically, our system first generates a candidate optimal trajectory to

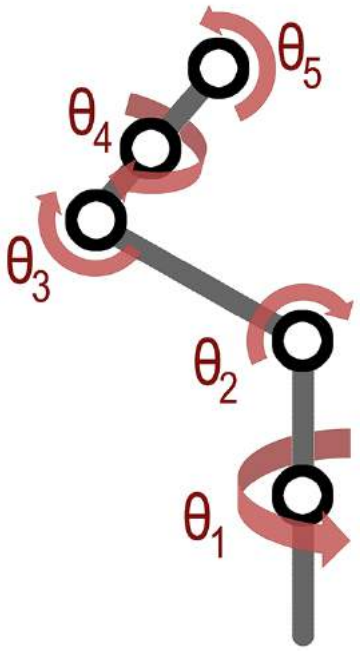


FIGURE 3.14: Five of the TX90L joints were instrumented, and the joint surfaces of each was selected in such a way to ensure the final controller mimicked the target robot's range of motion.

the target configuration, then runs collision detection on the trajectory (details in [41]). If the trajectory is collision free, it is accepted and the robot begins moving along it. If not, it is rejected and the robot continues along its current trajectory until the next input from the puppet is received.

3.5 Prototype Puppet for an Industrial Robot

Our prototype is a 30% scale controller for the Staübli TX90L robotic arm. The arm has six degrees of freedom, of which the puppet includes five. (Fig. 3.14) At this scale, the sixth joint is too small to contain the joint encoder assemblies and, thus, was not included. **CAD Preprocessing.** To keep the entry requirements for creating the controller low, we chose to use simple and free software (Freecad [37] and TinkerCAD [48]) to modify the robot's CAD files. A user with no previous 3D modeling or CAD experience was able to prepare each part for printing in approximately 15 minutes. This can be compared to 45 minutes or longer when not using premade pocket geometries. **Printing and assembly.** The device was printed on a Makerbot Replicator 2X using ABS 1.75 mm filament. 3D printing took 29 hours (Table 3.1) and required 0.483 kg of filament. Once the parts are printed, the assembly and joint encoder alignment can be completed in approximately 1.5 hours. Encoders and additional hardware were chosen to be affordable and readily available. All parts required for the model can be sourced

either from local stores or the internet and are chosen to be simple to use and wire. Printing is by far the most limiting step in our method. "Prosumer"-grade 3D printers

TABLE 3.1: Time taken for modifying the CAD mesh using pocket geometries and printing. (hours:minutes.seconds)

Link	**Digital Preprocessing**	**Printing**
L1	00:15	05:10
L2	00:20	06:37
L3 - top	00:10	02:57
L3 - bottom	00:10	04:14
L4	00:25	04:03
L5	00:15	04:07
L6	00:10	00:44
TOTAL	01:45	27:52

can require near-constant supervision, but rarely require intervention. As a result, most of the CAD processing and hardware installation tasks can be completed while waiting for the model to complete printing. **Cost.** Table 3.2 shows the costs of individual components, with a total puppet cost of approximately $85. For further savings, the most expensive component, the Arduino UNO, can be replaced with a less expensive microcontroller. **Preliminary testing.** The device and controller was implemented in

TABLE 3.2: Cost for all parts and materials.)

Equipment	**Cost**	**Amount**	**Total Cost**
5k-Ohm Linear Taper Potentiometer	3.49 each	5	$17.45
Stranded wire	10.00 kit	0.5	$5.00
Solder	6.49 spool	0.1	$0.65
Solderless Breadboard	8.90 each	1	$8.90
Arduino Uno R3	29.95 each	1	$29.95
ABS filament	0.048 per g	483	$23.18
TOTAL			$85.13

a physics simulator [42] as well as on the target robot, as shown in Fig. 3.12 and in the supplemental video. Undergraduate students, seniors, and children as young as 6 years old were able to perform a sequence of trial reaching tasks without instruction. Our observations are that large and medium scale movements are easily controllable, and collision filtering is necessary due to accidental movements, like dropping the puppet. We also observed that fine-grained positioning (within millimeters) is not yet possible due to the low fidelity of the encoders, but we imagine higher grade encoders could be used, or the puppet could be used in conjunction with alternative control methods.

3.6 Conclusions

This paper presented ROBOPuppet, a technique for building miniature robot control devices using 3D printing and low-cost electronics. It consists of a standardized encoder module, 3D printing procedure, and assembly process that is generalizable between a wide variety of robots. Novel methods for calibration and real-time input sanitization for collision prevention are also presented. A prototype puppet is presented for a 6DOF industrial robot with a materials cost of $85. In the near future we will produce another prototype for a more complex robot, such as the Rethink Robotics Baxter. We also intend to share parts and modifications [39], with the goal of cultivating an online community of researchers, citizen scientists, and robot enthusiasts. The current iteration of ROBOPuppet has a few limitations that we intend to address in future work.

- We used encoders with a range of 300°, which may not be sufficient to capture the joint range of a given robot. Multi-turn or continuous encoders could be used instead.
- Only hinge type joints or joints that can be represented using rotary motion can be replicated using the current method. We are exploring solutions for alternative joint types.
- Robot joint limits could be implemented as physical stops in the puppet itself, providing immediate feedback.
- The current joint encoder assemblies may not fit into small links, placing a lower bound on the size of the model. We are currently working on a smaller joint encoder assembly. that does not use the potentiometers as a structural element, allowing the use of miniature potentiometers. This will enable smaller puppets, and will expand the range of applicable robots.
- Mobile robot navigation control may be implemented with new encoder modules, such as the trackballs of computer mice.
- The current implementation does not provide haptic feedback. Standardized motor/encoder assemblies could provide this functionality.
- Direct control may be inappropriate for robots that must walk and/or maintain balance (e.g., bipeds). Separate balance control filtering may need to be implemented.

We also wish to perform user studies to test the hypothesis that ROBOPuppet makes robots easier to control than alternative input modalities, such as joysticks, mice, and

off-the-shelf haptic devices. It is our intention to foster a community of researchers and citizen scientists leveraging current rapid prototyping communities who can share, improve and use the ROBOPuppet pocket geometries, joint encoder assemblies and general methods. As a ROBOPuppet controller can be used to move robots in simulation, these controllers are an excellent way to work with robots that are not immediately accessible. Robots that need to support themselves or balance, particularly mobile robots or robots that are not affixed to a supporting platform, are not suitable candidates for the ROBOPuppet method without additional considerations in place. These robots can be modified during assembly/printing to be mounted on a supporting bracket that is affixed to a platform or only portions of the robot can be modeled(for example, a humanoid robot may benefit from having only the torso, arms and head modeled and assembled in the ROBOPuppet controller.)

Chapter 4

Adapting a Robotic Arm for Gait Rehabilitation

The purpose of this study was to adapt a multipurpose robotic arm for gait rehabilitation. An advantage of this approach is versatility: a robotic arm can be attached to almost any point on the body to assist with lower- and upperextremity rehabilitation. This may be more cost-effective than purchasing and training rehabilitation staff to use several specialized rehabilitation robots. Robotic arms also have a more human-like morphology, which may make them less intimidating or alien to patients. In this study a mechanical interface was developed that allows a fast, secure, and safe attachment between a robotic arm and a human limb. The effectiveness of this interface was assessed by having two healthy subjects walk on a treadmill with and without a robotic arm attached to their legs. The robot's ability to follow the subjects' swinging legs was evaluated at slow and fast walking speeds. Two different control schemes were evaluated: one using the standard manufacturer-provided control algorithm, and another using a custom algorithm that actively compensated for robot-human interaction forces. The results showed that both robot control schemes performed well for slow walking. There were negligible differences between subjects' gait kinematics with and without the robot. During fast walking with the robot, similar results were obtained for one subject; however, the second subject demonstrated noticeable gait modifications. Together, these results show the feasibility of adapting a multipurpose robotic arm for gait rehabilitation.

4.1 Robotic rehabilitation: a brief introduction

Restoration of walking ability is an important goal of rehabilitation following neurological disorders such as stroke or spinal cord injury. Conventional gait training programs are often labor intensive. For example, in body-weight-supported treadmill training, physical therapists provide manual assistance to move a patient's leg and/or pelvis in a desired trajectory, which can demand high therapist effort. There is a growing interest in using robots to automate gait training and relieve the physical burdens placed upon therapists. Consequentially, there has been a proliferation of robotic devices for gait rehabilitation applications. These robots are able to alter gait by pushing and/or pulling on a person, and have met with varying degrees of success in terms of rehabilitation outcomes [50–55]. Most gait rehabilitation robots are highly specialized devices that perform singular functions. Although many of these robots are quite sophisticated, as with any device, they have limitations. Some designs may be intimidating to patients (e.g. a large exoskeleton or several wires and push-rods); others may require long and involved patient setups. Instead of adding to the current stock of customized gait rehabilitation robots, an alternative approach may be fruitful. Specifically, adapting a commercially available robotic arm for gait rehabilitation may provide advantages over traditional approaches. A robotic arm could perform many functions, e.g. it could be attached to any point on the body for lower- and/or upper-body rehabilitation. For universities and clinics this may be more cost-effective than purchasing several specialized robots. Robotic arms also have a human-like morphology, which may make the robot less alien to patients (e.g. by resembling a therapist holding onto a patient's leg). Finally, using an off-the-shelf robotic arm would allow researchers to focus on improving robot control algorithms for greater success in gait rehabilitation. Using a multipurpose robotic arm for gait rehabilitation holds promise, but several challenges must be overcome. First, there must be a way to attach the robot arm to a human limb, preferably one that takes a minimum amount of time. Second, there should also be a mechanism to automatically and instantaneously detach the robotic arm if a patient stumbles. Finally, the robotic arm should be transparent, i.e. it should be able to follow a moving subject and not interfere with their nominal gait. This paper first presents a mechanical interface that allows for a safe connection between a robotic arm and a human limb. Next, the ability of a robotic arm to follow the leg of healthy subjects walking on a treadmill is evaluated, and the potential for gait rehabilitation assessed.

4.2 Robot-Human interface

4.2.1 Mechanism Overview

A mechanism was developed that allows a robotic arm to be attached to the human body in a quick and safe manner. The mechanism consisted of two parts: one that attached to the robot and another to the human body. The *robot attachment* included a ball joint to increase mobility and a rare-earth magnet for attachment/detachment. The *human attachment* consisted of a limb brace with a receptacle for the robot attachment. Both attachments, which coupled a Whole-Arm Manipulandum (WAM, Barrett Technology, Inc., Newton MA, USA) to a human leg, are shown in Figure 4.1.

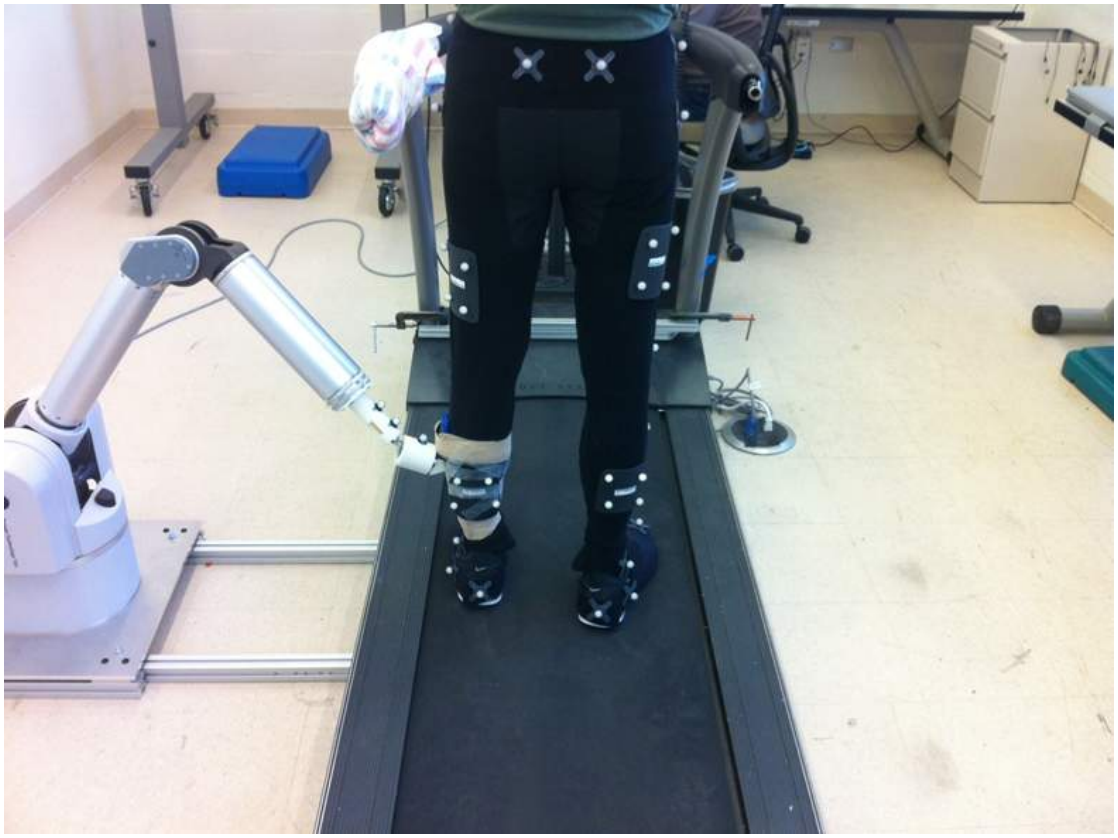


FIGURE 4.1: Robot attached to the lower leg.

4.2.2 Robot Attachment

A one-piece plastic base cylinder was made with a 3D printer (U-Print SE Plus, Stratasys, Eden Prairie, MN, USA) and attached to the WAM tool plate (Figure 4.3). A second plastic cylinder that housed a ball joint was attached to the base cylinder with a screw (Figure 4.4). The ball joint was added because the four degree-of-freedom WAM could

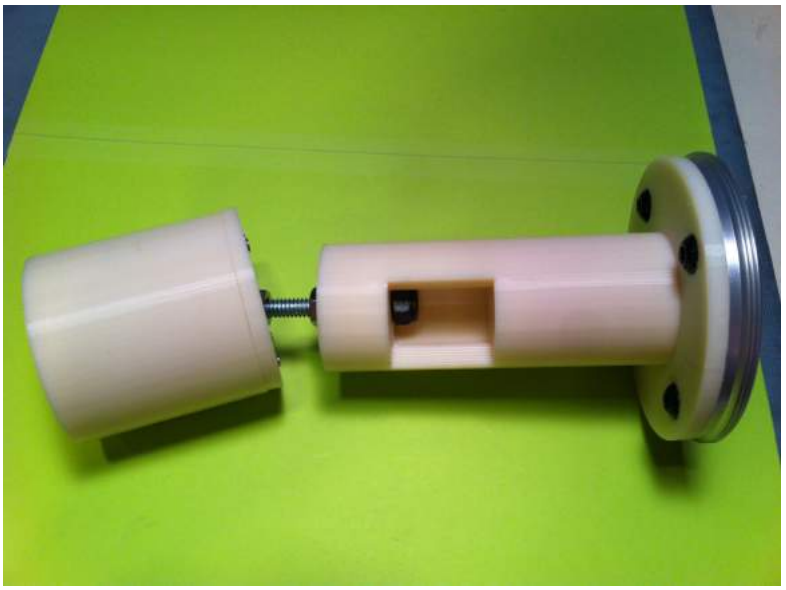


FIGURE 4.2: Robot attachment.

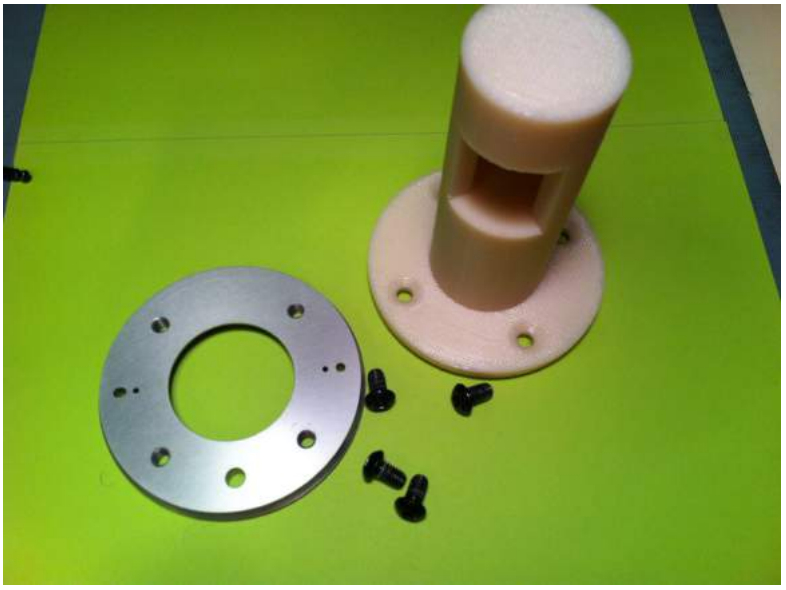


FIGURE 4.3: Plastic base cylinder.

not follow the leg of a person walking on a treadmill without reaching a range-of-motion limit. Although a powered "wrist" could be added, using a ball joint was a simpler and sufficient solution. The ball joint had a zinc-plated steel housing with a nickel-plated steel ball and oil-impregnated bronze insert; the maximum swivel angle was 42° (Part 4786T7; McMaster-Carr, Princeton, NJ USA). A rare-earth magnet was attached to the end of the plastic cylinder to connect the robot to the human (Figure 4.5). Trialand-error was used to select a magnet strong enough to stay attached during normal activities, but would also detach if the robot and/or subject behaved abnormally. A neodymium ring magnet (RX033CS-S; K&J Magnetic Inc.) was used with an axial pulling force of 20.51 lbs (single magnet vs. steel plate).

FIGURE 4.4: Ball joint and socket.

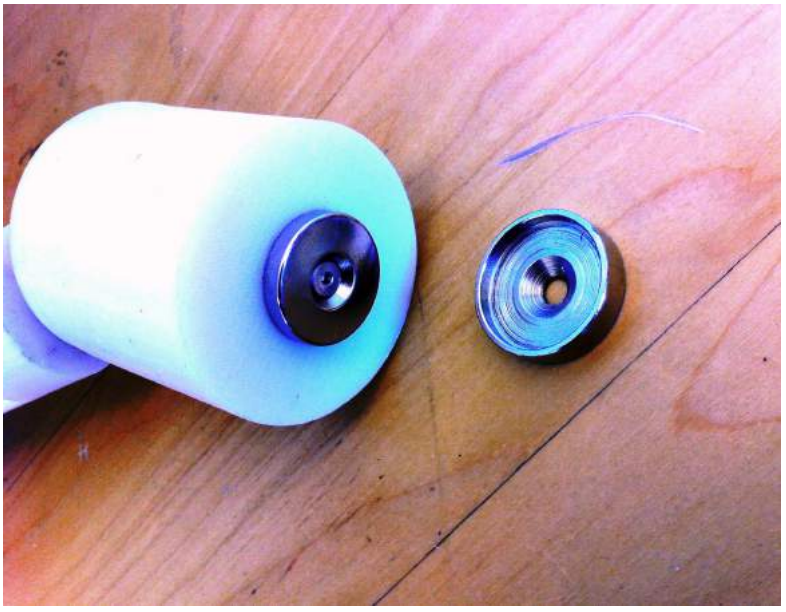


FIGURE 4.5: Rare-earth magnet.

4.2.3 Human Attachment

The human attachment (Figure 4.6) consisted of a standard air/gel ankle brace (Figure 4.7; DeRoyal Industries, Powell, TN), a 3D-printed plastic plate, and a steel cup (Figure 4.8). The plate was screwed on the brace, on which a cup was attached to receive the magnet. This plate was angled to be in line with the WAM to reduce the angle between the WAM and the ball-joint.

FIGURE 4.6: Human attachment.

4.3 Evaluation of robot-human interface

4.3.1 Robot Control

For gait rehabilitation with a robotic arm it is not only important to have a suitable attachment mechanism, the robot should also be able to follow a patient without imposing unwanted forces, i.e. the robot should be transparent. The WAM uses low friction cable drives that allow the mass of the motors to be located away from the end-effector, keeping inertia (and friction) low. This makes the WAM relatively transparent; however, the degree of transparency for gait rehabilitation applications is unknown: would a person be able to walk "normally" with the WAM attached to their leg? To answer this question the WAM's transparency was evaluated under two gait conditions: 1) with the WAM's nominal operation mode, which controls the motor currents, and 2) with a control scheme that also actively compensates for robot-human interaction forces. In

FIGURE 4.7: Ankle brace.

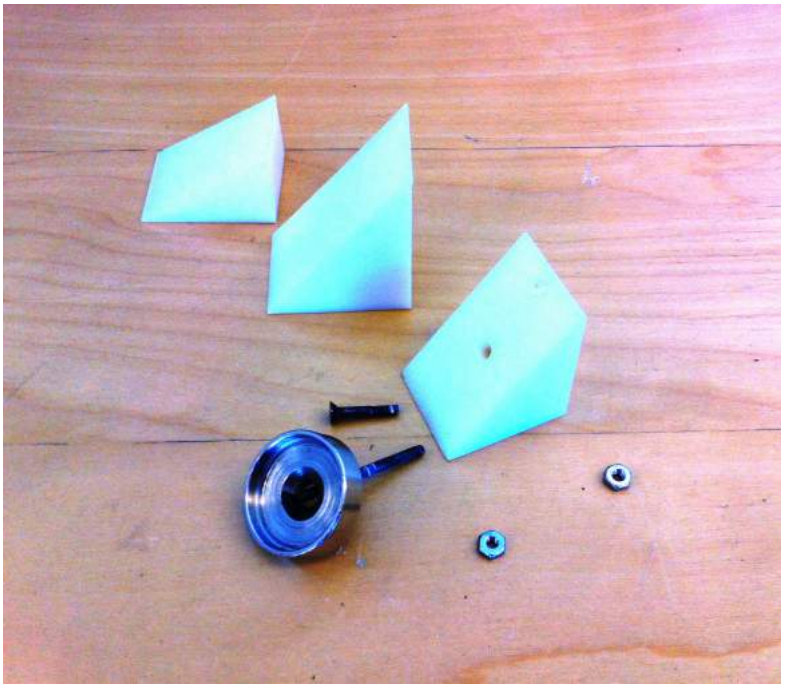


FIGURE 4.8: Socket for magnet and plastic plates for different angles.

this study the former is called *standard contro*l, and the latter, *force control*. **Standard Control**: In this control scheme the WAM controller commands a particular current to each motor to compensate for gravitational forces. When an external force is applied to one or more of the WAM linkages by a human, it turns the WAM motors and therefore the current within the motors changes, which generates torques that resist motion. However, when this occurs, the WAM controller immediately adjusts the motor currents to maintain the commanded currents. This way, to move the robot a human operator needs to only overcome relatively low inertial and frictional forces, and does not need to

exert additional force to move the motors. The net effect of this control scheme is that the WAM will maintain a position in space, yet can be easily moved by application of external forces. **Force Control**: The inclusion of a three-axis force and torque sensor on the WAM end-effector allows force control to be implemented. This was achieved by using the manufacturer-supplied Libbarrett C++ library to operate the WAM with a Jacobian transpose controller [56]. Given an external force F at the end-effector in Cartesian space and the Jacobian for a joint configuration, the robot joint torques τ are calculated as

$$\tau = J(q)^T kF \tag{4.1}$$

where $J(q)^T$ is the Jacobian transpose, q are the joint angles, and k is a proportional gain ($k = 1.6$). The torques are applied to the motors such that the end effector moves in the direction of the external force. This reduces robot-human interaction forces. The magnitude of τ is regulated by k; increasing k makes the robot follow the external force more aggressively. However, if the gain is too high instability may result.

4.3.2 Subjects

The ability of the WAM to follow the leg of two healthy subjects walking on a treadmill was evaluated. The subjects signed an informed consent document approved by the Northeastern University Institutional Review Board.

4.3.3 Experimental Setup

Subjects walked on a motorized treadmill (GK2200, Mobility Research, Tempe, AZ). Lower extremity motion was captured using an optical motion capture system (Oqus 300; Qualysis, Gothenburg, Sweden). Reflective markers were placed on both the legs and pelvis following the standard Visual 3D convention (C-Motion, Germantown, MD).

4.3.4 Protocol

There were three evaluation conditions: 1) no robot, 2) robot with standard control, 3) robot with added force control. Each condition was performed for one minute at 1 and 2 mph. The faster speed was close to the subjects' preferred walking speed; the slower speed was chosen because patients often walk at a slower speed. The WAM was attached to a point midway between the left lateral malleolus and left lateral femoral epicondyle, i.e. mid-shank (Figure 4.1).

4.3.5 Data Reduction and Analysis

The motion of the left ankle was used to characterize the influence of the robot on subjects' gait and assess transparency. To account for potential drifting of the subject on the treadmill, the anterior-poster and medial-lateral left ankle (lateral malleolus marker) positions were referenced to the left hip (iliac crest marker). The approximate time of heelstrike was identified as local minimums in the vertical displacement of the heel (heel marker). Using the heel-strike event, the ankle kinematics were segmented into individual gait cycles. Linear interpolation was used to determine the ankle position at integer percentages of the gait cycle (0 - 100%). For each of the experimental conditions, the cycles were averaged and 95% confidence intervals (across cycles) calculated. The velocity of the ankle was calculated by differentiating the position data with respect to time using the central difference method, and this data was segmented and averaged in the same way as the position data. The velocity calculation used a smoothed version of the position data, smoothed with a Savitzky-Golay FIR filter (polynomial orderof 4 and frame size of 21). The toe-off event was identified by finding local minimums in the toe marker anterior-posterior position data, which represent the time at which the foot transitions from moving backwards on the treadmill belt into swing. Swing time was calculated as the time between toe-off to the subsequent heel strike (of the same limb) and was expressed as a percentage of the gait cycle.

4.4 Results

The goal of the analysis was to evaluate the transparency of the WAM for gait rehabilitation. The ankle position and velocity are shown for Subject 1 in Figure 4.9. The no-robot (green line), robot with standard control (blue), and robot with force control (red) data are shown for the slow walking speed. The locomotor pattern of Subject 1 with the robot operating under both the standard and force control schemes was similar to their gait without the robot. The force control scheme provided a slightly better match during late-swing. For a different perspective, the average two-dimensional ankle kinematics for both subjects is shown in Figure 4.10 for the different robot control schemes and fast and slow walking. As shown in Figure 4.9, for Subject 1 there were only small differences between the no robot condition and those with the robot for slow walking; however, there were larger differences for fast walking (Figure 4.10; lower-left panel). In the latter case, the heel tended to lift off slightly earlier in the gait cycle with the robot, and as the leg swung through the leg appeared to drag, such that the ankle was behind its position without the robot. This is supported by the analysis of the average leg swing times, which became longer with the robot (Subject 1; fast walking

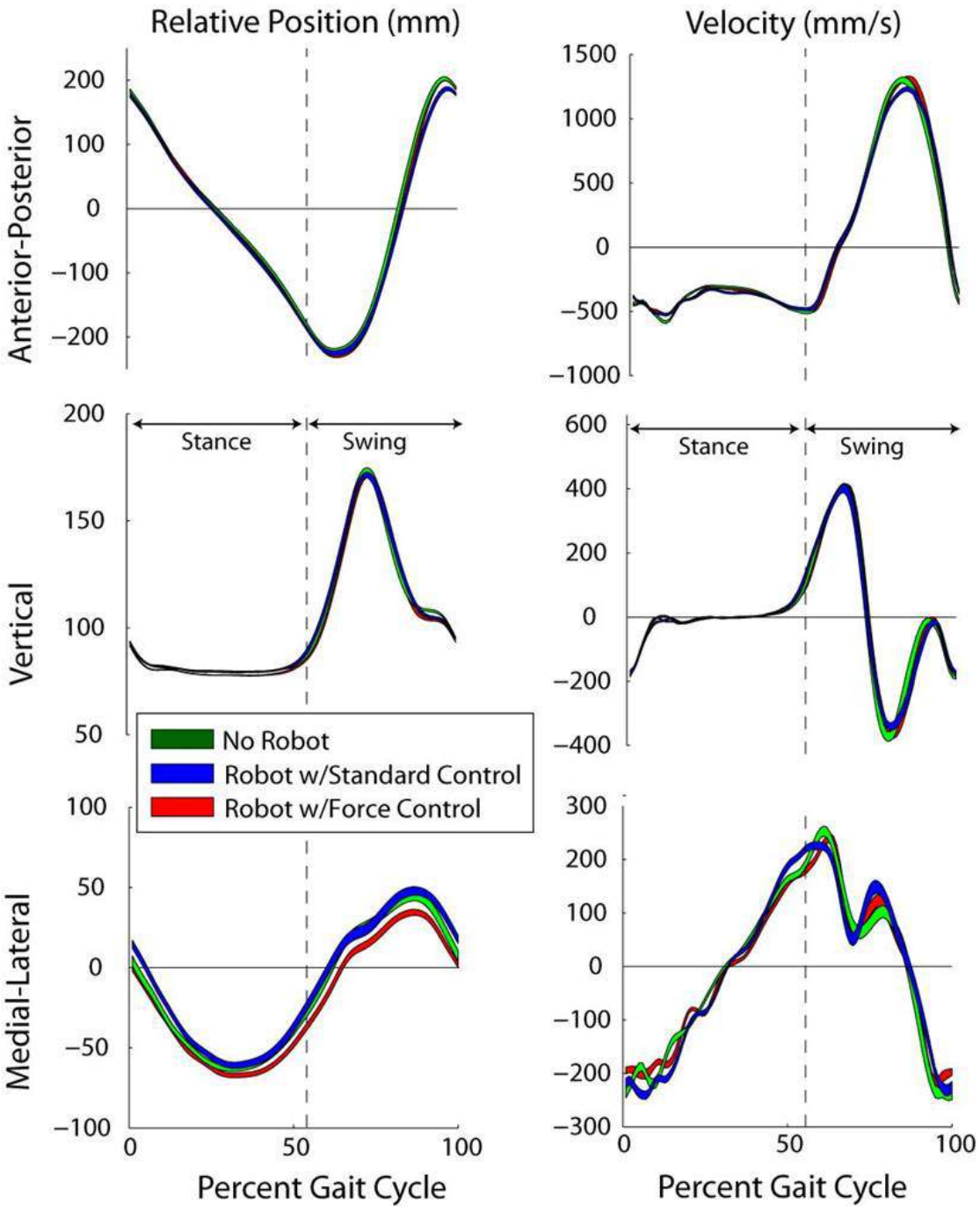


FIGURE 4.9: Left ankle kinematic.

condition; Figure 4.11). For the other subject, Subject 2, there was good correspondence between the no-robot and with-robot conditions for both slow and fast walking (Figure 4.10). The only noticeable difference was a longer swing time for slow walking with the robot under standard control (Figure 4.11).

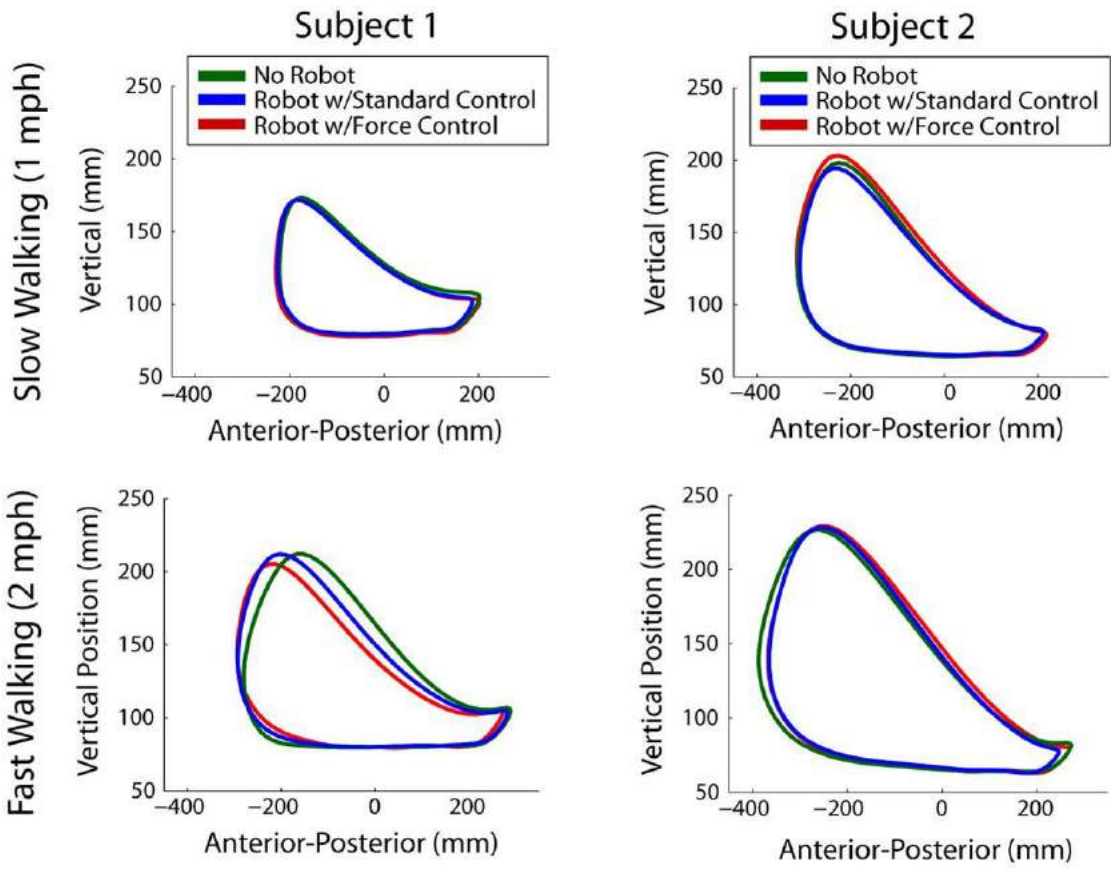


FIGURE 4.10: Average two-dimensional ankle kinematics.

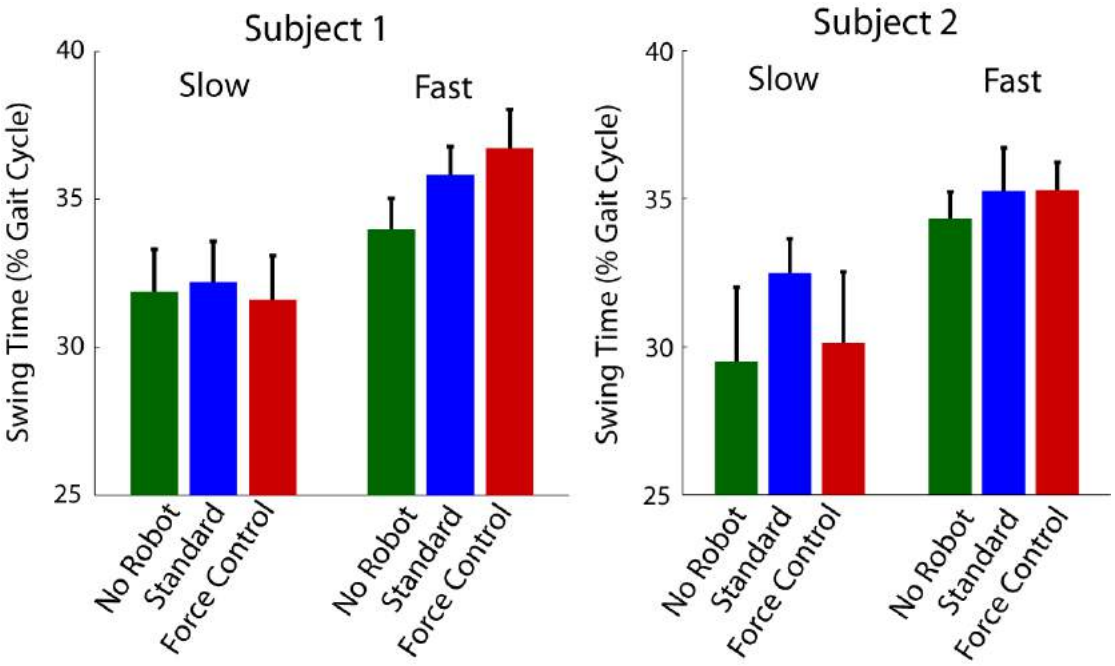


FIGURE 4.11: Average leg swing time.

4.5 Conclusions

The results show the feasibility of adapting a robotic arm for gait rehabilitation. A robot-human interface was developed, which allowed a firm mechanical connection between the robot and a human limb, yet also permitted freedom of movement for the human relative to the robot. A magnetic coupling allowed for fast attachment, as well as quick detachment in the event of an anomaly or malfunction in the robot or human controllers. The WAM, a commercially available multipurpose robotic arm, was used as the robotic platform. The robothuman interface performed well. Once a subject was ready on the treadmill, it took only seconds for the experimenter to securely attach the robot to the subject's leg, and the robot remained attached throughout the trial. Although only a single attachment site was used on the leg, it would be straightforward to use the same attachment system on a different body location, such as the upper leg or pelvis. The standard WAM controller allowed the robot to act in a transparent manner, such that it could follow a human's leg while walking on a treadmill. Transparency was acceptable for both test subjects during slow walking (1 mph). This is important because many patients tend to walk with a slower speed. When the walking speed was increased (2 mph) the effectiveness of the robot control algorithms became subjectdependent. In one subject the robot was able to keep up with the leg (Subject 2), but in another subject it appeared that the robot slightly dragged the leg (Subject 1). It was expected that adding an additional layer of forcebased compensation to the control scheme would improve the robot transparency. This is because the standard WAM controller does not compensate for friction and inertia. When the leg lifts from the treadmill and accelerates into swing, there will be an inertial force from the robot. Under force control, this force would be detected and the robot will try to move in the direction of the force, which should ultimately reduce the force felt by the subject. Contrary to this expectation, for Subject 1 during fast walking, the forcebased control seemed to drag the leg more than the standard control scheme. At this time the cause of this effect is unclear; it was not due to higher leg velocities - the peak ankle velocity was similar for both subjects in fast walking. Although these results are promising, they are based on a limited sample of healthy adults. Further experimentation is needed, particularly on patient populations. Also, the robotic interface was attached to only one leg. However, there is no reason to expect that the transparency would be significantly different if a robotic arm was attached to each leg. Nevertheless, a one-sided robotic interface would be suitable for some patient populations, such as those with chronic stroke, who primarily have unilateral deficits.

Chapter 5

The Baxter Easyhand

This paper introduces and characterizes the Baxter Easyhand, a new 3D printed hand derived from the Yale T42 hand [71], [77], but designed specifically to be mounted on the Baxter robot from Rethink robotics. Because this hand is designed specifically for Baxter, we are able to make some important simplifications in the design relative to other 3D printed hands. In particular, the Easyhand is smaller than most other 3D printed hands and it is powered by the native Baxter gripper actuator. As a result, our hand is cheaper, lighter, and easier to interface with than other robot hands available for Baxter. This paper details the design of the hand and its mechanical characteristics and reports results from experiments that characterize its grasping performance.

5.1 3D printed robotics hands

Underactuated robot hands produced using 3D printing have recently become a viable alternative to conventional commercially available robot hands. One of the key motivations for using a 3D printed hand rather than a commercial alternative is cost: 3D printed hands can typically be manufactured in a research lab for less than $1000 US dollars. In particular, the Yale OpenHand project makes available drawings, parts lists, and assembly instructions for a series of 3D printed hands [71], [77]. This is important because it gives many more researchers access to relatively sophisticated robot hands. The alternative is to spend tens of thousands of dollars to purchase a robot hand from a vendor such as Schunk, Barrett Technology, Robotiq, *etc.* Moreover, the compliant and underactuated nature of compliant 3D printed hands makes them suitable for a variety of robust grasping scenarios. In this paper, we present and make available a new underactuated robot hand specifically designed for use with the Baxter robot from Rethink Robotics. Our goal is to provide researchers who use the Baxter robot

with an inexpensive robot hand that is more flexible and robust than the native Baxter gripper. Easyhand assembly instructions will be available at the following URL: http://www.ccs.neu.edu/research/helpinghands/easyhand/. Our design is derived from the T42 hand (one of the Yale OpenHand hands [77]), but differs from the T42 hand in important ways:

- it is cheaper to build (approximately $150US in parts);
- it is lighter;
- it is smaller and therefore better suited to typical Baxter grasping scenarios;
- it is easier to use and control because it uses the native Baxter gripper actuator and drivers.

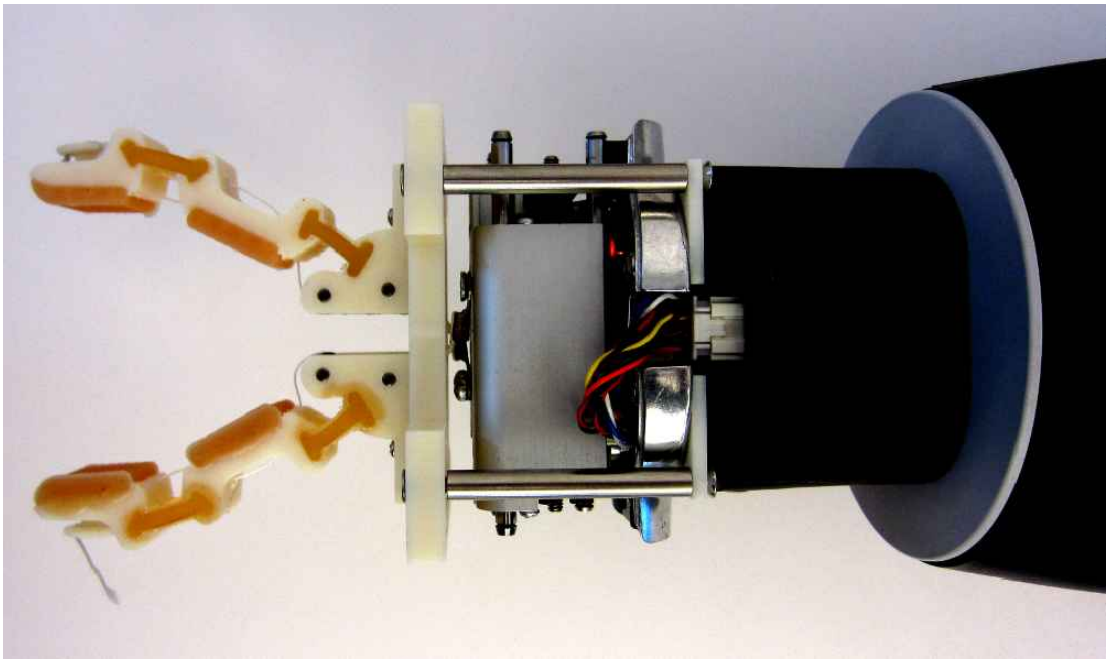


FIGURE 5.1: The Baxter Easyhand

To our knowledge, the Baxter Easyhand is currently the most inexpensive "research-grade" robot hand available. The key feature of the Baxter Easyhand relative to other 3D printed hands is that the fingers are actuated using the native Baxter gripper actuator rather than a Dynamixel servo. This results in the advantages cited above. It reduces the weight of the hand since it is no longer necessary to lift the additional servos. It makes the hand easier to use because we use the native Baxter gripper ROS driver. This driver already incorporates all the relevant hand speed/force settings. Finally, the Easyhand is smaller than most other 3D printed hands. This is important because, with a maximum payload of five pounds, the Baxter arm is essentially designed to lift small light objects.

FIGURE 5.2: The Baxter Easyhand in use

The smaller size of the Easyhand makes it more compatible with Baxter manipulation scenarios. Compared to the native Baxter parallel jaw gripper, the Easyhand has one additional critical advantage: it can grasp any object that fits within the 8 cm aperture between the fingers. Each finger on the native Baxter parallel jaw gripper can only translate a maximum of 2 cm. As a result, it is impossible to grasp objects larger than 4 cm in diameter without manually removing the fingers and re-attaching them to the hand in a configuration where they are further apart. However, if one does this, then the fingers will no longer touch when they are closed fully. In contrast, the Easyhand fingers will close from a 8 cm aperture all the way to a fully closed configuration. We have designed the tendon arrangement so that the maximum travel of the underlying native Baxter gripper corresponds roughly with the maximum travel of the Easyhand fingers.

5.1.1 Related Work

The design and manufacture of robot hands appropriate for industrial and research applications remains an important challenge. A number of robot hands are currently available commercially including the Barrett Hand [62], the Robotiq Adaptive Gripper [60], the Shunk hand [66], the Shadow Dexterous hand [78], and the Ottobock hand [61].

However, all these hands are relatively expensive (tens of thousands of US dollars) and several of them are heavy and large. Because many robot hands are significantly larger than human size, they typically have a hard time grasping many of the smaller objects that humans grasp easily (pens, credit cards, *etc.*). The Baxter robot comes equipped with a two-finger parallel jaw gripper that is light and easy to control. However, this gripper has a very limited stroke: each finger can only translate two cm. Compared with the Baxter gripper, the Easyhand has a larger stroke but retains a similar form factor. It is often the case that the most compact robot hands are integrated with an arm: the Meka complaint hand [64], [65] and the Robonaut 2 [79] hand are examples. Recently, there has been interest in underactuated 3D printed hands that can be produced using shape deposition manufacturing (SDM). The first robot hand to use SDM was the Harvard SDM hand [72]. The iRobot-Harvard Hand [74] from RightHand Robotics is a commercial hand based on SDM that was used in the DARPA Robotics Challenge. Recently, the Yale OpenHand Project [77] has made 3D printed hands based on SDM available to the wider public. Perhaps the most important feature of hands manufactured using SDM is that they can be extremely inexpensive to produce (assuming you have access to a 3D printer). All of the hands in the Yale OpenHand Project (model T, T42, O, and M2) can be built with a 3D printer, hardware costing approximately \$150 – \$200 US, and a set of Dynamixel actuators costing approximately \$400– \$800 US. These underactuated hands are able to grasp a variety of object shapes and sizes by passive adaptation of the finger conforming to the geometry of the object. The iRobot-Harvard Hand [74] and the Yale Open Hand [70], which both originated from the SDM hand [67], are examples of recently developed underactuated hands that are open-source, and can thus be reproduced with a 3D printer and a simple set of tools. These underactuated hands are able to grasp a variety of object shapes and sizes by passive adaptation of the finger conforming to the geometry of the object. In part, these hands are result of the new possibilities opened up by shape deposition manufacturing [67], [68] and 3D printing [69].

5.2 Design

Although the baxter easyhand mechanical design is derived from the Yale T42 hand, there are some key differences. Not only is the Easyhand smaller, but the tendon arrangement, routing, and termination is different. These differences result from our choice to actuate our hand using the same actuator as the native baxter gripper rather than using dynamixel servos. The Baxter Easyhand consists of two fingers with compliant flexure joints that are driven by the native Baxter gripper actuator. Our design incorporates a tendon arrangement that maximizes squeeing force given the travel available

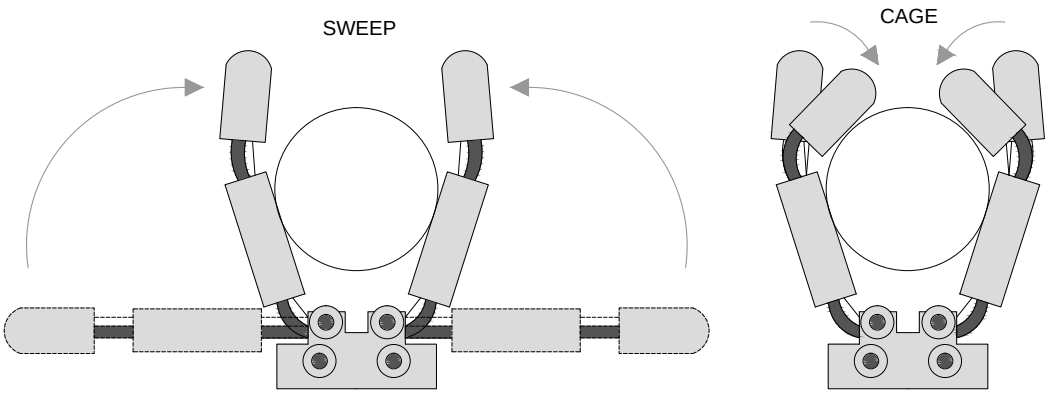


FIGURE 5.3: Sweep and Cage.

in the native Baxter gripper actuator. Our design also features a novel fingertip tendon termination and tension adjustment mechanism. Compliant flexure joints allow a larger range of motion for the fingers during the grasp. The main distinguishing characteristic

FIGURE 5.4: Native Baxter gripper.

of the Baxter Easyhand is that is it driven using the same actuator that is used by the native Baxter gripper. Other 3D printed hands such as the Yale T42 hand are driven by Dynamixel servos. Although the Dynamixel is a great actuator, it is costly, heavy, and requires using a separate ROS driver. In contrast, most Baxter users already own the

FIGURE 5.5: Sliding bars to which the gripper fingers are mounted.

native Baxter gripper. The gripper is fast and can be controlled in a precise way using a ROS node pre-installed on the Baxter. However, each jaw of the Baxter gripper has a maximum of only 2 cm travel (Figure 5.6). If the user wishes to grasp objects more than 4 cm in diameter then it is necessary to unscrew the parallel jaws from the sliders and manually mount them in a wider configuration. One way to view the Easyhand is as a way of replacing the parallel jaws with something that can grasp a wider variety of objects. The Easyhand can grasp any object up to 8 cm in diameter. Moreover, we have empirically observed that the Easyhand produces secure grasps and that it shares many of the compliance, flexibility, and robustness characteristics of the Yale OpenHand series [77] or the Harvard SDM hand [72].

5.2.1 Actuation

The native Baxter gripper is driven by a single motor that actuates two sliders in opposite directions via a wormgear (Figure 5.4 and Figure 5.5). Each slider has approximately 2 cm of translational travel as shown at the bottom of Figure 5.6. The speed and maximum force of the actuator can be controlled by interfacing with a ROS node that runs on the Baxter robot itself. Each finger of the Easyhand is actuated by one tendon that flexes (closes) the finger. Each tendon is connected to one of the sliders via a wire ring terminal crimped onto the tendon and the wire terminal is screwed into the slider (Figure 5.6).

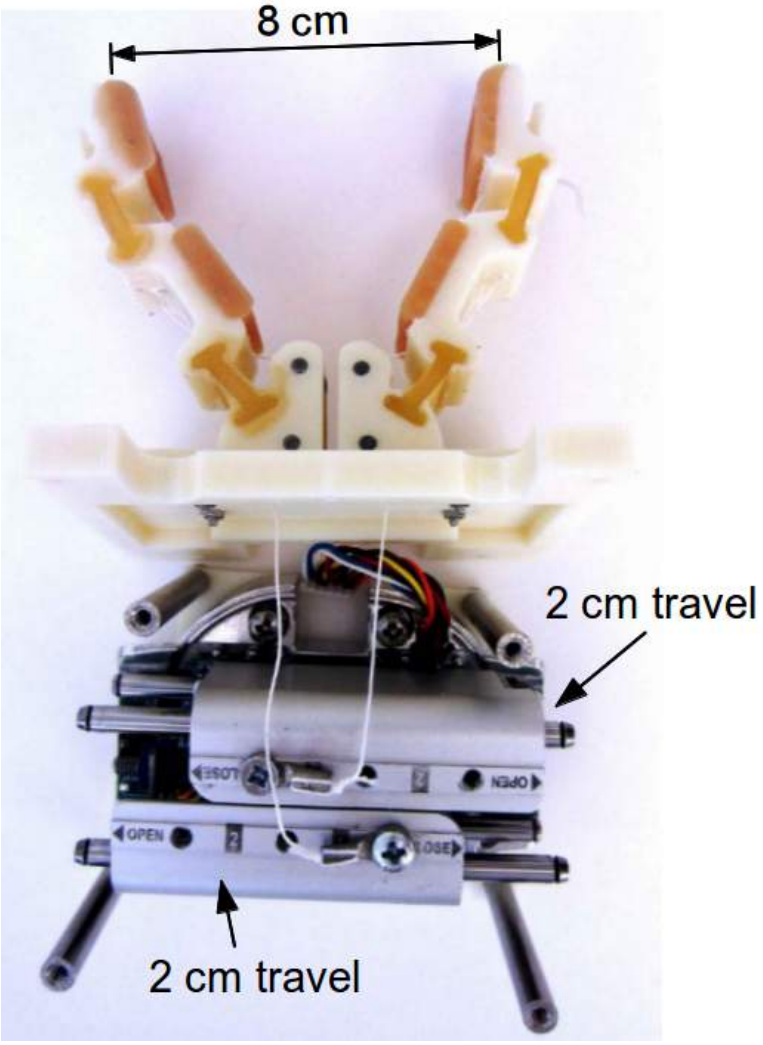


FIGURE 5.6: Tendon attachment to the sliders.

5.2.2 Tendon Arrangement

One disadvantage of using the native Baxter gripper actuator is the limited force that it can apply. Without any modification, we have found that the Baxter gripper can only squeeze an object with 11 Newtons of force (at the maximum gripper force setting in the driver). In the context of our design, this is a potential problem because tendons typically need to apply many times the force that the hand will ultimately be required to apply at the fingertip. Because of this, we have designed the tendon routing in order to maximize the ratio of the force applied by the finger with respect to the tendon tension (see Figure 5.7). In particular, the "first pulley" in Figure 5.7 routes the flexion cable such that the tendon pulls in a direction nearly parallel to the direction of finger closing. As a result, we obtain a nearly one-to-one ratio between finger-force and tendon tension for a contact near the tendon attachment point. As the point of contact moves up the finger, the finger force decreases linearly with distance from the proximal flexure joint.

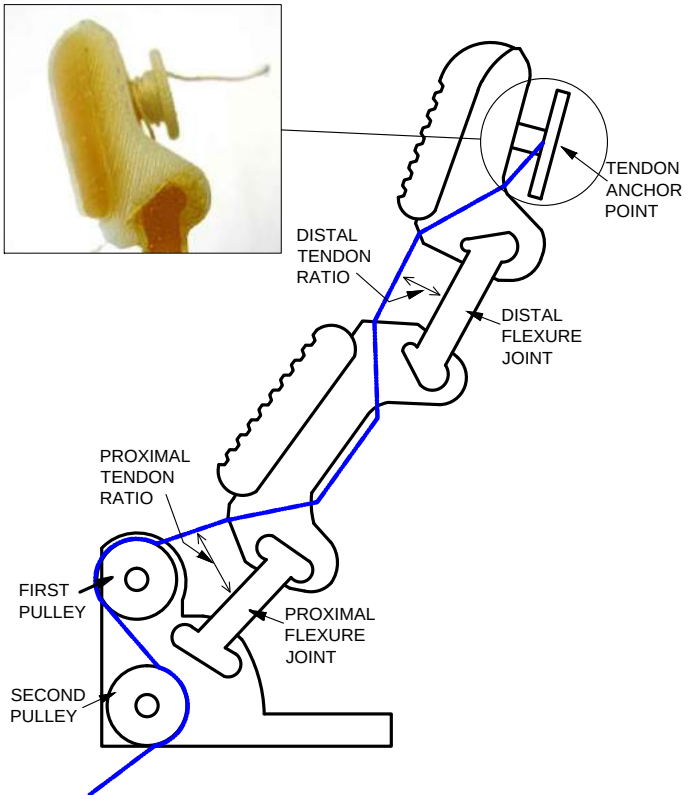


FIGURE 5.7: Finger design details.

We experimentally characterized the squeezing force that our hand can apply using the apparatus shown in Figure 5.8. A Nano25 Force/Torque sensor from ATI Industrial Automation was placed between the fingers and used to measure hand squeezing force. We designed and printed four sets of shells (30 mm, 40 mm, 50 mm, and 60 mm in diameter) that we mounted to the rectangular force sensor plates in order to measure squeezing force when grasping objects of different sizes. For each object, we performed a grasp in three different positions: close to the proximal link (case a), just in the middle of the two links (case b), and close to the distal link (case c).All results are averages over ten trials The results are shown in Figure 5.9. Note that the maximum squeezing force is roughly half the maximum tendon tension force in the actuator: all finger squeezing forces are between 4 Newtons and 7.5 Newtons. As might be expected squeezing forces are largest when grasping close to the palm of the hand (the left side of Figure 5.9). Here, the object makes contact with the finger very close to the tendon attachment point and we therefore expect the highest squeeze force. As the contact moves away from the palm (the center and right side of Figure 5.9), squeezing force drops somewhat. These results illustrate the trade-off between squeeze force and swept volume of the fingers. Since we're using the Baxter gripper actuator, we have limited force and limited actuator travel. The Easyhand trades some of this force for additional finger travel – essentially

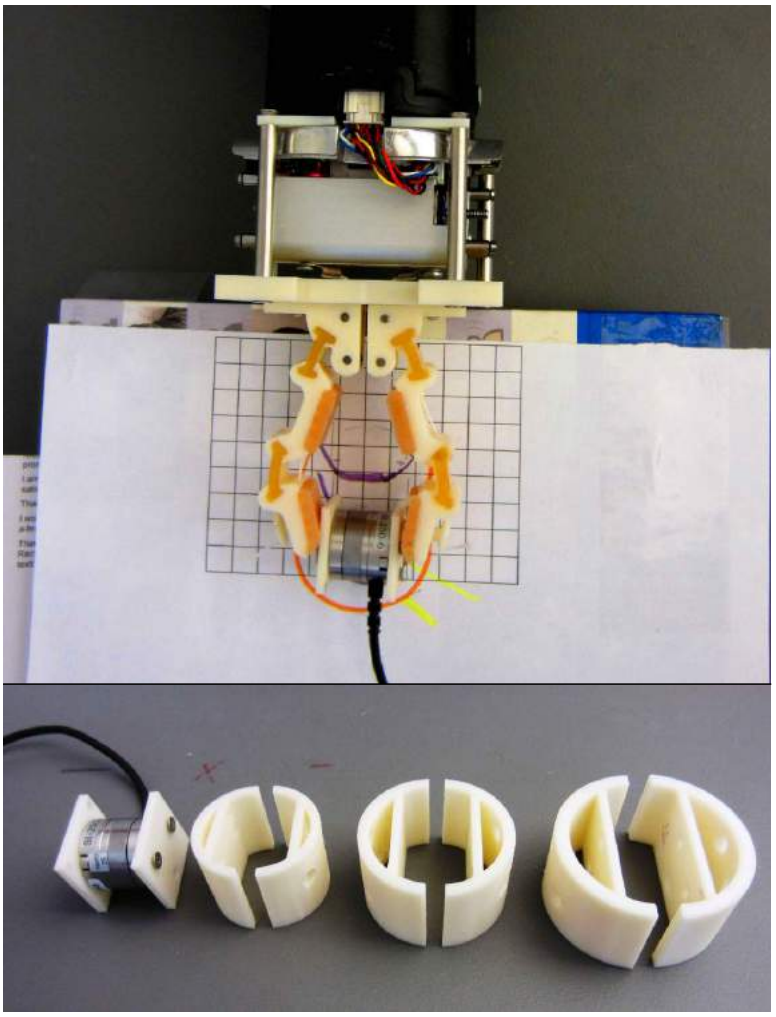


FIGURE 5.8: Apparatus for measuring the Easyhand squeezing force.

doubling the swept volume of the fingers while cutting the squeezing force in half. The squeezing force in the middle and distal positions is approximately the same, even though one would expect squeezing force to be less in the distal position. We hypothesize that this is because these grasps caused the distal joint to flex and move the contact point closer to the palm. Tendon termination and length adjustment is another important part of our design. It is common in robot hands to adjust tendon length at the point where the actuator attaches to the tendon. For example, in both the Robonaut 2 hand and the Yale OpenHand designs, tendon length is changed by adjusting the position at which the tendon is attached to the actuator [77], [79]. In contrast, we adjust tendon length at the tip of the finger, as illustrated in the insert of Figure 5.7. One end of the tendon is crimped to a wire ring terminator (Figure 5.6). In Figure 5.11 the three phases of thetendon mounting process are shown. First, tie tendons to the ring terminators; there are several ways to tie the knot. All that matters is that the tendon attaches securely to the terminator without abrasion. Second, pull the tendons through the two respective tendon paths in the fingers. Third, tension the tendons to the right length

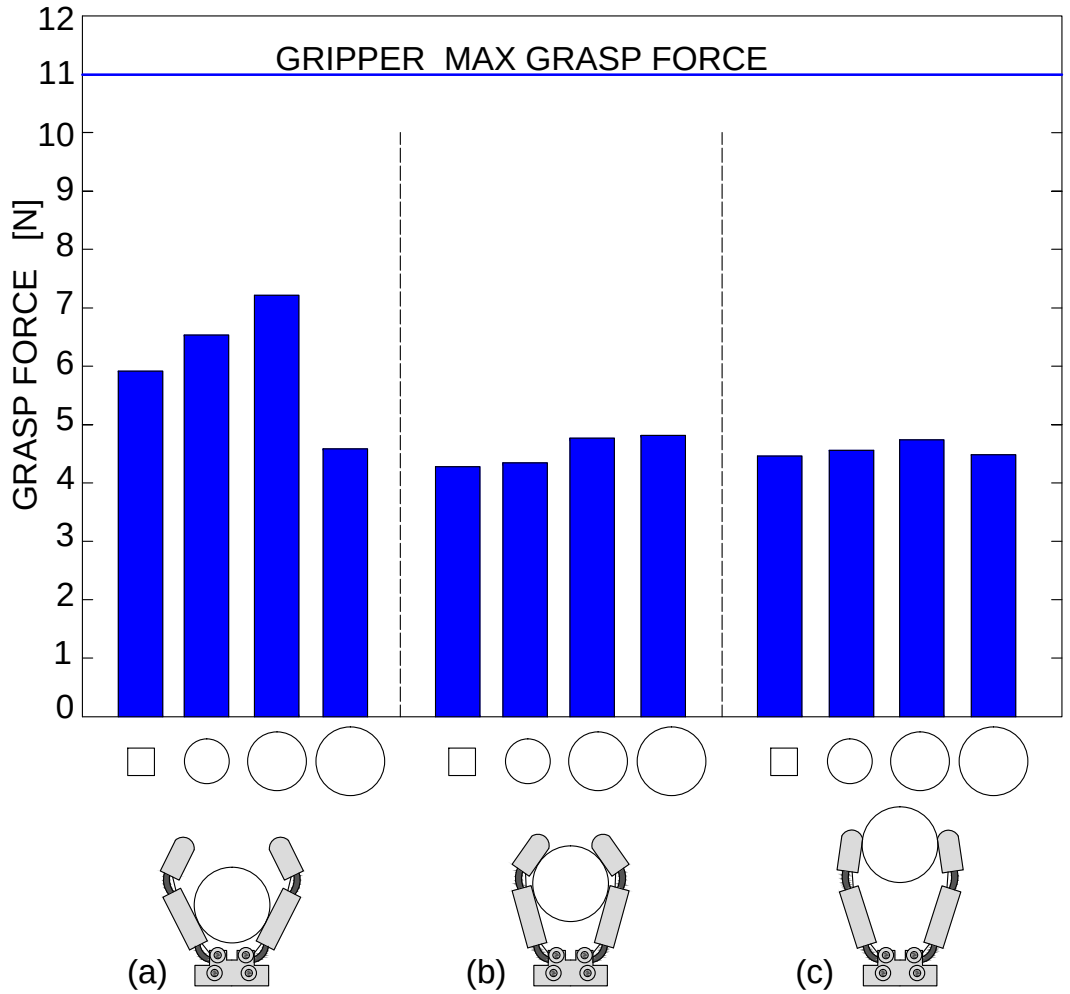


FIGURE 5.9: Squeezing force profile.

and wrap them around the round tie-off (the capstan) on the back of the finger. Make several wraps and then terminate the tendon in the narrow slot. The friction of this slot should be enough to prevent the tendon from slipping. The tendon length should be adjusted so that the EasyHand opens and closes when the Baxter gripper actuator opens and closes, for this reason is better to tension the tendon after calibration. The other end is pulled through the finger and terminated at the tendon anchor point on the back of the fingertip as shown in Figure 5.7. The tendon is wrapped around the anchor point several times and then tied off in a slot in the anchor point. The advantage of adjusting tendon length via a fingertip terminator is that it makes it relatively easy to adjust tendon length. Tendon length adjustment is the last step in finger assembly. It occurs after the fingers are assembled and the hand is mounted to the robot. If the tendon stretches or slips in the terminator, it is simple to unwrap the tendon on the fingertip terminator and adjust length. After length adjustment, it is only necessary to execute a standard Baxter gripper calibration sequence and the hand is ready to be used again.

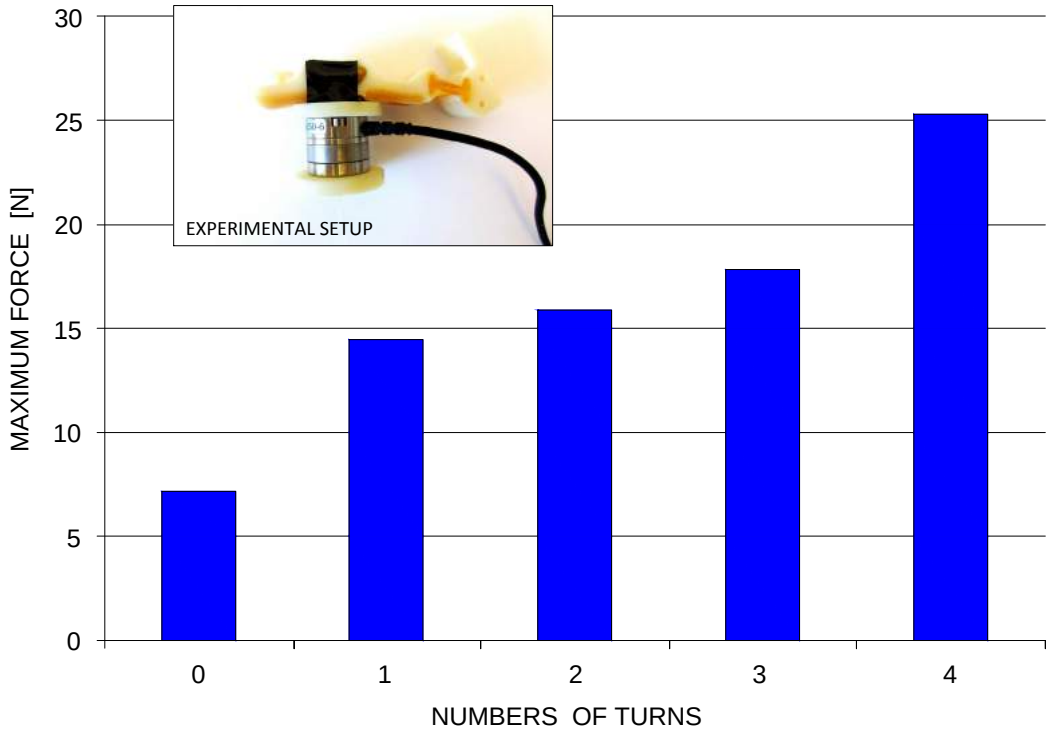


FIGURE 5.10: Tendon force measurement.

The entire process takes less than two minutes. We performed experiments to quantify the load that our terminator could react. We mounted a Nano25 force/torque sensor from ATI Industrial Automation in series with the tendon and measured the amount of force we could apply until the tendon started to slip in the terminator. We repeated this experiment for different numbers of (zero to four) "wraps" of the tendon around the fingertip terminator. The results are shown in Figure 5.10. At four wraps, we were able to exert a maximum of 25 Newtons of force. Since this is well above the 11 Newton maximum force that can be applied by the Baxter gripper actuator, we used a four-wrap termination in all of our subsequent work.

5.2.3 Hand Fabrication

The Baxter Easyhand is simple to build. All that is required is access to a 3D printer and the ability to order approximately $150 worth of hardware from standard vendors. Figure 5.12 illustrates all of the required parts. The fingers are similar to those in the Yale T42 hand. Essentially, the Easyhand fingers can be viewed as miniaturized versions of the T42 fingers with a slightly different tendon routing. As in the T42 hand, the fingers are manufactured using shape deposition manufacturing. Figure 5.13 illustrates the main steps of the manufacture process. First, the fingers are printed using a fused

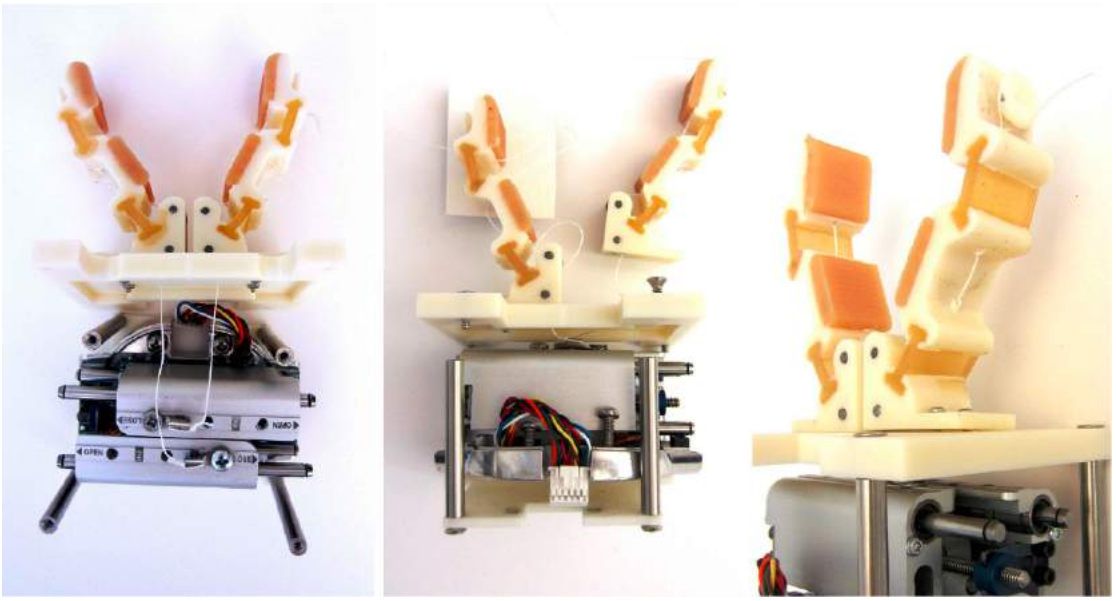


FIGURE 5.11: Tendon mounting phases

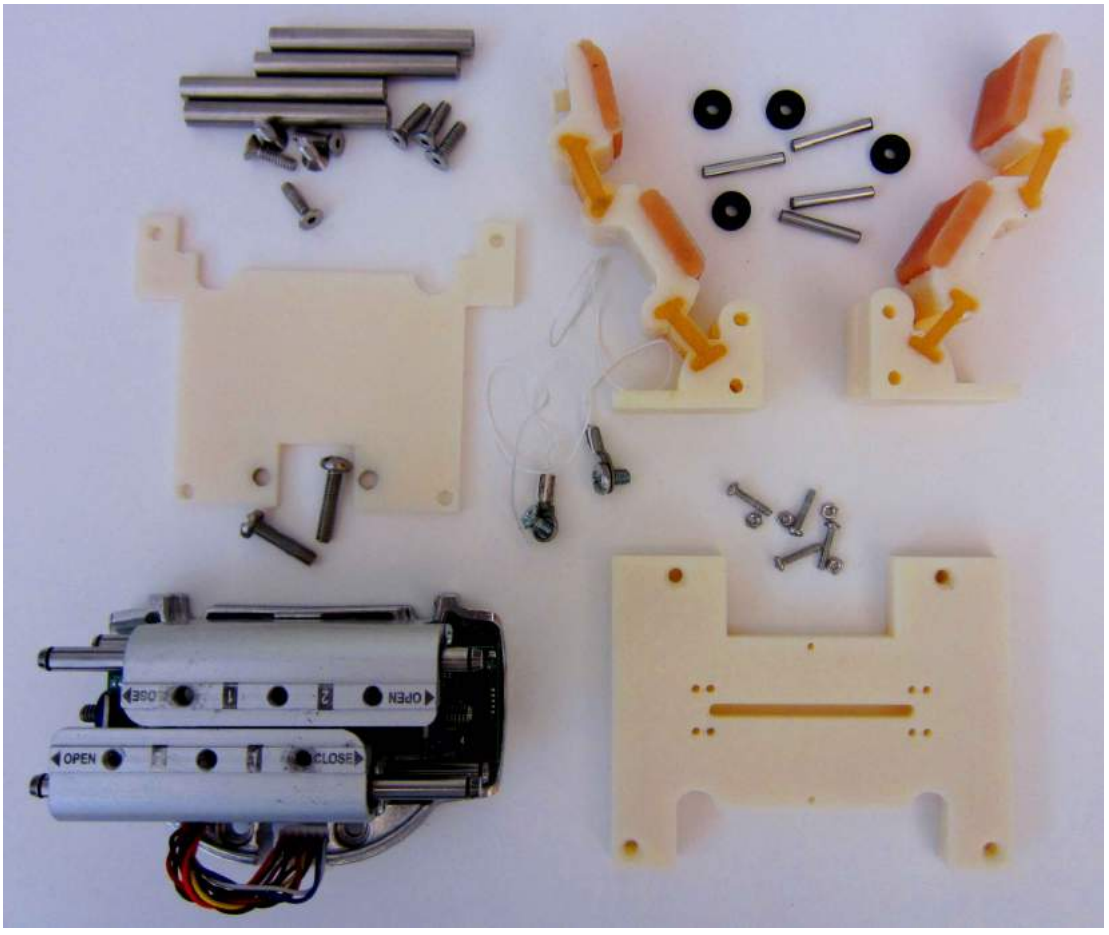


FIGURE 5.12: Baxter Easyhand's parts.

deposition modeling 3D printer (in our case, the U-Print SE Plus [75]). The fingers are printed with cavities that function as molds for Polyurethane material is poured

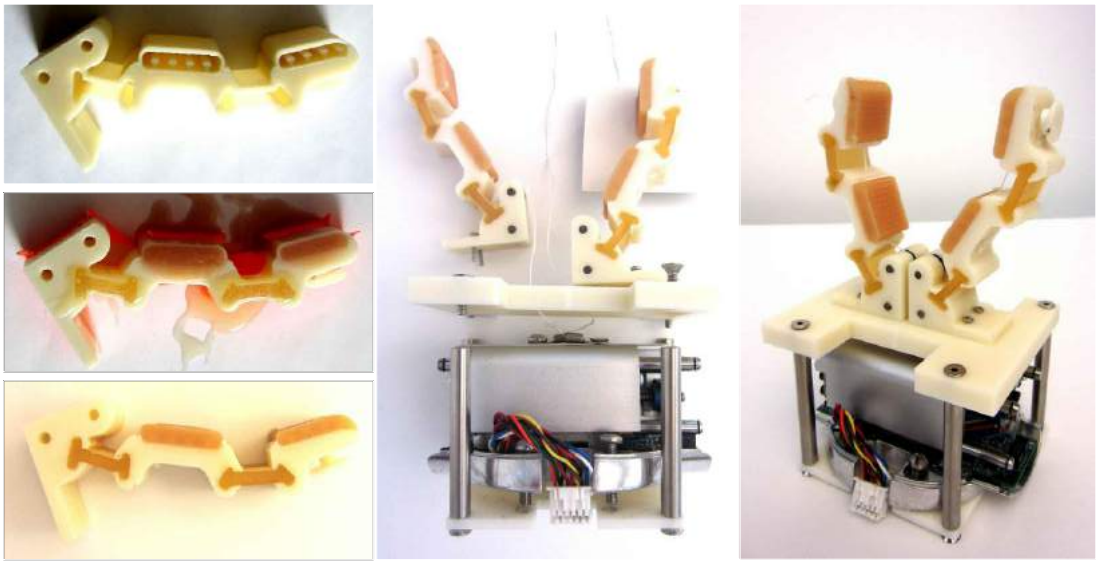


FIGURE 5.13: (a) Components used to create the Baxter Easyhand, (b) main steps of the building process: (left) pouring rubber for finger construction, (center) assembling fingers, tendons and bases, (right) fixing the tendon around the termination.

TABLE 5.1: Easyhand components

Part	**Usage**	**Vendor**
Power Pro Spectra	Tendon	Amazon
PMC-780 Urethane	Finger Joint	Smooth-On
Vytaflex 30 Urethanea	Finger Pad	Smooth-On
4 nylon pulley	Tendon Routing	McMaster-Carr
D3/8, Wd1/8		[3434T31]
4 Steel Dowel Pin	for Pulley	McMaster-Carr
D1/8, L13/16		[98381A173]
4 Round Head Screw	for base-finger	McMaster-Carr
2-56 X 7/16”		[91772A080]
4 Hex Nut 2-56a	for base-finger	McMaster-Carr
		[91841A003]
4 Female Standoff	for bases	McMaster-Carr
OD1/4”, L2”, 6-32		[91125A250]
4 Flat Head Machine Screw	for bases	McMaster-Carr
6-32 X 1/2”		[91500A148]

once printing is complete (0.7 mm thick shell). Two types of Polyurethane are used: a flexible type of material for the joint flexures and a gripping type for the finger pads. Once the Polyurethane is dry (almost 24 hours), the cavity shells are removed and finger assembly is completed by installing a pair of pins and pulleys into each finger. Then, tendons are routed and the fingers are mounted onto the palm plate. The palm plate is mounted to the arm via four screws that attach to four quarter-inch standoffs that mount onto another plate mounted below the Baxter gripper actuator.While working with the

Easyhand on the Baxter robot requires the same set of skills and knowledge as using the Baxter gripper, grasping is significantly improved by adding flexible fingers and by increasing the aperture range to twice the original range. In the baxter easyhand, each of the two fingers has two phalanges and is driven by a single tendon. The tendon acts to flex both phalanges simultaneously. The flexure in each joint is elastic and passively acts to resist the flexing motion and to supply extension force in the absence of flexure.

5.3 Grasping performance

The plot in Fig 5.14 show the region of configurations where the object would be captured by the hand. This was determined experimentally: fixing the hand in a certain spot just above the table and placing manually an object near the hand in different positions, we close the hand around the object and evaluate the grasp. More tests were repeated over a grid of object positions relative to the hand and the hand's capture region for the given object was plotted. We evaluated how well the Easyhand can grasp objects

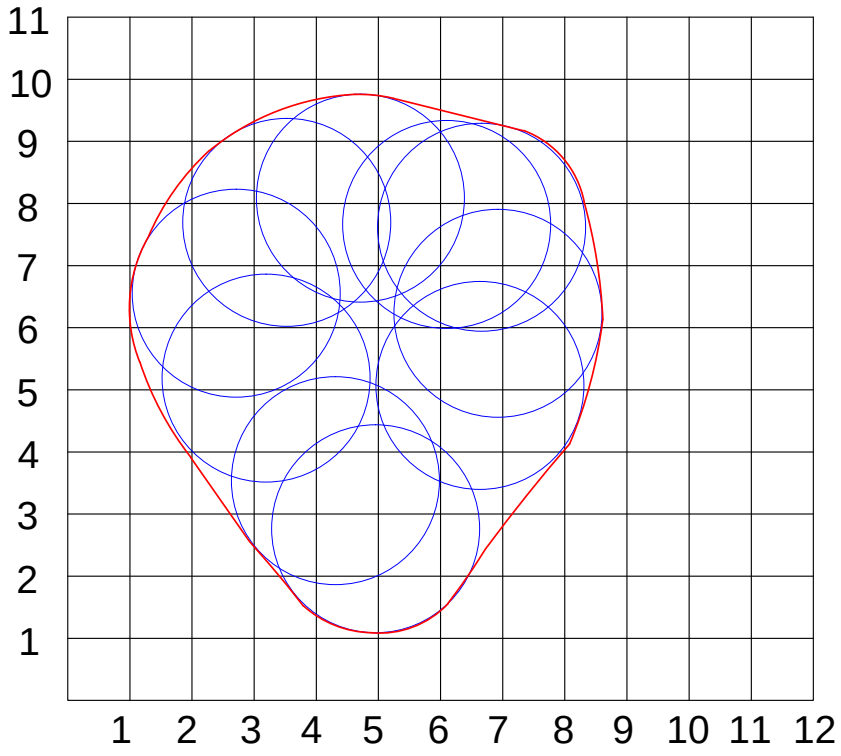


FIGURE 5.14: Esayhand Capture Region.

when under teleoperated control and autonomous control.

FIGURE 5.15: Set of different graspable objects.

FIGURE 5.16: Objects used in the teleoperation grasping.

FIGURE 5.17: Objects used in autonomous grasping.

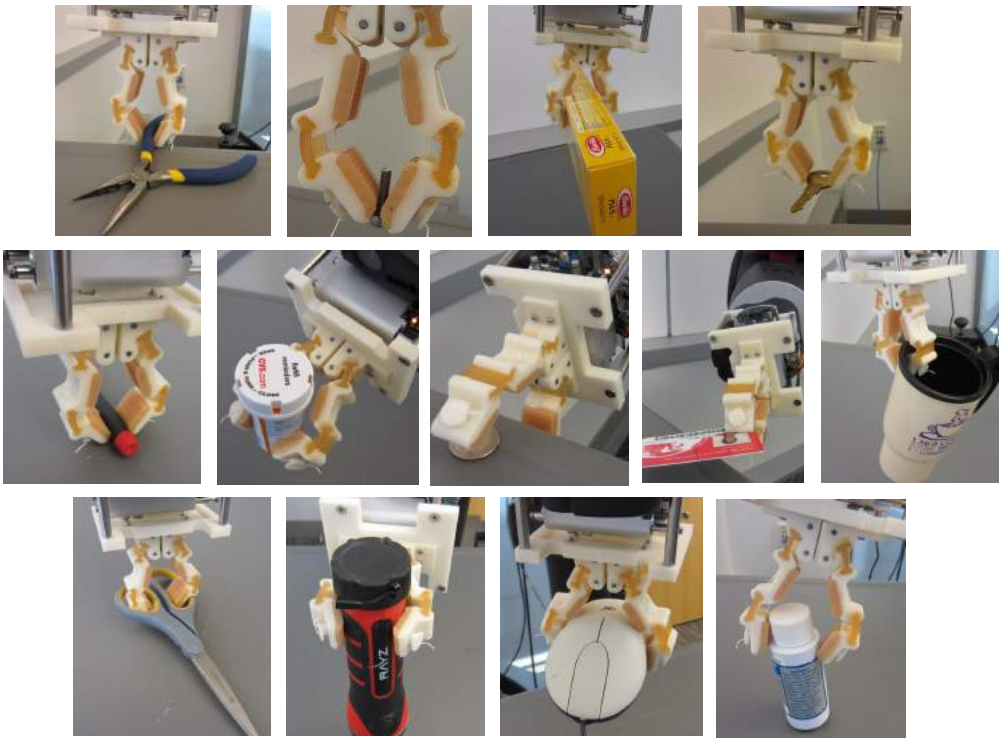


FIGURE 5.18: Teleoperated grasps.

5.3.1 Grasp performance under teleoperation

In teleoperated grasping, the joints of the robot arm and hand are controlled directly by a human. Since autonomous robot grasping still cannot match the capabilities of a human operator, teleoperation can give us a sense for the best-case performance of the hand under the control of an "ideal state of art" controller. In our experiment (see Figure 5.19 for the experiments setup), Baxter was placed a master-slave mode where one arm functioned as the master and the other as the slave (we used the "Puppet" example program that ships with Baxter). A human controller moved the master arm kinesthetically and this motion was reflected on a joint-by-joint basis to the slave arm. In addition, the human was able to open and close the hand by pressing a button on the side of the master arm. We were able to grasp each of the objects shown in Figure 5.16 under teleoperation. Figure 5.18 shows the teleoperated grasps that we obtained. These grasps indicate that the Easyhand is capable of graping a variety of objects including keys and small screws as well as larger objects such as the flashlight or the computer mouse. The most difficult objects for the Easyhand to grasp were thin objects such as the student ID, the coin, or the key. The teleoperator was only able to grasp these objects by first sliding them to the edge of the table (see Figure 5.18, for example), and then performing the grasp. This is a deficiency relative to other 3D printed hands (for

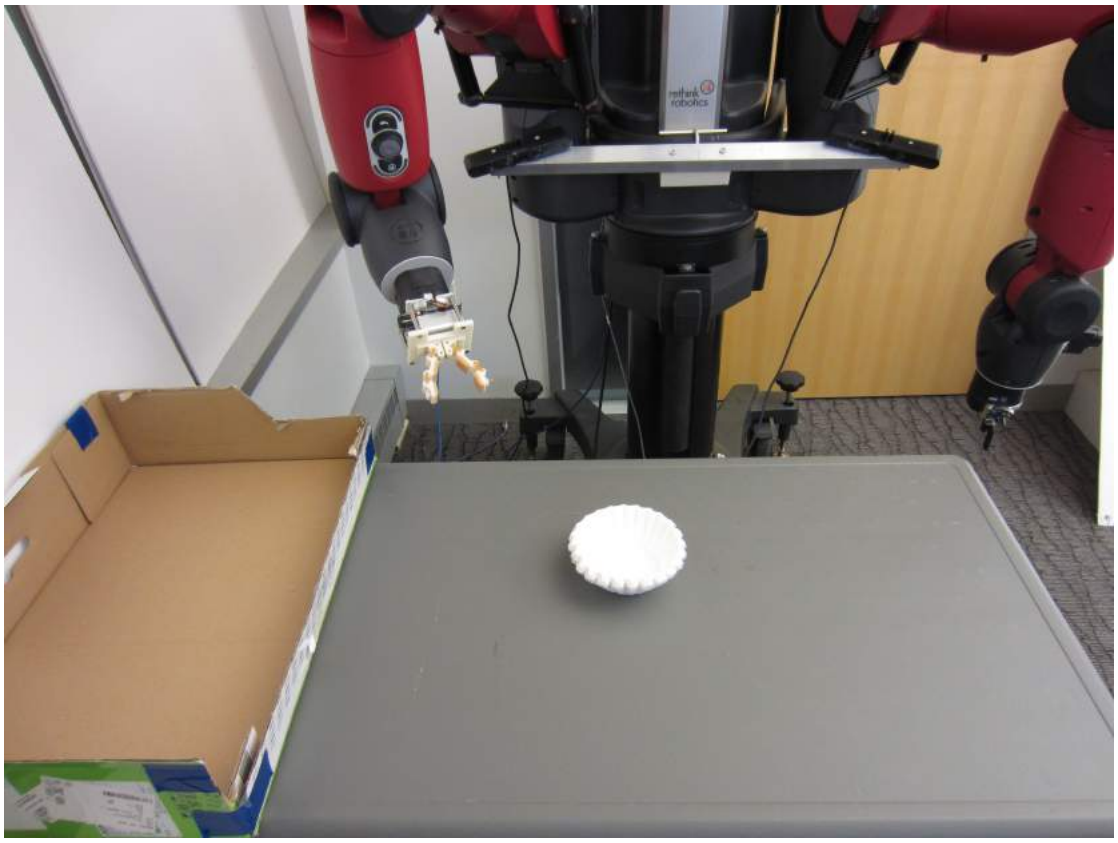


FIGURE 5.19: Experiments setup.

example, the Harvard-iRobot hand [74]) which are equipped with "fingernails" that can "scoop up" small objects.

5.3.2 Grasp performance under autonomous control

The teleoperated grasping results described above characterize the potential of the Easyhand given a very intelligent control system (*i.e.* a human). However, we are also interested in understanding what kind of objects the Easyhand can be expected to grasp under autonomous control, given the algorithms currently available. In order to accomplish this, we used a grasp localization system developed in conjunction with our recent antipodal grasp prediction work [76]. Essentially, our system uses two RGBD cameras to create a point cloud that characterizes the scene in front of the robot. Our algorithm searches the point cloud for hand configurations where antipodal grasps are predicted to exist. Once a set of potential grasp configurations is obtained, the algorithm selects one based on manipulator kinematics, obstacle configuration, *etc.*. Here, we performed single-object tests where a single object was placed in front of the robot and our system performed a grasp. Our test set consisted of the ten objects shown in Figure 5.17 (note the absence of hard-to-grasp items such as the key in Figure 5.16). We attempted to

TABLE 5.2: Autonomous grasping results.

Object	Success Rate
Lint Roller	100%
Britannia Box	100%
Yarn Roll	83.33%
Computer Mouse	83.33%
Green Tape Roll	83.33%
Red Tape Roll	66.66%
Coffee Filters	83.33%
Vacuum Cleaner Part	66.66%
Coffee Bag	83.33%
Pepper Dispenser	100%
Average	85%

grasp each object in six different poses. A trial was considered a success only if the robot successfully localized, grasped, lifted, and transported the object to a container on the side of the robot where the object was dropped. During our experiments, the perceptual system sometimes found spurious grasp targets in the air or elsewhere caused by occlusions or noise in the point cloud. We eliminated this effect from our results by reporting on only those grasp trials when the robot hand actually made contact with the object. The results of this experiment are shown in Table 5.2. The robot was able to successfully grasp, transport, and drop into a box objects approximately 85% of the time (8 failures out of 60 attempts). Of the 8 failures, one was caused by dropping the object after an initially successful grasp, and another one was caused by one of the fingers colliding with the target while attempting the grasp and thus pushing the object out of range of the hand. The other 6 failures were due to perception: missing data in the RGBD images caused our algorithm to select non-existent grasps or to reach to configurations where the Easyhand did not fully envelop the object. These results are particularly interesting when they are compared with the results of using the same grasping system with the native Baxter parallel jaw gripper as reported in [76]. There, we reported a 96% grasp success rate for a collection of 31 objects. Perhaps the main reason why the 85% success rate reported here is lower than the 96% success rate reported for the parallel jaw gripper is because the object set in this paper (Figure 5.17) has a wider variety of objects (including thin objects and deformable objects) relative to the 31-object test set used in [76]. The greater versatility of the Easyhand relative to the Baxter gripper enabled us to grasp these additional objects but our grasp success rate suffered somewhat. Another reason for the lower success rate is that we may not have spend as much time tuning various system parameters in the Easyhand scenario relative to the gripper scenario (this could account for a couple of percentage points). However, we also observed a couple of failure modes that are important to point out. First, we observed a rolling failure mode that occurred when attempting to grasp cylindrical objects presented horizontally. One finger would make contact with the surface of the object prior to the other. In some situations, the way that contacting finger complied with the object actually caused the object to roll out of the grasp (Figure 5.20 (left)). This failure did not occur with the

FIGURE 5.20: Failure modes

parallel jaw gripper because the gripper did not comply with the object in the same way. The object rolls away when one finger pad is making contact with object surface. Another failure mode occurs because the Easyhand grasps small-diameter objects with the fingertips (Figure 5.20 (right)), in this case we have an unstable pinch grasp. This is in contrast to the grasps generated by a parallel jaw gripper where the fingers are always parallel. Since Easyhand grasps small-diameter objects with the fingertips, these grasps are less stable than those generated by a gripper. While we first performed the experiment in the same way in which we use the Baxter gripper, i.e., move the arm to a grasp pose, close the gripper and lift the target object, we noticed that Easyhand performs significantly better if we use a different approach. With Easyhand once a grasp pose is reached, we close the hand, but then we move the hand up by a small amount, and close it again. This avoids slipping of some objects after an initially attempted grasp and leads to more stable grasping behavior.

5.4 Conclusions

In this paper we describe the Baxter Easyhand, a new 3D printed robot hand specifically designed for use with the Baxter robot. This is an underactuated, two finger, flexible joints, low cost robotic hand made through 3D printing and SDM that is derived from the Yale T42 hand [77]. Our goal is to provide those who use the Baxter robot with an inexpensive hand that is easy to build that is more robust and flexible than the native Baxter gripper (see Figure 5.21). We demonstrate via teleoperated and autonomous grasping experiments that the Easyhand is an effective hand. Videos and details of assembly instructions are available at:
http://www.ccs.neu.edu/research/helpinghands/easyhand/easyhand_assy_instructions.html

FIGURE 5.21: The Baxter robot with the native gripper and the Easyhand.

Chapter 6

Epilogue

In this conclusion and future work chapter, a brief review about the thesis aim and the obtained results is presented. Proposals for future researches are also discussed.

6.1 Conclusions

The construction of the mechanism and the robotic hand have the goal to produce a new rehabilitation device for optimal grasping. From the analysis of industrial robot grasping behavior, a general rule for performing a stable, human-like grasp with a robotic hand have been find. A different method of doing grasping with industrial robot, that help to find optimal grasps configuration with big robotic arms; have been provide. Finally a new mechanism to connect the human leg with a robotic arm for rehabilitation have been build and a new 3d printed robotic hand easy and cheap to build have been release. Studying grasping in industrial robots, a hybrid system model of an underactuated robotic hand have been developed, in the form of dynamic equations and analytical solutions. The model is verified with a vision-based data collection system to match the experimental data well within measurement error. The model is implemented in a demo program in Klamp't, which is freely available from http://klampt.org. A technique for building miniature robot control devices using 3D printing and low-cost electronics is presented. It consists of a standardized encoder module, 3D printing procedure, and assembly process that is generalizable between a wide variety of robots. As main goal of this dissertation, a robot-human interface was developed, which allowed a firm mechanical connection between a multipurpose robotic arm and a human limb, yet also permitted freedom of movement for the human relative to the robot. An advantage of adapting a multipurpose robotic arm for gait rehabilitation is versatility: a robotic arm can be attached to almost any point on the body to assist with lower-and

upper extremity rehabilitation. This may be more cost-effective than purchasing and training rehabilitation staff to use several specialized rehabilitation robots. Robotic arms also have a more human-like morphology, which may make them less intimidating or alien to patients. Several test with healthy subjects walk on a treadmill show the feasibility of adapting a multipurpose robotic arm for gait rehabilitation. Since the human hand is an example of high dexterity system above all others, after studying the grasping problem with robots and behind the construction of the robotic rehabilitation device, it is clear that we need a new hand able to replicate the human hand ability in the rehabilitation process. In this Thesis the Baxter Easyhand is presented. We demonstrate via teleoperated and autonomous grasping experiments that the Easyhand is an effective hand. Videos and details of assembly instructions are available at: http://www.ccs.neu.edu/research/helpinghands/easyhand/easyhand_assy_instructions.html

6.2 Futureworks

In future work we wish to perform a more precise empirical validation and to implement the model for grasp optimization and object recognition without additional sensors. We also intend to build a more complete model of the force characteristics of the gripper in contact with an object. We also wish to perform user studies to test the hypothesis that ROBOPuppet makes robots easier to control than alternative input modalities, such as joysticks, mice, and off-the-shelf haptic devices. It is our intention to foster a community of researchers and citizen scientists leveraging current rapid prototyping communities who can share, improve and use the ROBOPuppet pocket geometries, joint encoder assemblies and general methods. As a ROBOPuppet controller can be used to move robots in simulation, these controllers are an excellent way to work with robots that are not immediately accessible. Robots that need to support themselves or balance, particularly mobile robots or robots that are not affixed to a supporting platform, are not suitable candidates for the ROBOPuppet method without additional considerations in place. These robots can be modified during assembly/printing to be mounted on a supporting bracket that is affixed to a platform or only portions of the robot can be modeled (for example, a humanoid robot may benefit from having only the torso, arms and head modeled and assembled in the ROBOPuppet controller). In respect to the rehabilitation mechanism, although only the mechanism was used to attachment the robotic arm with the leg, it would be straightforward to integrate the Easyhand with the attachment system for optimal grasping. Taking advantage from the lightness, the construction and control ease and the dexterity of the Easyhand, it is possible to develop a human-like rehabilitation device to help patients to re-learn movements.

Publications

- Conference Proceedings
  - G. Franchi, A. ten Pas, R. Platt and S. Panzieri. The Baxter Easyhand: A Robot Hand That Costs 150 UDS in Parts. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2015), Sept.28 - Oct.2, 2015, Hamburg, Germany.
  - G. Franchi, U. Viereck, R. Platt, S. Yen and C. J. Hasson. An Arm for a Leg: Adapting a Robotic Arm for Gait Rehabilitation. 37th Annual International Conference of theIEEE Engineering in Medicine and Biology Society, Aug.25-29, 2015 - Milan, Italy.
  - G. Franchi, K. Hauser, S. Panzieri. A Hybrid System Model of the RobotiQ Adaptive Gripper. Grace Hopper Conference, Celebrationg of Woman in Computer, Oct.14-16, 2014 - Phoenix, AZ, USA.
  - A. Eilering, G. Franchi and K. Hauser. ROBOPuppet: Low-Cost, 3D Printed Miniatures for Teleoperating Full-Size Robots. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Sept. 14-18, 2014, Chicago, IL, USA.
- Technical Reports
  - G. Franchi, K. Hauser. Use of Hybrid Systems to model the RobotiQ Adaptive Gripper (School of Informatics and Computing, Indiana University, Bloomington).

Bibliography

[1] M. Hillman, *Rehabilitation robotics from past to present: a historical perspective*, Advances in Rehabilitation Robotics, vol. 306, 2004.

[2] H. van der Loos and D. Reinkensmeyer,*Rehabilitation and health care robotics*, Springer Handbook of Robotics, 2008.

[3] D. Erol, V. Mallapragada, N. Sarkar, and E. Taub, *A new control approach to robot assisted rehabilitation*, 9th International Conference on Rehabilitation Robotics, 2005.

[4] S. Subramanian, L. A. Knaut, C. Beaudoin, J. B. McFadyen, A. G. Fedman, and M. F. Levin, *Virtual reality environments for post-stroke arm rehabilitation*, Journal of Neuroengineering and Rehabilitation, vol. 4, no. 20, 2007.

[5] P. Lum, D. Reinkensmeyer, R. Mahoney, W. Z. Rymer, and C. Burgar, *Robotic devices for movement therapy after stroke*, Topics in Stroke Rehabilitation, vol. 8, no. 4, pp. 40-53, 2002.

[6] L. Zollo, M. Passalacqua, D. Formica, and E. Guglielmelli, *Performance analysis of adaptive interaction control laws for rehabilitation robotics*, Proceedings of the IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2008.

[7] M. W. O'Dell, C. Lin, and V. Harrison, *Stroke rehabilitation strategies to enhance motor recovery*, Annual Review of Medicine, vol. 60, 2009.

[8] R. Teasell, N.A.Bayona, and J. Bitensky, *Plasticity and reorganization of the brain poststroke*, Topics in Stroke Rehabilitation, vol. 12, no. 3, pp. 11-26, 2005.

[9] E. Guglielmelli, M. Johnson, and T. Shibata, *Guest editorial special issue on rehabilitation robotics*, IEEE Transactions on Robotics, vol. 25, 2009.

[10] B. Massa, S. Roccella, M. C. Carrozza, and P. Dario, *Design and development of an underactuated prosthetic hand*, Proceedings of the IEEE International Conference on Robotics and Automation, vol. 4, 2002.

[11] D. Gunji, Y. Mizoguchi, S. Teshigawara, M. Aiguo, A. Namiki, M. Ishikawaand, and M. Shimojo, *Grasping force control of multifingered robot hand based on slip detection using tactile sensor*, IEEE International Conference on Robotics and Automation, 2008.

[12] L. Zollo, S. Roccella, E. Guglielmelli, M. C. Carrozza, and P. Dario, *Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications*, IEEE/ASME Transactions on Mechatronics, vol. 12, no. 4, pp. 418-429, 2007.

[13] M. Carrozza, G. Cappiello, S. Micera, B. B. Edin, L. Beccai, and C. Cipriani, *Design of a cybernetic hand for perception and action*, Biological Cybernetics, vol. 95, no. 6, pp. 629-644, 2006.

[14]] C. L. Taylor and R. J. Schwarz, *The anatomy and mechanics of the human hand*, Artificial Limbs, vol. 2, no. 2, pp. 22-35, 1955.

[15] C.M. Gosselin and T. Lalibertè, *Underactuated Mechanical Finger with Return Actuation*, U.S. Patent 5 762 390, 1998.

[16] C. Melchiorri and M. Kaneko, *Robotic Hand*, pp. 345-360 (B.Siciliano and O. Khatib, Eds, Handbook of Robotics. Berlin: Spinger, 2008).

[17] M.G. Catalano, G. Grioli, A. Serio, E. Farnioli, C. Piazza and A. Bicchi,*Adaptive Synergies for a Humanoid Robot Hand*, IEEE-RAS International Conference on Humanoid Robots, 2012.

[18] L. Birglen, T. Lalibertè and C. Gosselin, *Underactuated robotic hands*, pp. 345-360 (B.Siciliano and O. Khatib, F. Groen, Eds, Springer Tracts in Advanced Robotics. Berlin: Spinger, 2008).

[19] W. Townsend, *The BarrettHand grasper - programmably flexible part handling and assembly*, vol. 27 (3), pp. 181-188 (Ind. Rob., 2000).]

[20] T. Lalibertè, L. Birglen and C.M. Gosselin,*Underactuation in robotic grasping hands*, Machine Intelligence and Robotic Control, Vol. 4, No. 3, 1-11 (2002).

[21] S. Engell, G. Frehse and E. Schnieder (Eds.),*Modeling, Analysis and Design of hybrid Systems*, pp. 436-465 (T. Schlegl, M. Buss and G. Schmidt, Hybrid Control of Multi-fingered Dextrous Robotic Hands), 2002.

[22] T. Schlegl, M. Buss and G. Schmidt, *A Hybrid Systems Approach Toward Modeling and Dynamical Simulation of Dextrous Manipulation*, IEEE/ASME Transaction on Mechatronics, 2003.

[23] Y. Yin, M. Svinin and S. Hosoe,*Modeling and control of multifingered robot hand for dexterous manipulation: a continuous and discrete hybrid approach*, Systems, Man and Cybernetics, IEEE International Conference, 2004.

[24] T. Schlegl and M. Buss, *Dextrous hand regrasping using Hybrid System Models*, Advanced intelligent Mechatronics, Tokyo, 1997.

[25] S.L. Ricker, N. Sarkar and K. Rudie, *A discrete-event systems approach to modeling dextrous manipulation*,vol 14, pp.515-525, Robotica, 1996.

[26] http://robotiq.com/en/products/industrial-robot-hand, (Technical sheets), (accessed Sep.10, 2013)

[27] http://www.indiana.edu/~motion/software.html, (accessed Jan.28, 2014)

[28] http://www.pointclouds.org/documentation/tutorials/basic_structures.phpbasic-structures, (accessed Oct.5, 2013)

[29] http://pointclouds.org/documentation/tutorials/statistical_outlier.php, (accessed Oct.5, 2013)

[30] J. Bruce, T. Balch and M. Veloso*Fast and inexpensive color image segmentation for interactive robots*, IEEE/RSJ International Conference on intelligent Robots and Systems, 2000 (IROS 2000).

[31] http://wiki.ros.org/cmvision_outlier.php, (accessed Oct.5, 2013)

[32] Arduino: an open-source electronic platform for protyping. http://www.arduino.cc/ (accessed Feb. 6, 2014).

[33] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. *Springer Handbook of Robotics*, chapter Robot Programming by Demonstration Springer-Verlag, 2008.

[34] B. Dariush, M. Gieger, A. Arumbakkam, Y. Zhu, B. Jian, K. Fujimura, and C. Goerick. Online transfer of human motion to humanoids. *International Journal of Humanoid Robotics*, 6(2):265–289, 2009.

[35] I. Ebert-Uphoff, C. M. Gosselin, D. W. Rosen, and T. Laliberte. *Cutting Edge Robotics*, chapter Rapid Prototyping for Robotics Pro Literatur Verlag, Germany, 2005.

[36] T. Fong and C. Thorpe. Vehicle teleoperation interfaces. *Autonomous Robots*, 11(1):9–18, 2001.

[37] Freecad:an open source parametric 3d cad modeler. http://www.freecadweb.org/ (accessed Feb. 6, 2014).

[38] Geomagic haptic devices. http://geomagic.com/en/products-landing-pages/haptic (accessed Feb. 6, 2014).

[39] Thingiverse. http://www.thingiverse.com/ (accessed Feb. 6, 2014).

[40] C. Guo. *New Paradigms for Human-Robot Interaction Using Tangible User Interfaces.* PhD thesis, University of Calgary, 2008.

[41] K. Hauser and V. Ng-Thow-Hing. Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts. In *Intl. Conf. Rob. Automation*, 2010.

[42] Klamp't. http://klampt.org/ (accessed Feb. 6, 2014).

[43] T. Kroger and F. M. Wahl. Online trajectory generation: basic concepts for instantaneous reactions to unforeseen events. *IEEE T. Robotics*, 26(1):94–111, 2010.

[44] S. Lee, G. Sukhatme, J. Kim, and C.-M. Park. Haptic control of a mobile robot: a user study. In *IEEE / RSJ International Conference on Intelligent Robots and Systems(IROS)*, Lausanne, Switzerland, 2002.

[45] A. Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 31(6):499–508, 2004.

[46] A. Payne. A five-axis robotic motion controller for designers. In *ACADIA Conference: Integration Through Computation*, Banff, Canada, October 13 - 16 2011.

[47] G. Reshko, M. Mason, and I. Nourbakhsh. Rapid prototyping of small robots. *Tech. Report CMU-RI-TR-02-11, Robotics Institute, Carnegie Mellon University*, 2002.

[48] Tinkercad. https://tinkercad.com/ (accessed Feb. 6, 2014). Member of Autodesk, 123D family of products.

[49] B. Yamauchi. Packbot: a versatile platform for military robotics. *Unmanned Ground Vehicle Technology VI*, 2004.

[50] G. Colombo, et al., *Treadmill training of paraplegic patients using a robotic orthosis.* J. Rehabil. Res. Dev, vol 37, pp. 693-700, 2000.

[51] G. Xu, A. Song, and H. Li, *Control system design for an upper-limb rehabilitation robot.* Adv. Robotics, vol 25, pp. 229-251, 2011.

[52] J. Patton, et al., *Robotics and virtual reality: the development of a life-sized 3-D system for the rehabilitation of motor function.* Conference Proceedings Engineering in Medicine and Biology Society, 2004. 7: p. 4840-4843.

[53] A. Song, et al., *Impedance Identification and Adaptive Control of Rehabilitation Robot for Upper-Limb Passive Training*, in Foundations and Applications of Intelligent Systems 2014, Springer. p.691-710.

[54] J. Veneman, et al. *Design and Evaluation of the LOPES Exoskeleton Robot for Interactive Gait Rehabilitation.* IEEE Trans. Neural Syst. Rehabil. Eng., vol. 15: pp. 379-386.

[55] S.K. Banala, et al., *Robot assisted gait training with active leg exoskeleton (ALEX).* IEEE Trans. Neural Syst. Rehabil. Eng., vol. 17, pp. 2-8, 2009.

[56] M. Takegaki and S. Arimoto, *A new feedback method for dynamic control of manipulators*, J. Dyn. Sys, Meas. Control, vol. 4, pp. 119-125, 1981.

[57] L. Birglen, C. Gosselin, T. Laliberte, *Underactuated Robotic Hands*, Springer Tract in Advanced Robotics, 2008.

[58] Kuka YouBot (2013) Available at: http://www.youbot-store.com/accessories/youbot-extensions/soft-two-finger-gripper (accessed 15 February 2015)

[59] N. Ulrich, R. Paul and R. Bajcsy, *A medium-complexity compliant end effector*, in Proc. IEEE Int. Conf. on Robotics and Automation, 1988, pp. 434-436.

[60] Robotiq, *Two-Finger Adaptive Robot Gripper*, 2011. [Online]. Available: http://robotiq.com/products/industrial-robot-gripper/ (accessed 15 February 2015)

[61] Ottobock, *Electric Greifer*, 2011. [Online]. Available: http://www.ottobock.com/cps/rde/xchg/ob_com_en/hs.xsl/3359.html (accessed 15 February 2015)

[62] W. Townsend, *The BarrettHand grasper - programmably flexible part handling and assembly*, Industrial Robot: An International Journal, 27 (3), 2000, pp. 181-88.

[63] Rubinger, B. et al., *Self-Adapting Robotic Auxiliary Hand (SARAH) for SPDM Operations on the International Space Station*, Proc. I-SAIRAS, Quebec, Canada, 2001.

[64] Meka, *H2 Compliant Hand*, 2011. [Online]. http://mekabot.com/products/compliant-hand/

[65] Meka, *G2 Compliant Gripper*, 2011. [Online]. Available: http://mekabot.com/products/compliant-gripper/

[66] Schunk, *SCHUNK PGN-plus Gripper*, [Online]. http://www.schunk-produkte.com/en/pneumatik/2-finger-parallelgreifer-pgn-plus.html (accessed 15 February 2015)

[67] A.M. Dollar and R.D. Howe, *A robust compliant grasper via shape deposition manufacturing*, ASME/IEEE Transactions on Mechatronics, 11(2): 154-161, 2006.

[68] R. Merz, F.B. Prinz, K. Ramaswami, M. Terk and L. Weiss, *Shape deposition manufacturing*, Proceedings of the Solid Freeform Fabrication Symposium, University of Texas at Austin, 8-10 August 1994.

[69] J. Won, K. DeLaurentis, C. Mavroidis, *Rapid Prototyping of Robotic Systems*, Proc. of Intern. Conf. on Robotics and Automation, San Francisco, CA, 2000.

[70] Rethink Robotics, *Baxter Electric Parallel Gripper.* [Online]. Available: http://www.rethinkrobotics.com/accessories/ (accessed 15 February 2015)

[71] R. Ma. Raymond, L. U. Odhner and A. M. Dollar, *A Modular, Open-Source 3D Printed Underactuated Hand*, Robotics and Automation, 2013. ICRA 2013. IEEE International Conference on. IEEE, 2013.

[72] A.M. Dollar, R. Howe, *The Highly Adaptive SDM Hand: Design and Performance Evaluation*, International Journal of Robotics Research, 29 (5), pp. 585-597, 2010.

[73] C. Gosselin, F. Pelletier and T. Laliberte, *An Anthropomorphic Underactuated Robotic Hand with 15 Dofs and a Single Actuator*, IEEE International Conference on Robotics and Automation, 2008

[74] L.U. Odhner, L.P. Jentoft, M.R. Claffee, N. Corson,Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe and A.M. Dollar, *A compliant, underactuated hand for robust manipulation* IJRR The International Journal of Robotics Research, 33(5): 736-752, 2014.

[75] Stratasys, *uPrint SE Plus*, 2012. [Online]. Available: http://www.stratasys.com/3d-printers/idea-series/uprint-se-plus (accessed 15 February 2015)

[76] A. ten Pas and R. Platt, *Localizing Antipodal Grasps in Point Clouds* 2015. [Online]. Available: http://arxiv.org/abs/1501.03100 (accessed 22 February 2015)

[77] Yale Grab Lab, *Yale OpenHand Project.* [Online]. Available: http://www.eng.yale.edu/grablab/openhand/ (accessed 3 March 2015)

[78] Shadow Co., *The Shadow Dexterous Hand.* [Online]. Available: http://www.shadowrobot.com/products/dexterous-hand/ (accessed 4 March 2015)

[79] Diftler, M., Mehling, J., Abdallah, M., Radford, N., Bridgwater, L., Sanders, A., Askew, S., Linn, D., Yamokoski, J., Permenter, F., Hargrave, B., Platt, R., Savely, R., Ambrose, R., *Robonaut 2: The First Humanoid Robot in Space*, IEEE Int'l Conf. on Robotics and Automation, Shanghai, May, 2011