Supernumerary Robotic Limbs: Biomechanical Analysis and Human-Robot Coordination Training

by

Clark Davenport

B.S., Engineering as Recommended by the Department of Mechanical Engineering,
Massachusetts Institute of Technology, 2011

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author .....
Department of Mechanical Engineering
May 5, 2013

Certified by .....
H. Harry Asada
Ford Professor of Mechanical Engineering
Thesis Supervisor

Accepted by .....
David E. Hardt
Chairman, Department Committee on Graduate Students
Department of Mechanical Engineering

Supernumerary Robotic Limbs: Biomechanical Analysis and Human-Robot Coordination Training

by

Clark Davenport

Submitted to the Department of Mechanical Engineering
on May 5, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

As the workforce within manufacturing grows older, especially within aircraft manufacturing, the need for new technologies to assist workers arises. If a technology could offer improvements to an aircraft manufacturing laborer's efficiency, as well as reduce the load on his body, it could potentially see vast use. This thesis discusses a potential solution to these issues - the Supernumerary Robotic Limbs (SRL). These limbs could potentially increase the workspace of the human operator to him more efficient, as well as reduce the load on the human while he performs staining tasks. It accomplishes this by providing the worker with extra arms in the form of a wearable backpack. This thesis first evaluates how the torques imposed on a human are affected when he uses an SRL-like device to help bear a static load. It is shown that the human work load necessary to bear such a load is reduced substantially. The second focus of this thesis is the skill acquisition. A data-driven approach is taken to learn trajectories and a leader-follower coordination relationship. This is done by generating teaching data representing trajectories and coordination information with two humans, then transferring the pertinent information to a robot that assumes the role of the follower. This coordination is validated in a simple one-dimension example, and is implemented on a robot that coordinates with a human leader during a control-box wiring task.

Thesis Supervisor: H. Harry Asada
Title: Ford Professor of Mechanical Engineering

Acknowledgments

First and foremost I would like to thank my fellow students and visiting engineers in the d'Arbeloff lab, whose constant help and support guided me over these last two years. A very special thanks to Shin and James for going far out of their way in order to help me when needed, Ani for his electronics expertise, and our UROP Kam for staying with us in the lab through ridiculously long hours.

Of course my parents deserve very much credit for this, who saw me through all of my schooling and pushed me even when I didn't want to be. I would not be not anywhere close to completing something like this without your love and support.

A very large thank you to my girlfriend Shireen who was always there with an open ear and food during those long nights in lab prepping for PSets, sponsor visits, and this thesis.

I would like to thank the Boeing Corporation for its financial support, without which, I would not have been able to complete these my degree as a Research Assistant.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation for New Assistive Technologies in Aircraft Assembly

In the past two decades, there has been a resurgence in the desire to establish manufacturing superiority in the United States [1]. This desire is especially true within the context of aircraft manufacturing, where companies like Boeing wish to expand their domestic manufacturing capabilities within the US. Boeing, in collaboration with various universities, is investigating technologies within the field of robotics to accomplish this goal. When using robotics to expand these manufacturing capabilities, there are two approaches that can be taken. The first approach is to employ heavily autonomous robotic systems, which have been widely adopted in some industries. The second, starkly different approach is to instead keep humans in the loop and provide them with technologies that would increase their efficiency, enhance their accuracy, and reduce their fatigue while performing their tasks.

Given all the potential benefits of heavily autonomous systems such as repeatability of performance and precision monitoring, a valid question is why are efforts not fully devoted to the development of a robust autonomous system? The reality of the situation in the aircraft manufacturing and assembly world is that many jobs cannot be automated due to task and spacial constraints. One example task that would be

quite difficult to automate is the assembly of titanium beams, which requires assistance from a support structure, as well as demands the labor of multiple workers. Small-lot production is another challenge. There are many other tasks like this that could not be easily accomplished with autonomous systems.

Because all tasks are not able to automated within the aircraft manufacturing process, potential issues and problems that arise from the requirement of manual laborers must be investigated. The first major concern is that the average worker age in all industries is roughly 40 years old, but the median age of workers in aircraft manufacturing in 48.4 years. Taxing aging bodies is of obvious concern in the context of both stamina and health.

In both general and aircraft manufacturing, there are a number of especially straining or dangerous tasks that the workers must perform. One issue inherent with manual labor in this context is the difficulty in reaching various positions. Some tasks may require the worker to adopt unergonomic positions. A second issue inherent to manual labor tasks is the physically demanding nature of the work, which can fatigue workers, especially the older workers in this industry, quickly. Additionally, a real danger that aircraft manual laborers are exposed to is the inhalation of the carbon fiber particles that are the result of drilling into carbon fiber material. These particles are extremely toxic to humans. In order to present the spread of these particles, the task of drilling into carbon fiber becomes a two person job. While one worker drills, another worker must be constantly vacuuming up the particles. This leads to an obvious loss in overall worker productivity.

1.2 The Supernumerary Robotic Limbs (SRL)

Taking into consideration all these problems with manual labor, the d'Arbeloff lab has developed the Supernumerary Robotic Limbs (SRL), which is described in the three proceeding subsections.

1.2.1 Concept of the SRL

The SRL is a wearable robot that provides extra limbs to a human worker with the intent of assisting him in manufacturing and various other tasks. A concept drawing of the SRL is shown in Figure 1-1, while the realization of the SRL is shown in Figure 1-2.

Figure 1-1: Concept drawing of the Supernumerary robotic limbs. Picture by F Parietti.

These extra limbs can take the roles of either extra arms, legs, or both, while assisting the worker in a workspace that potentially exceeds that of the human. The aim of the SRL is that it will be so closely integrated with the human body that it will increase both the efficiency and productivity of the worker. Because it will be worn directly by the user at the iliac crest (in the same fashion of a hiking backpack), as well as share much of the workspace of the human, it is designed to operate at safe and comfortable ranges. Ultimately, we wish for the human to work so closely with the SRL that the SRL is perceived as an extension of the human body.

1.2.2 Potential Uses of the SRL

There are many tasks in the aircraft manufacturing setting where the SRL could potentially benefit the manual laborer.

- Imagine the situation where a human needs to hold up a heavy piece to be

Figure 1-2: Realization of the Supernumerary robotic limbs. Picture by F Parietti and K Chan.

drilled. Using the SRL, the human can focus on carefully operating the drill while the SRL bears the load of the piece.

- Imagine the task of wiring a control box. With the SRL, the human can focus on positioning the wires into the appropriate control box terminals while the SRL comes in and affixes the wires.

- Imagine the task of drilling in an unergonomic position. Using the SRL to brace or otherwise support the worker, he can complete his drilling job in a tight, hard to manage space without fear of falling [2].

- Imagine the task of drilling and vacuuming carbon fiber as previously discussed. With the SRL, it would take only one human to drill holes in the carbon fiber because the SRL limbs would simultaneously vacuum the toxic carbon residue.

One can easily envision potential benefits of the SRL in other contexts outside the aircraft manufacturing setting.

- One can envision an astronaut using the SRL to assist work on space shuttles and the International Space Station.

- One can envision the SRL assisting a surgeon in the operating room, providing him an extra hand to hold tools or assist in the cutting operations.

- One can envision the SRL providing help to an assisted living worker, by either helping him support an elderly's weight or grabbing one of the elderly's belongings.

1.2.3 Potential Benefits of the SRL

Ultimately, the SRL is designed to provide the worker the benefits of having additional robotic limbs. The first potential benefit is the increase in productivity due to being able to accomplish more without having to rely on other workers. The second potential benefit is the expansion of the human workspace, which could subsequently allow humans to perform new tasks that they could not otherwise. The third potential benefit is the elimination of the poor postures due to the SRL's weight-bearing ability. Similarly, the fourth potential benefit is the reduction of the human work load, which should increase a worker's safety and reduce his fatigue. The fifth potential benefit is the increase of the accuracy and work quality of the human due to the robotic limb's enhanced positioning. The sixth and last potential benefit is the addition of in situ error detection and correction made possible by limbs fully equipped with task-monitoring sensors.

1.3 Thesis Layout

This work will focus specifically on a few potential benefits that the SRL has to offer. The first focus is the validation that the SRL in fact does reduce the work load of the human worker in the case of static loading. In order to realize many of the potential benefits of the SRL, however, the implementation of its control scheme must be addressed. More specifically, the details close interaction of the human and the robot must be determined. Therefore, the second focus of this paper is on the coordination of the human worker with the SRL by means of learning in order to

realize those benefits described above. This coordination has two main parts. The first is learning the robotic motions from human demonstrations of a task. The second is learning how the SRL adjusts its behavior based on the actions of the human, again from human demonstrations of a task.

The following are the chapters of this thesis.

1. Chapter 1 motivates the development of the SRL and its research areas.
2. Chapter 2 explores the prior art in the areas of SRL-like robots, biomechanics, and trajectory generation.
3. Chapter 3 briefly introduces the design of the SRL.
4. Chapter 4 examines the biomechanics of using the SRL for static loading cases.
5. Chapter 5 explores a method of generating endpoint trajectories in a reduced dimensional space using Partial Least Squares Regression.
6. Chapter 6 develops a method for coordinating the human with the SRL.
7. Chapter 7 concludes and lists potential future work.

This concludes the introduction and the details of this work of the last two years follow.

Chapter 2

Prior Art Relating to the Objectives of the SRL

2.1 Adopting the SRL as Part of the Human Body

Although not the focal point of this particular thesis, we hope that one day the extra limbs provided by the SRL can be accepted as parts of the body. Therefore, psychological work concerning the ownership of corporeal and non-corporeal objects is of interest.

2.1.1 The Right Hand Illusion

The standard experiment for controlled manipulation of body image and ownership is the right hand illusion (RHI), developed originally by Botvinick and Cohen [3]. To induce the sense of ownership of a rubber hand, they simultaneously touched the participants' real hand, which was hidden out of view, and a rubber hand. When this occurred, participants reported feeling the strokes of the experimenter on the rubber hand.

The general structure of the experiment has been repeated over the last decade and a half in order to discover the conditions necessary to induce the illusion and subsequently understand the brain's mental model of one's appendages. Currently,

there are two competing appendage model. Armel and Ramachandran proposed that as long as visual and tactile correlations are established, the illusion can be induced [4]. Subsequently, they predicted that any object can be adopted as part of one's own body. Tsakiris et al. ran an experiment testing the sense of ownership on increasing lifelike objects and found that participants only experienced ownership of a rubber hand. This suggests that although the visual and tactile stimulation is necessary in experiencing the RHI, only corporeal objects are included in one's model of self [5]. This last test has important implications on designing a supernumerary device that the user perceives as his own. Indeed, it suggests that the adoption of the arm into the model of one's self would be best accomplished with a robot that is similar in both form factor and dynamic performance to the human arm or leg.

2.1.2 The Right Hand Illusion with Additional Limbs

Quite recently, the RHI was extended to test the ownership experience of extra limbs in work done by Guterstam et al [6]. Instead of hiding the participants' right hands as done in the traditional RHI experiments, Guterstam et al. placed an object next to the real hands while hiding the space between the participants' shoulder and the beginning of the extra object with a cloth. As in the traditional RHI experiments, both the real and hands and object were stimulated via touch. This stroking invoked a feeling of having two right hands, instead of a strong disownership of the real right hand. Supporting Tsakiris et al's suggestion, the Guterstam et al. work found that the sense of ownership in the extra object was only invoked when both the object and the right hand were stroked synchronously, and when the extra object was anatomically similar to the right hand. Such a finding suggest that one may be able to perceive a robot arm as an extra limb if it is made and aligned in an anatomically similar fashion to his real arms.

2.2 Systems Similar to the SRL

When evaluating the potential utility of the SRL, one must examine other robotic systems that operate closely with a human operator for comparison. Over the last 50 yeas, there has been much progress in the development of exoskeletons. There have been two general areas of advancement - in exoskeletons and active orthoses.

2.2.1 Exoskeletons

An "exoskeleton" is generally thought of as a device that enhances the abilities of its wearer. With these types of devices, the design focus seems to be on load-bearing capacity, speed, and stamina [7, 8, 9, 10, 11]. Although these certainly do assist users with physically taxing tasks such as bearing loads of up to 84kg for a sustained period of time (something many humans cannot do without substantial fatigue), the exoskeletons do not come without their drawbacks. Current technical difficulties include minimizing the weight of the exoskeleton, designing lightweight, transportable, and long-lasting power supplies, and creating more efficient transmission systems [11]. It is important to note that while exoskeletons do increase the strength of humans, they do not increase the workspace of the human - in fact, a sizeable problem of this technology is kinematic constraints caused by attachments of joints and actuators. [7]

2.2.2 Active Orthoses

An "active orthosis" is generally thought of as a device to assist someone with leg problems [7]. Although these devices are designed to increase the weight bearing capacity of a damaged leg, the focus of these devices is not to provide strength that far surpasses that of a normal human. To this effect, the objective of these devices is far different than those of the SRL. This detail notwithstanding, these devices have to be light enough to be worn for extremely long time periods and thus the SRL can take design cues from solutions to this constraint.

2.3 Programming by Demonstration (PbD)

Because a substantial focus of this work concerns endpoint trajectory and coordination learning, it is useful to look at data-driven, programming by demonstration, and teaching-by-showing approaches to trajectory learning and trajectory recognition in robotics.

Robot programming by demonstration has been used in a number of studies to reduce the amount of explicit commands that are needed to be programmed into the robot, especially in repetitive assembly tasks [12]. Vision has been used to discern high-level plans from hand data [13]. Contact positions between a robot and its environment were found also using vision by Miura and Ikeuchi [14]. Virtual environments have been used to learn assembly strategies by looking at a robot's contact transitions with its environment [15]. Force-based states were used by Skubic and Volz so that skill execution is reliant on changes in force and not absolute position [16]. Using this approach, detailed geometric information of the environment does not need to be known a-priori. Using the covariance that was exhibited between learning trials, Dong learned primitive motions using a representation called probabilistic flow tubes, automatically detected relevant features in the environment, used temporal information to handle traversing across learned trajectories, and recognized in real time the current classification of motion [17].

Different PbD and other data-driven techniques have not only been used to generate assembly and manufacturing tasks, but have also been used to identify various types of hand gestures and grasps.Many of these techniques are heavily rooted in computer science and statistics. Hidden Markov Models were used to continuously identify sequences of hand gestures in work by Bernardin et al [18]. Gaussian mixture models (GMMs) were used by Palm et al. to model fingertip positions as grasp primitives. Using these primitives, they recognized various hand gestures [19]. The Glove-Talk project showed the power of neural network techniques in recognizing hand grasps [20]. Fuzzy rule-based approaches have been used by Bedregal et al. [21]. Ju et al. produced impressive results when identifying continuous hand ges-

tures with a time-clustering method that used finger angle trajectories and the angle models [22]. Although all the these techniques concern hand positions, they are quite related to the question of discerning operator intent with the SRL so that it may react appropriately.

2.4 Input Space Reduction

When dealing with such a complex system as the SRL, the size of one's input space which decisions are based off of becomes of interest. In gesture recognition and robotic arm traversal this has been explored using the technique of Principal Component Analysis (PCA), which seeks to reduce the size of one's input space by describing his data with a smaller number of variables. Fod et al. represented human movements using linear primitives [23]. Jenkins et al and Lim et al used these primitives to reduce computation when replicating human movements in robot arms [24, 25]. Jiang et al. extended the PCA technique so that the primitives do not need to be recalculated every time new data is captured [26].

2.5 Departure from Prior Work

As mentioned in the previous chapter, one focus of this work is in the impact of using the SRL during the operation of static loading. The departs from the prior work in that a device like the SRL has not been used to assist in load bearing - its design differs in structure and intent from that of the exoskeletons and the active orthoses and therefore needs to be examined. More generally, it is of interest to provide a quantitative biomechanical benefit of using our device. The second focus of this paper is motion generation. Most importantly, we explore the real-time updating of the motion of a robot that interacts closely with a human. It does so by constantly examining information about the position of a human or other objects of interest. My focus here differs with the prior work mentioned above in the emphasis on using learning to generate motions in real time, as opposed to classification.

Chapter 3

Brief Design Introduction

Before the biomechanical analysis and the control of the SRL are presented, it is beneficial to include a brief introduction of the design of the SRL so the reader understands the system to be analyzed. However, the design is not the point of this particular thesis, nor my personal work. This is largely the work of Ph.D. student Federico Parietti and undergraduate student Kameron Chan.

3.1 Design Concept

As explained in the introduction, the purpose of the SRL is to assist workers in the context of aircraft manufacturing and final assembly. It achieves this via augmenting the workspace of the human and bearing a load that would otherwise be born by the human. Both of these capabilities should allow a single operator to accomplish jobs that were previously impossible to accomplish with a single human or that imparted large loads onto the human. Lastly, the robot can potentially offer additional safety features to the workers by offering an extra leg if the worker is slipping, or by grabbing a scaffold when the human adopts unsafe postures.

3.2 Functional Requirements

The largest functional requirement of the SRL is to well-mimic a human arm. The reasons for this are three-fold:

1. In order to give the user the highest probability in perceiving the SRL as an extension of the human body, the robot dynamics should well match those of a human arm or leg.

2. Because part of the intended purpose of the SRL is to take over fatiguing jobs from the human, the SRL bandwidth, velocities, and torques must be comparable or better to that of the human.

3. In order to interface well with tools and fixtures that are designed by and for humans, a robotic arm similar to that of a human can be quickly integrated into aircraft assembly facilities.

Because the workers will likely be wearing this robot for substantial periods of time, another functional requirement is that the robot is lightweight and able to be worn for hours with minimal external forces exerted on the worker during both periods of rest and periods of operation. Next, another purpose of the SRL is to augment the workspace of the human, rather than to interfere with it. To this effect, interference in the humans workspace must be minimized. Lastly, the robot must be safe. This will be working in extremely close proximity with the human, and potential forces and hazardous motions that can be applied to the human during cases of malfunction must be minimized, if not eliminated entirely.

3.3 Prototype Realization

The SRL was realized with two, three degree-of-freedom robot arms affixed to a hiking backpack-like harness, which is worn by the operator at the hip. Each arm has two rotational joints at the base and one at the middle of the arm. The torque characteristics are modeled after the shoulder and the elbow to achieve the first

functional requirement. The arms are actuated with flat brushless motors. Using large amounts of carbon fiber tubing, the mass of the system is kept near 20kg, which achieves the weight functional requirement. Because the motors are largely housed behind the operator, it interferes very little with the humans workspace and due to its size, expands it. In order to achieve safety, series viscoelastic components are placed between the gearhead output of the motor and the actuator output, which minimize impact forces in the event that the robot malfunctions. An operator wearing the SRL in a rest position is shown in Figure 3-1. An example task for the SRL, which is to hold a piece of material while the human operator drills, is shown in Figure 3-2.

Figure 3-1: Supernumerary robotic limbs being worn by an operator. Picture by F Parietti and K Chan.

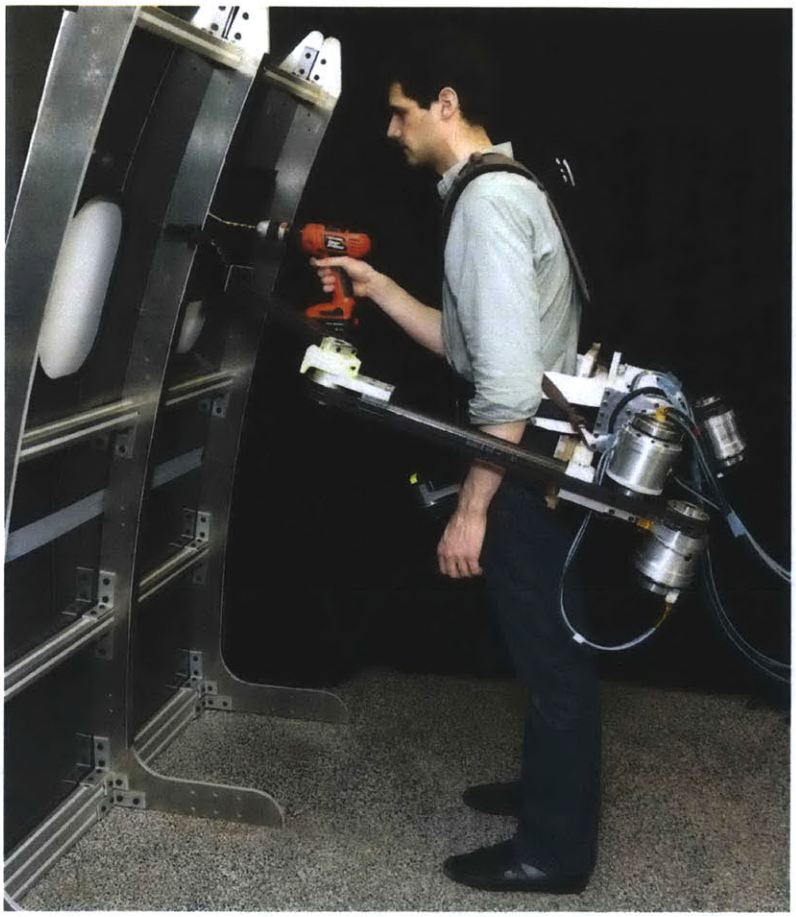

Figure 3-2: Supernumerary robotic limbs assisting an operator with a drilling task. Picture by F Parietti and K Chan.

Chapter 4

Biomechanic Analysis of the Supernumerary Robotic Limbs

4.1 Introduction

As stated in Chapter 1 as one of the objectives of the SRL, reducing the load of the worker is of great importance. When looking at the potential use cases of the SRL, one quickly finds that effort reduction is especially pertinent in static load-bearing cases. A static loading case can be considered the situation where one must hold a mass steadily at a particular distance away. Consider again Figure 3-2, where the SRL is helping bear the load of the piece of aluminum to be drilling. It would obviously be beneficial to the human if his work load was less while performing this task with the SRL than without the SRL.

In order to evaluate the work load of the SRL in the static load-bearing case, the subject of this chapter, there are a number of things to take into consideration. The first is determining how to quantify this work load. The joint torques at each major joint provide good insight into how much load the operator is bearing. It was the initial hopes that the joint torques in the aggregate should be reduced. However, a major obstacle facing the SRL is its substantial mass, which is a non-negligible 18kg in its first iteration and one that would surely induce an increase in joint torques. In order for the SRL to therefore be useful, the positive effects of the load reduction

must counteract the negative affects of the induced joint torques brought about by the mass of the SRL.

The precise manner in which the joint torques were derived and the work load of the human quantified during a static loading task with and without the SRL are described in the following sections of this chapter.

4.2 Biomechanic Model

A biomechancial model of the human performing the static loading task with and without the SRL was developed and shown in Figure 4-1.

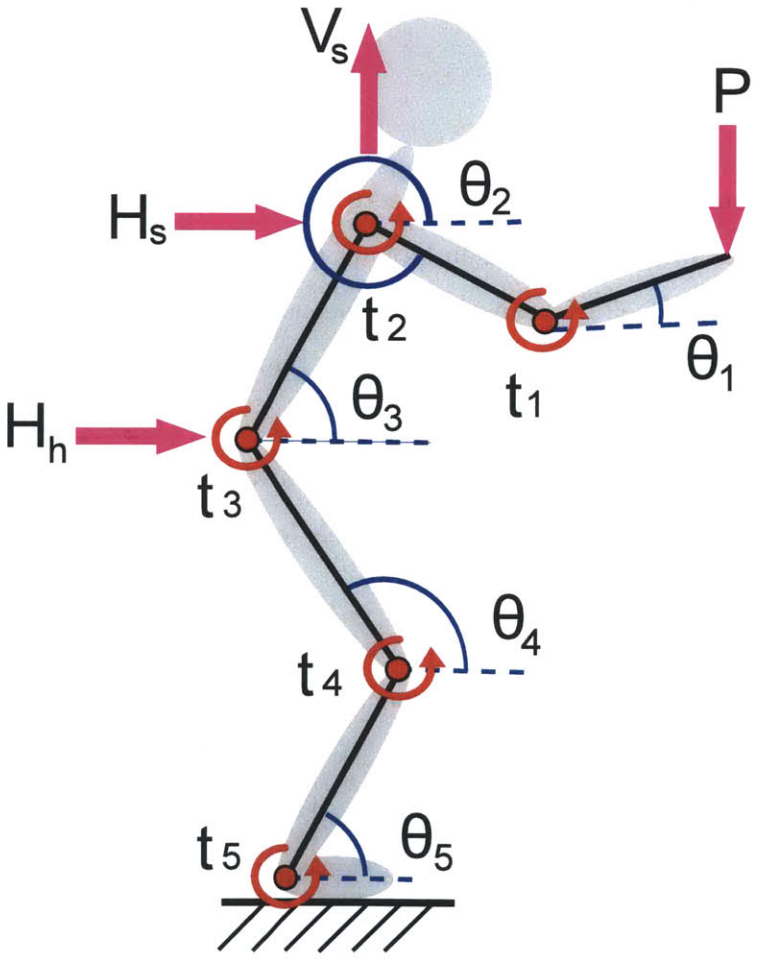

Figure 4-1: Biomechanical model of the human utilizing the SRL in a static load bearing task.

In order to reduce the complexity of the human to a manageable order while still offering richness that would allow for insight into the impact of the SRL on the human, the human was reduced down to five rigid bodies that were connected into a kinematic chain that exists in the sagittal plane. The five rigid bodies included in the model were abstractions of the leg shanks, the thighs, the torso, the upper arm, and the forearms. The body number of these segments are 5, 4, 3, 2, and 1 respectively. Lengths and mass properties of these segments were found in works by Boling et al. and Wang et al. [27, 28]. They are presented here in Table 4.1.

Table 4.1: Biomechanical model rigid body segment names and physical properties

Joint Number	Name	Length (m)	Length to CoM (m)	Mass (kg)
1	*Forearms*	0.472	0.322	1.650
2	*Upper Arms*	0.346	0.151	2.100
3	*Torso*	0.536	0.268	21.675
4	*Thighs*	0.456	0.258	7.500
5	*Shanks*	0.530	0.209	4.575

Modeling the interaction of the SRL with the back can potentially be quite complex. Because the SRL is to be worn like a hiking backpack, looking at prior works in the field of modeling backpack interactions with the human provided insight into this issue. Ren et al. developed a dynamic model to describe a pack's response to trunk motions by using a nonlinear pack suspension equation. The force that a pack exerted on the axis parallel with the spine was captured with cubic polynomials, whose parameters were calculated from dynamic test data. Using this technique, it was possible to evaluate the resultant pack forces and moments imposed on torso motions [29]. Despite the utility in dynamic tests, this method is a bit complicated for a first pass at understanding the SRL effects and requires both dynamic data and a complex test rig. Instead, Foissac et al. modeled the movement of a hanging backpack as a damped harmonic oscillator moving in a single direction [30].

Taking Fissac's work as inspiration, the forces imposed on the human were modeled simply. Taking into consideration that the supporting frame of the SRL comes into contact with the human body predominantly at the shoulders and the hip, the

SRL was abstracted into point forces that acted on the human kinematic chain at these locations. At the shoulder, the user feels forces imposed by the SRL in the direction along the shoulder and thus the force imposed on the human is modeled as one point force acting vertically at the shoulder joint and one point force acting horizontally at the shoulder joint. At the hip, the SRL is secured at the iliac crest and thus the force imposed on the human here is one point force acting horizontally at the hip joint.

The static load to be born is modeled simply as a point force acting at the beginning of rigid body 1 in the vertical direction.

From this model, the joint torque at each joint can be calculated using the following equations:

$$\tau_1 = PL_1 \cos\theta_1 + m_1 g d_1 \cos\theta_1 \tag{4.1}$$

$$\begin{aligned} \tau_i = \tau_{i-1} + \left[\left(P + \sum_{k=1}^{i-1} m_k g\right) L_i + m_i g d_i\right] \cos\theta_i + \\ [-V_S^* L_i \cos\theta_i + H_S^* L_i \sin\theta_i](i \geq 3) + \\ [-V_H^* L_i \cos\theta_i + H_H^* L_i \sin\theta_i](i \geq 4) \end{aligned} \tag{4.2}$$

where Equation 4.1 describes the torque on joint 1 and Equation 4.2 describes the torques on the rest of the joints, where $i = 2, .., 5$. In Equation 4.1, τ_1 is the torque at joint 1, P is the static load to be born, g is the acceleration of gravity, m_1 is the mass of the first joint, d_1 is the distance from joint 1 to rigid body 1's center of mass, L_1 is the length of rigid body 1, and θ_1 is the angle of the first rigid body with respect to the horizontal. In Equation 4.2, τ_i is the torque at joint i, L_i is the length of rigid body i, m_i is the mass of rigid body i, d_i is the distance from the joint i to rigid body i's center of mass, θ_i is the angle formed between rigid body i and the horizontal, and g is the acceleration of gravity. For the point forces, P is the static load to be born as in the previous equation, variables V and H denote horizontal and vertical forces,

subscripts S and H denote the shoulder and the hip, respectively, and superscript $*$ indicates the force is evaluated in the global frame.

4.3 Experimental Setup

This section describes how the joint torques were calculated experimentally.

4.3.1 Testing Apparatus

Because the SRL was not completed at the time of this experiment, a test rig that mimicked the mass properties and the geometry of the SRL was created. The test rig is shown in Figure 4-2.

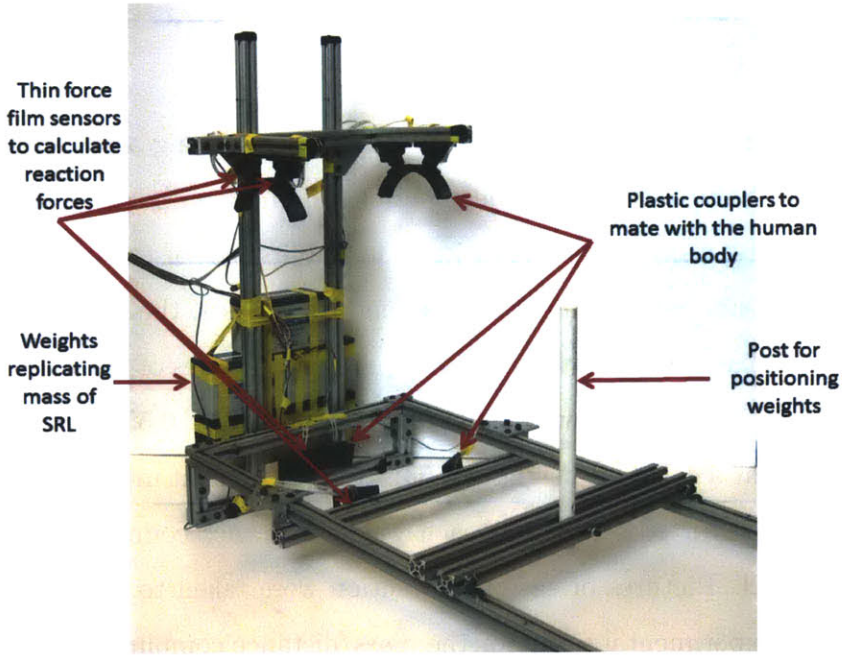

Figure 4-2: Test rig used in experiments to find effects of bearing a static load with the SRL on the human

To mimic the weight and position of the motors and gear train of the SRL, 15kg masses were attached to the bottom of the rig where it made contact with the hip. Because 80/20 does not mate well with the human, 3D printed shoulder cups and hip plates were attached to the 80/20 frame. These can be seen as the black pieces

in Figure 4-2. In order to capture the forces exerted by the SRL onto the human, FlexiForce® sensors (model A201) were positioned throughout the testing apparatus. To capture the shoulder forces, two force sensors were places on the opposite ends of each 3D printed shoulder support 30° from the vertical. To capture the horizontal hip forces, two force sensors lined the abdomen and two were placed along the lower back in the horizontal direction.

To use the biomechanical model, the angles of the human joints needed to be found. For that, a vision system was used. At each joint a marker was placed. To capture the position of the marker and thus the joint angles in the sagittal plane, a simple camera was used. The measurements from the force sensors were all captured with LabView data acquisition software running at a sampling rate of 50 Hz.

4.3.2 Experimental Task Description

In order to capture the differences in human work load during a static loading task with and without the SRL, the subject performed the load-bearing task with and without the test rig in many configurations. In the first set of tests, the test rig was not used to assist in bearing load. In the second set of tests, the user wore the rig to bear the load. In each set of tests, the participant bore masses of 0 to 9.07kg in increments of 2.268kg. The weights were held at distances of 0.425, 0.55, 0.675, and 0.800m from the human, at a constant height of 0.9m. The participant maintained a comfortable static position for 20 seconds at each weight/distance combination while force data was collected. Pictures of each combination were taken to capture posture information. The experimental setup for the mass/distance combination $9.07kg/0.55m$ is shown in 4-3.

4.3.3 Data Processing

Joint torques could be calculated after raw force data was collected in the manner described in the previous section. First, the raw force data from the testing apparatus was used to find the resultant point shoulder and hip forces. Joint angles were

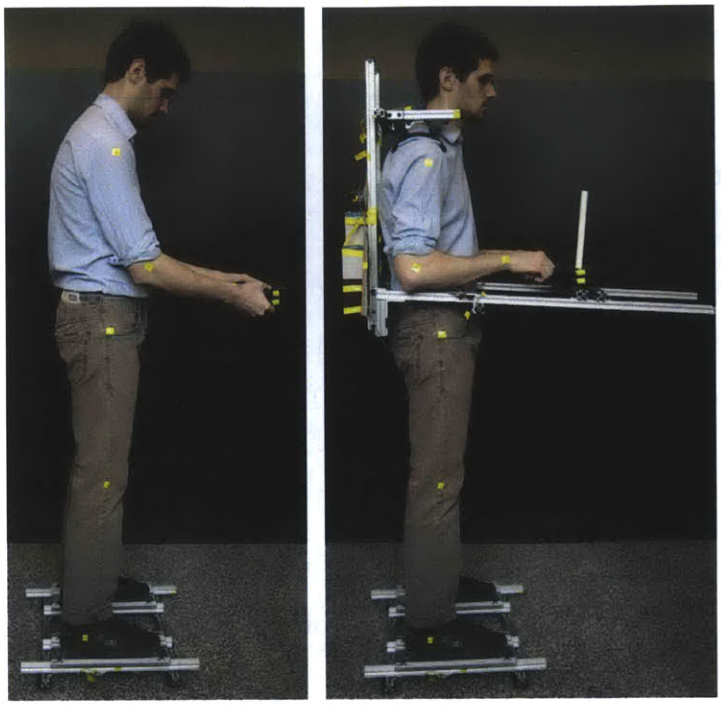

Figure 4-3: Experimental setup for the static loading task for mass/distance combination $9.07kg/0.55m$ with and without the testing apparatus.

calculated via processing of posture images with the MATLAB® Image Processing Toolbox. These experimentally derived forces and angle measurements were used as inputs into Equations 4.1 and 4.2, which directly calculated the torques at each joint of interest.

There was an imperfection with the testing rig that needed addressing. This was the fact that the horizontal forces that were calculated at the hip and the shoulder were not equal, as would be expected in the static loading case. This may have been the result of the back and/or the stomach not being in full contact with the force sensors. Given that the test rig was held against the human's hips via a bar of 80/20, this is not all that unlikely. In order to compensate for this problem, a force equal in magnitude and acting in the direction opposite to the more accurate (due to geometric constraints) horizontal shoulder reaction force was assumed to be acting at the hip to ensure that the static equilibrium condition was met. The torques at each joint were then calculated.

4.4 Raw Joint Torques

4.4.1 Numerical Results

The calculated torques for bearing a static load without the test rig are summarized in Table 4.2, while the torques for bearing a static load with the test rig are summarized in Table 4.3. The torques given are for joints 1, 2, 3, 4, and 5, and have units of Nm.

Table 4.2: Calculated raw torques (Nm) from bearing static loads without test rig.

Distance (m)	Mass (kg)	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5
0.425	**0**	5.04	6.75	-4.36	-14.94	-2.79
0.425	**2.27**	5.57	6.55	-2.42	-2.30	5.50
0.425	**4.54**	5.96	7.10	-3.86	-4.39	3.27
0.425	**6.80**	6.61	7.22	-3.21	-6.05	6.56
0.425	**9.07**	7.13	6.24	-0.09	-2.25	-29.04
0.550	**0**	4.98	9.24	5.56	8.35	23.56
0.550	**2.27**	5.44	8.43	1.61	8.99	32.30
0.550	**4.54**	6.03	8.89	5.58	7.20	29.61
0.550	**6.80**	6.54	9.76	4.62	6.14	26.92
0.550	**9.07**	7.05	10.01	11.63	12.00	34.13
0.675	**0**	4.68	11.55	10.80	10.87	21.41
0.675	**2.27**	5.12	11.03	18.10	17.34	30.65
0.675	**4.54**	5.74	11.40	17.76	17.67	34.46
0.675	**6.80**	6.15	11.66	29.82	28.72	46.31
0.675	**9.07**	6.79	12.31	21.97	20.50	39.31
0.800	**0**	4.94	10.63	27.61	20.30	39.63
0.800	**2.27**	5.00	12.35	41.13	15.00	34.91
0.800	**4.54**	5.80	12.68	46.67	26.53	40.54
0.800	**6.80**	6.04	14.01	50.25	7.93	36.04
0.800	**9.07**	6.38	14.62	55.81	15.33	29.74

Results for illustrative absolute joint torque distance/mass pairs for loading with and without the test rig are shown in Figure 4-4. The mass/distance combination pairs in the figure are $2.27kg/0.55m$ and $9.07kg/0.800m$.

Table 4.3: Calculated raw torques (Nm) from bearing static loads with test rig.

Distance (m)	Mass (kg)	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5
0.425	**0**	5.20	3.64	-2.74	-3.98	24.61
0.425	**2.27**	5.21	3.05	1.12	-1.58	23.92
0.425	**4.54**	5.21	2.56	3.41	2.90	37.55
0.425	**6.80**	5.20	3.37	4.43	-2.03	37.08
0.425	**9.07**	5.21	2.69	6.69	3.15	35.13
0.550	**0**	5.18	2.74	-1.99	4.77	16.66
0.550	**2.27**	5.20	3.51	0.13	8.43	25.04
0.550	**4.54**	5.20	2.97	3.59	7.83	26.59
0.550	**6.80**	5.15	3.84	2.01	10.34	30.31
0.550	**9.07**	5.19	4.35	14.05	18.70	36.37
0.675	**0**	5.17	4.74	-5.67	-1.42	11.98
0.675	**2.27**	5.21	3.67	0.33	7.88	20.13
0.675	**4.54**	5.19	2.55	12.01	14.03	22.99
0.675	**6.80**	5.14	2.81	9.90	1.87	23.71
0.675	**9.07**	5.20	3.91	9.54	7.82	45.51
0.800	**0**	5.14	2.17	-3.71	11.63	24.03
0.800	**2.27**	5.20	3.39	-0.36	7.65	22.36
0.800	**4.54**	5.11	1.40	1.57	0.75	20.02
0.800	**6.80**	5.15	1.78	13.36	2.97	38.96
0.800	**9.07**	5.17	3.48	20.09	5.78	50.40

4.4.2 Discussion

The first trend of note is that while using the SRL rig, the torques acting on joints 4 and 5 are typically higher than the torques acting on the other joints. These resultant higher torques likely occur on these joints of the lower body because the weight is being borne by the testing rig instead of the arms. The second trend of note is that the torques caused by the masses 0 to 6.8kg at a distance of 0.55m were quite low. Such low torques in these configurations were likely a result of the torques caused by the loading mass counteracting the torques caused by the mass on the back of the testing apparatus.

The third trend of note is that the static load-imposed joint torques tend to be smaller for the same mass/distance combination while wearing the rig compared to those torques imposed when not wearing the rig. However, this trend is not without exception. The torques imposed on joint 5 while bearing the load with the test

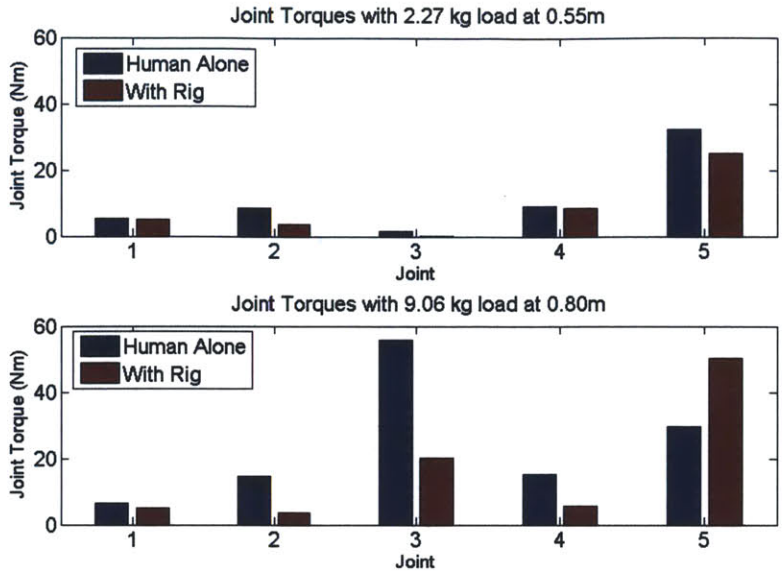

Figure 4-4: Illustrative joint torques for static loading task for mass/distance combination $2.27kg/0.55m$ with and without the testing apparatus.

rig at the distances of 0.550m, 0.675m, and 0.800m for the mass of 9.07kg tend to comparable but slightly higher to those torques imposed while bearing the load without the rig. This is likely due to the load being borne predominantly by the lower body instead of the upper body, which is the case when bearing the load without the test rig. The final trend of note is that the torques on joint 5 tend to be far higher when the load is borne at a distance of 0.425m from the body. This is likely caused by the lever arm of the load being too small to effectively help counterbalance the torques imposed by the mass of the rig.

4.5 Normalized Joint Torques

4.5.1 Numerical Results

When attempting to evaluate the effectiveness of baring the static load with the rig when compared to not using the rig, one needs to consider that the torques imposed on the joint 1 will not have the same perceived effect on the human's exertion because humans are able to bear more torques on some joints than others. Therefore, a metric

other than the sum of absolute joint torques needs to be developed. If the imposed torque on each joint is normalized by the maximum torque that the joint is able to bare, the imposed torques on each joint can be directly summed to obtain a single metric to allow one to compare the torques from trials of the same mass/distance combination imposed with and without the test rig.

The metric Overall normalized Exertion (OE) for each mass/distance pair can be therefore expressed as

$$OE = \sum_{i=1}^{5} \left(\frac{\tau_i}{\tau_{MAX,i}} \right)^2 \tag{4.3}$$

where $\tau_{MAX,i}$ is the maximum possible torque output for joint i, τ_i is the calculated joint torque of joint i, and OE is the overall exertion. The maximum torques for each joint used in the calculations for overall exertion are given in Table 4.4. These values were obtained from [27, 28].

Table 4.4: Maximum exertable torques for each joint in biomechanical model.

Joint Number	Maximum Torque
1	70
2	40
3	200
4	360
5	150

Using the results of the absolute joint torques and Equation 4.3, the overall exertion of each trial was calculated. The values for each trial are given in Table 4.5 and the results are given in graphical form in Figure 4-5.

4.5.2 Discussion

Using the OE metric described in this section, one clearly sees the benefit of using a load-bearing rig to bear loads. The OE was found to be lower for every weight while using the rig to bear the load at distances of 0.550m, 0.675m, and 0.800m compared

Table 4.5: Calculated overall exertion from bearing static loads with and without the rig.

Distance (m)	Mass (kg)	OE without Rig	OE with Rig
0.425	**0**	0.036	0.041
0.425	**2.27**	0.035	0.037
0.425	**4.54**	0.040	0.073
0.425	**6.80**	0.044	0.077
0.425	**9.07**	0.072	0.066
0.550	**0**	0.084	0.023
0.550	**2.27**	0.098	0.042
0.550	**4.54**	0.097	0.043
0.550	**6.80**	0.101	0.056
0.550	**9.07**	0.129	0.084
0.675	**0**	0.112	0.027
0.675	**2.27**	0.134	0.033
0.675	**4.54**	0.151	0.038
0.675	**6.80**	0.217	0.038
0.675	**9.07**	0.188	0.110
0.800	**0**	0.168	0.035
0.800	**2.27**	0.199	0.035
0.800	**4.54**	0.240	0.024
0.800	**6.80**	0.252	0.079
0.800	**9.07**	0.261	0.136

to without it. Insight into why this occurs comes from looking at the equations of static equilibrium. When using the test rig, the imposed torques caused by the load were transferred to the lower joints. Because these lower body joints can exert more torque than their upper body counterparts, the normalized exertion typically decreases when using the test rig. The OE is only typically decreased because at a distance of 0.425m, the OE was higher when using the test rig. Here, the lever arm from the human to the load was small, and therefore the imposed torques are small. Despite the rig still transferring the weight to the lower body, these smaller torques are unable to counteract the imposed torque contribution from the mass of the test rig. Such a result shows that the SRL can be useful in bearing static loads at particular distances. The biomechanical analysis can be extended to find the optimal placement of the load in order to reduce the human exertion as defined by the Overall

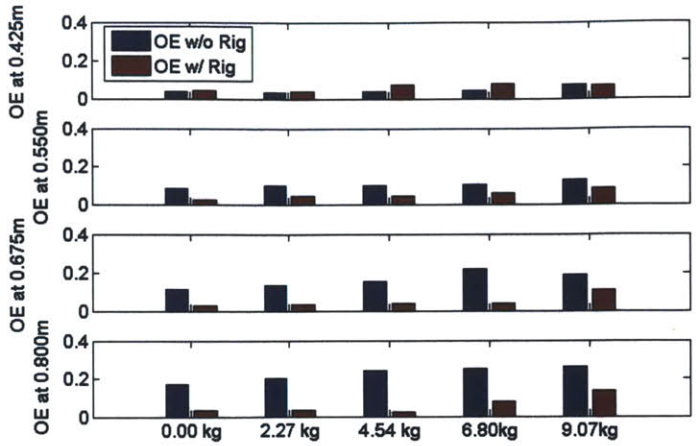

Figure 4-5: Overall exertion with and without the test rig for each mass/distance pair.

Exertion metric when using a load-assistance mechanism such as the SRL. This can be potentially combined with trajectory generators, which would allow the SRL to minimize human exertion while traversing its trajectory. As far as future work is concerned, this evaluation should be conducted with more subjects, and eventually the analysis should be extended to dynamic load-bearing. This evaluation should also be done with the actual SRL. However, this analysis shows that the SRL will reach its objective of reducing the human work load of its operator, at least in this particular use case.

Chapter 5

Endpoint Trajectory Generation

5.1 Relationship to Human-SRL Coordination

To be of use to the operator, the robot obviously must have some means of deciding where to go. The process of assigning a path, whether dynamic or static, is an extensively researched task within the context of robotics and artificial intelligence. Nevertheless, trajectories must be obtained for the SRL for each task and thus the methodology for this process should be developed explicitly. This is the subject of the rest of this chapter.

Generating a trajectory does not paint the complete picture, however. Once a trajectory has been generated for the SRL, feedback from the human should be used to decide how the SRL traverses that trajectory in order to best coordinate the SRL with human actions. This coordination step is one whose importance cannot be overlooked. If the SRL blindly follows trajectories, it can in no way respond to the action of a human. For instance, a cue given from the human may be used to command the SRL to go to its goal location. However, while the SRL is traversing its commanded trajectory the human may falter or decide the action needs to be dynamically updated. In these two instances, it would be desirable if the SRL updated its position along its trajectory given the new actions of the human instead of blindly stopping or continuing on its path. Because of this, the process in which human information is used to update the SRL's traversal along its trajectories is explored

in Chapter 6. Coordination between stationary robots and humans sharing the same workspace has been explored previously in [31].

5.2 Approaches

When determining robot trajectories, there are typically two different approaches taken: the model driven approach and the empirical-data driven approach. The two approaches are briefly summarized here.

5.2.1 Model Driven Trajectory Generation

The model-driven approach is the one where the equations of motion of a system are derived and used to dictate the motion. This approach is quite beneficial for providing physical insight into what the underlying physics of a system are. Once they are understood, they can then be easily combined with other metrics in an optimal motion planner in order to optimize the trajectory to a particular criteria. Another benefit of this approach is that it is a well-developed field, especially in terms of state prediction. Using techniques such as the Kalman Filter, one can easily predict the future positions and states of the robot at time points in the future, in addition to enabling the estimation of the states not directly sensed by the system.

However, a major drawback to this approach is that the equations of motion must be generated for every task that one wants to accomplish. If there are many submovements within a particular task, this can be quite tedious. Moreover, if the task is complex (as many real physical systems are), the derivation of these equations can be not only quite tedious but also quite challenging as well. Within the context of aircraft manufacturing, this may very well be the case.

5.2.2 Empirical-data Driven Trajectory Generation

Contrasting the model-driven approach is the empirical-data driven approach. Here, the trajectories of a robot are derived from a set or multiple sets of training data

obtained from demonstrations of trajectories via backdriving, simulation, or replication of a human. Once these sets are obtained, they are used as input into an algorithm of choice, which extracts relevant information and outputs a trajectory that the robot is to follow. A large benefit of using the data-driven approach is that once a technique has been developed to evaluate the data, it can be used over and over again with different tasks without the need to develop complex models of the tasks to be accomplished. Of course, new training data for each task would still need to be obtained.

As with the model-driven approach, there are some drawbacks of using the data-driven approach. One issue that was faced in the research was that depending on the representation of the trajectory, it may be difficult to predict future positions of the robot. The problem of dealing with initial conditions (ICs) during runtime that were unlike those of training data must also be addressed. One hopes that the set of training data is rich enough to include the typical ICs that one may encounter, however, this is not guaranteed and nevertheless must still be handled. The last issue with the data-driven approach is that is provides little insight into the underlying physics behind a system, which are often desirable for use during the process of control or within the greater context of engineering as a whole.

5.3 Data-Driven Control Scheme

For this work, a data-driven approach to learning tasks related to aircraft manufacturing was taken. The scheme for controlling the SRL or any other robot during an arbitrary task using the data-driven approach is summarized in Figure 5-1. Because the robot must be coordinated with the human, desired trajectories for both the human and the robot must be found.

The process of trajectory generation is denoted by the black dot-dashed line. During the offline portion of the trajectory generation, three main steps occur:

1. Trial data for the desired task to be learned is generated for both the human and the robot. Here, the desired path of the robot is shown to the robot via

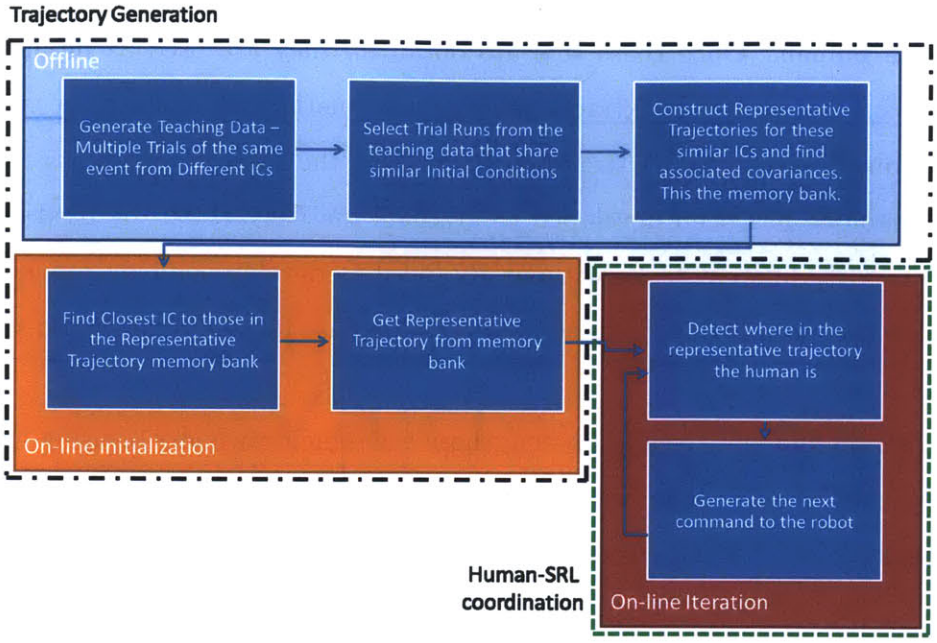

Figure 5-1: Controlling the SRL using a data-driven approach to task modeling.

backdriving it and recording the positions, obtaining positions via a simulation, or learning from the positions of another human. The desired path for the human is determined via whatever sensing mechanism is appropriate for the given task. It is beneficial if the training set obtained from this step has a variety of likely ICs because the task may have different trajectories depending on different starting conditions. The total data obtained during this stage is considered the "*teaching*" or "*training*" set.

2. Trials from the training set are grouped together based on the similarity of their initial conditions.

3. Representative trajectories are generated for each of the initial condition clusters determined in the previous step and are stored in memory.

Once the memory bank of trajectories has been created offline, the appropriate trajectory can be chosen on-line in a process defined by the next two steps:

1. The initial conditions of both the human and the robot are sensed in real time and the closest initial conditions from the memory bank for each are found.

2. The representative trajectory that corresponds to the IC that best matches the human and the robots' current ICs are selected from the memory bank.

Once the human and robot's desired trajectories are chosen, the process of human-robot coordination in real time can begin. This is denoted by the green dashed line. Here, it is abstracted into two extremely simple steps, although it will be elaborated in great detail in the next chapter.

1. The position along the trajectory of the human along his desired trajectory is determined.

2. Based on this human position, the desired corresponding position along the robot's trajectory is determined and is given as a command to the robot.

5.4 Task Description

5.4.1 Drilling Task

The process of generating single trajectories via trial data was attempted in the context of the wiring task, which is the affixing of wires to control boxes in the aircraft. This is an essential task because there are hundreds of miles of wires to be strung across the aircraft [32]. If the operator can hold the wires while the SRL affixes them, the operator's efficiency will be increased. In order to teach the robot its trajectories, the drilling task was completed with two humans - one assuming the role of the leader, and the other the follower. This data was processed in the manner described above, where the data obtained from the follower human is used as the teaching data for the robot. The two-person wiring process is summarized in Figure 5-2 and is explained below.

There are six main steps in the two-person wiring process:

1. The first human grabs two wires to be affixed to the wiring control box. He then approaches the control box with the wires in hand.

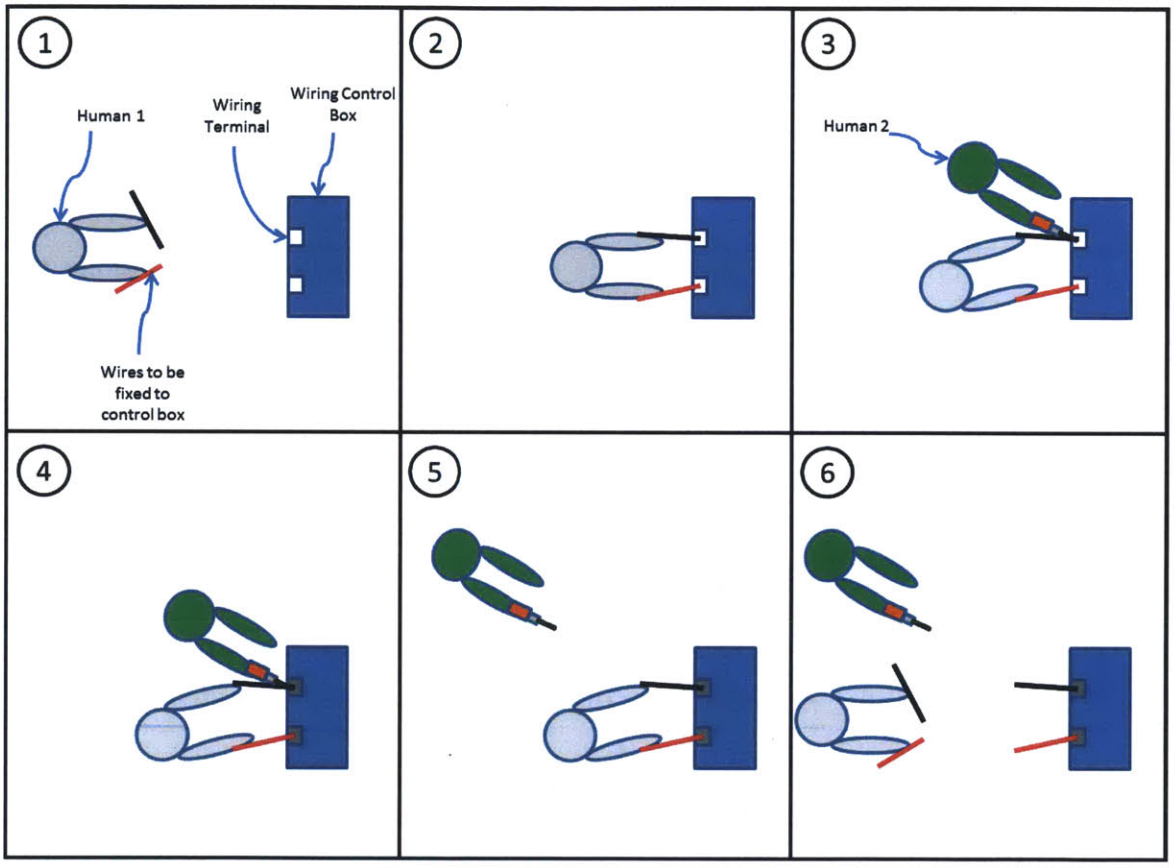

Figure 5-2: Summary of the two-person drilling task used to train the SRL trajectories. Step 1 begins with human 1 obtaining wires. Step 2 sees human 1 holding wires up to the control box terminals. Step 3 sees human 2 approaching the wires held by human 1. Step 4 is the affixing of the wires. Step 5 is the clearing of the workspace by human 2. Step 6 sees human 1 retrieving more wires to be affixed.

2. Once at the control box, human 1 holds the wires at the appropriate wiring terminals and waits for a predetermined amount of time to signal that the wires are in place.

3. Once the wires are in place, the second human comes in with an automatic screwdriver.

4. Via the screwing in of the terminals, human 2 affixes the wires to the control box.

5. At this point, the second human clears himself from the area near the control box in order to clear up the workspace for human one.

6. Free now to move about, the first human grabs more wires to be affixed to the control box and repeats steps (1)-(5) until all wires are affixed.

5.4.2 Simulation

As a first pass at obtaining the trajectories, a simulation of the two person wiring task was created to allow the author full authority over the relationship between the first human and the second human. The parameters of interest in the relationship between the two humans were the X,Y, and Z coordinates and the roll, pitch, and yaw values of both the wires and the drill. Because this is a trajectory, time is obviously important. A data set containing ten trials starting at random initial conditions and a sampling frequency of 100Hz was generated. This simulation is summarized in Figure 5-3.

5.5 Analysis of the Drilling Task

Once a training set of data from the wiring task was obtained, the trajectories of the drill and the human could then be extracted. The method of trajectory extraction is explained in the remaining sections of this chapter.

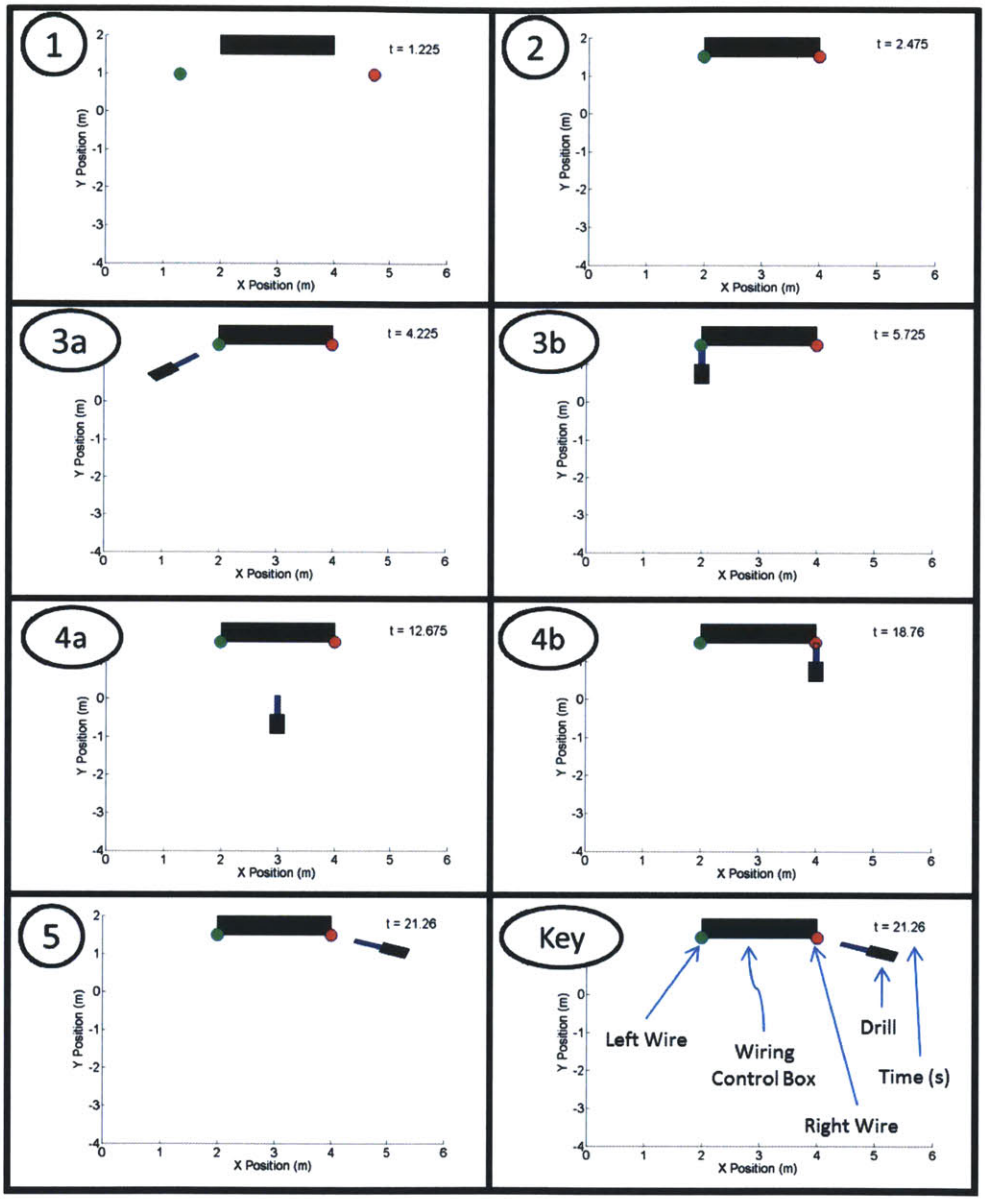

Figure 5-3: Simulation of the wiring task. The pictures show the movement of the left and right wires held by human one and the drill held by human two. The control box (shown in black) is also in the picture. The key in the bottom right square explains the items seen throughout the figure. The numbers at the top-left hand corner of each square indicate each stage of the wiring task, as explained in Figure 5-2. Human 2 fixes both the left and right wires of human 1, hence the steps (3) and (4) (approaching a wiring with a drill and affixing it) happen twice.

5.5.1 Introduction to Partial Least Squares Regression

The technique chosen to extract the trajectories is Partial Least Squares Regression (PLSR). Developed originally as an economic tool, PLSR relates data in two matrices with a linear multivariate model, while capturing their structure. It can accomplish this even in the presence of noisy and correlated data, which separates it from traditional multiple linear regression (MLR).

Let the first matrix, X, be the input space of predictors, while the output space of responses is called Y. The whole goal of this process is to predict Y from X while evaluating the structural similarities between the two matrices. It does this by calculating latent vectors that decompose X and Y while at the same time ensuring these latent vectors explain the maximum possible covariance between X and Y. It assumes that the data has the underlying structure

$$X = TP^T + E \tag{5.1}$$

$$Y = UQ^T + F \tag{5.2}$$

,where X and Y are the predictor and output matrices as previously explained, T and U are the inferred latent variables, P and C are the loadings or weights of the latent variables, and E and F are the residual matrices, which are the parts of X and Y that the latent variables cannot explain. In one way or another, PLSR algorithms work to minimize these residual matrices.

Geometrically, the PLSR is a reduction of the original data of the X space into a lower dimensional hyperplane as defined by loadings. These loadings associated with the latent variables also inform one about how strongly correlated the X and Y are. The latent variable T contains the projections of the X space into the lower-dimensional hyperplane. The positions of these projections are strongly related to the values of matrix Y [33, 34].

There are a number of different algorithms that can be used in order to calculate the latent variables and the weights. $PLS1$ attempts to find T, which is assumed to

be othonormal. One benefit of this particular algorithm is that X and Y do not need to be centered prior to using, which is convenient for prediction later on. Despite its widespread use, the NIPALS algorithm will be used due to its prevalence in statistical packages. The basic steps of this algorithm are:

1. Estimate the weights of X.
2. Estimate the latent variable T.
3. Estimate the weights of Y.
4. Estimate the latent variable U.

Further details of this algorithm can be found in [33].

5.5.2 Adapting Partial Least Squares Regression to Trajectory Data

The formulation of this algorithm assumes snapshots of data - as it stands, PLSR is not useful for estimates of time-series data. However, with modifications made to the matrices X and Y, PLSR can still be useful in the application of extracting dynamic trajectories. Spatial-temporal analysis with PLSR was first investigated by [35] in the context of fMRI analysis. In order to use it in this context, fMRIs with a fixed number of images ($1..T$) were taken. There were $1..M$ points of interest in the fMRI throughout trials $1..K$. At each observation time point t, a submatrix was constructed of the form shown in Figure 5-4. The measurements of trial k constituted a row within submatrix X_t, and the trials $1..K$ formed a matrix of size K x M in the case of the input space.

Once this matrix was constructed for each of the observation time points, the submatrices $X_1..X_t..X_T$ were then concatenated to form the single input matrix X, as shown in Figure 5-5. This process was done for both the input space X and the output space Y.

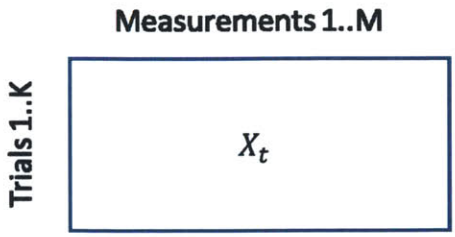

Figure 5-4: Submatrix construction for PLSR at time point t.

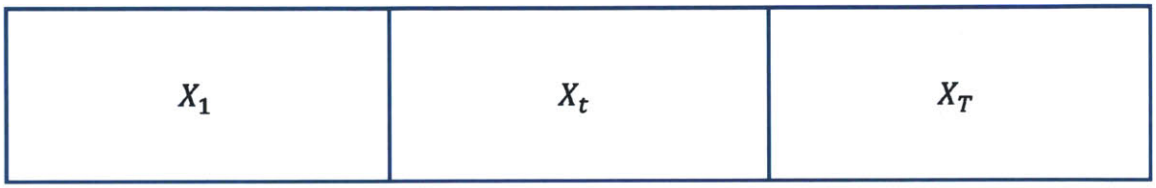

Figure 5-5: Submatrix concatenation to form input matrix X.

However useful, the conditions in which this technique was used were quite ideal. For instance, the fMRIs each had the same number of samples and they were taken at the same sample interval. In our work, it is inevitable that each trial length differ due to the varying time it takes to complete each step of the task across trials. This can be seen both as a negative and a positive - the negative being that this is further indication that humans are imperfect in their timing abilities, but the positive being that having multiple examples of this imperfection will likely make the results more generalized. In order to overcome this issue of trials of varying length, each trial was reduced down to the number of sample points in the shortest trial by down-sampling the longer trials. This had the result of creating trials that have different effective sampling rates.

Ideally, the trial would be segmented in order to allow each phase of a trial to occur in the same submatrices across trials. This could be accomplished with two approaches - the data could be segemented manually and interpolated within each segment. This is not desirable and goes against the goal of making this an automatic process. The other approach is to use a technique like Dynamic Time Warping to align the key features of the trajectories [36]. Dynamic Time Warping could also be used instead of PLSR because it has been proven to be an effective way to generate

trajectories for a data set [17]. However, the unsegmented trial data organized in the manner described at the beginning of this subsection produced surprisingly good results with the NIPALS algorithm. These results are given in the next subsection.

5.5.3 Predictive Ability for Drill Trajectory

As described previously, a data set of 10 simulated trials of the wiring task was generated. The trial data was rearranged in the manner described in the previous section and put into the PLSR implementation within MATLAB® . Once the PLSR representation was established, a test trajectory of the wires was provided, and the resultant drill positions were calculated. There are a number of qualities used to evaluate the effectiveness of this representation. The first is that for the representation to useful, it must capture the goal positions well; because the accuracy with which the goal positions must be approached with is extremely high, the ability to capture their exact position is imperative. Second, we hoped that PLSR would capture the "triggering" relationship between the wires and the drill; this relationship shown during the trials and is defined by the drill not approaching the wiring control box until after the wires were held at their goal positions for a specified amount of time.

Indeed, both these important relationships were found. In Figure 5-6, the desired left and right drill position coordinates are shown and compared to the outputted coordinates of the PLSR fit. In the trials, the drill did not approach the wires until one second after they were affixed. This can be clearly seen as the non-motion exhibited in the fitted trajectories. To give a better sense of how well the reconstructed trajectories reached the import points, the reconstructed trajectory of the drill is plotted and compared to the actual trajectory of the drill in that particular trial in Figure 5-7. For the most part, the trajectories are quite well matched. This is especially apparent in the desired and reconstructed trajectories matching almost exactly at the the points representing the positions of the fixing of the wires to the control box.

The major problem with this approach as it stands is that the drill trajectories can only be constructed once the entire trajectory of the hands has been generated. This is obviously not useful at all when attempting to determine the trajectories in real

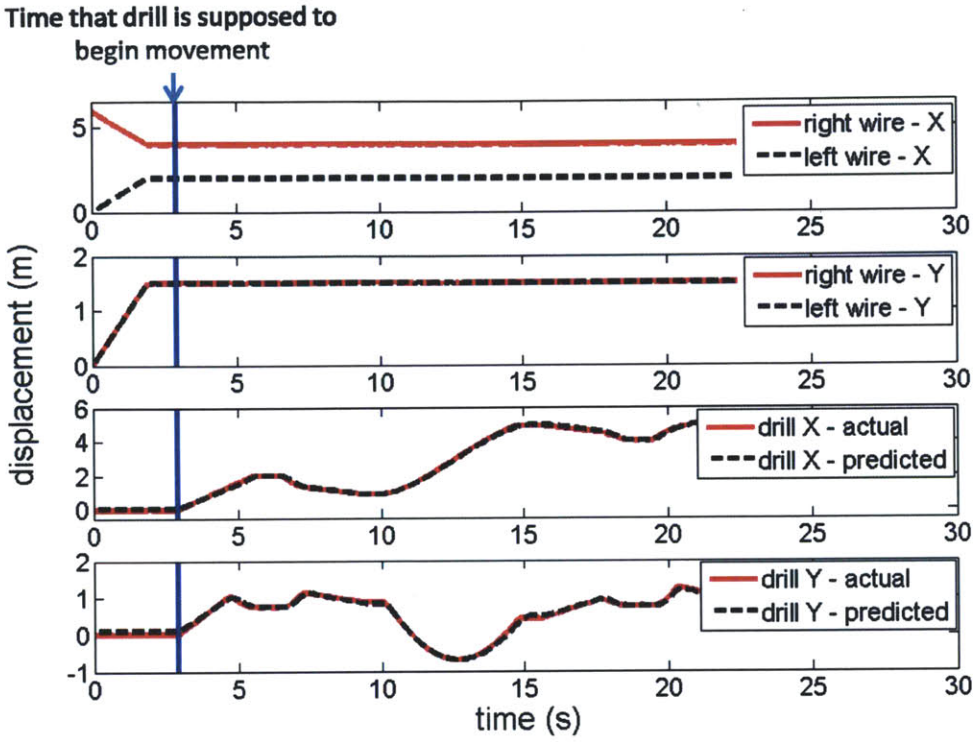

Figure 5-6: Individual reconstructed vs. actual drill angles for wiring task.

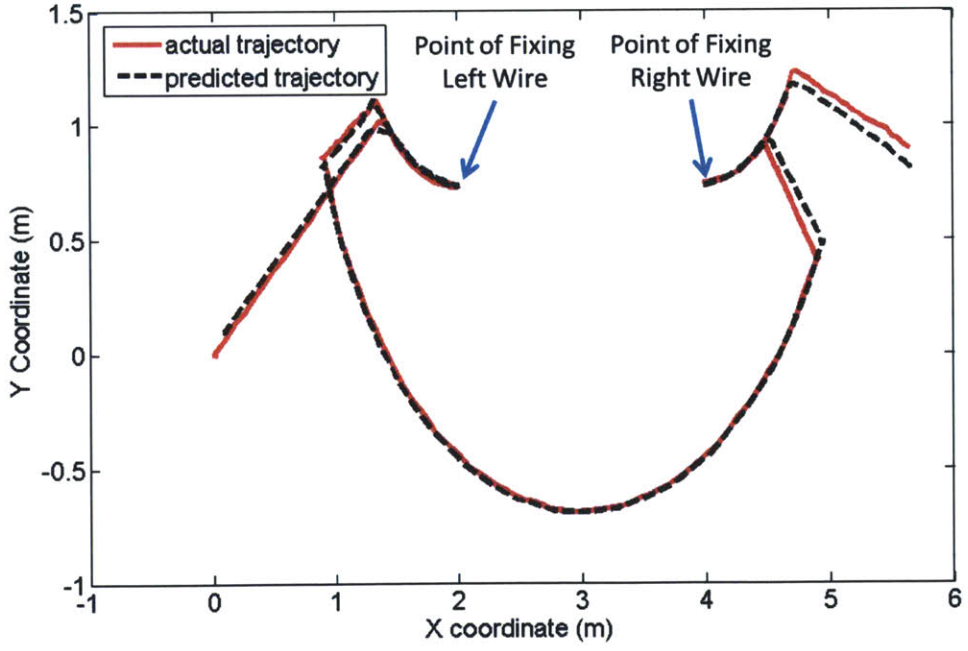

Figure 5-7: Reconstructed vs. actual drill trajectory for wiring task.

time. However, not all is lost here. The method of created the trajectory bank can still be used to generate the sample trajectories offline. However, online recognition will have to be done in some other manner. For this real time coordination, the representation of regression trees is used to determine where along a precalculated trajectory the robot is supposed to go based off the position of the human.

Chapter 6

Human-Robot Coordination

This chapter explains the process of how human-robot coordination can be derived for any robot once a trajectory has been generated in the manner described in the previous chapter, especially within the context coordinating the wiring task with the SRL. Like with the trajectory learning, here the goal is to teach the robot coordination with the human by generating training data with two humans, then transfering the role of one of the humans over to the robot.

6.1 Defining a Simple Relationship

As a first pass, the wiring task as described in the previous chapter seems a bit too complicated to tackle. This is because it has a substantial number of degrees of freedom. Before this particular task is tackled, it should be verified that a relationship within a one degree of freedom task can be captured accurately and in a representation that is of use for the real-time coordination of a robot with a human. It would be of particular interest if in the qualitative sense this task well captures the relationship that is expected to exist between human one and human two during the wiring task teaching phase.

To this end, we proposed an extremely simplified leader-follower task as depicted in Figure 6-1. There are two humans - one who takes the role of the leader, and another who takes the role of a follower. Both the leader and the follower turn a

handle attached to a rotating shaft from an initial position to a goal position. The leader moves as he wants from his random initial angle condition towards a pre-set goal angle. The follower is to then go to his goal position from his random initial angle condition by reacting to the leaders actions in a very well-defined manner. The one degree of freedom for each human in this task is the angular distance between his current position and his goal position.

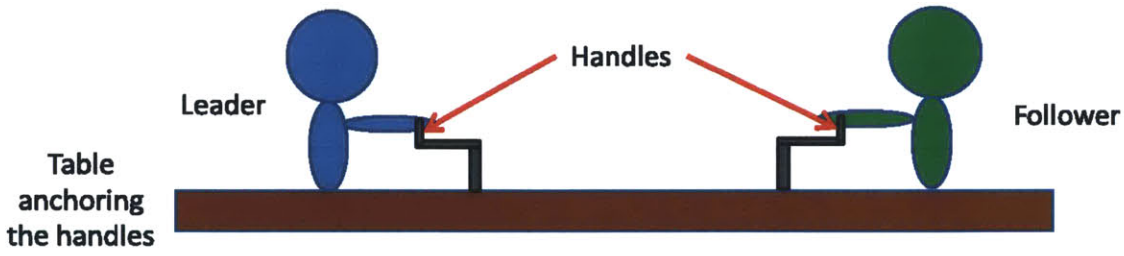

Figure 6-1: Illustration of simplified leader-follower task to be learned.

The set of rules that relate the reaction of the follower human to the actions of the leader human are summarized in Figure 6-2, as well as listed below.

1. In general, the follower should approach his goal angle as long as the leader human is either approaching his own goal or at his goal angle. In the context of the wiring task, this simulates the simultaneous approach of the leader human carrying his wires to their goal positions and the follower human carrying the drill towards the wire terminals to affix the wires.

2. If the leader human begins oscillating his angular distance away from his goal position, the follower should pause at his current angle. In the context of the wiring task, this simulates the follower human being unable to infer the intentions of the leader human. If the follower human or robot in a real scenario cannot infer the intentions of the leader human, it could be desirable for the follower to pause his current actions. This rule simulates such a response.

3. If the leader human is consistently moving away from his goal angle, the follower human should start moving away from his goal angle as well. In the context of the wiring task, this simulates the follower human clearing up the workspace

of the leader human once the follower human has determined that the leader human wants to move away from the wiring control box.

4. If the leader human is at his goal angle, the follower human should approach his respective goal angle quickly. In the context of the wiring task, this simulates the follower human attempting to be as efficient with his time as possible by quickly approaching the terminals with the drill once the wires are in place.

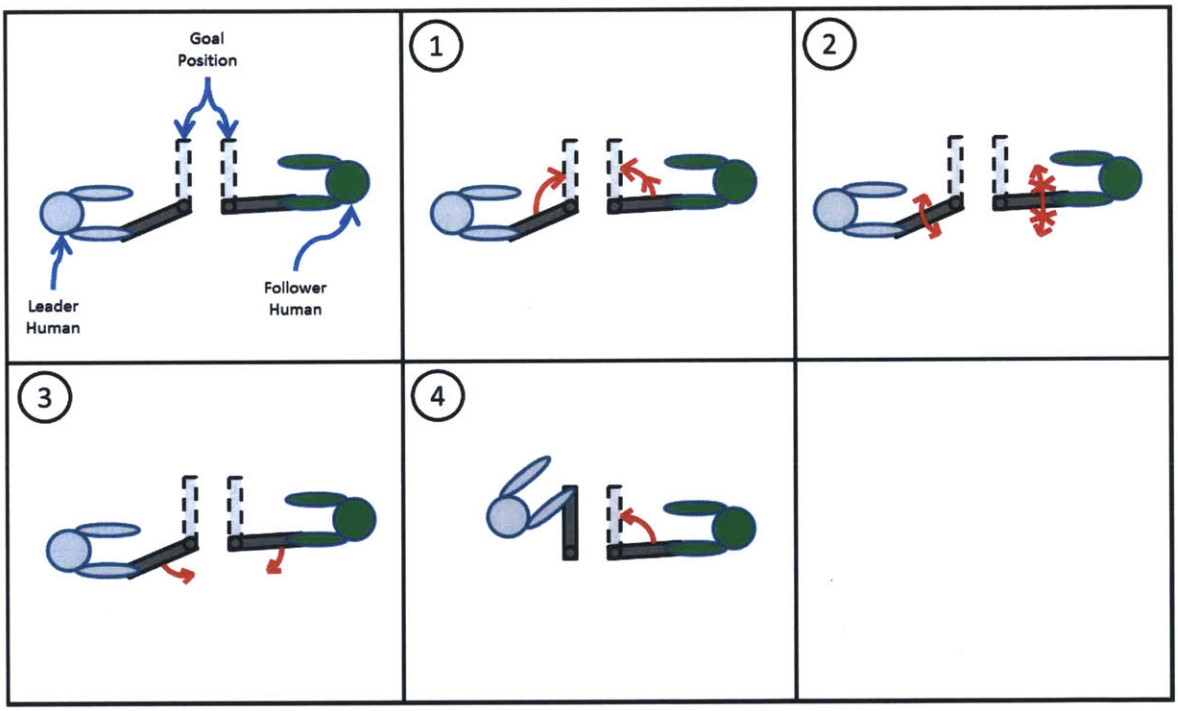

Figure 6-2: Rules describing the coordination between the leader and follower humans in the simplified coordination task. The numbered frames provide an illustration of their respectively described rule.

6.2 Experimental Setup

To capture the angle of each human while he turns a handle, an extremely simple testbed was made.

6.2.1 Design

The device used to capture the human coordination during the simplified task is shown in Figure 6-3. The main component of the device is an absolute magnetic encoder with PWM output (US Digital, model MAE3), which captured the angles of the handle. An aluminum shaft was mated to the handle, and at the encoder end of the shaft was the magnet that lay inside the encoder. In order to prevent play in the aluminum shaft, which could potentially add noise the the encoder readings, two bearings were used. The encoder was given enough room for its cables to reach it by using 2" standoffs. In order to use the device in an accurate and repeatable way, it was held to the desk via c-clamps. However, using these clamps could potentially interfere with the motion of the handle. To provide extra clearance for these c-clamps, threaded rods separated the plates containing the bearings for the aluminum shaft. Lastly, a standoff was attached to the top plate in order to mark the goal position of the handle.

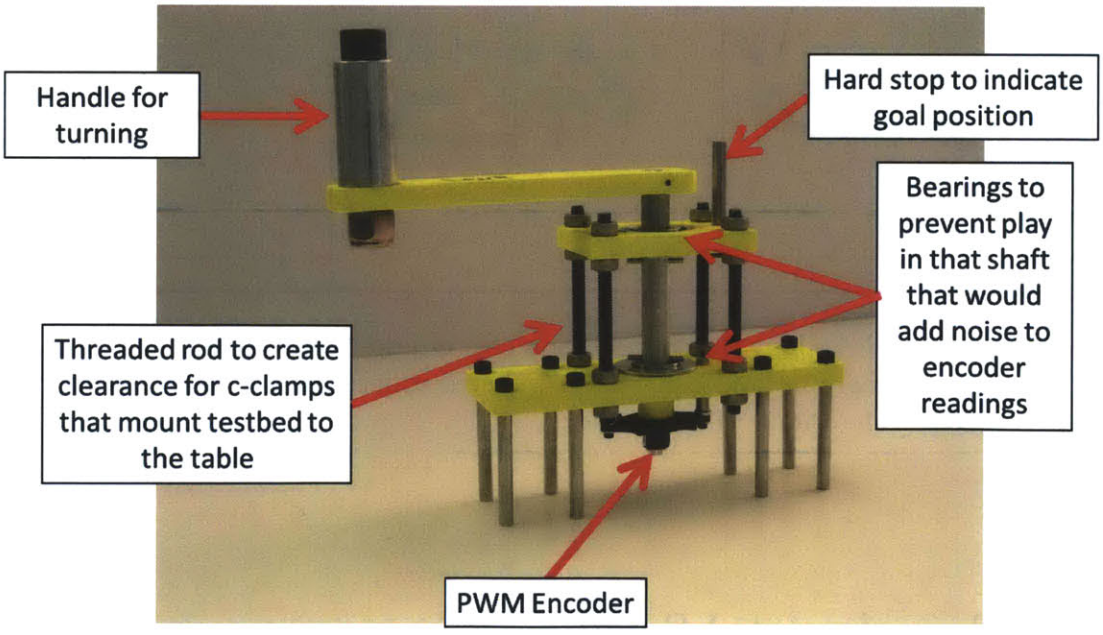

Figure 6-3: Testbed used to capture the human coordination exhibited during the simplified task.

6.2.2 Data Acquisition

In order to process the data output of the PWM encoder, a data acquisition device was necessary. A schematic of the setup of this device is shown in Figure 6-4. The PWM signal of the encoder was captured by a digital input card (National Instruments 9403) that took up a slot in a National Instruments compactRio device (NI cRIO-9047). The signal was sampled at 100Hz. The cRIO device communicated with the computer via an ethernet chord. A computer running Labview version 12 controlled the sampling process.

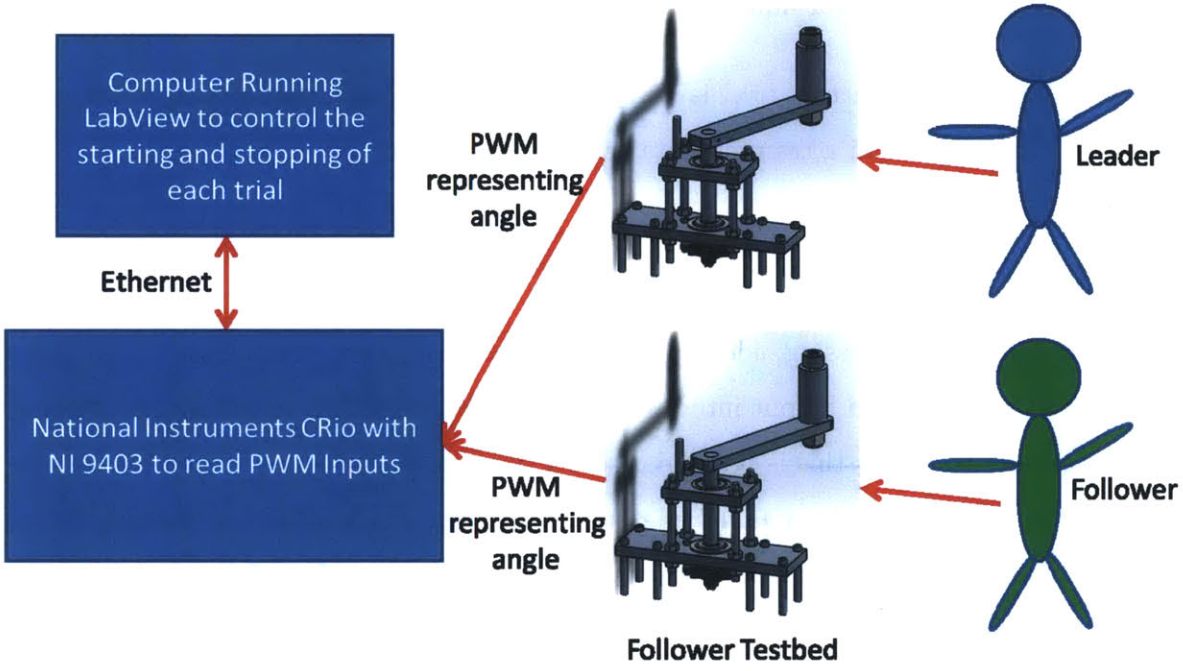

Figure 6-4: Data acquisition setup used to capture the human coordination exhibited during the simplified task.

6.2.3 Trial Process

Obtaining the training data set with the two humans was a five-step process, which is explained below.

1. The leader human and the follower human move their handles to random initial distances away from the pre-set goal position. The leader human picks his angle

independently from the follower human.

2. The leader human begins his approach towards his goal position. This initial movement of the leader human marks the start of each trial.

3. The leader human and the follower human move towards their respective goal positions. The leader human approaches his goal as he pleases. The follower approaches his goal in a manner determined by the rules described previously.

4. When the follower human reaches his goal, the trial is considered finished.

5. The data from this trial is saved. Steps (1) - (4) are then repeated 99 more times to obtain a training data set that consists of 100 trials.

6.2.4 Preprocessing

After the trail processing, ideally no further preprocessing would have to be done before the data could be put into a structure to be used by the coordination-learning algorithm. However, this is not the case because the digital input channel was quite noisy, which lead to many faulty measurements. The angle of the handle was calculated using information about the pulse width and pulse period of the output of the encoder. The noisy channel led to many false readings in both the pulse width and period, which lead to either incorrect or completely nonsensical angle measurements. Because of these erroneous readings, preprocessing had to be done on the raw angle signal in a process described below and summarized in Figure 6-5.

1. The calculations involving the PWM encoder are only supposed to produce angles between 0 and 360. The first step was therefore discarding angles that were outside this range. Also, any angle found that was a result of using an unreasonable pulse width or pulse period was also discarded.

2. Given the very high sampling frequency (100Hz), the angles between sample points should not change much. Therefore, angles that were not discarded as

a result of the initial filtering were put through a low-pass filter to remove high-frequency angle noise.

3. After discarding the non-negligible number of error-containing angles, there were many points missing in the data. Additionally, the sampling period was inconsistent, which was obviously not desirable. Therefore, the angle data was interpolated at a fixed interval to mimic consistent sampling. The data was downsampled to 10Hz.

4. The output of the encoder only gave absolute angles. However, the degree of freedom of interest was not the arbitrary angle of the handle but the angular distance of the handle of its current position to its goal position. The goal position of both the leader and the follower were therefore normalized to 0, which led to the angle measurements corresponding to the angular distance instead of their own arbitrary angles.

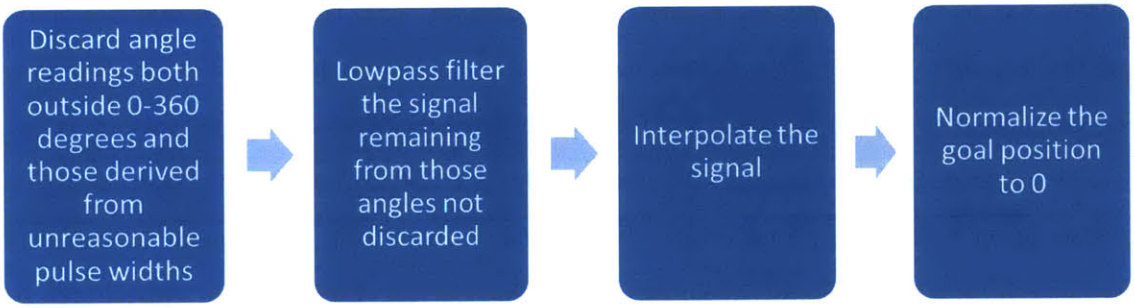

Figure 6-5: Preprocessing and filtering of the raw single output of the PWM encoder.

6.3 Learning of the Leader/Follower Relationship

Now the that training data had been collected, it was time to extract the relationship between the leader and the follower. However, before choosing or developing an algorithm to collect it, it would be quite helpful to first determine an underlying relationship structure or model that the follower human followed.

6.3.1 Defining the Leader/Follower Relationship

Considering that the follower human's actions were a result of first inferring whether the leader human was approaching his goal, it becomes clear that the follower human had to keep track of the leader's prior positions. As a guess, it is likely that the follower human also based his actions off his own prior positions. For this reason, we assume that the position of the follower human at time point t follows an autoregressive model of the form:

$$\begin{aligned} f(t) = g_{abs}[&f(t-1), f(t-2), ..., f(t-n_b), \\ &l(t-1), l(t-2), ..., l(t-n_a)] \end{aligned} \tag{6.1}$$

where $f(t)$ is the location of the follower human at time t, $l(t)$ is the location of the leader human at time t, and $g_{abs}[..]$ is the function describing the location of the follower human using the absolute angle inputs. It is assumed that we are interested in the n_a previous positions of the leader, and the n_b previous positions of the follower. What these values are is not known ahead of time, and they can be considered a design choice when formulating the input space of the training data, which is described in the next section. Although this formulation makes intuitive sense, upon further examination the differential change in angles between time points is more important that the absolute angles themselves. Equation 6.1 can therefore be reformulated to

$$\begin{aligned} \Delta f(t) = g_{diff}[&\Delta f(t-1), \Delta f(t-2), ..., \Delta f(t-n_b), \\ &\Delta l(t-1), \Delta l(t-2), ..., \Delta l(t-n_a)] \end{aligned} \tag{6.2}$$

where $\Delta f(t)$ is the change in the angle of the follower from time $t-1$ to t , $\Delta l(t)$ is the change in the angle of the leader from time $t-1$ to t, and $g_{diff}[..]$ is the function describing the change in angle of the follower human using the differential angle inputs. However, if a triggering relationship is to be found, that is, one where

the follower cannot move unless the leader has passed a particular threshold, then Equations 6.1 & 6.2 can be refined further to include absolute and differential angle information:

$$\begin{aligned}\Delta f(t) = g_{mix}[&f(t-1), f(t-2), ..., f(t-n_b),\\ &l(t-1), l(t-2), ..., l(t-n_a),\\ &\Delta f(t-1), \Delta f(t-2), ..., \Delta f(t-n_b),\\ &\Delta l(t-1), \Delta l(t-2), ..., \Delta l(t-n_a)]\end{aligned} \tag{6.3}$$

where $f(t)$, $l(t)$, $\Delta f(t)$, and $\Delta l(t)$ all are as described previously, and $g_{mix}[..]$ is the function describing the change in angle of the follower human using the combination of differential and absolute angle as inputs. In Equation 6.3, it is of interest as to why $\Delta f(t)$ was chosen to be on the left side of the equation instead of $f(t)$. Certainly $f(t)$ can be chosen instead, but as stated before, how the angle changes between time points is of greater interest than its absolute angle counterpart.

6.3.2 Idea of Regression/Classification Trees

Now that a general model of the relationship between the follower and the leader humans has been established, a representation that well captures the underlying nature of the relationship needs to be selected. Here it is important to reemphasize the very conditional nature of how the follower human selects his action. Because of this conditional nature, regression trees are chosen to represent the relationship. The concept of using classification/regression trees to relate an output to an input space is shown in Figure 6-6.

The purpose of the classification/regression tree is to predict the values of a target variable from several values of an input variable. It does this with a tree-like structure. Conceptually, the input space is partitioned with binary splits and each partition is assigned an output value. Each leaf of the tree represents the value of a target variable given the values of the chosen input variables, found by traversing the tree from the

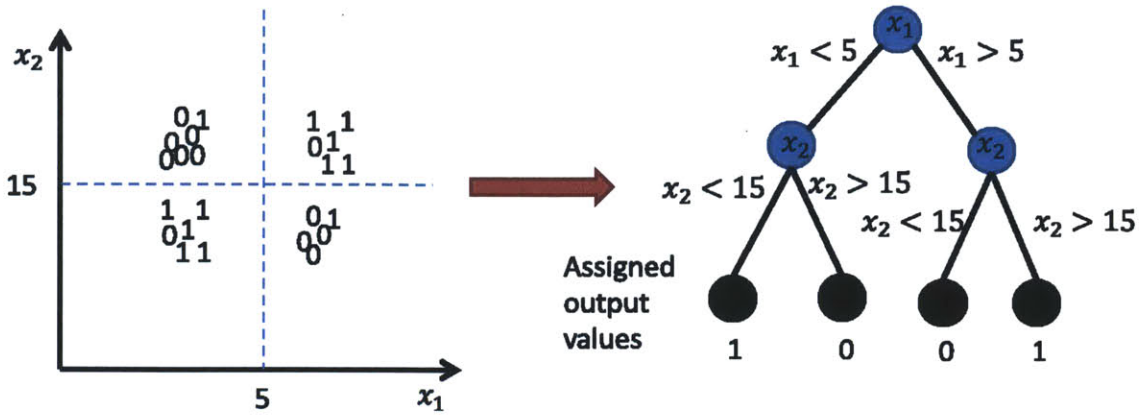

Figure 6-6: A simple example of using classification trees to map an output space to an input space. The data set with an $\Re^2$ input space and a binary (0/1) outputs is shown on the left. The classification tree representation of this data is shown on the right.

root to the leaf. For instance, if the variable x_1 takes a value of 1 and x_2 takes a value of 1, then the output from the tree is 1 because x_1 is < 5 and x_2 is < 15. Similarly, if x_1 takes a value of 10 and x_2 takes a value of 5, then the output from the tree is 0 because x_1 is > 5 and x_2 is < 15. In the left side of the figure, the dashed blue lines in the input space represent the tests at the nodes and partitions.

It should be noted that many classification/regression trees can be generated for a given data set. There is nothing preventing one from creating a regression tree for the data in the figure that first checks the value of x_2 and then checks the values of x_1. It should also be noted that not all variables need to be partitioned. An even simpler tree could simply choose output values based on the value of x_1. Also, multiple splits can occur along a single variable. For instance, the tree can first check if x_1 is < 5, then underneath its subbranch test whether x_1 is < 2.

Lastly, it should be noted that the example in Figure 6-6 is technically classification tree. This is because the outputs can take on *classifications*, rather than real numbers. However, a *regression* tree only differs in that the output takes on numerical values. The tree structure for the two is exactly the same. The only other difference between the two is that the algorithms for finding each tree take slightly different approaches.

6.3.3 Algorithms for Finding Regression/Classification Trees

The problem of finding the optimal classification/regression tree is an NP-complete problem - i.e., no efficient solution to this problem is known. However, there have been a number of algorithms that take a greedy approach to this problem. One widely used algorithm used for classification/regression tree building is the CART algorithm developed by Breiman et al [37], which serves at inspiration for this work. In order to build the regression tree from the input data for this problem, a variant of the CART algorithm was implemented in MATLAB® . The stages of this algorithm are as follows:

1. All input data is considered. For each variable, examine a split at all points as defined by a particular resolution. Looking at Figure 6-6 for example, assume that we choose a resolution of 0.1. Then the splits considered are at 0, 0.1, ..., 10 for x_1 and 0, 0.1, ..., 20 for x_2, assuming the largest values in the input data for x_1 and x_2 are 10 and 20, respectively. The resolution used for this implementation was 0.01 .

2. Possible outputs are tested for each split. The split whose output results in the lowest mean squared error (MSE) between the actual outputs and the assigned possible outputs is selected.

3. The selected split is applied to the data. In this way, the input space is partitioned.

4. This process is repeated recursively for the two children notes that are created as a result of the split.

5. Splitting is stopped when the MSE of the current tree drops below a threshold and the regression tree is returned.

There are a number of tradeoffs in deciding the threshold balance. The first is having it small enough that it maps the input well. However, if it is too small, the tree can have a very large depth due to a likely increased number of splits. A

tree with an extremely large depth is undesirable for two reasons. The first is that it is hard to implement in a controller if the splits must be programmed manually. The second is that the length of time in which an output is obtained from a tree is directly proportional to the depth of the tree. This becomes important when the tree is traversed in real time to obtain commands for the robot. Therefore, the MSE must be balanced in such a way to both well-map the input data and run quickly in real time. The exact balance obviously depends on the real-time needs of the application. One thing that is not explored is the biasing of particular variables in the splitting. For instance, more splits across a more relevant variable may be desirable. Even without taking this information into account, the performance of this algorithm was still acceptable.

The algorithm was structured to take data in a similar fashion to the majority of learning algorithms. This structure is explained in the following section.

6.3.4 Structuring of Input Data

Before the input data for the regression trees is structured, the underlying model model must be chosen between Equations 6.1, 6.2, and 6.3. During the generation of the first set of training data via human trials, a thresholding condition of the movement of the follower human was not imposed. Therefore, the underlying model proposed in equation 6.2 was chosen. This representation has the benefit of having a smaller input space of only differential angles when compared to the input space of both differential and absolute angles. This is important for two reasons. The first is in learning the tree. The number of splits performed during the building of the tree increases with each variable, and makes the building of the tree take longer. Additionally, more splits can potentially lead to a tree with larger depth, which as described before has ramifications for real-time performance.

Typically, the process of learning the tree structure that represents a set of training data takes m observations of a variable Y that is caused by p predictor variables, x_1, ..., x_p. The inputs to most algorithms are matrices X and Y. X is a matrix of size m x p, whose columns are the predictor variables and the rows are the observations.

Y is a column vector of size m x 1. This is illustrated in Figure 6-7.

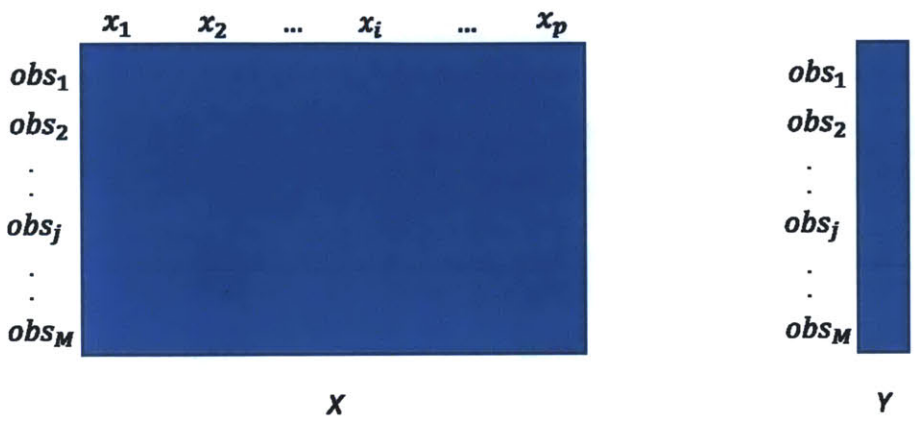

Figure 6-7: Typical training data structure for learning regression trees.

Because in our particular example we are concerned with predicting the follower angle from previous differential follower and leader angles, we choose the predictor variables to be $\Delta f(t-1)$, ..., $\Delta f(t-n_b)$, $\Delta l(t-1)$, ..., $\Delta l(t-n_a)$, and the predicted values to be $\Delta f(t)$. It is assumed that for each trial, each sample point constitutes an observation an observation as defined previously. For a particular trial X_n, there are n_K observations. Therefore, the observation matrix X_n is a n_K x $(n_a + n_b)$ matrix, while the output matrix Y_n is a n_K x 1 column vector. This is illustrated in Figure 6-8.

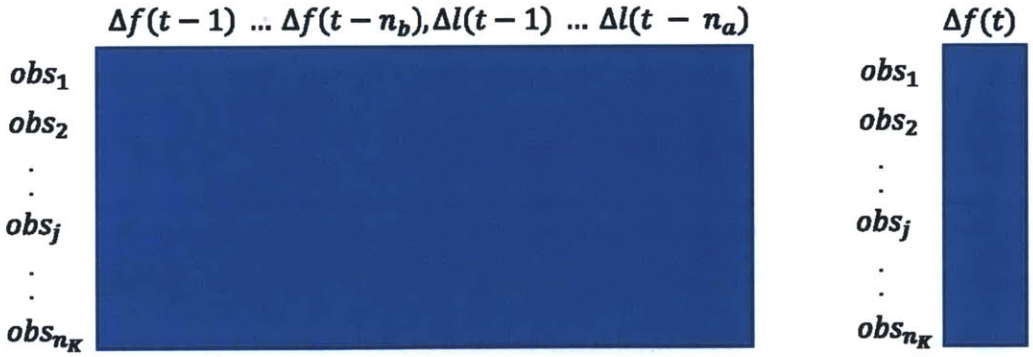

Figure 6-8: Training data organization of a single trial for use in regression tree learning algorithm.

Because the leader and the follower share the same relationship both between

each trial during the process of training data generation and across each time slice within a given trial, the individual trial input matrices X_n for each trial can be vertically concatenated with the other input matrices to form a single input matrix: $X = X_1, ..., X_n$. The same is dowe for the output matrices to form a single output matrix: $Y = Y_1, ..., Y_n$. This is illustrated in Figure 6-9.

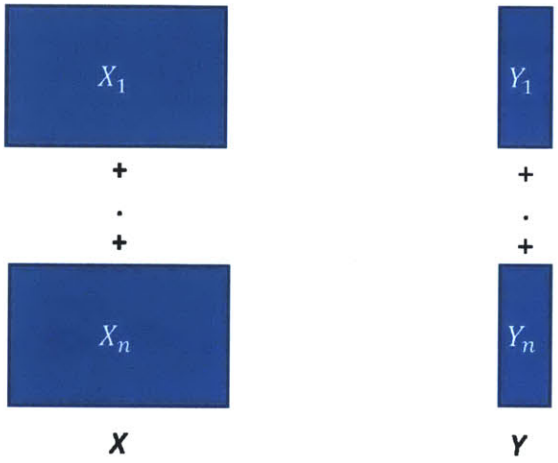

Figure 6-9: Concatenation of individual input matrices X_n and Y_n to form regression tree algorithm inputs X and Y.

The data set generated through the 100 trials as described above was arranged in the manner just described to allow for use with the regression tree algorithm.

6.3.5 Results of Using Regression Trees for Prediction in Simplified Follower Task

With the objective of using the regression tree output as commands for an actual system, the stopping criterion of the regression tree was chosen to create a tree of depth 20 when using the training data set from the 100 trials. In order to validate the use of regression trees in predicting the change of the follower angle at time t from previous follower and leader angle changes, the training data segment was divided into two sets - one used for training, and one used for validation (50 for training, 50 for validation).

An example reconstruction of the follower angle based on real-time regression tree-based calculations is shown in Figure 6-10. The upper graph shows the differential follower human angle output from the regression tree. The bottom graph shows the reconstruction of the follower angle based on the output of the regression tree at each time point. For reference, the differential and absolute angle of the leader is shown in each graph.

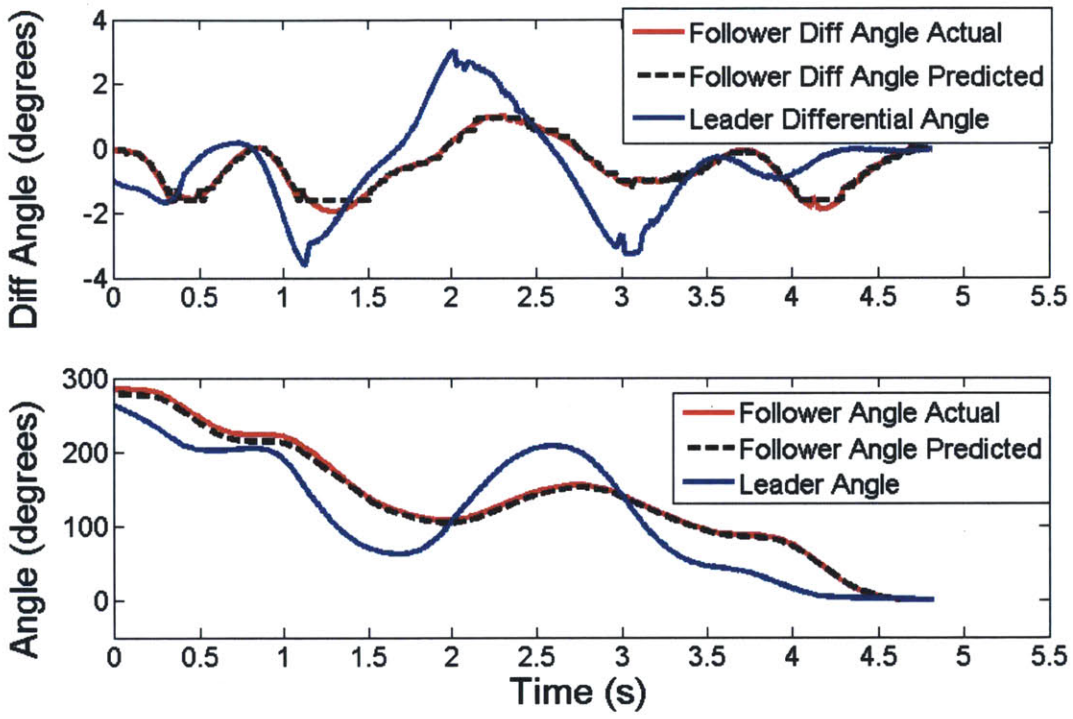

Figure 6-10: Follower angle outputs calculated by the learned regression tree.

The graph shows the success of the algorithm in capturing the leader-follower relationship of the simplified task. Although the differential output has occasional spikes that do not well capture the actual differential angles, the reconstructed angle still very closely follows the actual angle of the follower human during the trial. Because of the imperfect following of the rules by the follower human, whether the angle adheres to these rules is difficult to tell. What's important here, however, is the qualitative response of the follower human. If we assume that the follower human followed these rules in a relatively close manner, then the relationship should

be captured. Upon a qualitative inspection of the figure, all of the rules previously specified are followed. This gives hope in capturing leader-follower relationships of more complicated tasks, whose learning can potentially be the subject of future work.

6.4 Implementation of Leader/Follower Relationship of SRL-Mimicking Robot

Given that the leader-follower relationship could be captured in the one-dimension case, transferring this follower behavior a robot was attempted for use in the wiring task. In order to do this, the wiring task had to be reduced to a single dimension, which was considered the distance of a component away from its goal. Taking the place of the leader human was the position of a wire, which started arbitrarily far away from a wire control box and approached in a linear fashion. Taking the place of the follower human was a miniature robot that traveled from its arbitrary initial condition to a wire terminal in order to simulate the affixing of the wire to its terminal. It did so in a piece-wise linear trajectory. The regression tree, which dictated the motion of the robot across its trajectory based on the position of the wire, was scaled to match both the sampling rate and magnitudes of distances that were relevant given the wiring task.

The task simulated the affixing of three wires to a wiring control box. Each wire had a random starting position and a predetermined goal position. The equipment setup used in the implementation of the wiring task is shown in Figure 6-11. An Opti-Track motion capture system consisting of four high-speed cameras was used to track the wires, which were outfitted with markers that identified the wires uniquely. The cameras ran at a speed of 100Hz and tracked objects of interest in six degrees of freedom. The robot used was a Denso Robot (Denso Robotics Teaching Edition), which is a seven degree of freedom robot. The Denso robot had some less than ideal performance characteristics - its joint angle commands were open loop because that lack of sensory information at the joints, and due to proprietary communication

protocol constraints, it could only be run at 1 Hz.

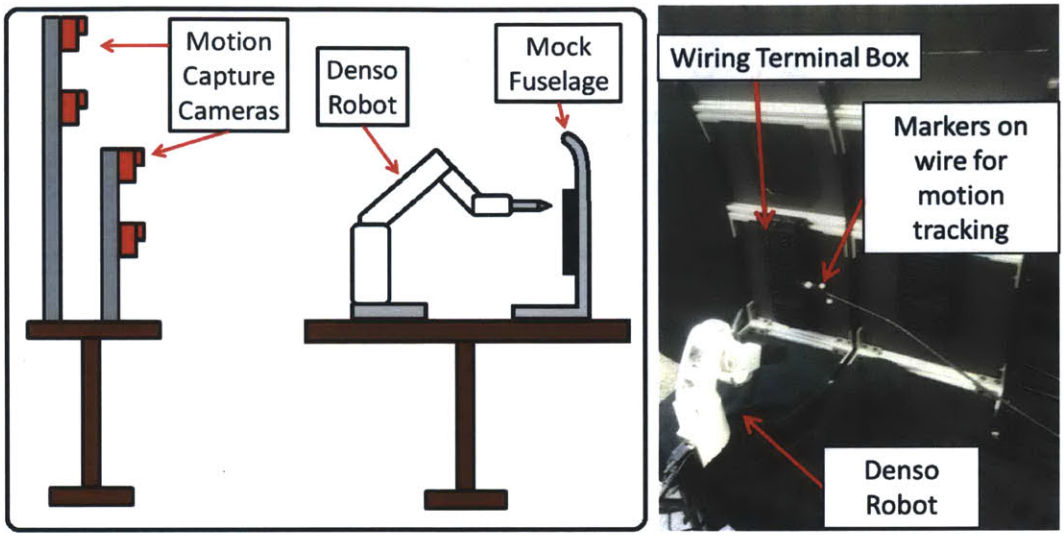

Figure 6-11: Setup for the implementation of the wiring task completed with human-robot coordination.

Despite these glaring drawbacks, the coordination task was still accomplished in a fluid and cohesive manner. The robot followed the motions of the wires quite well. As the wire approached the wiring terminal, the robot approached the terminal as well. If the wire was moving away from the control box, the robot went back to its initial condition. If there was a substantial amount of noise in the distance of a wire away from its goal position, then the robot paused at its current position. This behavior is the desired behavior seen in the simplified coordination task as defined before. The only noticeable downside of the implementation set-up was an occasional lag in responding to a wire switching from approaching a goal to moving away from a goal and vice versa. This is almost certainly due to the poor sampling rate, since the robot still travels forward between the sampling intervals. Such a result is very promising because it implies that a very simple one-dimensional relationship can be captured from two human demonstrators and transferred to a robot that coordinates with a human in six dimensions.

Chapter 7

Conclusion

In this work, the concept of the Supernumerary Robotic Limbs (SRL) was introduced. This newly developed SRL is a set of extra limbs in the form of a wearable robot. The initial goal of this robot is to assist a human in aircraft manufacturing and assembly tasks. One can envision the SRL being used to assist a human operator in a slew of tasks, such as the wiring of a control box, bearing the load of a window to be affixed to a fuselage, and the vacuuming of extremely hazardous carbon fiber particles that are a result of drilling. Outside of aircraft assembly, one can envision the SRL being used in the operating room, in space, or other manufacturing contexts. The SRL promises to assist the human in at least two large ways. The first is to augment his workspace. The second is to assist in load bearing to ultimately reduce the workload imposed on the human. These additional limbs were realized in the form of two, three degree of freedom robotic arms that have the load-bearing capacity of human arms and attach to the human operator much like a hiking backpack. An element of safety was introduced to these limbs via the coupling of an elastic element with the output shaft of each DC brushless motor.

The biomechanics of using the SRL to assist in the static load-bearing task were then explored in depth. A test rig that mimicked the mass and geometric properties was used to evaluate the reaction forces imposed on the human by the SRL. Sensory information derived from this rig was used in a biomechanical model that related those reaction forces and the angles of major human joints to the torques at those

joints during a static load bearing task with and without assistance from an SRL-like apparatus. It was found that the raw torques were higher on the lower joints but smaller on the upper body joints when using the test rig to assist in bearing weight compared to when not using the rig. A metric to summarize the impact of imposed torques on the human was developed. Using this new metric, it was shown that the work load of the human was reduced during the static load bearing task when using an SRL-like rig.

Within the context of endpoint trajectory learning, the model-based approach and the data-driven approaches were compared in terms of utility for the SRL in its aircraft assembly setting. A data-driven approach was selected, and the process with which a robot (especially the SRL) can both empirically derive its trajectories to follow and coordinate with its human leader was proposed. The specific task of wiring a control box was chosen as an exemplar task. The endpoint trajectory was found via Partial Least Squares Regression, which represented the trajectory in a lower dimensional space. This representation well captured the important points along the trajectory but provided little insight as how the robot was to coordinate with the human along that trajectory.

Before tackling the capturing of the leader-follower relationship in the wiring task, a simplified task that captured the essence of the original task was proposed. A testbed used to capture the relationship that dictated the actions of the leader and the follower in the simplified task was created and used to generate 100 demonstrations of the task. An autoregressive model relating the differential movement of the follower human to both his past differential movements and the past differential movements of the leader was chosen to capture the underlying relationship between the humans. The representation of regression trees was chosen to predict the change in follower human position from the past differential movements of both the leader and follower, and a variant of the CART algorithm was developed to generate a regression tree from the trial data. Once it was verified that the regression tree could well predict the follower human position changes in this simple case, the full wiring task was attempted. Here, the robot took the position of the follower human and guiding its

actions based on the positions of the wires was the scaled regression tree derived from the simplified task. It was found that this generated the desired qualitative behavior for the robot; thus, the potential extension of the regression tree learning strategy to more complex tasks looks promising.

Bibliography

[1] A. Teich, A. A. for the Advancement of Science, E. AAAS Committee on Science, and P. Policy, *AAAS science and technology policy yearbook, 1993.* Committee on Science, Engineering, and Public Policy, American Association for the Advancement of Science, 1993.

[2] F. Parietti and H. Asada, "Dynamic analysis and state estimation for wearable robotic limbs subject to human-induced disturbances," *Proceedings of IEEE International Conference on Robotics and Automation*, 2013.

[3] M. Botvinick and J. Cohen, "Rubber hands "feel" touch that eyes see," *Nature*, no. 6669, pp. 756–756, 1998.

[4] K. Armel and V. Ramachandran, "Projecting sensations to external objects: evidence from skin conductance response," *Proceedings of the Royal Society B: Biological Sciences*, vol. 270, no. 1523, pp. 1499–1506, 2003.

[5] M. Tsakiris, L. Carpenter, D. James, and A. Fotopoulou, "Hands only illusion: multisensory integration elicits sense of owndership for body parts but not for non-corporeal objects," *Experimental Brain Research*, vol. 204, no. 3, pp. 343 352, 2010.

[6] A. Guterstam, V. Petkova, and H. Ehrsson, "The illusion of owning a third arm," *PloS ONE*, vol. 2, no. 6, 2011.

[7] A. Dollar and H. Herr, "Lower extremity exoskeletons and active orthoses: challenges and state-of-the-art," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 144–158, 2008.

[8] A. Zoss, H. Kazerooni, and A. Chu, "Biomechanical design of the berkeley lower extremity exoskeleton (bleex)," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 2, pp. 128–138, 2006.

[9] E. Guizzo and H. Goldstein, "The rise of the body bots," *IEEE Spectr.*, vol. 42, no. 10, pp. 50–56, 2006.

[10] C. Walsh, K. Pasch, and H. Herr, "An autonomous, underactuated exoskeleton for load-carrying augmentation," *Processings of IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, pp. 1410–1415, 2006.

[11] S. Mohammed and Y. Amirat, "Towards intelligent lower limb wearble robots: challenges and perspectives - state of the art," *Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetrics*, pp. 312–317, 2009.

[12] H. Asada and H. Izumi, "Automatic program generation from teaching data for the hybrid control of robots," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 166 173, 1989.

[13] Y. Kunihoshi, M. Inaba, and H. Inoue, "Learning by watching: extracting reusable task knowledge from visual observation of human performance," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 799–822, 1994.

[14] J. Miura and K. Ikeuchi, "Task-oriented generation of visual sensing strategies in assembly tasks," *IEEE Transactions on Pattern Analsis and Machine Intelligence*, vol. 20, pp. 126–138, 1998.

[15] H. Onda, H. Hirukawa, F. Tomita, T. Suehiro, and K. Takase, "Assembly motion teaching system using position/force simulator-generating control program," *Processings of IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 2, pp. 938–945, 1997.

[16] M. Skubic and R. Volz, "Aquiring robust, force-based assembly skills from human demonstration," *IEEE Transactions on Robotics and Automation*, vol. 16, pp. 772–781, 2000.

[17] S. Dong, *Learning and Recognition of Hybrid Manipulation Tasks in Variable Environments using Probabilistic Flow Tubes.* PhD thesis, 2012.

[18] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillman, "A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models," *IEEE Transactions on Robotics and automation*, vol. 21, pp. 47–57, 2005.

[19] R. Palm and B. Iliev, "Segmentation and recognition of human grasps for programming by demonstration using time-clustering and fuzzy modeling," *Proceedings of IEEE Fuzzy Systems Conference*, pp. 1–6, 2007.

[20] S. Fels and G. Hinton, "Glove-talk ii, a neural-network interface which maps gestures to parallel formant speech synthesizer controls," *IEEE Transactions on Neural Networks*, vol. 9, pp. 205–212, 1998.

[21] B. Bedregal, A. Costa, and G. Dimuro, "Fuzzy rule-based hand gesture recognition," *Artificial Intelligence in Theory and Practice*, vol. 217, pp. 285–294, 2006.

[22] Z. J. abd H Liu, X. Zhu, and Y. Xiong, "Dynamic grasp recognition using time clustering, gaussian mixture models, and hidden markvov models," *Advanced Robotics*, vol. 23, pp. 1359 1371, 2009.

[23] A. Fod, M. Mataric, and O. Jenkins, "Automated derivation of primitives for movement classsification," *Autonomous Robots*, vol. 12, pp. 39–54, 2002.

[24] O. Jenkins and M. Mataric, "Deriving action and behavior primitives from huamn motion data," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2551–2556, 2002.

[25] B. Lim, S. Ra, and F. Park, "Movement primitives, principal component analysis, and the efficient generation of natural motions," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4630–4635, 2005.

[26] X. Jiang and Y. Motai, "Learning by observation of robotic tasks using online pca-based eigenbehavior," *Proceedings of IEEE International Symposium on Computation Intelligence in Robotics and Automation*, pp. 391–396, 2005.

[27] M. Boling, D. Padua, and R. A. Creighton, "Concentric and eccentric torque of the hip musculature in individuals with and without patellofemoral pain," *Journal of Athletic Training*, vol. 44, pp. 7–13, 2009.

[28] X. Wang and D. Lu, "Isokinetic peak torque of concentric and eccentric contraction," *ISBS Conference Proceedings*, 2009.

[29] "Dynamic analysis of load carriage biomechanics during level walking," *Journal of Biomechanics*, vol. 38, pp. 853–863, 2005.

[30] M. Foissac, G. Millet, A. Geyssant, P. Freychat, and A. Belli, "Characterization of the mechanical properties of backpacks and their influences on the energetics of walking," *Journal of Biomechanics*, vol. 42, pp. 125–130, 2009.

[31] J. Lew, Y. Jou, and H. Pasic, "Interactive control of human/robot sharing same workspace," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 535 540, 2000.

[32] D. Evans, "Wiring matters: An overview of the aircraft wiring issue," *Aviation Maintenance*, pp. 30–34, August 2006.

[33] H. Abdi, "Partial least squares regression and project on latent structure regression (pls resgression)," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, pp. 97–106, 2010.

[34] "Pls-regression: a basic tool of chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 58, pp. 109 130, 2001.

[35] A. McIntosh, W. Chau, and A. Protzner, "Spatiotemporal analysis of event-related fmri data using partial least squares," *Neuroimage*, vol. 23, pp. 764–775, 2004.

[36] P. Senin, "Dynamic time warping algorithm review," Tech. Rep. CSDL-08-04, 2008.

[37] L. Breiman, *Classification and regression trees.* Wadsworth International Group, 1984.