Development and Testing of an Impedance Controller on an Anthropomorphic Robot for Extreme Environment Operations

by

Lucille Aileen Hosford

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

Masters of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Signature redacted

Author . . . .
Department of Mechanical Engineering
May 3, 2016

Signature redacted

Certified by . . . .
Neville Hogan
Sun Jae Professor of Mechanical Engineering
Thesis Supervisor

Signature redacted

Accepted by . . . .
Rohan Abeyaratne
Professor of Mechanical Engineering Graduate Officer

Development and Testing of an Impedance Controller on an Anthropomorphic Robot for Extreme Environment Operations

by

Lucille Aileen Hosford

Submitted to the Department of Mechanical Engineering
on May 3, 2016, in partial fulfillment of the
requirements for the degree of
Masters of Science in Mechanical Engineering

Abstract

This thesis explores the use of impedance control on an anthropomorphic robot for operations in extreme, poorly mapped environments. First, a dynamic model was developed for a Baxter Research Robot. This model improved on standard dynamic models for similar robots by including the dynamics of the actuators in the system. Specifically, it was demonstrated that when the effective inertia of the actuators is neglected, the system will transmit 1.6 times more force to the environment than the model predicts. A force based Cartesian impedance controller was then implemented on Baxter, and numerous ways to modulate the endpoint impedance, including feedback and geometric configuration, were discussed and compared. Finally, a series of scaled down tasks similar to ones which are required in the decommissioning of offshore oil fields were then completed on Baxter using the Cartesian impedance controller. Overall, it was demonstrated that by using this more advanced control scheme, Baxter was (1) capable of satisfactorily completing the scaled down tasks, (2) more robust against errors in the map of the environment than with traditional controllers, and (3) capable of improving the map of its environment while completing the task.

Thesis Supervisor: Neville Hogan
Title: Sun Jae Professor of Mechanical Engineering

Acknowledgments

This Thesis would not have been possible without support from both Newman Labs and the Hover Research Group.

In Newman Labs, I would first like to thank my advisor, Professor Neville Hogan. Neville's understanding of the dynamics of interactive systems is unmatched, and I am thankful to have had the opportunity to work for him. I am also thankful for my colleagues, Will, Julie, and David, who were always willing to lend a helping hand with Baxter.

In the Hover Research Group, I would like to thank my undergraduate advisor, Professor Franz Hover, for encouraging me to pursue a graduate education and recruiting me for this project. Many thanks to Chris Welch for sharing both his knowledge of robotics and his coffee supply.

Finally, thank you to my parents for deciding that their children would have the opportunity to go to college, and doing everything possible to make that a reality.

This work was supported by Chevron Energy Technology Company under research grant E12 UPP PDEP: Robotic and Autonomous Decommissioning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Challenge of Decommissioning

There are 2900 offshore oil platforms in the Gulf of Mexico alone. As of 2013, 629 of those are currently eligible for decommissioning (BSEE, 2013). The overall process of decommissioning an offshore oil platform is both dangerous and expensive. In order to inspect and ultimately disassemble the pilings, saturation divers have to work at pressures of up to 19 times that of atmospheric pressure. Decompressing from these pressures can take up to 6 days and cause a significant amount of stress on the body. To save time and stress on the body, divers will frequently live at pressure for up to a month at a time. In order to accomplish this, support ships have to stay on site for the entire duration of the decommissioning process. This supporting infrastructure is extremely expensive. Furthermore, in the future offshore oil platforms will be at depths which are significantly deeper than they currently are now. The deepest offshore oil well is 2880m (Goodley, 2013). In contrast, the world record saturation dive is 701m (Clark, 2016). The physical cost on the human body and financial supporting costs will only increase.

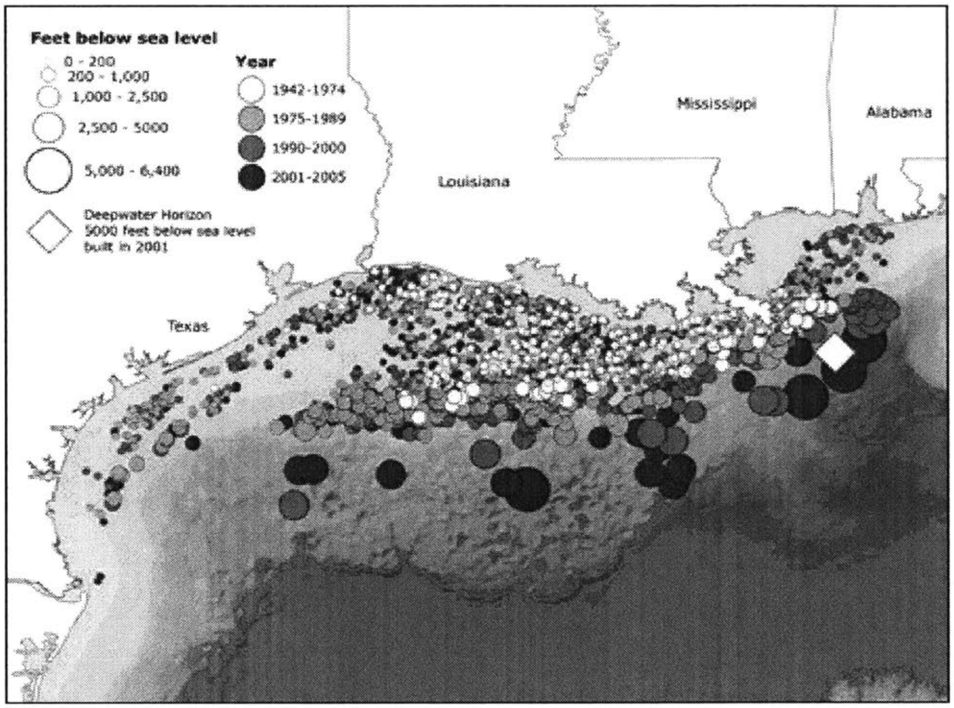


Figure 1-1: Offshore oil platforms in Gulf of Mexico (DeepSeaNews, 2010).

Robotic decommissioning could provide a safer and less expensive alternative to the current situation. Fewer divers in the water would mean fewer lives at risk. Leaving robots to complete the mission and coming back at specified check-in times to see how the work has progressed would drive down operation costs of the support crew.

1.2 Specific Tasks Required for Decommissioning

In order for robots to complete this mission, they would need to accomplish a number of tasks which humans can complete quite easily. These tasks include but are not limited to (Chevron, 2014):

1. Securing itself to an unprepared (biofouled) surface. These surfaces could be flat, cylindrical, or "saddle-shaped" such as at a gusset/junction between cylindrical members.

2. Removing/scrubbing biofouling from a local area, possibly with a complex geometry.

3. Despite biofouling, find a nominally known feature: valve, lever, etc. Test its function, i.e. whether it can be operated with acceptable force/torque.

4. Using tactile/haptic exploration, refine a geometric model of a shape.

1.3 Specific Technical Challenges in Accomplishing These Tasks

While many of these technical challenges are straightforward for a human being, they can prove surprisingly challenging for a robotic system. For instance, before a robot can complete any of these tasks, it must first make and maintain stable contact with the object in question. While humans do this all the time, finding the maximum closing velocity (and optimum impedance) of a robot to make and maintain contact with an ill-defined surface is a non-trivial challenge. Moving slowly until contact with the environment is detected—a process known as "guarded moves" (Mason, 1981)—and using hybrid control thereafter (Craig & Raibert, 1979) has been proposed as a sufficient framework for transitioning between free and constrained motion. However, due to the non-zero closing velocity at the transition between free and constrained motion, there will always be an initial impact no matter how slowly the robotic arm is moving. Understanding when and how that initial impact becomes important in designing a control system is vital to the success of the task.

Each of these tasks presents its own unique set of challenges. For instance, scrubbing biofouling could potentially be accomplished by controlling force along the normal of the surface and position along the tangent—hybrid control. In contrast, when drilling through a surface, the reaction torque from the drill bit will become one of the most significant dynamics to consider. An alternative strategy may be required. Furthermore, a fully constrained task where the robot is gripping an object—such as turning a valve—will produce very different reaction dynamics then when it is simply pressing against a surface. One way to view the problem is that each of these individual tasks will require its own unique suspension system.

Furthermore, due to the age and decay of the structures, Computer Aided Design (CAD) drawings of the offshore oil structures will no longer provide an accurate description of the operating environment. Visual feedback in the ocean environment can be highly limited. Mapping techniques utilizing sonar and lasers continue to progress and provide more detailed descriptions of the environment. However, they do not determine the rigidity of the object they are mapping. For instance, if a structure is covered in biofouling, a map of that structure will not provide information on whether the biofouling is hard or soft. The robot must instead be able to operate in a poorly mapped environment, as well as feel the object it is working with to develop a haptic map—or an impedance field—in which the impedance of the surroundings is a function of location.

1.4 Strategies for Approaching This Challenge

In approaching this challenge, two different design strategies become apparent. First, a different robot could be designed to address each of these individual tasks. Much like in a manufacturing plant, the design of the robot could be optimized around one specific task. Frequently, robots actually surpass a human's capabilities in completing a task when utilized in this manner. However, it would be expensive to design and maintain multiple types of robots. Furthermore, while this design strategy has proven successful in structured manufacturing and lab environments, operations in an ocean environment will require a higher level of robustness than has been previously required of this approach.

An alternative approach would be to create an "all in one" solution that can accomplish many or all of these tasks while being robust enough to operate in a poorly mapped environment. Human beings have yet to be surpassed by any robotic design in either versatility or robustness against an ill-defined environment. As such it stands to reason that certain components of an anthropomorphic design could be useful in accomplishing this task.

Robotic arms are already heavily utilized in ocean engineering projects. However, there are a number of key differences between their current design and what makes a human be able to interact stably with so many different environments. First, there is evidence to suggest that humans are able to interact passively with their environment (Lee, 2013). Passivity is important as it ensures stable interaction (Colgate & Hogan, 1988). Some robotic arms do utilize force feedback to implement haptic control (BluHaptics, 2016). However the robotic arm itself copies the human's movement using velocity control. It is still the human being who adjusts the interaction with the environment based on the reaction forces sent back to them. However, while the details of the controls for currents systems used today are not known, communications delays in the system could result in severe limitations in the operating bandwidth of the system (Van De Vegte, Milgram, & Kwong, 1990).

Second, the human arm has kinematic redundancies: more controllable degrees of freedom than the minimum number required to describe the spatial position of its end-effector (e.g. the hand). This redundancy appears to be a prominent contributing factor to why people are able to stably interact with so many different environments. Being able to maintain a single position of the hand while changing the configuration of the arm is an effective way to modulate endpoint impedance. In a way, the arm can act as a variable suspension system. However, most robotic arms do not incorporate kinematic redundancies because of the additional challenges they introduce into the control scheme.

As engineers, an overarching goal of any design challenge is to make the solution as simple as possible. Additionally, when creating a mathematical model of the system in question, the model should be as simple as possible, but not simpler (Einstein, 1933). This thesis utilizes the Baxter Research Robot (an anthropomorphic robot) to assess whether common assumptions made in the creation of a dynamic model of a robot are still valid in this challenging environment. Once a dynamic model of Baxter was created, a force-based Cartesian impedance controller was implemented on Baxter and used in a series of scaled down decommissioning tasks. The controller's performance was compared to more traditional control methods. Finally, position and force data recorded by Baxter were analyzed to determine whether or not they could be utilized in mapping and control decisions.

1.5 Overview of Thesis

The remainder of this thesis is divided into five chapters. Chapter 2 provides an overview of the Baxter Research Robot and its supporting software. It also documents the derivation of Baxter's kinematic model, and compares the author's model to pre-existing models. Chapter 3 documents Baxter's dynamic model. Specifically, it assesses the common assumption that Baxter's actuators may be treated as perfect force/torque sources. Chapter 4 justifies and describes the force-based impedance controller implemented on Baxter, and compares various methods of modulating endpoint impedance. Chapter 5 documents the two scaled down tasks completed using this control scheme: (1) making stable contact and operating in a poorly mapped environment, (2) and sanding down a piece of wood. It also provides an assessment of the impedance controller, compares its performance to more traditional controllers, and determines whether a map of the environment could be improved upon while completing these tasks. Finally, Chapter 6 contains conclusions and recommendations for future work.

Chapter 2

The Baxter Research Robot and Supporting Software

The Baxter Research Robot is an anthropomorphic robot developed by Rethink Robotics (RethinkRobotics, 2016). It was originally designed for light "pick and place" tasks in a manufacturing setting. As it was specifically designed to operate in close proximity to humans, it also comes with a number of safety features. It is clearly not the final robot which would be used in any extreme environment operation. However, its Series Elastic Actuators which provide force feedback at each joint, redundant number of joints, and many safety features allowing close proximity to humans make it an ideal piece of hardware on which to test different control systems in a controlled, scaled-down setting.

This chapter provides a review of the Baxter Research Robot's hardware, control systems, and supporting software. It also provides baseline documentation verifying the accuracy of Baxter's dynamic model.

2.1 Hardware

As shown in figure 2-1, Baxter has seven degrees of freedom in each arm. The number of degrees of freedom in the joint space (7) is greater than the degrees of freedom in the end-effector space (6). Because of this, each arm is kinematically redundant.

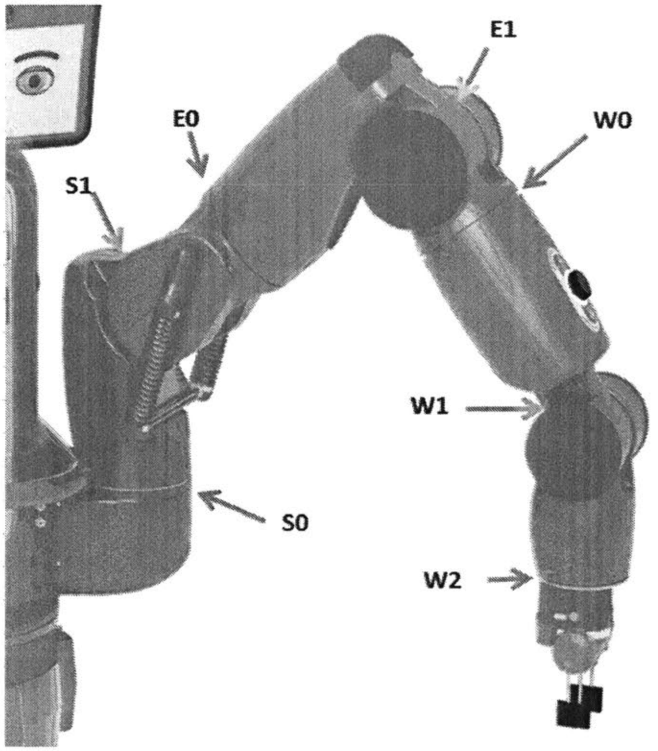


Figure 2-1: Baxter's arm with Rethink Robotic's joint name convention (sdk, 2015).

Baxter's joints are driven by Series Elastic Actuators (SEAs). Originally developed by (Pratt & Williamson, 1995), SEAs introduce compliance into the system by placing a flexure in series between the motor shaft and the output load. The torque applied to each joint can be determined by measuring the deflection of the spring. Figure 2-2 shows the SEA for joint E1. Joints S0, S1, E0, and E1 have a peak torque of 50 Nm and a spring stiffness of approximately $843\frac{Nm}{rad}$. Joints W0, W1, and W2 have a peak torque of $15Nm$ and a stiffness of approximately $250\frac{Nm}{rad}$ (Hardware, 2015). In all joints, at peak torque the elastic element deflects no more than $0.06rad$, just under 3.5^o. As such, the SEAs serve more as robust torque sensors than as a method for absorbing impacts or creating mechanical compliance. This will be discussed more rigorously in Chapter 3. The gear ratios and motor inertia are not provided by Rethink Robotics.

In addition to the springs at each of Baxter's SEAs, each of Baxter's arms has two large springs in parallel with the shoulder joint. Each spring has a stiffness of

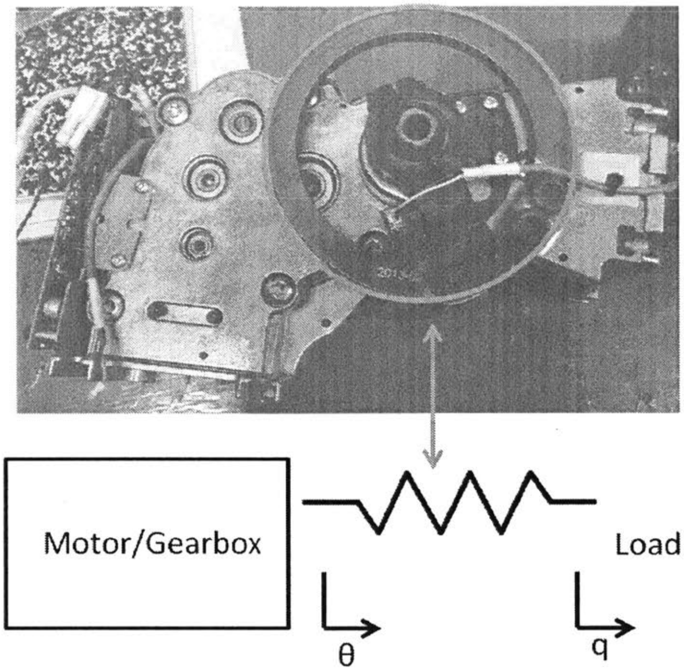


Figure 2-2: SEA driving Joint E1. The motor and gearbox are connected to the load via the "G" shaped spring.

9.6N/mm. These springs provide even more compliance at the end effector. The position of each joint is measured by an absolute magnetic angle encoder which is placed on the load side of the flexure. Velocity is derived from the position data.

2.2 Baxter's Supporting Kinematic and Dynamics Tools

The dimensions, masses, and inertias of each limb that are available to calculate the kinematics and dynamics of the system are all taken directly from Baxter's SolidWorks model. All of these parameters, along with information regarding the sensors, are stored in a Robotic Operating System (ROS) document called the Unified Robot Description Format (URDF). The URDF is a standardized Extensible Markup Language (XML) format that is used for constructing a robot's model. In particular, it provides a standard method in which to describe the kinematic relationships and track the coordinate frames of the links and joints throughout the robot.

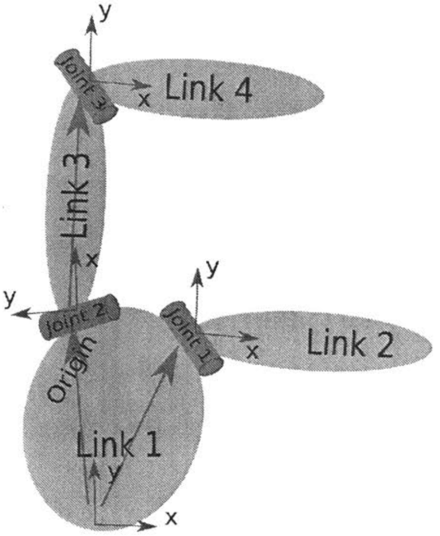


Figure 2-3: Example of a joint/link system as created by the URDF (ROS, 2012).

While this is clearly a powerful computational tool, it has certain limitations which are important to acknowledge. Specifically, none of the dynamic parameters of the SEAs are included in the URDF. This assumption to neglect actuator dynamics is not unique to the ROS framework. To the best of the author's knowledge, no dynamic simulator on the market today accounts for actuator dynamics. Instead, it is always assumed the control system at each joint allows the actuator to be modeled as an "ideal" position, velocity, or torque source—the commanded motion is assumed to be independent of effort and commanded effort is independent of motion. The consequences of this omission for interactive tasks will be considered in Chapter 3.

The URDF contains kinematic and dynamic data for the arm. The computation of the kinematic structure is then completed by the Orocos Python Kinematics and Dynamics Library (PyKDL), an open source framework for modeling and computation of open loop kinematic chains. Originally created by (Hawkins, 2013) this model can provide the forward kinematics, inverse kinematics, Jacobian, Jacobian Transpose, Jacobian Pseudo-Inverse, and Joint and Cartesian Inertias in real time at up to $1000Hz$. There is minimal documentation on the derivation of the PyKDL. Furthermore, Rethink Robotics does not currently guarantee its continued support of the system. As such, the author felt it was necessary to verify the accuracy of the forward kinematics and joint space inertia matrix before utilizing these tools to perform tasks. By doing so, there is now baseline documentation of how these parameters were derived should the PyKDL system at any time no longer be supported.

2.2.1 Verification of the Forward Kinematics

Using the parameters in Baxter's URDF, the Denavit Hartenberg convention (Denavit, 1955) was used to construct the Forward Kinematics of Baxter's arm. The DH parameters for each joint are listed below. d and a are in units of meters.

$$\theta = (q_1, q_2 + \frac{\pi}{2}, q_3, q_4, q_5, q_6, q_7) \tag{2.1}$$

where q_n is the joint angle relative to the body frame from proximal—closest to the body—to distal—furthest from the body—joints.

$$d = (0.27035, 0, 0.36435, 0, 0.37429, 0, 0.229525) \tag{2.2}$$

$$a = (0.069, 0, 0.069, 0, 0.01, 00) \tag{2.3}$$

$$\alpha = (-\frac{\pi}{2}, \frac{\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{2}, 0) \tag{2.4}$$

These parameters can be used to transform between coordinate frames at each joint using the standard DH convention as follows

$$\mathrm{A}_i = R_{\theta_i} Trans_{d_i} Trans_{a_i} R_{\alpha_i}$$

$$= \begin{bmatrix} c_{\theta_i} & s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(2.5)

Baxter's Origin is located at its center where the robot connects with the stand (see figure 2-4). While not included in Baxter's URDF, the final translation from joint S0 to Baxter's origin of the left arm is

$$A_l = \begin{bmatrix} 0.70711 & -0.70711 & 0 & 0.064027 \\ 0.70711 & 0.70711 & 0 & 0.25903 \\ 0 & 0 & 1 & 0.09066 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Final translation from joint S0 to Baxter's origin of the right arm is

$$A_r = \begin{bmatrix} 0.70711 & 0.70711 & 0 & 0.064027 \\ -0.70711 & 0.70711 & 0 & -0.25903 \\ 0 & 0 & 1 & 0.09066 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

These final translations from the shoulder joint to Baxter's origin were taken from work done by (Rupert, 2016).

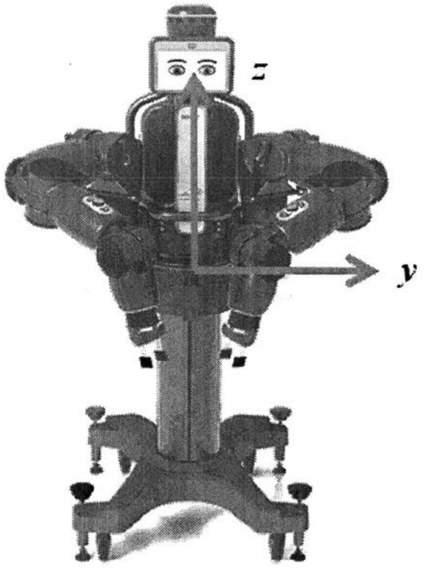


Figure 2-4: Body Coordinates for Baxter. The origin is located at the center of the robot where the body connects with the stand (Teq, 2016).

By multiplying these matrices together for a given set of coordinates a matrix is created of the form

$$T_b^7 = A_l A_1 A_2 A_3 A_4 A_5 A_6 A_7 = \begin{bmatrix} X_x & Y_x & Z_x & T_x \\ X_y & Y_y & Z_y & T_y \\ X_z & Y_z & Z_z & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

Where the last column provides the x, y, and z position of the end effector with respect to Baxter's origin and the upper left 3x3 matrix provides the orientation referenced from Baxter's coordinate frame.

Both the forward kinematics derived from the DH parameters and the PyKDL were used to calculate the endpoint position and orientation of Baxter's arm for three different configurations. As shown in Table 2.1, the results were comparable.

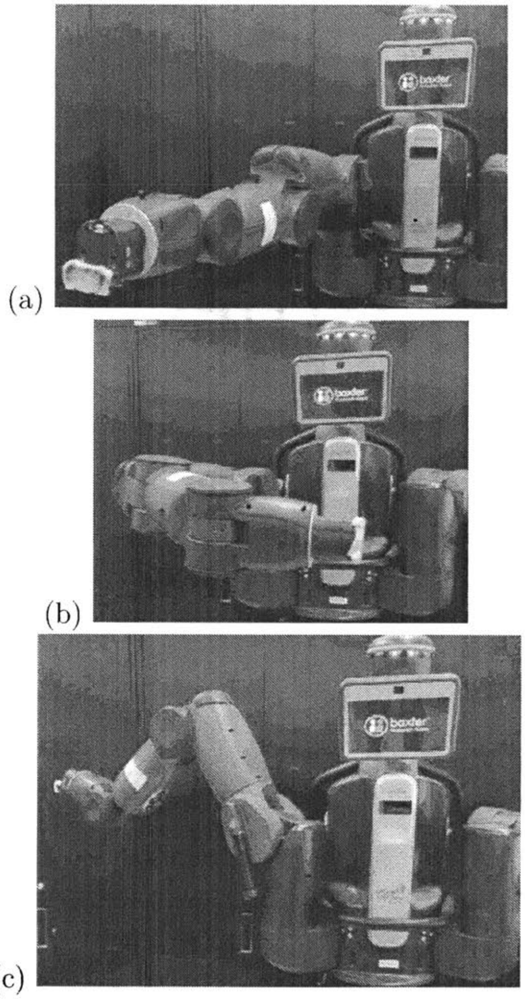


Figure 2-5: Three separate poses used to verify Baxter's forward kinematics.

Table 2.1: Comparison of endpoint position and orientation as calculated by the dynamic library vs. the Denavit Hartenberg convention, pose "a." Results were similar for different configurations.

Model (right arm)	X(m)	Y(m)	Z(m)	Roll(deg)	Pitch(deg)	Yaw(deg)
Dynamics Library	0.786	-1.037	0.335	-81.2	84.2	-30.5
DH convention	0.7742	-1.01	0.3154	-77.2	85.1	-33.5
Percent difference	1.5	2.6	5.9	2.2	1.5	3.4

2.2.2 Verification of Joint Space Inertia Matrix

The Joint Space Inertia Matrix (JSIM) describes the inertia seen at each joint of a robot due to the coupling of the linkages. With seven degrees of freedom, Baxter's full JSIM is a 7x7 matrix. For clarity, a simplified example of how to derive a JSIM for a planar 3 dof robot is first provided

Example for Planar Case

Consider a three linkage planar mechanism as shown in figure 2-6.

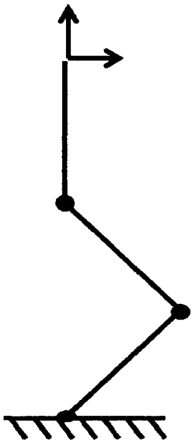


Figure 2-6: Simple planar example for deriving the Joint Space Inertia Matrix.

In a global reference frame, fully uncoupled from each other, the positions of each limb can be fully described as

$$x_u = \begin{bmatrix} x_{c1} \\ y_{c1} \\ \theta_1 \\ x_{c2} \\ y_{c2} \\ \theta_2 \\ x_{c3} \\ y_{c3} \\ \theta_3 \end{bmatrix} \tag{2.9}$$

To simplify calculations, assume each limb can be modeled as a cylindrical rod with a mass m_n, inertia I_c, and length l_n. An uncoupled inertial parameter matrix, M, can be constructed

$$M = \begin{bmatrix} m_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{c1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{c2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{c3} \end{bmatrix} \tag{2.10}$$

This matrix represents the inertia which each limb contributes to the system independently. It can be used to express the kinetic co-energy of the system as a whole as

$$E_k^* = \frac{1}{2} v^t M v \tag{2.11}$$

Where v is the derivative of x_u. Working with a robotic arm, it is desirable to operate in a set of generalized coordinates which account for the coupling of the linkages. A kinematic relationship between the position of the center of mass of each link and the global coordinate frame can be derived as shown below.

$$x_u = \begin{bmatrix} x_{c1} \\ y_{c1} \\ \theta_1 \\ x_{c2} \\ y_{c2} \\ \theta_2 \\ x_{c3} \\ y_{c3} \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \frac{l_1}{2}\cos(\theta_1) \\ \frac{l_1}{2}\sin(\theta_1) \\ \theta_1 \\ l_1\cos(\theta_1) + \frac{l_2}{2}\cos(\theta_2) \\ l_1\sin(\theta_1) + \frac{l_2}{2}\sin(\theta_2) \\ \theta_2 \\ l_1\cos(\theta_1) + l_2\cos(\theta_2) + \frac{l_3}{2}\cos(\theta_3) \\ l_1\sin(\theta_1) + l_2\sin(\theta_2) + \frac{l_3}{2}\sin(\theta_3) \\ \theta_3 \end{bmatrix} \tag{2.12}$$

By operating in these coordinates, only three variables—θ_1, θ_2, and θ_3—are required to fully define the position of the arm at any time. The Jacobian of this kinematic relation, $j(\theta)$, provides a transformation between uncoupled and generalized velocities such that

$$v = j(\theta)w \tag{2.13}$$

Finally, because kinetic co-energy is the same in any coordinate frame, it may now be expressed as

$$E_k^* = \frac{1}{2}v^t M v = \frac{1}{2}\omega^t j^t(\theta) M j(\theta)\omega \tag{2.14}$$

The inertia matrix in a generalized coordinate frame, which is now a function of configuration, may be written as

$$JSIM = j^t(\theta) M j(\theta) \tag{2.15}$$

Derivation for Baxter

Each of Baxter's arms have 7 joints will full 6 DoF motion. This means one arm's motion vector will have 42 terms! Additionally, the centers of mass of each link were not on the lines joining adjacent joint centers, further complicating the calculations. Instead of deriving all 42 kinematic relations by hand, the Denavit Hartenberg parameters were once again used to calculate the position of the each link's center of mass with respect to Baxter's origin. Baxter's JSIM was then calculated for pose "a" in figure 2-5. The diagonal of the JSIM matrix is shown in table 2.2, and the results were compared to the PyKDL. Some of the joints had errors as large as 25%. As it was not derived from the URDF, the largest source of difference between the two is most likely from the final transformation from the joint S0 to the torso. The full documentation of the DH parameters, forward kinematics, and JSIM, can be found in Appendix A. The URDF treats the SEAs as perfect motion or effort sources. As such, any dynamic parameters for the SEAs—such as the rotational inertia of the motors reflected through the gearbox—are neglected. The validity and consequences of this assumption will be discussed in Chapter 3.

Table 2.2: Comparison of diagonal entries of the JSIM for pose "a" calculated by the dynamic library vs. the Denavit Hartenberg convention. The results for other poses were similar.

Dynamics Library	3.89	3.07	0.048	0.628	0.009	0.027	0.0006
DH Convention	3.97	3.17	0.0433	0.763	0.0112	0.032	0.0007
Percent Difference	2%	3.25%	10%	17.7%	24.4%	18.5%	17%

2.2.3 Onboard Control Systems

The Baxter Research Robot's control system has four layers (ControlOverview, 2015):

1) User Code running via workstation or a Secure Shell SSH (Python Joint Control or Joint Trajectory Action Server or your custom interface)

2) Joint Control Listeners via ROS topic (on Baxter's internal Gentoo Linux PC)

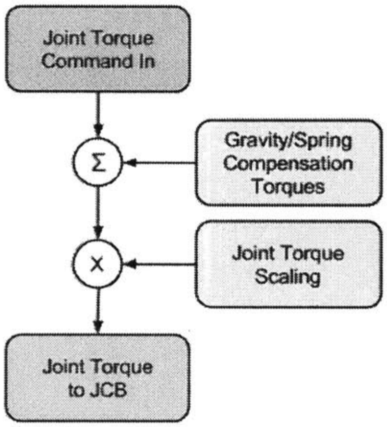
3) RealTime Motor Control Loop (the highest priority process on Baxter's internal Gentoo Linux PC)

4) Joint Control Boards (microcontrollers attached to each arm joint)

The bottom three layers of the control system cannot be accessed by users. However, the real-time motor control loop has four "joint control modes" which provide varying levels of access to the Joint Controller Boards (JCBs) at each individual SEA: position control, torque control, velocity control, and "raw" position control. For the purposes of this project, three of the control modes were used: position, velocity, and torque control. The specific gains of all controllers at the JCBs are not known. It is important to note the Baxter Research Robot does not come with the same software package as the Baxter designed for industry. Any details about the controllers used on the Research Robot are not necessarily applicable to the industry package.

Torque Control

The diagram describing the torque control mode is shown in figure 2-7. However, the specific gains of the torque controller at the JCBs are not known. According to Rethink Robotics, the controller has a reliable loop closure rate of up to $1000Hz$.



Gravity/Spring Compensation: The joint torque command is applied in addition to the gravity and S1 spring compensation torques.

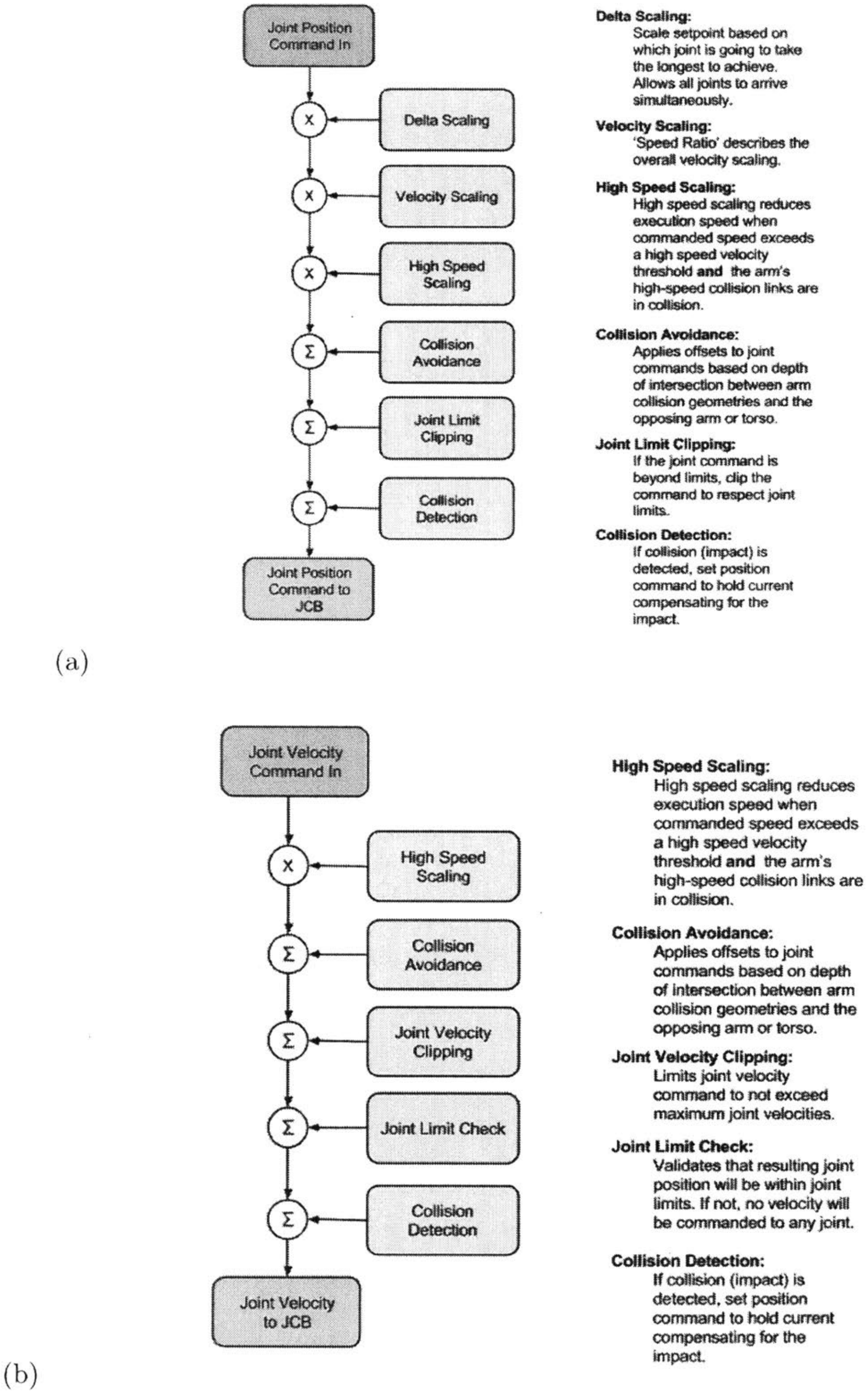
Joint Torque Scaling: Scales all joint torques if a torque command exceeds the maximum allowable torque for that joint. This scaling ratio is defined as torque_max / torque_command.

Figure 2-7: Diagram of JCB Torque Control Mode (ArmControlMode, 2015).

This torque controller was used to develop and implement a force-based impedance controller which will be described in Chapter 4 and assessed in Chapter 5.

Position and Velocity Control

The majority of robotic control systems, particularly in deep sea applications, utilize position or velocity control. Before implementing a new control scheme, it is important to understand what the standard control system is capable of. To that end, the scaled down tasks which were attempted with the new controller were also attempted using Baxter's joint trajectory playback mode. In this mode, the user can guide Baxter's arm through a desired series of motions. Both the joint positions and velocities are recorded throughout the movement. The motion of the arm can then be "played back" using either Baxter's positions controllers or velocity controllers. The results of the use of these controllers are presented in Chapter 5. Figure 2-8 shows a diagram of the position controller at each joint.



Delta Scaling:
Scale setpoint based on which joint is going to take the longest to achieve. Allows all joints to arrive simultaneously.

Velocity Scaling:
'Speed Ratio' describes the overall velocity scaling.

High Speed Scaling:
High speed scaling reduces execution speed when commanded speed exceeds a high speed velocity threshold **and** the arm's high-speed collision links are in collision.

Collision Avoidance:
Applies offsets to joint commands based on depth of intersection between arm collision geometries and the opposing arm or torso.

Joint Limit Clipping:
If the joint command is beyond limits, clip the command to respect joint limits.

Collision Detection:
If collision (impact) is detected, set position command to hold current compensating for the impact.

High Speed Scaling:
High speed scaling reduces execution speed when commanded speed exceeds a high speed velocity threshold **and** the arm's high-speed collision links are in collision.

Collision Avoidance:
Applies offsets to joint commands based on depth of intersection between arm collision geometries and the opposing arm or torso.

Joint Velocity Clipping:
Limits joint velocity command to not exceed maximum joint velocities.

Joint Limit Check:
Validates that resulting joint position will be within joint limits. If not, no velocity will be commanded to any joint.

Collision Detection:
If collision (impact) is detected, set position command to hold current compensating for the impact.

Figure 2-8: Diagram of JCBs (a) Position and (b) Velocity Control Modes (ArmControlMode, 2015).

Chapter 3

Dynamic Model of Baxter

Before Baxter could be used in contact experiments, a dynamic model of its arms had to be constructed. Robots are most commonly and successfully used in a highly structured environment to accomplish "pick and place" tasks. In these tasks, the dynamics between the robot and the environment can be handled almost solely by compliance at the end effector. As such, a detailed dynamic model of the robot is not always required. However, when a task requires dynamics which utilize the whole arm, or the robot is operating in a poorly mapped environment that is likely to evoke unplanned collisions, it becomes necessary to have a thorough understanding of how the arm will react. A more rigorous model may be required.

This chapter will provide an overview of the model of a robot with SEAs, and specifically address the assumptions made in Baxter's model. It will discuss two separate methods which were used to characterize the reflected motor inertia—the rotational inertia of the motors as seen through the gearbox—of Baxter's SEAs and assess the importance of these parameters in a dynamic model.

3.1 Model of a Robot with Series Elastic Actuators

All of Baxter's joints are revolute, and its linkages were assumed to be rigid. With these assumptions in mind, the equations of motion in joint space for a series of linkages forming a robotic limb with SEAs are

$$M(q)\ddot{q} + C(\dot{q}, q) + B\dot{q} + g(q) = \tau_{measured} + \tau_{ext} \tag{3.1}$$

$$I_r\ddot{\theta} + D\dot{\theta} = \tau_{motor} - \tau_{measured} \tag{3.2}$$

$$\tau_{measured} = K_{sea}(\theta - q) \tag{3.3}$$

M is the joint space inertia matrix and I_r is the reflected inertia of the motors. C is the Coriolis force, B and D are any friction—assumed to be linear—in the joints, and g is the force due to gravity. $\tau_{external}$ is any external torque on the system, and τ_{motors} is the torque from the motors. $\tau_{measured}$ is the torque measured at the SEA which has a stiffness K_{SEA}. Finally, q and θ are positions distal and proximal to the SEA's spring respectively.

This reflected inertia represents the rotational inertia of the motors that are driving the actuator as seen through the gearbox such that

$$I_r = N^2 I_{motor} \tag{3.4}$$

Where I_{motor} is the actual rotational inertia of the motor and N is the gear ratio of the gearbox. From a strictly mechanical perspective, the reflected inertia would be exactly equal to this value. However, as there is a control system with unknown gains at this actuator, the reflected inertia may be changed slightly. This will be further discussed later in this chapter.

Baxter has an internal gravity compensation model which applies compensation torques for both gravity and the large springs located at the shoulders. As such, $g(q)$ may be neglected, at least as a first approximation. Coriolis forces will also be initially neglected, but this assumption will only hold for small motions. Eventually much larger motions will be desired. With these assumptions, the equations of motion simplify to

$$M(q)\ddot{q} + B\dot{q} = \tau_{measured} + \tau_{ext} \tag{3.5}$$

$$I_r\ddot{\theta} + D\dot{\theta} = \tau_{motor} - \tau_{measured} \tag{3.6}$$

$$\tau_{measured} = K_{sea}(\theta - q) \tag{3.7}$$

As shown in Chapter 2, $M(q)$ is highly dependent on the configuration of the arm. However, for a given configuration, it becomes a constant. Linearizing about a set of joint positions using the Taylor series expansion, the model may be written in state space form as

$$\frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 \\ -M^{-1}K_{sea} & -M^{-1}C & M^{-1}K_{sea} & 0 \\ 0 & 0 & 0 & I \\ I_r^{-1}K_{sea} & 0 & -I_r^{-1}K_{sea} & -I_r^{-1} * D \end{bmatrix}\begin{bmatrix} q \\ \dot{q} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ I_r^{-1} \end{bmatrix}\tau_{motor} \tag{3.8}$$

$$\dot{x} = \begin{bmatrix} 0 & J & 0 & 0 \end{bmatrix}\begin{bmatrix} q \\ \dot{q} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix}\tau_{motor} \tag{3.9}$$

where J is the Jacobian, also linearized the about the operating point, which transforms the joint velocity output to end-effector space. Each actuator is a fourth order system. Neglecting structural vibrations, one of Baxter's arms would be modeled as a 28^{th} order system.

3.1.1 Assumptions in Modeling SEAs

The general model for a single SEA in shown in figure 3-1. The linearized equation of motion for a single SEA has the same general structure as that of the whole arm.

$$\frac{d}{dt}\begin{bmatrix} x_2 \\ \dot{x_2} \\ x_1 \\ \dot{x_1} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 \\ -m_2^{-1}K_{sea} & -m_2^{-1}b_1 & m_2^{-1}K_{sea} & 0 \\ 0 & 0 & 0 & I \\ m_1^{-1}K_{sea} & 0 & -m_1^{-1}K_{sea} & -m_1^{-1}*b_2 \end{bmatrix}\begin{bmatrix} x_2 \\ \dot{x_2} \\ x_1 \\ \dot{x_1} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & m_2^{-1} \\ 0 & 0 \\ m_1^{-1} & 0 \end{bmatrix}\begin{bmatrix} F_{motor} \\ F_{external} \end{bmatrix} \tag{3.10}$$

The system will have two modes: one where the two masses move in opposite directions, $(x_1 = -c * x_2)$ and a rigid body mode $(x_1 = x_2)$.

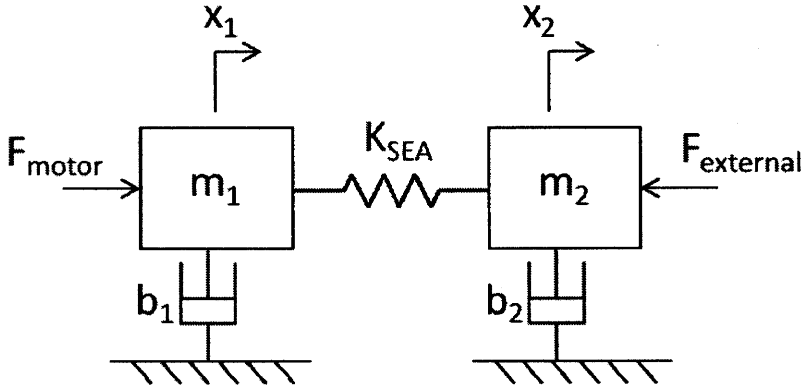


Figure 3-1: Diagram of an SEA. A motor and gearbox with some inertia and damping m_1 and b_1 drives a spring connected to the linkage with inertia and damping m_2 and b_2.

The springs on Baxter's SEAs are extremely stiff; as discussed in Chapter 2, maximum deflection under peak torque is less than 3.5°. The frequency response for a single SEA is represented in the Bode plots shown in figure 3-2. Because of the high stiffness of the spring, the rigid body mode is the dominant behavior up to approximately $90\frac{rad}{s}$. For planned operations, this is far above Baxter's maximum operating frequency.

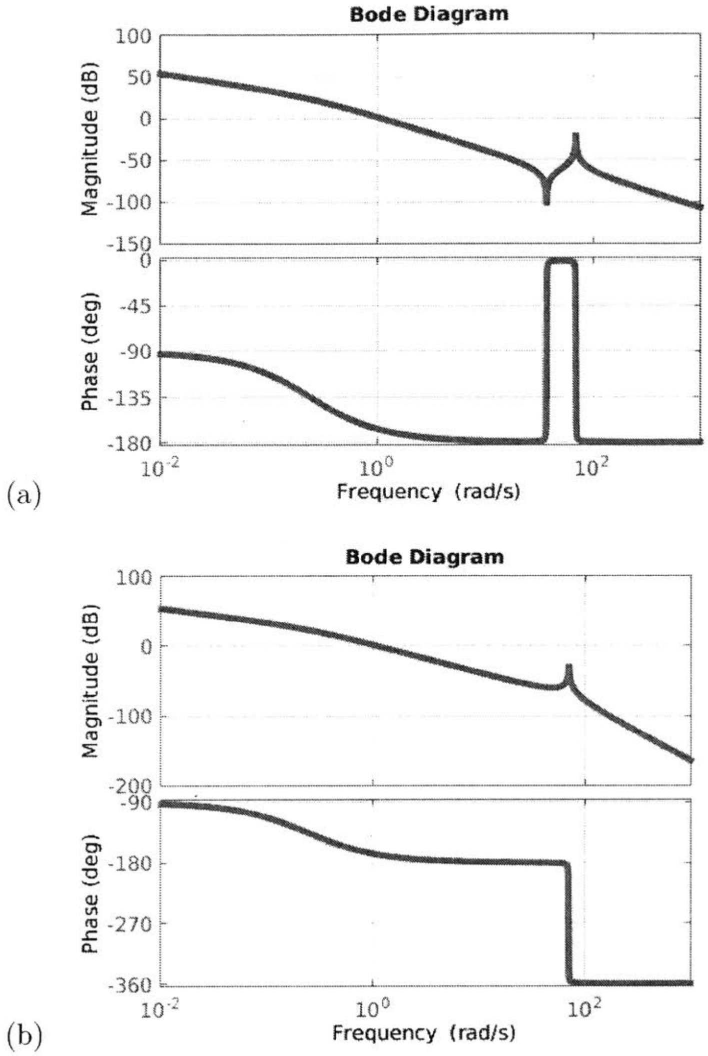


Figure 3-2: Bode plots for the simplified model with position "x_2" as the output and (a) an external force and (b) the motor as the input. In case (a) the rigid body mode dominates. The system reacts like a first order model except for at approximately $90\frac{rad}{sec}$. In case (b), the rigid body mode also dominates until approximately $90\frac{rad}{sec}$. Because the input and the output are separated (non-collocated) passivity is compromised for case "b." For both cases, the stiffness and the inertias were based on actual parameters of Baxter's SEAs and linkages. A low damping was arbitrarily assigned.

However, this does not necessarily mean that the second mode is negligible. For instance, an unexpected collision with the environment could have a very short impact time. As an example, figure 3-3 shows the force profile from an impact with a rigid surface, which lasted approximately 0.07 seconds.

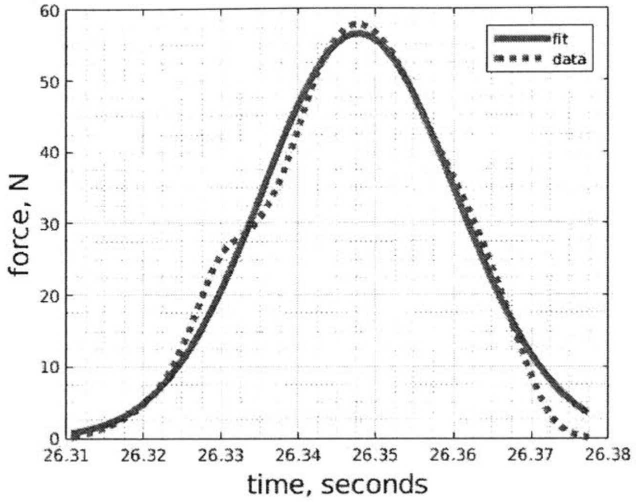


Figure 3-3: Force profile of a collision event. The impact lasted approximately 0.07 seconds.

This impact was approximated to be a gaussian curve, and fit to a model with 95% confidence bounds such that

$$fit = 56.52e^{-(\frac{t-26.35}{0.01775})^2} \tag{3.11}$$

The simplified model's response to this external force was then simulated in matlab. The results are shown in figure 3-4. Even with this extremely short impact time and low damping parameters, the model responds like a standard first order system, with only the slightest oscillation visible from the second mode. The first order approximation holds.

Despite the fact that Baxter will typically operate at low frequencies, if the second mode is not accounted for in the control of the SEA, high gains could still drive the system to instability. The pole zero map of a single SEA is shown in figure 3-5. Despite the 2nd order behavior, there are still two poles close to the origin which, if not accounted for in the control system, could be driven into the right half plane (RHP) at even modest gains.

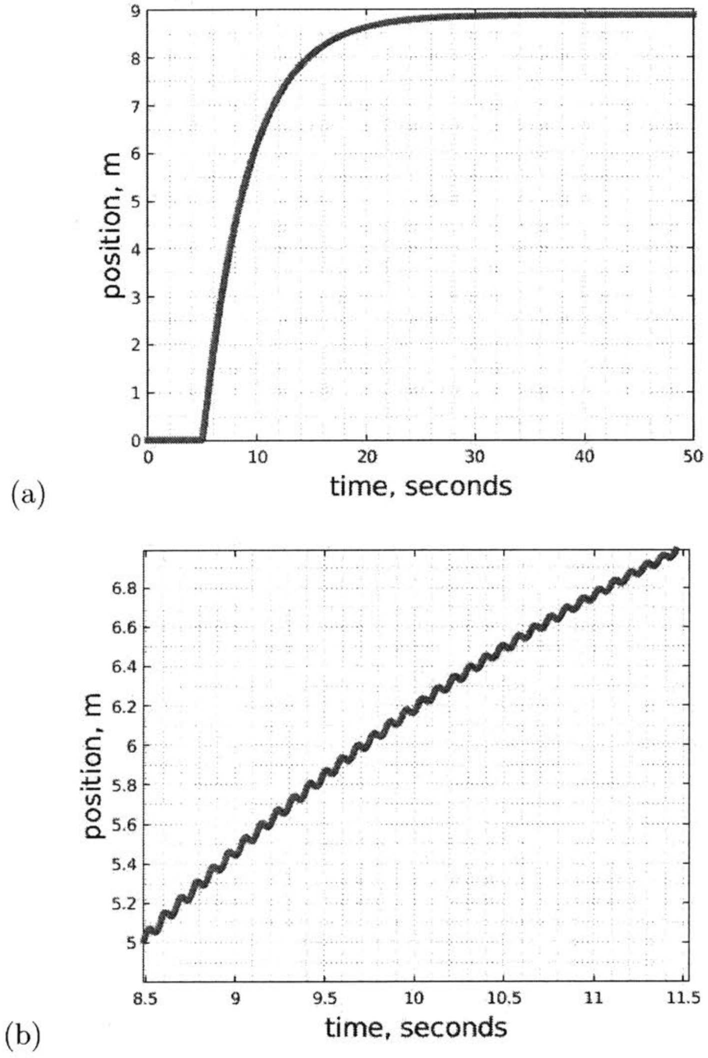


Figure 3-4: SEA's response to a short pulse at 5 sec. The model responds like (a) a first order system. The second mode creates minimal oscillations which can only be seen by magnifying the graph (b). The effect of the second mode on the system is negligible.

As discussed in Chapter 2, the Baxter Research Robot comes with position and torque controllers at the joint level which are not documented and cannot be adjusted by the user. It is not known how the dynamics of the SEAs are addressed in these controllers. As a first approximation, the stiffness of the flexure of the SEA will be neglected. This reduces a single arm to a 14th order system (2nd order in each DOF). Additionally, as the actuator and the linkage are assumed to move as a rigid body,

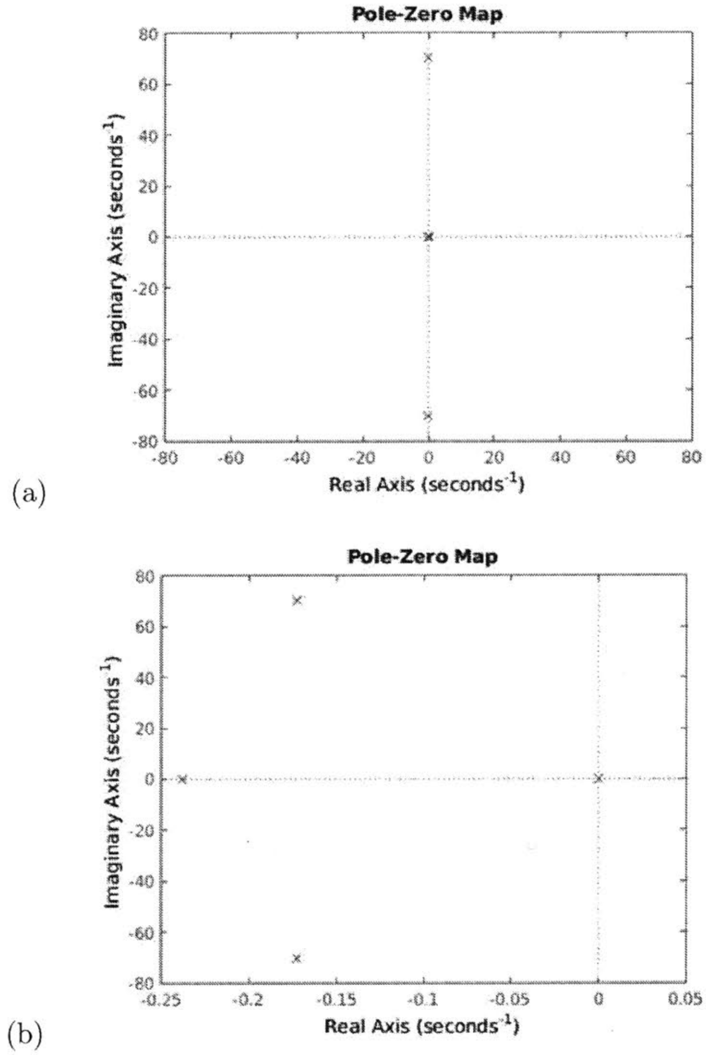


Figure 3-5: Pole Zero plot of SEAs with (a) equal axis and (b) zoomed in. The complex poles are close enough to the origin that at even modest gains they could cross into the right half plane.

their inertias may now be added such that the equations for the whole robot simplify to

$$(M(q) + I_r)\ddot{q} + D\dot{q} = \tau_{motors} \tag{3.12}$$

However, it is important to note that the total apparent inertia and damping at each joint are not only caused by the mechanical properties of the system, but also by the control system implemented at each actuator.

In state space form, the simplified, linearized model may be written as

$$\frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & -(M+I_r)^{-1}D \end{bmatrix}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ (M+I_r)^{-1} \end{bmatrix}\tau_{motor} \tag{3.13}$$

$$\dot{x} = \begin{bmatrix} 0 & J \end{bmatrix}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix}\tau_{motor} \tag{3.14}$$

3.2 Characterization of Reflected Motor Inertia: Theory, Experiment Setup, and Methods

As discussed in Chapter 2, the dynamics of Baxter's SEAs are not included in the URDF. This is not unique to Rethink Robotics. When creating a dynamic model of a robot, the reflected inertia of the motors, I_r, is often neglected. For instance, many of the dynamic simulation tools on the market today such as Open Dynamics Engine (ODE, 2007) were not initially designed to account for actuation dynamics. While there are ways to account for this, such as placing an infinitely small link with the desired apparent mass of the motor between the transmission and the actual link, there is still no infrastructure in place to account for the interactive dynamics of a robot's actuators.

Depending on the gear ratios used in a robot's design, the reflected inertia of the motors could be a nontrivial source of inertia. For large enough gear ratios, it could be the primary source. The fact that current, commonly used dynamic modeling tools make it inordinately difficult to account for parameters of the transmission systems which will have a nontrivial effect on the overall reaction of the robot implies there is a misconception in the field as to the significance of these dynamics on the robot's performance.

Granted, it is possible to lower the apparent inertia of an actuator that has force/torque feedback by increasing its controller gains. However, (Colgate, 1988) has shown that if the actuator's loop gain is greater than 2, passivity—an effective method to guarantee coupled stability—is compromised. As a loop gain of 2 is extremely low, any gain that would significantly alter the apparent inertia of an actuator would compromise passivity. This will be discussed further in Chapter 4. Passivity is not the only way to achieve coupled stability. However, it is an effective tool which could be of great use as the field of robotics works towards achieving more contact tasks. This will be further discussed in Chapter 4. It would be better to include the apparent inertia of a motor in a robot's dynamic model and reserve the use of passivity than to neglect the apparent inertia and lose such an effective tool by increasing force-feedback gains.

No specific technical data such as motor inertia or feedback gains pertaining to the SEAs are publicly available to users of the Baxter Research Robot. As such, before the significance of the reflected motor inertia could be evaluated, it first needed to be determined on Baxter. This was completed for two of Baxter's joints.

3.2.1 Experiment Setup

The inertia of two of Baxter's joints was characterized in situ: Joint W1 and Joint E1. For joint W1, two distinct methods were applied. The first assesses the interactive dynamics of the robotic limb based on direct measurement of the end-point behavior—interactive dynamics. The second considers the forward path dynamics by exciting the robot using its own actuators. Inertia should not differ between these two methods. However, it should be noted that not all dynamic parameters can be characterized using forward path dynamics. For instance, non-linear friction parameters can differ between interactive and forward path dynamics.

In the first method, the distal link was isolated from the rest of the arm so there would be no interactive dynamics from the rest of the linkages and oriented to rotate perpendicular to gravity (see figure 3-6). A torsion spring was connected to the arm such that its axis of rotation was collinear with Joint W1. A range of torque step commands were applied to the actuator, and the resulting motion was recorded. Using the resulting step responses, the inertia, motor damping, and directional coulomb friction were then estimated using a least-squares fitting procedure. This process was then repeated with three other springs.

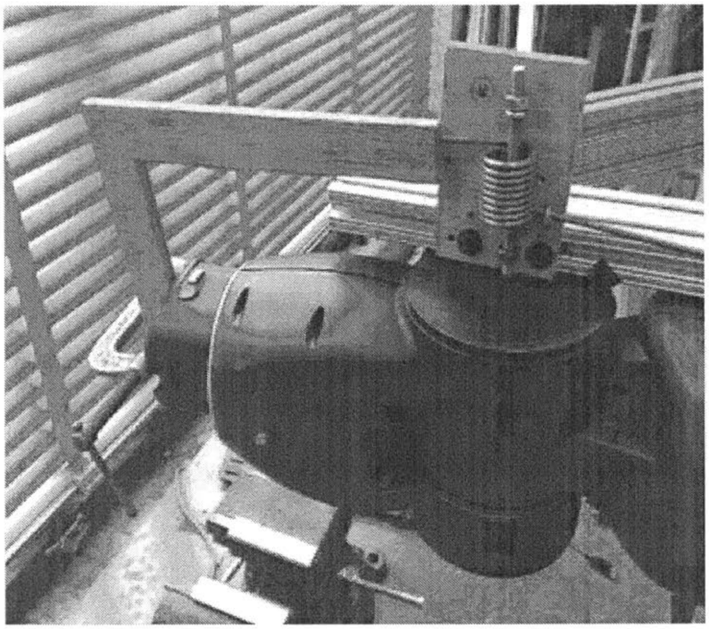


Figure 3-6: Side view of experiment setup for characterizing Baxter's inertia using an external excitation. The link proximal to Joint W1 was constrained to eliminate any reaction forces in the upper linkages.

In the second method, an approach was taken very similar to that of (Armstrong, Khatib, & Burdick, 1986). With the proximal link still in the clamp, Joint W1 was given a virtual stiffness utilizing force feedback to simulate a torsion spring being placed at each joint. An external impulse was applied and the resulting frequency was used to calculate the total apparent inertia.

For both methods, the inertia due to the distal linkages was then subtracted off, leaving the reflected inertia due to the transmission system of the rotating joint. This process was repeated for multiple virtual joint stiffnesses to ensure the inertia did not vary.

Once it was verified that the virtual spring method provided the same inertia values as with the real spring (see Results), the virtual spring method was then repeated for Joint E1. Joints which were distal to Joint E1 were given a very high virtual stiffness so the arm could be treated as one solid linkage.

3.2.2 Results

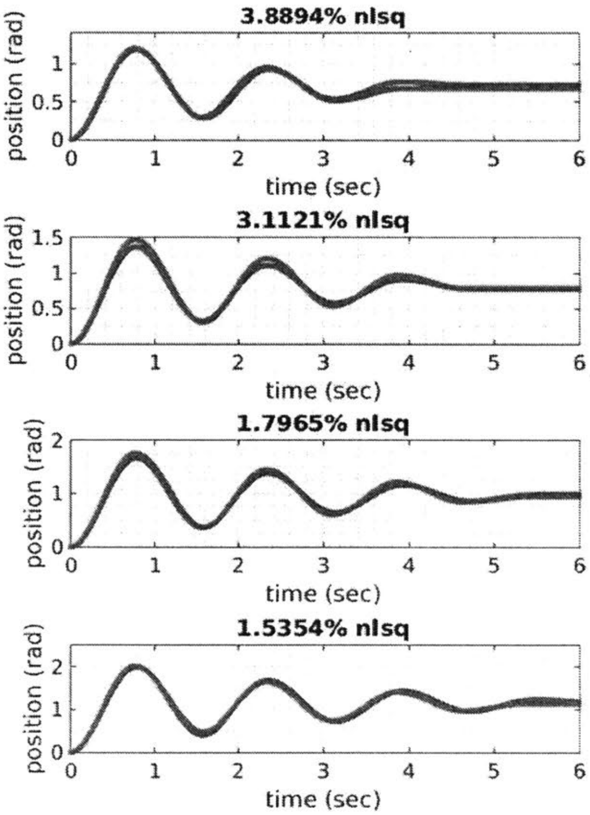


Figure 3-7: Fitting the model (blue) to experimental data (red) of a step response for four different applied torques. The variance accounted for (VAF) in the second order model is 96%. The normalized least square (nlsq) fit is the least squares fit of the model normalized to the steady state error.

As shown previously, the equations of motion for SEAs are a fourth order model. However, given the stiffness of the flexure in the SEA, it had been assumed that the rigid body mode would dominate the behavior. The model was treated as a second order system for the parameter estimation. This assumption was verified experimentally. The results of the least squares fit from using a real spring are shown in figure 3-7. Using the real spring, the normalized least squares fit accounted for 96% of the variance with an inertia of $0.047kgm^2$ for Joint W1.

Using the virtual spring method, the mean of the inertia measurements for Joint W1 was found to be $0.044 \pm 0.003 kgm^2$. The mean of the inertia for Joint E1 using the virtual spring was $0.60 \pm 0.039 kgm^2$. Given that the peak torque in the two sets of joints differed by an order of magnitude, and this is most likely achieved by increasing the gear ratio, it is reasonable that inertia would change by a similar magnitude.

3.3 Assessing the Importance of Reflected Motor Inertia

Once it was verified that Baxter's actuators had measurable reflected inertia, the importance of including these parameters in the model was assessed. For interactive tasks, having an accurate model of the inertia of an arm is arguably most important when the end effector transitions from free to constrained motion. Before any of the tasks outlined in Chapter 1 can be accomplished, the robot must be able to make contact with the surface in question. In making this transition, the robot must also not transmit too much force to its environment. For this to be accomplished, a more detailed model of Baxter may be required than what is currently used.

Granted, there are some methods which allow for neglecting this initial contact. For instance, moving slowly until contact with the environment is detected—a process known as guarded moves (Mason, 1981)—and hybrid control (Craig & Raibert, 1979) have both been proposed as a sufficient framework for transitioning between free and constrained motion. However, due to the non-zero closing velocity at the transition between free and constrained motion, there will always be an initial impact no matter how slowly the robotic arm is moving. Having an accurate model of the magnitude and principal directions of the apparent mass can be useful to understand how the robot will react to that impact, as well as determine at what speed it becomes important to consider the impact in the design of the control system.

Furthermore, an accurate model of a robotic arm's inertia has become a limiting factor in some modern control techniques. One of the most common methods of utilizing kinematic redundancies in robotics is to implement a hierarchical control system using a dynamically consistent null space projector first developed by (Khatib, 1987) (Khatib, 1995). However, for these techniques to be implemented effectively, an accurate model of the inertia matrix is required. (Dietrich, Ott, & Albu-Schäffer, 2015) have demonstrated that while dynamically consistent approaches utilizing the inertia work better in simulation, a poor model of the inertia matrix negates any benefits in practice.

3.3.1 Constructing the Cartesian Inertia Matrix

Before understanding how the inertia of an arm affects an impact, the model of the inertia must first be transformed into the end-effector coordinate system: the Cartesian Inertia Matrix, Λ. The complete model for the Cartesian Inertia Matrix is

$$\Lambda = (J(q)(M(q) + I_r)^{-1}J(q)^T)^{-1} \tag{3.15}$$

M is the joint space inertia matrix due to the linkages which was derived and explained in Chapter 2, I_r is the inertia of the motors reflected through a transmission system, and J is the Jacobian mapping the velocity from joint space to the end effector.

The Importance of Reflected Inertia

To understand how the reflected inertia of its actuators would affect Baxter's apparent inertia, a model of Baxter was constrained to planar motion by only allowing joints S1, E1, and W1 to move.

This planar model was then analyzed both with and without the calculated apparent inertia. The apparent inertia of joint S1 has not been determined. However, it is rated for the same "peak torque" as joint E1. Given that gear ratios most likely increase further up the arm, the apparent inertia of joint S1 will most likely be larger than that of joint E1. Because of this, a conservative assumption is that the inertia from joint S1 is equal to that of E1. If anything, the apparent mass will actually be

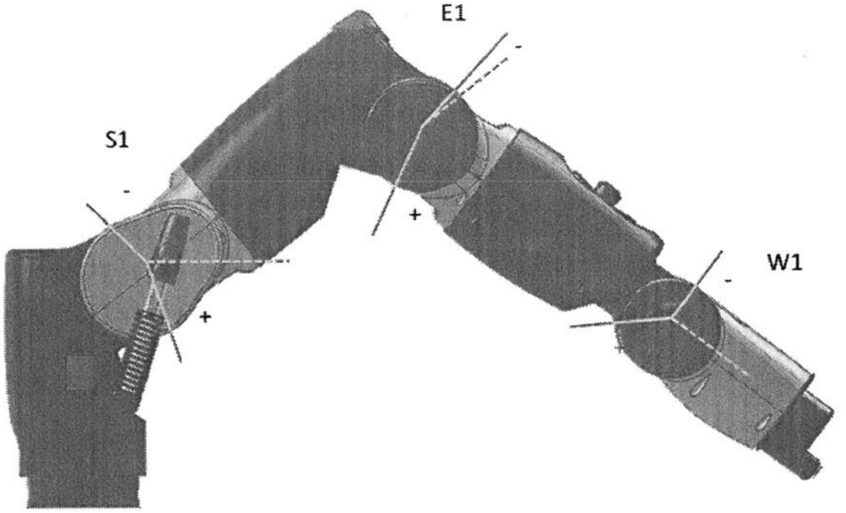


Figure 3-8: A model of Baxter was constrained to planar motion by only letting joints S1, E1, and W1 move (Hardware, 2015).

larger. The translational apparent masses both with and without the motors for a given configuration is shown below. The overall effect of including the motor inertia is about a 63% increase in the maximum mass and a 58% increase in minimal mass.

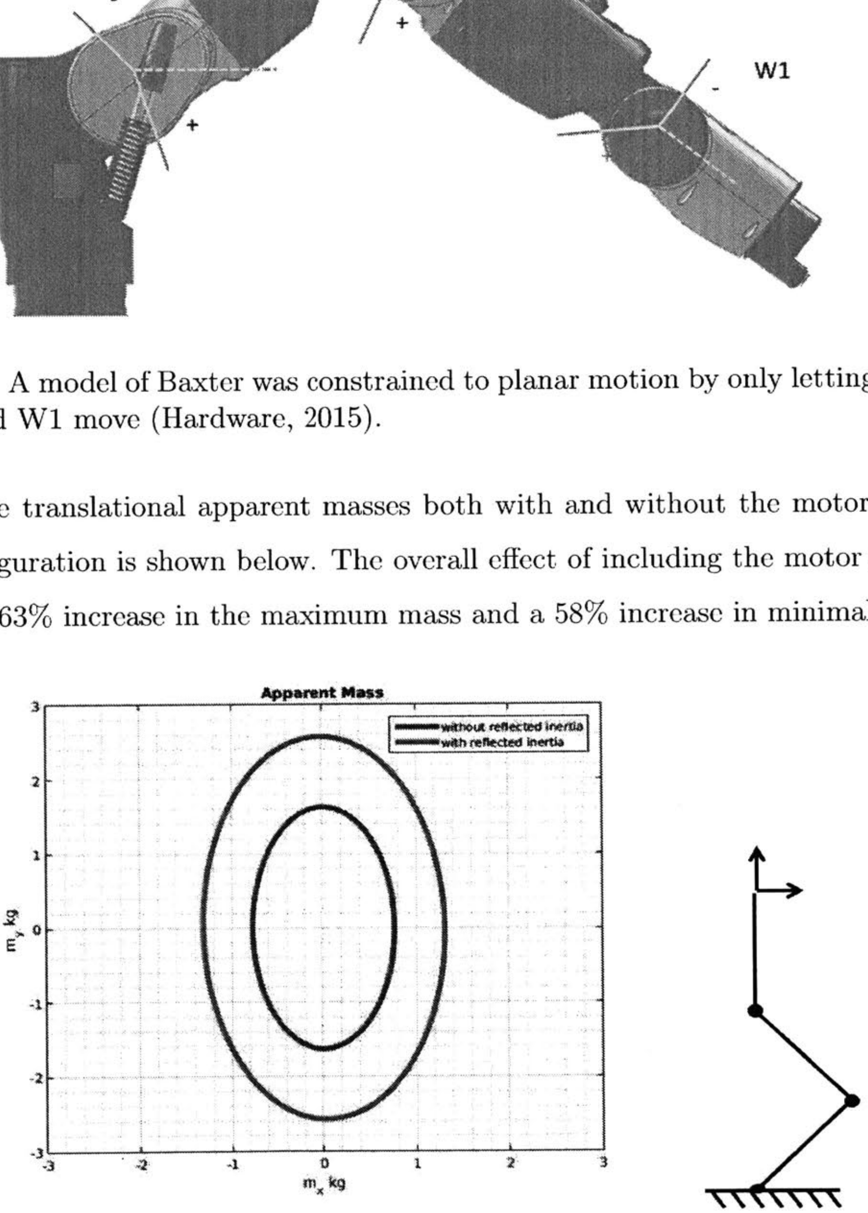


Figure 3-9: Apparent mass with (red) and without (blue) the apparent inertia of the model for the shown configuration. Referenced to ground, Joint S1 is $\frac{3\pi}{4} rad$, joint E1 is $\frac{3\pi}{4} rad$ and joint W1 is $\frac{\pi}{2} rad$. While the apparent mass of the motors did not affect the orientation, the change in magnitude is nontrivial.

Table 3.1: Apparent Mass at End Effector

Parameters	With Reflected Inertia	Without Reflected Inertia	Ratio
Maximum mass (kg)	2.574	1.628	1.58
Minimum mass (kg)	1.31	0.7683	1.71
Shape Factor	1.96	2.12	0.92

To provide a general sense of the significance of the difference in the apparent mass, consider an example of Baxter completing a contact task. Using the Cartesian impedance controller which will be described in Chapter 4, Baxter can maintain a trajectory through free space, make contact with a hard surface, slide along that surface, and then transition back into uncoupled motion. When Baxter makes the transition from free to constrained motion, its end effector is moving at approximately $0.1\frac{m}{s}$. Assuming the contact time with a hard surface is similar to that shown in figure 3-3—around $70msec$—and the apparent mass of the model is representative of that of the complete system, the force due to the impact at that general configuration would be either $3.67N$ or $2.34N$. The impact would transmit 1.6 times more force than the current model predicts. Even with a motion as slow as this, the apparent inertia of the motors is significant.

Chapter 4

Developing an Impedance Controller for Baxter

Once a dynamic model was created for Baxter, a control system needed to be implemented which could accomplish the tasks outlined in Chapter 1. All the tasks require that the robot's end effector is either partially or fully coupled to the environment in which it is interacting. For these tasks to be successful, the robot must maintain coupled stability throughout the interaction. One effective method to guarantee coupled stability is to design a control system whose external port dynamics interact passively with its environment. An external port is a point on the system where energy may be exchanged with the environment. A passive system is one which can store and release energy, but cannot continuously supply power. For instance, any combination of masses, springs, and dashpots is a passive system.

To achieve passivity, a control scheme which enforces a specific dynamic interaction instead of a position or force is required. Essentially the arm needs to respond to a disturbance like a suspension system. Two such control schemes exist: (1) Admittance control, which specifies a dynamic motion reaction to an input force and (2) Impedance control, which specifies a dynamic force reaction to an input motion. While mathematically these two control schemes are the inverse of each other, impedance control is frequently the more desirable method to implement. This is because a robot will most often be interacting in an environment composed of position constraints instead of force constraints. If a robot is moving in contact with a solid surface, it cannot impose a specific position on that surface. For this reason, impedance control was chosen as the preferred method of creating a passive system.

However impedance control alone still does not guarantee passivity. For instance if a system has a net negative stiffness, it meets the definition of an impedance controller: a dynamic response to an input motion. However by driving the system away from an equilibrium point instead of towards it, the control system could continuously supply power. Passivity would be compromised. To avoid this, a feedback controlled system must have a positive, real impedance to guarantee passivity. By imposing this constraint on an impedance controller, a system will have guaranteed coupled stability with any passive environment.

Because of this constraint, feedback control of impedance can only provide a limited range of impedances. (Colgate & Hogan, 1989) have shown that if the force feedback loop gain increases above 2, the passivity of the interactive behavior is compromised. As a loop gain of 2 is extremely low, this places a large restriction on achievable impedances. However, feedback control is not the only way to modulate impedance. As shown by (Hogan, 1985), the configuration of an arm can highly affect the impedance at the endpoint. When an arm has kinematic redundancies, the impedance can even be modulated by the configuration of the arm without changing the endpoint position or orientation.

With this goal of coupled stability via a passive system in mind, a force based Cartesian impedance controller as outlined in the block diagram of figure 4-1 was developed for Baxter using both the Joint Torque Control Mode and the PyKDL described in Chapter 2. For clarity, a simplified example of a force based impedance controller will first be described. This example of force based impedance control will then be expanded to Baxter. Next, alternative methods of modulating Baxter's endpoint impedance other than through feedback such as utilizing redundancies in the arm will be discussed. Finally, a description of "pseudo" hybrid control will be provided.

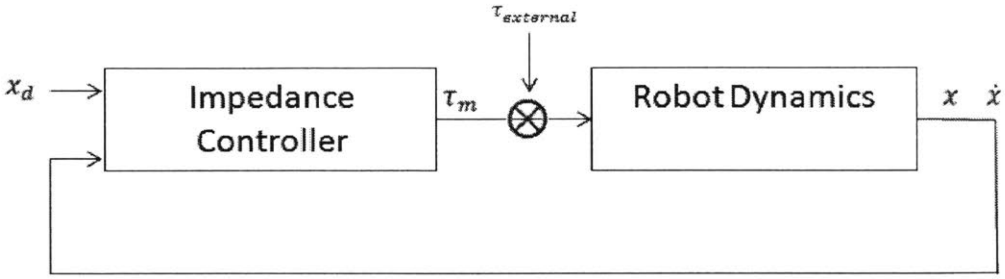


Figure 4-1: Simplified diagram of Baxter's force based impedance controller. A desired position x_d, and the actual position and velocity are fed into the impedance controller which then calculates the required torque to be applied to the SEAs.

4.1 A Simplified Example of Impedance Control Using Force Feedback

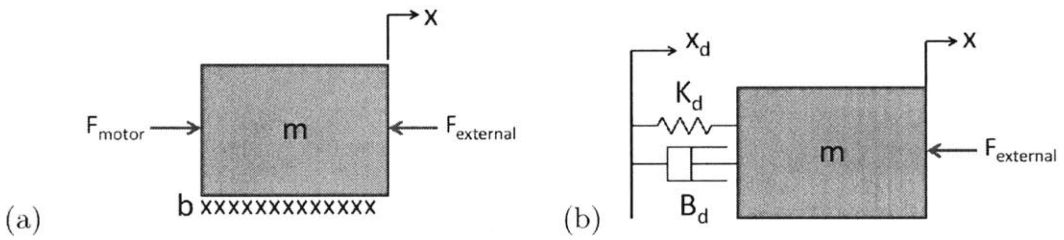


Figure 4-2: General concept of impedance control which will be applied to Baxter. Force feedback is used to make (a) a simple mass with some damping react like (b) a desired mass, spring, dashpot system.

Consider a single mass driven by a motor with some inherent damping constrained to translate in 1 DoF as shown in figure 4-2. The overall goal is to have the system achieve a specified dynamic behavior. This behavior can be any well-defined impedance, however for this case it will be to react like a linear mass, spring, and dashpot system.

To accomplish this, the equations of motion of the specific system must be set equal to the desired reaction to some external force. The equation of motion for the general mass/damping system is

$$m\ddot{x} = -b\dot{x} + F_{motor} - F_{external} \tag{4.1}$$

where F_{motor} is the force due to the motor and $F_{external}$ is any external force acting on the system. The desired reaction to some input force is

$$M_d(\ddot{x}_d - \ddot{x}) = B_d(\dot{x}_d - \dot{x}) + K_d(x_d - x) - F_{external} \tag{4.2}$$

where x_d, $\dot{x}_d$, and $\ddot{x}_d$ are desired position, velocity, and acceleration respectively. For the purposes of this project, $\dot{x}_d$ and $\ddot{x}_d$ will both be set to zero. As such, both M_d and B_d will always oppose the desired motion, x_d. The equation simplifies to

$$m\ddot{x} = -b\dot{x} + F_{motor} - F_{external} \tag{4.3}$$

Setting the two equal to each other results in

$$m\ddot{x} = -b\dot{x} + F_{motor} - K_d(x_d - x) + (B_d - b)\dot{x} \tag{4.4}$$

The force commanded to the motor is

$$F_{motor} = (m - M_d)\ddot{x} + K_d(x_d - x) + (b - B_d)\dot{x} \tag{4.5}$$

To implement this control scheme, the object's, position, velocity, and acceleration are required. This can either be determined by sensors on the system or through the creation of observers. For the purpose of this project, the inertia of the system was not shaped via feedback. This was accomplished by setting the desired inertia, M_d, to the actual inertia, m. Apparent inertia at Baxter's endpoint will instead be shaped via geometric configuration of the arm. This simplifies the control signal to

$$F_{motor} = K_d(x_d - x) + (b - B_d)\dot{x} \tag{4.6}$$

Ideally, the damping in the model, b, would be equal to the damping of the actual system. However, there will most likely be some small error. To differentiate between the modeled friction and the actual friction, the modeled friction will be called $\check{b}$ henceforth. The new equation of motion the system is

$$m\ddot{x} = -b\dot{x} - F_{external} + K_d(x_d - x) + (\check{b} - B_d)\dot{x} \tag{4.7}$$

Written in state space form, the closed loop equation of motion for this system is

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -m^{-1}K_d & -m^{-1}(B_d + \check{b} - b) \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ m^{-1} \end{bmatrix} \begin{bmatrix} K_d x_d - F_{external} \end{bmatrix} \tag{4.8}$$

In practice, what frequently occurs is that the actual damping of the motors will be canceled via feedback and completely replaced by a desired damping, B_d. However, this does not necessarily have to be the case. The real damping of a system—as well as the stiffness and the inertia—could be utilized to achieve the desired impedance instead of being completely replaced.

4.2 Closed Loop Control of Baxter

The concepts presented in this general 1 DoF case can be implemented on much more complex systems. In the case of a robotic arm, this requires mapping the external forces and desired endpoint dynamics from end effector space to configuration space via the Jacobian. If there is a desired stiffness and damping at the end effector such that

$$F_{external} = K_d(x_d - x) + B_d\dot{x} \tag{4.9}$$

The simplified 1 DoF case can be used to create the impedance controller. The state space equation has the same general form as the 1 DoF case

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\Lambda^{-1}K_d & -\Lambda^{-1}(B_d + \check{b} - b) \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ m^{-1} \end{bmatrix} \begin{bmatrix} K_d x_d - F_{external} \end{bmatrix} \tag{4.10}$$

where Λ is the Cartesian Inertia Matrix as derived in Chapter 3.

4.3 Controlling and Utilizing Redundancies to Modulate Impedances

This control system would be sufficient for a robot with no redundancies. However, each of Baxter's arms has an extra DoF. The endpoint impedance controller only controls 6 DoF. As such it cannot affect the null space created by this extra joint. Adding redundancies to the arm requires further care in designing the control system. More traditional controllers also encounter this dilemma. For instance, in velocity control, the end effector velocity is typically mapped to the joints using a Jacobian pseudo-inverse developed by (Whitney, 1969). However, as demonstrated by (Klein & Huang, 1983), redundancies in the arm cause the pseudo-inverse to be non-integrable. This leads to drift in the joint space during tasks. The joints can be driven to unpredictable configurations and at least one joint may approach its limit.

One method both addressing and utilizing kinematic redundancies in robotics is to implement a hierarchical control system. A primary command determines what the endpoint is to do. Second—and third—level commands are then projected into the null space established by the primary command (Dietrich, Albu-Schäffer, & Hirzinger, 2012). For instance, if a robot was tasked with picking up a cup of coffee, it could have a primary command to send the end effector to a specific position to grab the cup, and the secondary command to keep the arm from running into the table the coffee was sitting on.

An alternative approach, enabled by impedance control, takes advantage of the "compositionality" of impedances acting on an inertial body: the principle of superposition. The combined action of multiple impedances, whether linear or nonlinear, is obtained simply by summing the forces they generate. Thus one impedance—a "base" or "reference" joint stiffness—can be specified to manage the redundant degrees of freedom while other impedances implement the desired interactive behavior. An example of this is depicted in figure 4-3.

By mapping the joint compliance from configuration space to end effector space, the stiffness at the end effector may be determined. When no external torque is acting

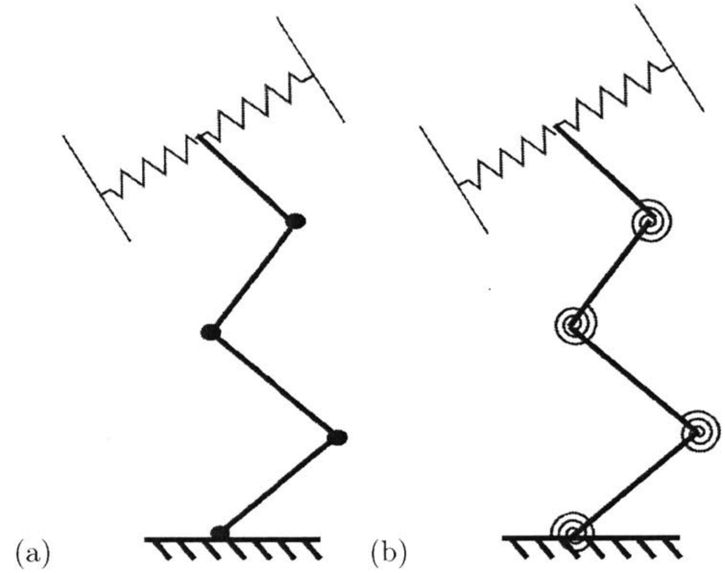


Figure 4-3: a. An example of a planar root with an additional degree of freedom. Only specifying the endpoint stiffness will leave an unused degree of freedom in the configuration space that could potentially drive the system to unpredictable configurations. b. By applying a stiffness to one or more of the joints, the additional degree of freedom is removed and can be utilized as an additional method to shape the endpoint impedance

on the system, the torque at the joints from the virtual springs is

$$\tau = K_{joints} dq \tag{4.11}$$

Using the force map from configuration space to the end effector, and assuming the joint stiffness, K_{joints}, is non-singular, this may be rewritten as

$$K_{joint_s}^{-1} J^T(q) F = dq \tag{4.12}$$

Multiplying by the Jacobian

$$J(q) K_{joint_s}^{-1} J^T(q) F = dx \tag{4.13}$$

Rearranging, and assuming the end effector compliance is nonsingular, the force at the end effector from the displacement of the joint springs is

$$F = (J(q)K_{joint_s}^{-1}J^T(q))^{-1}dx \tag{4.14}$$

By the principle of superposition, the effective stiffness from the joints may be added to the stiffness of the virtual springs at the end effector such that

$$K_{net} = (J(q)K_{joint_s}^{-1}J^T(q))^{-1} + K_{end} \tag{4.15}$$

However, as shown by (Mussa-Ivaldi & Hogan, 1991), when there is a change in force applied to the end effector, the change in torque at the joints is

$$d\tau = J^T dF + dJ^T F \tag{4.16}$$

This additional term, $dJ^T F$, will have an effect on the net stiffness. The i^{th} component of $dJ^T F$ may be written in terms of the robot's forward kinematics, x, as

$$(dJ^T F)_i = \sum_{m=1}^{K}(\sum_{j=1}^{N}\frac{\delta^2 x_m}{\delta q_j \delta q_i}\delta q_k)F_m = \sum_{j=1}^{N}(\sum_{m=1}^{K}\frac{\delta^2 x_m}{\delta q_j \delta q_i}F_m)\delta q_k \tag{4.17}$$

Defining the internal summation as Γ, where the i,j component is

$$\Gamma_{i,j} = \sum_{m=1}^{K}\frac{\delta^2 x_m}{\delta q_j \delta q_i}F_m \tag{4.18}$$

The change in torque may be written as

$$d\tau = J^T dF + \Gamma dq = J^T dF + \Gamma K_{joint_s}^{-1} d\tau \tag{4.19}$$

re-arranging

$$d\tau = (I - \Gamma K_{joint_s}^{-1})^{-1} J^T dF \tag{4.20}$$

Multiplying both sides by the inverse of the joint stiffness

$$dq = K_{joint_s}^{-1}(I - \Gamma K_{joint_s}^{-1})^{-1} J^T dF = (K_{joints} - \Gamma)^{-1} J^T dF \tag{4.21}$$

Finally, multiplying both sides by the Jacobian,

$$dx = (J(K_{joints} - \Gamma)^{-1} J^T) dF \tag{4.22}$$

re-arranging, the restoring force at the end effector from the effective stiffness of the virtual springs at the joints now contains an additional force dependent term

$$dF = (J(K_{joints} - \Gamma)^{-1} J^T)^{-1} dx \tag{4.23}$$

By principle of superposition, the effective stiffness from these spring may be added to the stiffness of the virtual springs at the end effector such that

$$K_{net} = (J(K_{joints} - \Gamma)^{-1} J^T)^{-1} + K_{end} \tag{4.24}$$

As such, the net stiffness is dependent on both configuration and force applied to the endpoint. This could introduce further complications in completing a contact task—as the robot increases the force it applies to the surface, the stiffness could change. The control signal sent to the joints is

$$\tau_{commanded} = J(q)^T K_{end} J(q) dx + K_{joints} dq \tag{4.25}$$

where *dq* is the deflection of the joints from equilibrium position. Overall, there are two ways to control the redundant degrees of freedom: joint stiffness, K_{joints}, and the joint equilibrium position, q_d. Setting different equilibrium positions for joint stiffness by varying *dq* is yet another factor to consider when designing the endpoint impedance. As this method is less likely to change the endpoint stiffness, it may be preferable. However, it would then require solving inverse kinematics to find q_d corresponding to x_d.

Finally, for simpler tasks which do not require a full 6 DoF motion, the control of the arm could be simplified by assigning a high stiffness to some of the joints: effectively removing those degrees of freedom from the system.

4.4 Shaping Inertia Using Geometric Configuration

Once a controller was built which could control the redundancies in the system, it could be utilized to shape the effective endpoint impedance via geometric configuration. As previously stated, for passivity to be preserved, feedback control can only provide a limited range of impedances. However, as is evident from the derivation of Baxter's equations of motion, the end effector inertia, stiffness, and damping are all highly dependent on the configuration of the arm. For the purposes of this project, the inertia was shaped solely by using geometric configuration. As discussed in the previous chapter, the Cartesian Inertia Matrix is

$$\Lambda = (J(q)(M(q) + I_r)^{-1}J(q)^T)^{-1} \tag{4.26}$$

This 6x6 matrix may be broken down into four 3x3 sub-matrix blocks such that

$$\Lambda = \begin{bmatrix} J_v(M + I_r)^{-1}J_v^T & J_v(M + I_r)^{-1}J_w^T \\ (J_v(M + I_r)^{-1}J_w^T)^T & J_w(M + I_r)^{-1}J_w^T \end{bmatrix}^{-1} \tag{4.27}$$

where J_v is the translational component of the Jacobian and J_w is the rotational component (Khatib, 1995). The upper left quadrant of the matrix represents the translational component of the inertia matrix. The lower right is the rotational inertia. The other two quadrants of the matrix represent the coupling between the rotational and translational inertia. The magnitude and direction of both the translational and rotational components of the matrix may be described using the inertial ellipsoid as developed by (Hogan, 1985). While the concept of inertia is more widely used and understood, it is sometimes preferable to instead modulate the inverse of the inertia—mobility. This is due to the fact that the effective inertia is not always defined. For instance, when a robotic arm is fully extended, the effective inertia reaches a singularity. The mobility, however, is still defined.

As an example of the inertia's geometric dependence, Baxter's translational inertia ellipsoids are show for three different arm configurations in figure 4-4. As can be seen, the effective inertia can be highly anisotropic. In cases a and b, the location of the end effector is not changed. Because of the redundancy in the system, the apparent inertia can be modulated without changing the position of the end effector.

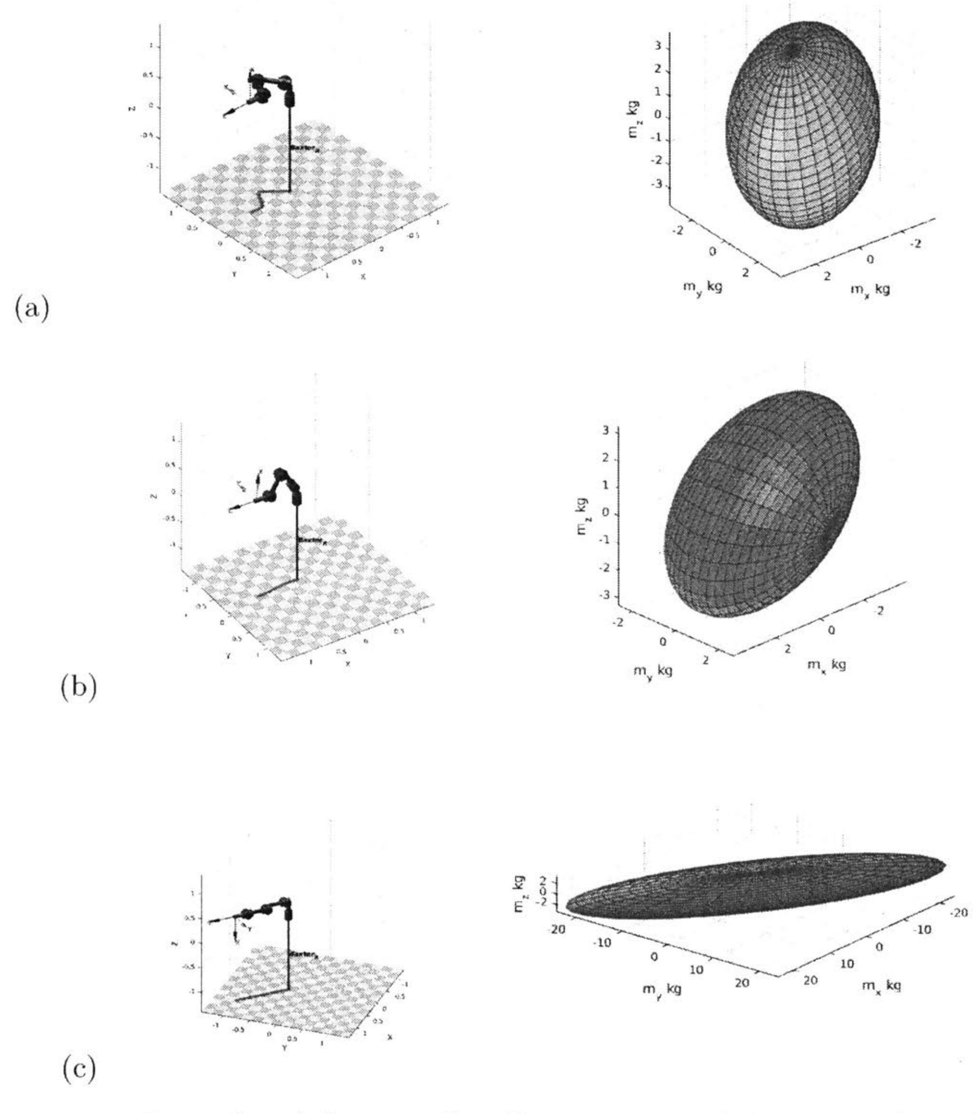


Figure 4-4: Examples of changing the effective inertia of the arm without changing the endpoint position. By moving Baxer's elbow from (a) the horizontal plane to (b) nearly the vertical plane, the effective inertia in the x and y direction changes in both magnitude and orientation. (c) Extending the arm close to full extension such that it is nearing a singularity causes the inertial ellipsoid to be highly anisotropic.

The academic community does not yet have a strong understanding of how to interpret the coupling terms of the Cartesian Inertia Matrix. They are quite often considered negligible (Muhammad, 2011). Certain light weight robots such as the Puma have been proven to be diagonally dominant (Roveda, 2015). However, for Baxter there are many configurations where the coupling terms are the same order of magnitude as the values on the diagonal. In practical terms, this means that when Baxter impacts a surface, even if the end-effector motion is strictly normal to the surface, there will be a measurable reaction torque at the end effector.

4.5 Planning Control of Both Arms

Up to this point, the net impedance of only one arm has been considered. With 14 DoF between both arms, making control decisions with Baxter can present an overwhelming number of options. A humanoid bipedal robot would have even more. For instance, the Atlas robot has 28 DoF (BostonDynamics, 2016). It would be useful to have a simplified framework for making control decisions under these circumstances.

The principal of superposition has already been applied to one arm in utilizing the redundancies of the joint. The stiffness from the virtual springs at the joints was added to the stiffness of the virtual spring at the end effector. In a similar manner, it is possible to design total impedance using a "modular" approach. The implementation of this concept on Baxter is deferred to future research. However, each arm could be treated as a single suspension system, and the net impedance at the end effector would be the sum of both arm impedances. This would simplify the overall control of a multi-degree of freedom robot. By considering each limb as its own "impedance module" and adding the net results together, a more intuitive framework for making control decisions could be created.

One example of how this method would be useful is in eliminating unwanted effective dynamics of the arm. Consider the simplified, three-linkage planar mechanism used as an example for deriving the JSIM in Chapter 2. In the configuration shown in figure 4-5, the effective endpoint inertia has a coupling term in both the x and y direction such that

$$\Lambda_{right} = \begin{bmatrix} 9.18 & -2.5 & 6.68 \\ -2.5 & 9.18 & -2.5 \\ 6.67 & -2.5 & 5.86 \end{bmatrix} \tag{4.28}$$

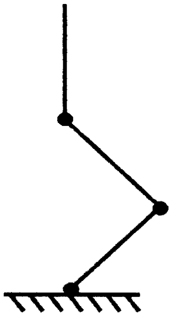


Figure 4-5: Right arm configuration

While the author has yet to find a useful analytical interpretation of these coupling terms, it should be noted that a symmetric design of the configuration of the arms can result in their reduction or elimination. By having the left arm mirror the configuration of the right arm, as shown in figure 4-6, the coupling term in the y direction will be equal and opposite such that

$$\Lambda_{left} = \begin{bmatrix} 9.18 & 2.5 & 6.68 \\ 2.5 & 9.18 & 2.5 \\ 6.67 & 2.5 & 5.86 \end{bmatrix} \tag{4.29}$$

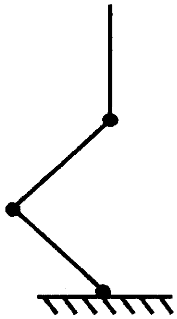


Figure 4-6: Left arm configuration. By having the left arm mirror the configuration of the right arm, the coupling term in the y direction will be equal and opposite.

Further analysis is required to understand the meaning and source of these coupling terms. However, consider the case shown in figure 4-7 where the arms are connected at the end effector. Traditionally, this situation would be analyzed as a closed loop kinematic chain. As discussed in Chapter 2, dynamic libraries such as Orocos do not currently handle closed loop chains (Orocos, 2016). A simpler approach would be to instead treat the arms as two "impedance modules." As the end effectors have to move together, their inertias can be added such that

$$\Lambda_{total} = \begin{bmatrix} 18.36 & 0 & 13.36 \\ 0 & 18.36 & 0 \\ 13.36 & 0 & 11.72 \end{bmatrix} \tag{4.30}$$

The coupling term in the y direction will be eliminated, and the end effector will be able to come into contact with a surface by moving normal to it without the need to compensate for joint torques induced by the impact.

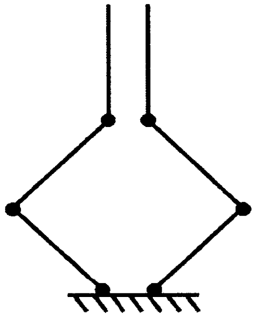


Figure 4-7: By connecting both arms, the coupling in the x direction cancels. Calculating the net impedance using this modular approach eliminates the need for closed loop kinematic chains. The separation of the arms is for graphical clarity.

4.6 Summary of Cartesian Impedance Controller

To summarize, using both feedback and geometric configuration, Baxter's end effector will respond to a disturbance as the following mass, spring, dashpot system

$$\Lambda\ddot{x} + B\dot{x} + K_{net}(x_d - x) = F_{ext} \tag{4.31}$$

The equilibrium position of the end effector is determined via x_d. Strategies for placement of the virtual equilibrium position will be discussed in the next section. The inertia is solely controlled by geometric configuration such that

$$\Lambda = (J(q)(M(q) + I_r)^{-1}J(q)^T)^{-1} \tag{4.32}$$

The stiffness is controlled by both feedback and geometric configuration

$$K_{net} = (J(q)K_{joints}^{-1}J^T(q))^{-1} + K_{end} \tag{4.33}$$

The damping is controlled solely via feedback. There is some inherent damping in the joints. In particular, joint S0 displays a non-trivial amount of stiction. However for an initial design it was assumed that the damping assigned via feedback would be much greater than the inherent damping. Characterization of Baxter's non-linear damping parameters is deferred to future work.

4.7 Pseudo Hybrid Control

The feedback impedance controller essentially makes the endpoint act like a suspension system about some position such that

$$F = K(x_0 - x) - B\dot{x} \tag{4.34}$$

However, for certain contact tasks it is required that the arm exerts force on its environment. This has previously been accomplished using a method called hybrid control. Originally developed by (Craig & Raibert, 1979), hybrid control imposes a specified force normal to the surface and a position tangent to the surface.

In the event that either the position control tangent to the surface (y) or force control normal to the surface (x) needs to display some degree of compliance (instead of pure force or position control), an impedance controller could be used to create a "pseudo hybrid" controller where the position and force controllers have their own desired suspension systems added to them such that

$$F_x = f_0 + K_x(x_0 - x) - B_x\dot{x} \tag{4.35}$$

and

$$F_y = K_y(y_0 - y) - B_y\dot{y} \tag{4.36}$$

The stiffness and damping in each direction can be tuned independently of each other. By commanding an anisotropic stiffness so that the end effector is stiff in one direction and compliant in the other, the arm will be capable of applying a relatively constant force in once direction while maintaining an accurate position in the other.

While this can be a very useful control scheme, one major drawback to both true hybrid control and this version of pseudo hybrid control is its inability to transition between free and constrained motion. Another form of pseudo hybrid control would be to not have a nominal force in the x direction independent of position such that

$$F_x = K_x(x_0 - x) - B\dot{x} \tag{4.37}$$

To apply a force to the surface using this method, the position of the virtual spring can be moved below the surface such that the new nominal force, f_k is

$$f_k = Kx_0 \tag{4.38}$$

This nominal force can now be modulated by either the stiffness or the position of the virtual spring. It also has the added benefit of being able to transition from free to constrained motion without changing control schemes. In free motion, the position of the end effector is controlled by moving the position of the virtual spring. Upon transitioning to constrained motion, the nominal force is controlled by moving the virtual position below the surface with which the end effector is interacting.

This nominal force is now dependent on both the stiffness and the virtual equilibrium position of the spring. As such, the accuracy of the force is dependent on the map of the environment. For instance, if the map of the environment has a nominal location for the surface that is off by some value δx, the desired force applied to the surface will have an error $K_x \delta x$.

Additionally, when using this method the force profile in the x direction must be taken into consideration when choosing K_x and x_0. Figure 4-8 compares two different spring/virtual equilibrium positions. While they would both apply the same nominal force to a surface, the configuration with the lower stiffness and the equilibrium position further from the nominal surface has a more constant force profile. However, this method is also not without its drawbacks. By selecting a low stiffness and stretching the "virtual spring" further below the surface, the controller stores significantly more energy than when the spring is closer to the contact surface. This could cause dangerously fast velocities should the robot lose contact with the surface. (Schindlbeck & Haddadin, 2015) address this force overshoot problem by using "energy tanks."

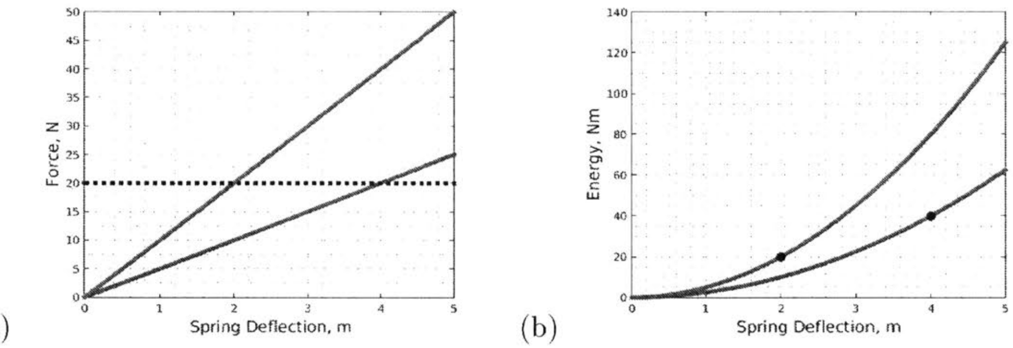


Figure 4-8: (a) Example force profile for two different springs. To achieve the same force (20N) one must be stretched twice as far as the other. For small variations in the deflection, the force will change less. (b) Example energy profile of same springs. Because the spring with lower stiffness has been stretched twice as far, it stores twice the amount of energy. This would result in larger velocities if the end effector were to lose contact with the surface

As with true hybrid control, a pseudo hybrid controller still enforces that the robot operates in a specific coordinate system. Because it commands a force along with a suspension system perpendicular to the surface, and a position tangent to the surface, the robot will not be robust to errors in the orientation of the map of its environment.

This force based impedance controller was implemented on Baxter and used in a number of scaled down decommissioning tasks. Its performance will be evaluated in the next chapter.

Chapter 5

Proof of Concept: Scaled Down Decomissioning Tasks

Once the force-based impedance controller described in Chapter 4 was implemented on Baxter, two scaled-down "proof of concept" tasks similar to those outlined in Chapter 1 were completed: 1) operating in a poorly mapped environment; and 2) scrubbing biofouling. Baxter has no force or motion sensors at its end effector. Mapping data between end-point and joint coordinates is accomplished via the Jacobian matrix. This design is what would most likely be implemented on a robot intended to operate in an extreme environment. It would be better to have the more expensive equipment—sensors—away from the primary point of impact. However, this may introduce limitations to how much a robot could use "tactile feedback" to understand its environment. To determine the importance of this factor, the position and force data collected during Baxter's scaled-down tasks were analyzed to see how or whether they could contribute to the task.

This Chapter will provide a description of the two scaled down tasks completed on Baxter, as well as an assessment of the performance of the controllers used. In addition, it will assess the feasibility of improving a geometric map during an operation and quantifying the roughness of a surface using Baxter's force sensors.

5.1 Scaled Down Task 1: Operating in a Poorly Mapped Environment

5.1.1 Motivation

When operating in the ocean, visibility may be extremely low or nonexistent. Despite vast improvements in sonar and laser technology, these technologies do not currently look at a biofouled surface and immediately tell if that surface is hard or soft. In order for a robot to be able to complete a desired task in such an environment and eventually improve the map of that environment, it must show some level of robustness against errors in the map. This can be accomplished by creating compliance in the arm. While the large shoulder springs on Baxter's arm provide some compliance even with just the position controller, this is not the case for most arms used in marine applications. To see how much additional compliance could be created with impedance control, Baxter interacted with a "poorly defined" surface using an anisotropic—pseudo hybrid—controller described in Chapter 4.

5.1.2 Experiment Setup

Baxter was programmed to complete a very simple task: make contact with a surface and move back and forth in the y-z (horizontal) plane across a flat, smooth table in the configuration shown in figure 5-1. This was first done using an anisotropic impedance controller.

Most, if not all, of the robotic arms used in the marine industry today use position or velocity control. To demonstrate how the pseudo hybrid controller would compare operating in this environment, the tasks were also completed using Baxter's Joint Trajectory Action Server which was described in Chapter 2. Both the position mode and the velocity mode were used.

Baseline data of interacting with the flat surface was taken to confirm that these controllers could stably interact with the "perfectly mapped" flat surface. Then the surface was curved by placing rigid objects under a flexible sheet of Aluminum (a

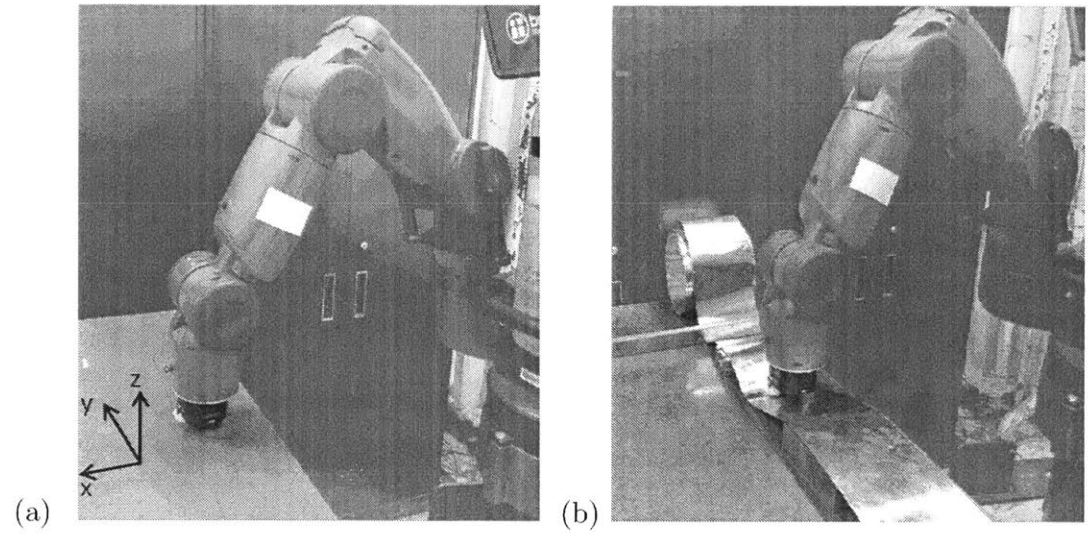


Figure 5-1: Arm configuration used in the poorly defined surface experiment. Baxter slid across a "perfectly mapped" flat surface and a surface with "errors" in the map created by an aluminum spline.

"spline") to determine how robust the controllers were against errors in its map of its environment. This spline also introduced a variation of stiffness with position—something which would be found in a real-world task. In this experiment, failure was defined as losing contact with the surface.

5.1.3 Pseudo Hybrid Controller Design

An anisotropic impedance—pseudo hybrid—controller was designed using the theory presented in Chapter 4 to complete this task. The controller was designed such that it would apply 25 N of force to the surface in a static configuration. The equilibrium position was 24 cm below the table.

As discussed in Chapter 4, there are two variations of pseudo hybrid control: (1) making the nominal force normal to the surface dependent of the distance of the virtual equilibrium position below the surface, and (2) commanding a nominal force normal to the surface which is independent of position. As a primary goal of this project is designing a controller which can transition from free to constrained motion, option 1 was chosen. However, it is acknowledged that this strategy has limitations. For instance, as the force is dependent on an equilibrium position, even with an anisotropic stiffness Baxter will not be able to maintain a constant force on a surface that has errors in its map. This could be accomplished using option 2 if the restoring stiffness normal to the surface was set to zero.

As discussed at length in Chapter 4, there are many ways to modulate the effective stiffness at the end effector. Variables include the stiffness of the virtual springs at the joints, the equilibrium orientation of the arm, and the stiffness of the virtual springs at the end effector. Furthermore, it was shown that the net stiffness is also dependent on the total force applied to the end effector. With so many variables, it was not intuitively obvious how the parameters would affect the net stiffness, or even which parameters would dominate the overall behavior.

It was desired that the behavior of the end effector be dominated by the stiffness and position of the virtual springs at the end effector. This way, should Baxter need to apply more or less force to the object it was interacting with, force could be adjusted quickly while performing the experiments by increasing or decreasing the stiffness or the virtual equilibrium position of the virtual springs at the end effector. To this end, joints S1, E0, W1 and W2 were empirically assigned a virtual stiffness just high enough so that when pressing on the surface there would be a restoring stiffness to keep the linkages from folding over on themselves. However they were kept as low as possible so that the net stiffness would primarily depend on the stiffness of the virtual springs at the end effector. For the purposes of this project, this was accomplished experimentally. Later work should explore this trade-off, perhaps based on work by (Rancourt & Hogan, 2001).

By assigning a stiffness to the four joints and in all three directions at the end effector, all seven DoF of Baxter's joints were utilized. However, joints S0, E1, and W0 were also assigned a small stiffness so that the stiffness matrix would be invertible. Tables 5.1 and 5.2 show the nominal stiffness applied to the joints at the right arm and at the end effector respectively. The stiffness at the end effector was chosen experimentally. However it should be noted that the upper limit of acheivable endpoint stiffness is approximately 900. Any higher will drive the system to an instability. The lower limit is approximately 10. Any lower, and errors in Baxter's gravity compensation torques will cause the arm to drift.

Table 5.1: Stiffness of Virtual Springs at Joints

S0	S1	E0	E1	W0	W1	W2
0.0005	5	5	0.005	0.005	5	5

Table 5.2: Stiffness of Virtual Springs at End Effector

	Y	X	Z
Stiffness	300	300	50

From the equations derived in Chapter 4, with this stiffness at the joints, the symmetric net stiffness matrix when the endpoint is at its equilibrium is

$$K_{net} = \begin{bmatrix} 296.96 & 6.34 & -8.46 & -7.76 & 14.93 & -3.73 \\ . & 296.12 & -4.34 & -2.86 & 3.25 & -1.65 \\ . & . & 71.84 & 5.63 & -5.69 & 2.27 \\ . & . & . & 1.03 & -2.70 & 1.42 \\ . & . & . & . & 8.35 & -1.41 \\ . & . & . & . & . & 0.84 \end{bmatrix}$$

(5.1)

As with the Cartesian inertia matrix discussed in Chapter 4, the net stiffness has coupling between rotation and translation.

To determine whether the values at the end effector dominated, as well as how much the stiffness varied with orientation and force applied, Baxter's resulting net stiffness was measured in each direction. This was accomplished by recording the force at the end effector in each direction as it was manually moved about its equilibrium position. Figure 5-2 shows the results in the x, y, and z direction. By keeping the joint stiffness relatively small in comparison to the virtual spring at the endpoint, the relation between force and stiffness appears close to linear within the operating space of the task. When the virtual spring at the end effector has a lower stiffness–such as in the z direction–the total apparent stiffness deviates from linear. As can be seen in the graphs, the slope and the y intercept of the force-position data is directionally dependent. The change in the y intercept is most likely due to static or dynamic friction. While friction could also account for the change in slope, another possible explanation for its directionality is the variation of the force applied at the end effector. As it was applied manually, its variability was undetermined.

A linear regression of the data in each direction was taken onto a straight line. The results are shown and a comparison of each with the calculated stiffness K_{net} is in table 5.3.

Table 5.3: Linear Regression of force-position data onto a straight line.

	Slope of Linear Regression	Norm of Residuals	Percent Difference	Calculated Stiffness from Model	Percent Difference
X	329.59 371.68	102.5 82.6	11%	296.96	10% 20.1%
Y	301.5 292.82	67.25 46.51	2.97%	296.12	1.2% 1.12%
Z	79.80 81.54	83.511 49.2	2.2%	71.84	9.97% 11.9%

For all but one outlier, the calculated net stiffness was within 12% of the actual stiffness. However, for one direction in the x frame, the difference was 20.1%. Maintaining the same orientation while manually translating the end effector in the x frame was by far the most challenging. This is the most likely source of the discrepancy. Overall, the change in configuration and force applied to the endpoint had a negligible affect on the effective stiffness within Baxter's range of motion.

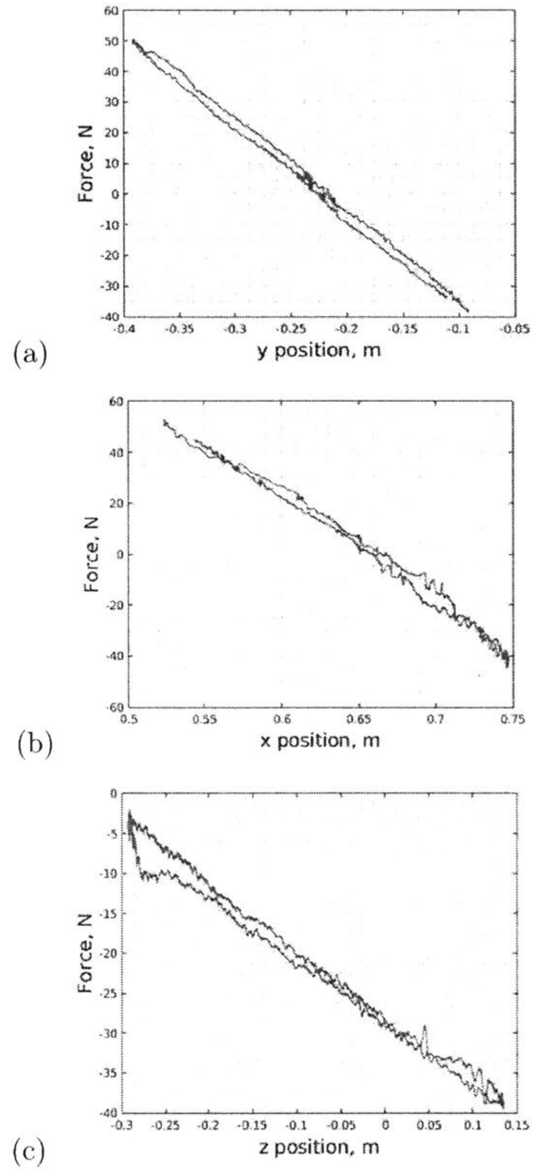


Figure 5-2: The end-effector force in the x, y, and z direction as a function of position. As the magnitude of the virtual spring at the end-effector is much greater than the virtual spring at the joints, the stiffness was relatively linear throughout the range of motion.

5.1.4 Results

Once the pseudo hybrid controller was characterized, its performance in a poorly mapped environment was compared to more standard controllers: position and velocity. To this end, many different spline configurations were tested on the controllers.

Figure 5-3 shows the cross section of one such configuration which highlights the difference in performance.

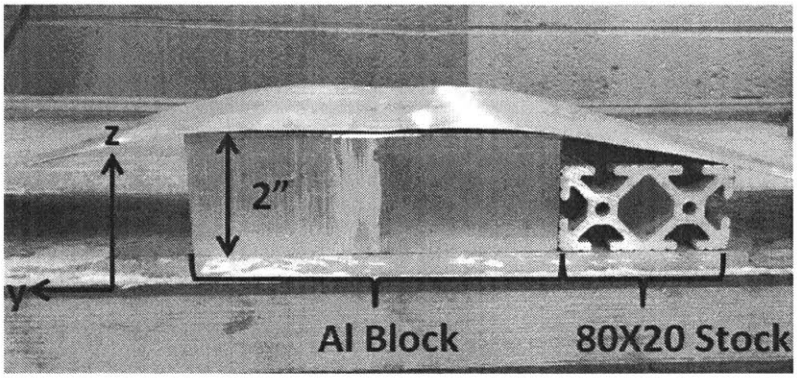


Figure 5-3: Example spline configurations of Baxter interacting with a poorly defined surface. The end effector started on the table to the right of the 80X20 Stock, swept up to the middle of the 2" Aluminum block, then back to its original position.

In this configuration, the pseudo hybrid controller was able to make and maintain contact with the surface throughout the trial. Furthermore, as can be seen in figure 5-4, the deviation in the z direction had minimal influence on the trajectory in the y direction. The end effector was able to maintain the same trajectory in the y direction as when it was interacting with the "perfectly mapped" flat surface. The Root Mean Squared (RMS) deviation between the two trajectories was $0.024m$.

This experiment was repeated using Baxter's Joint Trajectory Playback setting. As discussed in Chapter 2, the user can guide Baxter's arm through a series of motions, and then "play back" those joint values using either a "position mode" or a "velocity mode."

For the position mode, the end effector immediately failed when it came in contact with the 80X20 stock. This is due to the fact that when the joints exceed a specified deviation from the recorded joint positions, the arm shuts down. This could in theory be avoided by increasing the maximum allowable deviation. However Baxter's Joint Trajectory Playback setting does not allow for this. As this is a feature specific to Baxter's controller, there is not a broader conclusion to be reached than the controller's inability to complete the trajectory.

For velocity trajectory playback, the controller failed as the end-effector lost contact with the surface when it first reached the 80X20 stock. In addition to losing contact with the surface, the end effector was also unable to complete the desired Y trajectory. After losing contact with the surface, it was unable to move up and over the 2" block. Instead, as can be seen in figure 5-4, it stopped at the 80X20 stock before sliding back in the other direction. The RMS deviation between the two trajectories was 0.119m.

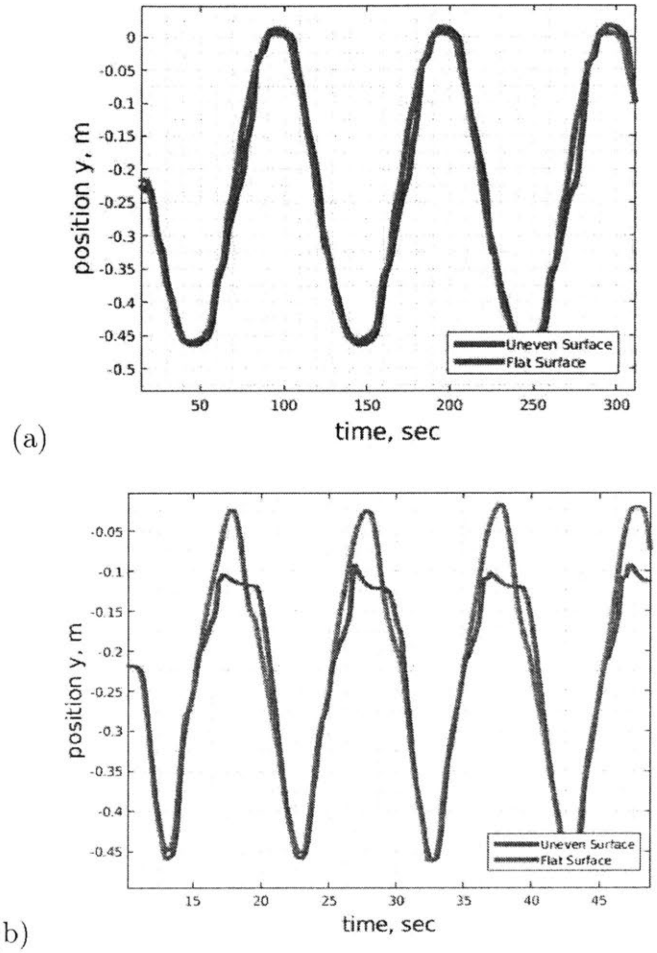


Figure 5-4: (a) Y trajectory (horizontal) on "perfectly mapped" and poorly mapped surfaces using pseudo hybrid control. The RMS deviation between the two trajectories was 0.024m. (b) Y trajectory (horizontal) on "perfectly mapped" and poorly mapped surfaces using velocity mode in the Joint Trajectory Playback setting. As can be seen at about -0.1m, the end effector was unable to traverse the 2" block. Instead, it both lost contact with the surface and paused at the transition from the 80X20 Stock and the 2" block before returning to its specified trajectory as it slid back to the flat surface. The RMS deviation between the two trajectories was 0.119m.

5.1.5 Improving the Map of the Environment While Completing the Task

To complete any contact task, it is likely that a robot will need to know the position of its end effector. As previously stated, despite vast improvements in sonar and laser technology, these sensors do not look at a biofouled surface and immediately tell if that surface is hard or soft. Since at the very least the end effector position data is already required, and the system has been proven to maintain coupled stability in a poorly mapped environment, there is an opportunity to improve the map of the environment as the robot completes its operations.

One of the specific goals listed in Chapter 1 was to use tactile exploration to refine a geometric model of a shape. To demonstrate the feasibility of this, a pseudo hybrid controller was used to move across a spline which neither the velocity or position modes of the Joint Trajectory Playback Setting could traverse. The endpoint data of Baxter interacting with the spline was recorded. Figure 5-5 shows the endpoint data of one full right-to-left motion across the Aluminum spline. As shown in the figure, this not only captures the general shape, but also the deflection of the spline. Given that the end effector force is also known, it would be possible to determine the stiffness of the spline. Further exploration of this possibility is deferred to future work.

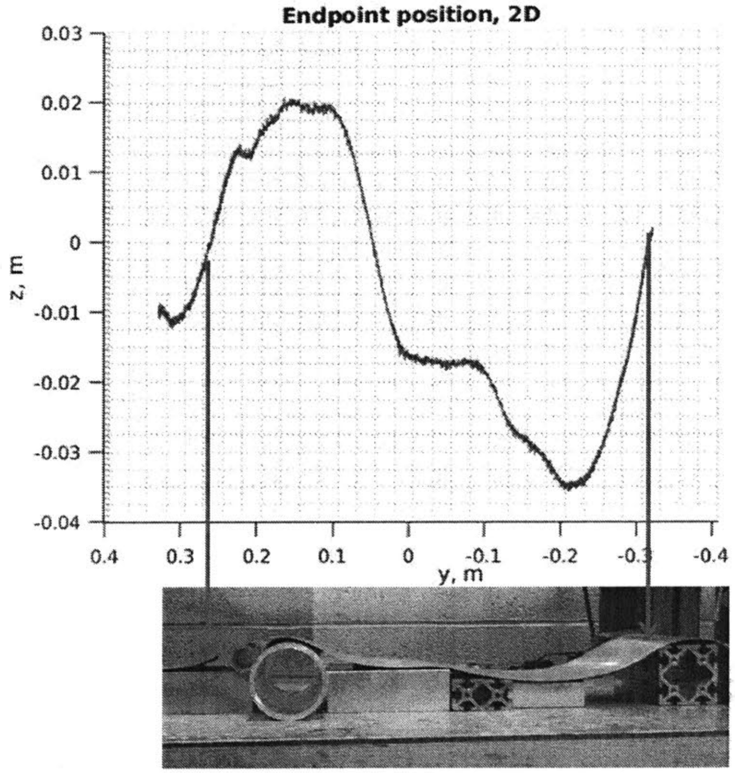


Figure 5-5: Tactile map of the Aluminum spline. The position of the end effector is able to capture the general shape of the Aluminum spline, improving the geometric map of the environment while interacting with it.

5.2 Scaled Down Task 2: Scraping Biofouling

5.2.1 Motivation

The underwater pilings in deep sea oil fields will frequently have developed a layer of biofouling which must be removed before a thorough inspection of the nodes can be completed. This biofouling is extremely dependent on depth. In deeper waters where there is no light, it can simply be sand and silt which has been caked on. However in shallower regions it can be extremely thick marine growth that has developed over the years. That marine growth may be an unknown combination of hard material (e.g. coral skeletons) and soft material (e.g. seaweeds, anemones). Removal of the biofouling is typically accomplished by divers who are equipped with either a waterjet or a wire brush. To determine whether an impedance controller is capable of accomplishing such a task, Baxter was given a scaled-down version of biofouling removal: sanding a piece of rough wood.

5.2.2 Experiment Setup

A piece of sandpaper was attached to Baxter's end effector using Velcro. The end effector was placed a few inches above the piece of wood to be sanded as shown in figure 5-6. Similar to the experiment when Baxter was interacting with a poorly mapped environment, Baxter was programmed to make contact with and wipe across the flat surface. The experiment was completed twice: once where the equilibrium position of the end effector was nearly at the surface of the wood being sanded down, and once where the equilibrium position was below the surface. The virtual equilibrium position's motion was sinusoidal at a frequency of 0.3 Hz. For reference, this experiment was also repeated on a smooth piece of wood. The stiffness of the joints and endpoint were chosen experimentally in the same manner as described in Scaled Down Task 1. Details of the stiffness chosen for this controller may be found in Appendix B.

5.2.3 Results

The controller was run for 15 minutes. Given the nominal equilibrium position frequency of 0.3 Hz, there were about 270 cycles. Baxter had nearly identical results with both the equilibrium positions. As seen in figure 5-7, Baxter was able to complete the sanding task.

In addition to completing the task, under this control scheme Baxter was also able to recover from a potential failure mode. When the end effector initially made contact with the surface, it could not push through the shards of wood. Because of this, the virtual spring moved back and forth on the piece of wood while the end effector remained stationary, resulting in a tangential force profile seen in figure 5-8. Due to the impedance controller, tangential force was slowly increased in both directions until the end effector was eventually able to break free. The controller was robust enough so that even when the robot was initially unable to move, it was able

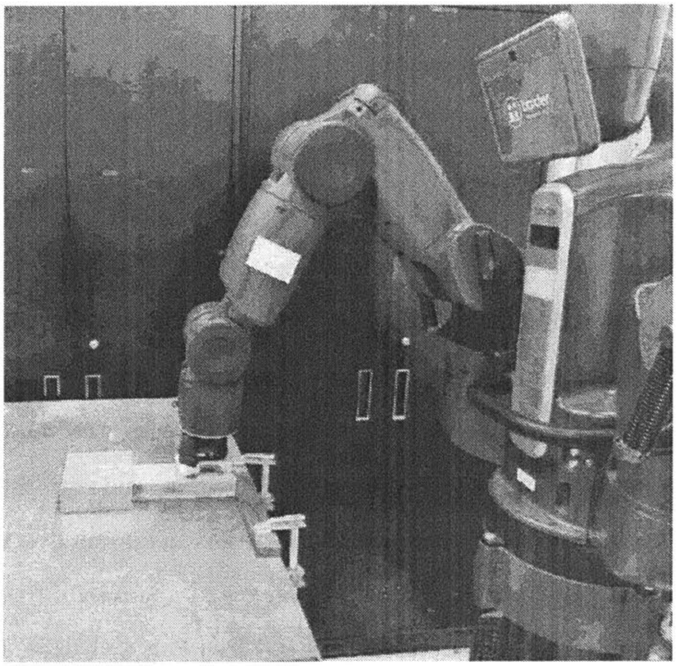


Figure 5-6: Sanding experiment setup. A splintered wood block was secured to the table. A piece of Velcro was attached to Baxter's end effector, which was programmed to move back and forth across a flat surface.

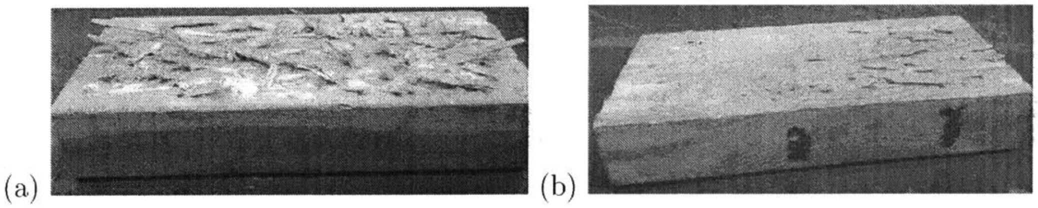


Figure 5-7: Wood sanded down by Baxter (a) before and (b) after

to continue the task without even needing to perceive the initial failure.

As can be seen by the tangential force data, Baxter was actually doing work on the system. To the best of the author's knowledge, this is the first time Baxter has been proven capable of doing this.

5.2.4 Differentiating Between Surface Roughness

Given the limited visibility at operating depths, it is not known if a robot would always be able to verify visually whether biofouling had satisfactorily been removed from the surface of a piling. In addition, removing biofouling from the surface could

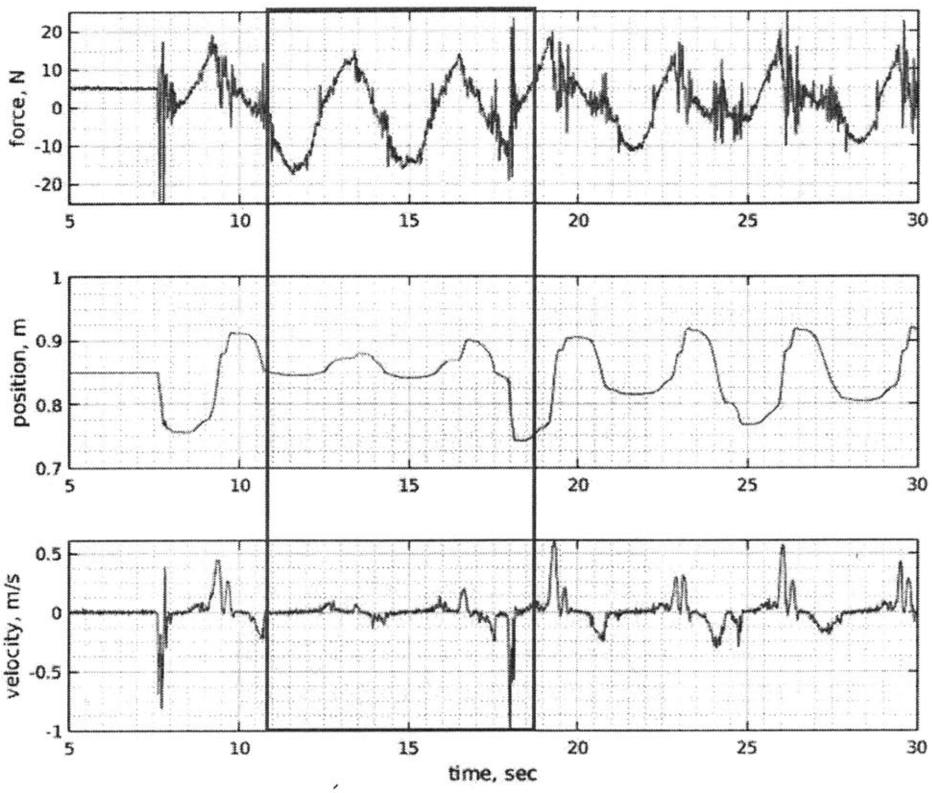


Figure 5-8: Initial impact and attempt to sand wood by Baxter when the equilibrium position was nearly at the surface. From 11 to 18 seconds (boxed in the graph) the endpoint became stuck on some splinters of wood. Instead of this becoming a failure mode, the "virtual spring" attached to the endpoint continued to move back and forth across the wood, as shown by the approximately triangular shape of the force data. Eventually, the shards of wood broke and Baxter was able to continue and eventually complete the sanding task.

generate suspended debris which could compromise visibility. It would be useful to be able to distinguish between the roughness of surfaces, and even tell if a task has been satisfactorily completed, based on the interaction forces at the endpoint. With this goal in mind, the force data from Baxter's joints during the sanding tasks were analyzed to see if there was a significant difference between interacting with a rough surface and interacting with a smooth surface. Three cases were analyzed: Baxter initially sanding rough wood, Baxter finishing sanding rough wood, and Baxter sanding a smooth piece of wood. Figure 5-9 shows the orientation of the end effector used in the sanding experiments.

It is important to note that end effector orientation had a significant effect on this task, as it changes the interactive dynamics: particularly inertia. While Baxter was able to satisfactorily complete the sanding task for both configurations shown in figure 5-9, the interactive dynamics were significantly different. In particular, when the end effector was in orientation "b," Baxter routinely lost contact with the wood when it was pushing away from itself.

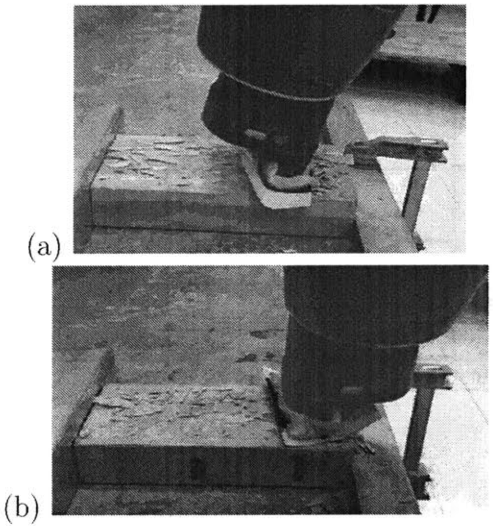


Figure 5-9: Two orientations of end effector used in the sanding task. The orientation in (a) provided much smoother results. In orientation (b), when Baxter pushed away from itself, the end effector actually lost contact with the wood. Despite this failure mode, Baxter was still capable of completing the sanding task in either orientation.

Figure 5-10 shows the force, position, and velocity data of Baxter in orientation (a) as it initially sanded the rough piece of wood using the impedance controller where the virtual equilibrium point was nearly at the surface. The interactive dynamics of Baxter sanding away from itself were markedly different from Baxter sanding towards itself. Particularly, as can be seen in the velocity data, as Baxter pushed away from itself the endpoint tended to become stuck momentarily about halfway across the wood block.

As the force data of Baxter pushing away from itself included zero velocity about halfway through each pass, it was decided that only the section with Baxter pulling towards itself—the data boxed in red in figure 5-10—would be analyzed.

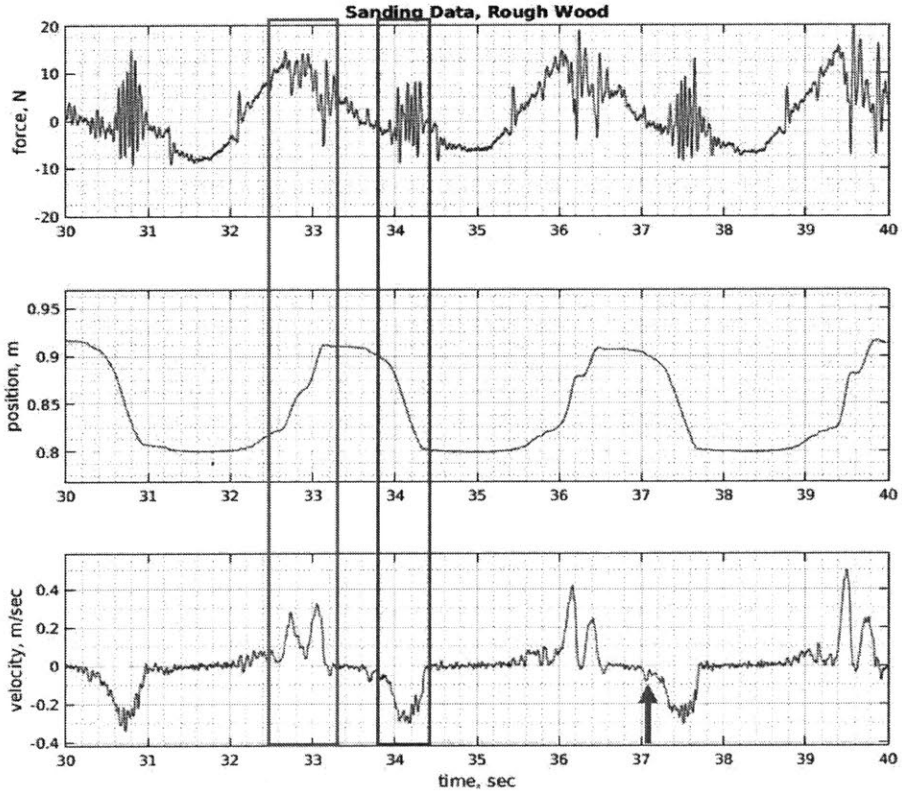


Figure 5-10: Baxter sanding rough wood using an impedance controller. The green boxes represent Baxter sanding away from itself, the red represents Baxter sanding towards itself. The red arrow marks the initial spike in velocity which was used to identify where the 0.8 second window of data was started.

Eight, 0.8 second windows of data of Baxter sanding towards itself were taken for analysis. An example window is shown in the red box in figure 5-10. Looking at the velocity data, at the beginning of each sanding motion, there is first a small increase in velocity where the end effector attempts to move, but must overcome static friction to do so. In an attempt to encompass both the static and dynamic friction components, this brief increase in velocity was used as the marker for when each 0.8 second window began. This was done for three separate cases: 1) Baxter initially starting the sanding task; 2) Baxter completing the sanding task; and 3) Baxter sanding the smooth piece of wood. The average and standard deviation for each is shown below. There is a clear difference between the three trials. However, the overall magnitude of the friction force did not change in a manner that would be expected for a standard linear friction model such that

$$F_{friction} = \mu F_{normal} \tag{5.2}$$

where $F_{friction}$ is the frictional force, F_{normal} is the normal force the end effector is applying to the wood, and μ is either the static or dynamic coefficient. Instead, the force oscillated about zero, and the most marked difference in the behavior between the three appeared in the frequency domain.

Once it was clear that the most marked difference was due to the high frequency content, the force data were analyzed in the frequency domain. The data were first detrended in the time domain. The Fast Fourier Transform, FFT, was used to transform to the frequency domain. The results are shown below. As can be seen in the graphs, there is a distinct frequency domain difference between the three trials. While all have a local maximum at 20Hz, the magnitude at that frequency of the trial at the end of the sanding task is almost half that of the magnitude at the beginning. Results of analyses in both the time and frequency domains may be seen in figure 5-11.

The results were quite different for the case where Baxter's equilibrium position was further below the surface. While the wood was sanded down to the same extent, the time course of the motion and force were quite different. As can be seen in figure 5-12, there is significantly less frequency content in the data. This allows for a clearer interpretation of the interactive dynamics. Looking at the velocity data, it is much more evident that the end effector initially attempts to move, gets stuck, then requires a large amount of force to overcome static friction.

The same analysis was then completed for this control scheme. There was a clear difference between the magnitude of the tangential force in each case. The results are shown below in figure 5-13. As the magnitude of the tangential force appeared to be the primary difference between the cases instead of the frequency content, the

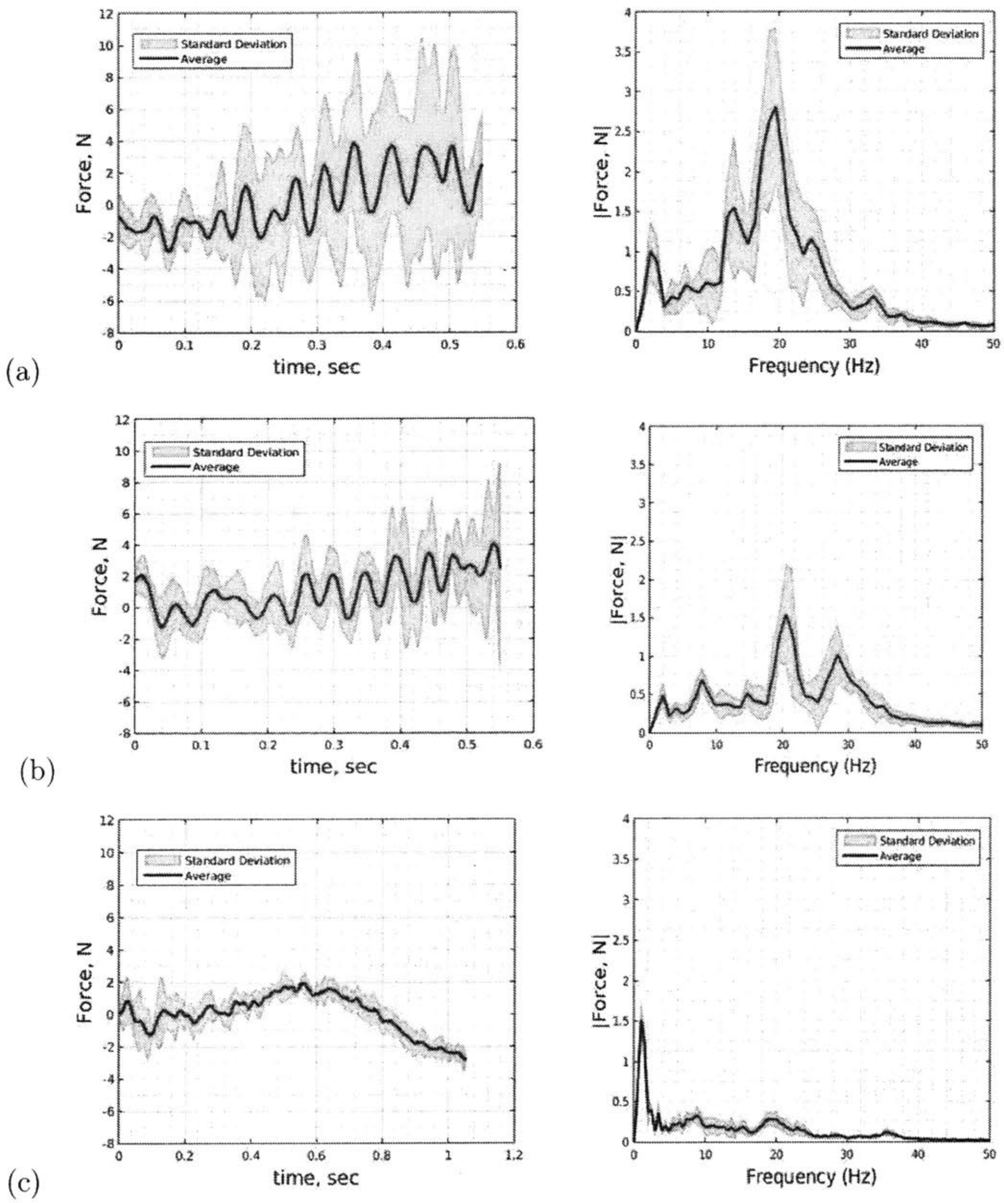


Figure 5-11: Time and frequency domain representations of the sanding data when (a) initially beginning the sanding task (b) completing the sanding task (c) sanding the smooth piece of wood.

tangential force was scaled to the normal force applied to the wood to determine a friction coefficient.

These preliminary results require further analysis. For instance, an immediate question to be answered is to determine the source of the high frequency content

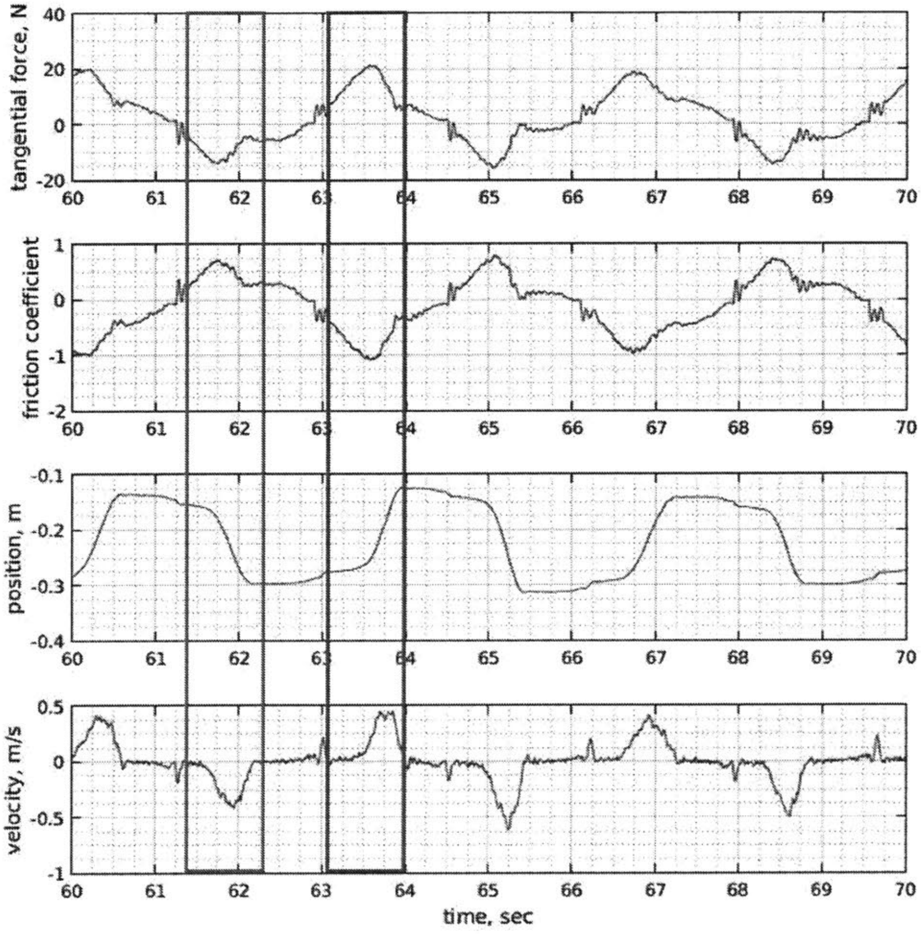
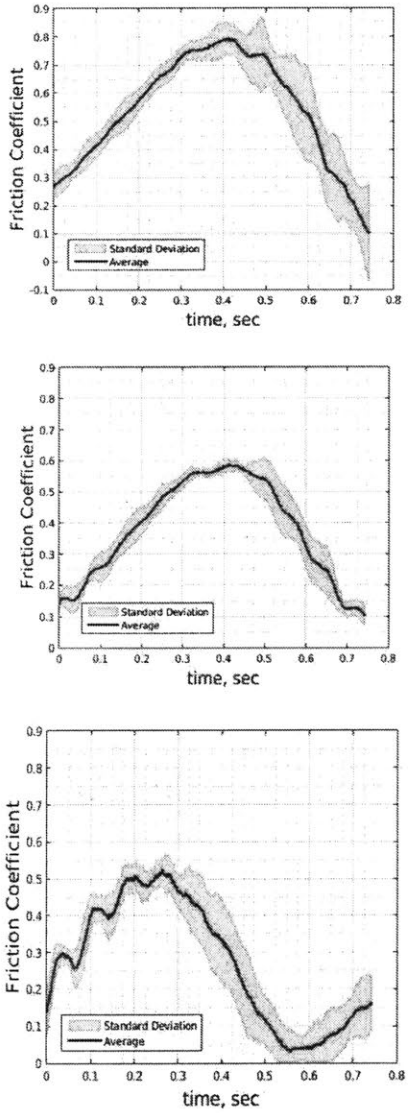


Figure 5-12: Force, position, and velocity data of Baxter initially sanding a rough piece of wood. The green boxes represent Baxter pushing away from itself, the red represent Baxter pulling towards itself.

which was analyzed in the first sanding trial.

However, it is clear that regardless of hardware constraints and interactive dynamics, the force data shows marked differences between the three different cases. Baxter is the least expensive robot of its kind on the market today. If it is capable of producing these kinds of results, implementing tactile feedback on robots to both improve the geometric map and make control decisions based on force data should be possible.



(a)

(b)

(c)

Figure 5-13: Time domain representations of the sanding data when (a) initially beginning the sanding task (b) completing the sanding task (c) sanding the smooth piece of wood.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis investigated a scaled-down version of the development and implementation of a control system for an anthropomorphic robot which could be used to decommission deep sea oil platforms. First, a dynamic and kinematic model of a Baxter Research Robot was developed and compared to Baxter's pre-existing KDL. If at any point the KDL is no longer supported by Rethink Robotics (as has been suggested), the documentation for the model will still exist.

Next, the assumption of Baxter's SEAs being a perfect force or motion source was analyzed. Overall, it was determined that for planned operations, the rigid body mode of the SEAs would dominate. The dynamics from the springs in the SEAs are negligible. Additionally, the reflected inertia of the SEAs was characterized using both forward path and interactive dynamics. The reflected inertia was a significant source of the apparent inertia at the end effector. It was determined that in applications where a robot may have unexpected collisions with its environment, a more rigorous dynamic model than is currently used in simulators is required to accurately model the force which will be transmitted to the environment.

After the model was developed and analyzed, a force-based Cartesian impedance controller was implemented on Baxter. To the best of the author's knowledge, this is the first and only controller of its kind to be implemented on a Baxter Research

Robot. Using this controller, Baxter was able to complete a scaled-down version of the following decommissioning tasks outlined in Chapter 1: making and maintaining stable contact with the environment, scrubbing biofouling, and using haptic exploration to refine a geometric model.

Preliminary use of the impedance controller in these scaled down tasks proved it to be more robust in poorly mapped environments than traditional control methods using position and velocity control. Furthermore, in completing the sanding task, Baxter actually performed mechanical work on the system it was interacting with. To the best of the author's knowledge, this is the first time Baxter was proven capable of doing this. Finally, analysis of the endpoint force data collected while completing these tasks showed that Baxter's force data could be used to determine when a task such as scrubbing was completed.

6.2 Future Work

In developing a dynamic model of, creating a control system for, and completing preliminary tests on Baxter, a number of potential research topics became apparent. They are summarized below.

6.2.1 Characterizing Nonlinear Damping Parameters

To improve Baxter's dynamic model, the damping parameters of Baxter's SEAs should be characterized. These parameters are most likely nonlinear. Additionally, they will most likely differ for forward path dynamics and interactive dynamics. As such, it is suggested to use a method similar to that of Chapter 3 to characterize the inertia using interactive dynamics. In fact, when fitting the data of Baxter interacting with a spring, both linear and nonlinear damping were included in the parameters to be estimated by the model. However, the experiment setup introduced a nontrivial amount of damping into the system. Before determining Baxter's damping parameters, either the damping from the existing experiment setup should be characterized, or a better setup with minimal friction should be designed.

6.2.2 Modulating Net Stiffness

As discussed in Chapter 4, the stiffness at Baxter's endpoint depends on the stiffness of the joints, the configuration of the arm, and the force applied at the endpoint. For the purposes of this project, these parameters were chosen empirically. Future work should be done to find the preferred configurations, as well as the range of restoring joint stiffness, such that when pressing on a surface the links of the arm will not fold over on themselves. This could potentially be based on work by (Rancourt & Hogan, 2001).

6.2.3 Creating an Impedance Map of the Environment

Baxter's endpoint position data was shown to be capable of improving a geometric map of its environment. Additionally, it was shown that Baxter's force data could be used to determine the "roughness" of a surface. Combining these data, it should be possible to create an "impedance map" of a surface which could characterize what parts of a surface are "hard" or "soft." For instance, when Baxter interacted with the Aluminum spline as discussed in Chapter 5, the end effector data not only captured the geometric shape of the spline, but the deflection of the spline where it was not fully supported by a rigid body. See figure 6-1 for an example. Given that the end effector force is also known, it would be possible to determine the stiffness of the spline, creating an impedance map of the surface.

In completing this initial task of operating with a poorly mapped environment, it was decided that the force acting normal to the surface be dependent on an equilibrium position. This was done so that Baxter would be able to transition from free to constrained motion. It should still be possible to develop an impedance map under these circumstances. As long as Baxter's end effector remains in contact with the surface, the stiffness of the controller could be subtracted from the stiffness of the environment. However to simplify the initial analysis, it is suggested that when developing the first impedance map using Baxter, the robot (1) applies a force that is independent of position and (2) interacts with a surface that has no errors in its geometric map. Work could be based on research completed by (Bosworth, 2016).

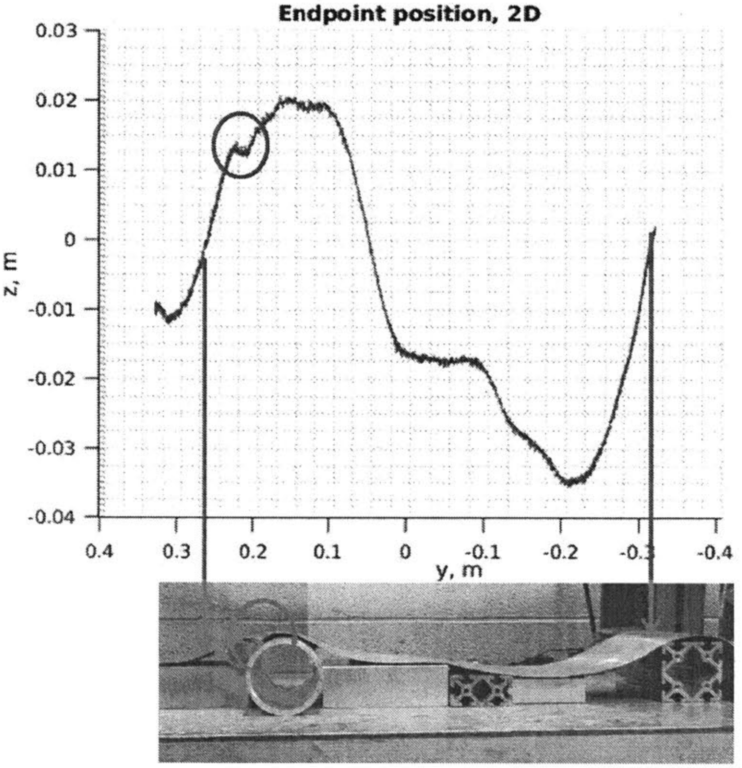


Figure 6-1: Position data of Baxter interacting with a poorly defined surface: an Aluminum Spline. as highlighted by the red circles, the position data captured not only the geometric shape of the spline, but also its deflection. This could be used to determine stiffness.

6.2.4 Creating a Simplified Framework for Large DoF by Using Modular Impedances

As shown in a brief example in Chapter 4, "modular impedances" could be used as a way to create a simplified framework for working with robots with a large numbers of DoF. This could be particularly useful in situations where traditional methods would require the computation of closed chain kinematics. When experiments with Baxter shift to utilizing both hands, it is suggested that this framework be developed. This way, instead of working with a single 14DoF system with closed chain kinematics, the user can instead design the controller as if they are superimposing two mass, spring, dashpot systems. It is suggested that as an initial experiment, the simplified example outlined in Chapter 4 be implemented on Baxter and the effective inertia compared. Further work could also be based on research completed by (Schneider & Cannon Jr, 1992). Specifically, this research could be extended to consider redundant DoF.

6.2.5 Transitioning to an Ocean Environment

The Baxter Research Robot is clearly not the hardware which would be used in any extreme environment. It was chosen for its (1) redundant limbs, (2) force feedback, and (3) safety features allowing for close human interaction. To the best of the author's knowledge, there is currently no robotic arm on the market today which can operate in the ocean that has either force feedback or mechanical redundancy. In order for the overarching control strategy of passivity to be implemented, force feedback is imperative. (Welch, 2015) has begun research on this front by combining commercial off-the-shelf (COTS) products to create an undersea robotic arm which has force feedback at its base joint. Further research is required to determine how effective this would be.

References

ArmControlMode. (2015). http://sdk.rethinkrobotics.com/wiki/Arm_Control_Modes. (Accessed: 2016-04-10)

Armstrong, B., Khatib, O., & Burdick, J. (1986). The explicit dynamic model and inertial parameters of the puma 560 arm. In *Robotics and automation. proceedings. 1986 ieee international conference on* (Vol. 3, pp. 510–518).

BluHaptics. (2016). http://www.bluhaptics.com/. (Accessed: 2016-04-10)

BostonDynamics. (2016). http://www.bostondynamics.com/. (Accessed: 2016-04-10)

Bosworth. (2016). *New methods in "ground aware" robot legged locomotion* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

BSEE. (2013). http://www.bsee.gov/About-BSEE/Budget/FY2013BudgetJustification/. (Accessed: 2016-04-10)

Chevron. (2014, June 23). *Chevron environmental management company.* (Personal Interview)

Clark. (2016). *Why deep saturation diving is like going to the moon, and beyond.* http://johnclarkeonline.com/2011/07/17/why-deep-saturation-diving-is-like-going-to-the-moon-and-beyond/. (Accessed: 2016-04-10)

Colgate. (1988). *The control of dynamically interacting systems* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Colgate, & Hogan. (1988). Robust control of dynamically interacting systems. *International journal of Control*, *48*(1), 65–88.

Colgate, & Hogan. (1989). The interaction of robots with passive environments: Application to force feedback control. In *Advanced robotics: 1989* (pp. 465–

474). Springer.

ControlOverview. (2015). http://sdk.rethinkrobotics.com/wiki/Arm_Control_Overview. (Accessed: 2016-04-10)

Craig, J. J., & Raibert, M. H. (1979). A systematic method of hybrid position/force control of a manipulator. In *Computer software and applications conference, 1979. proceedings. compsac 79. the ieee computer society's third international* (pp. 446–451).

DeepSeaNews. (2010). *gulf map.* http://www.deepseanews.com/wp-content/uploads/2010/06/gulfmap1.jpeg. (Accessed: 2016-04-10)

Denavit, J. (1955). A kinematic notation for lower-pair mechanisms based on matrices. *Trans. of the ASME. Journal of Applied Mechanics*, *22*, 215–221.

Dietrich, A., Albu-Schäffer, A., & Hirzinger, G. (2012). On continuous null space projections for torque-based, hierarchical, multi-objective manipulation. In *Robotics and automation (icra), 2012 ieee international conference on* (pp. 2978–2985).

Dietrich, A., Ott, C., & Albu-Schäffer, A. (2015). An overview of null space projections for redundant, torque-controlled robots. *The International Journal of Robotics Research*, 0278364914566516.

Einstein. (1933). https://en.wikiquote.org/wiki/Albert_Einstein. (Accessed: 2016-04-10)

Goodley. (2013). *Shell presses ahead with world's deepest offshore oil well.* http://www.theguardian.com/business/2013/may/08/shell-deepest-offshore-oil-well. (Accessed: 2016-04-10)

Hardware. (2015). http://sdk.rethinkrobotics.com/wiki/Hardware_Specifications. (Accessed: 2016-04-10)

Hawkins. (2013). http://wiki.ros.org/pykdl_utils. (Accessed: 2016-04-10)

Hogan, N. (1985). Impedance control: An approach to manipulation: Part iiâĂŤimplementation. *Journal of dynamic systems, measurement, and control*, *107*(1), 8–16.

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *Robotics and Automation, IEEE Journal of*, *3*(1), 43–53.

Khatib, O. (1995). Inertial properties in robotic manipulation: An object-level framework. *The International Journal of Robotics Research*, *14*(1), 19–36.

Klein, C. A., & Huang, C.-H. (1983). Review of pseudoinverse control for use with kinematically redundant manipulators. *Systems, Man and Cybernetics, IEEE Transactions on*(2), 245–250.

Lee, H. (2013). *Quantitative characterization of multi-variable human ankle mechanical impedance* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Mason, M. T. (1981). Compliance and force control for computer controlled manipulators. *Systems, Man and Cybernetics, IEEE Transactions on*, *11*(6), 418–432.

Muhammad, A. (2011). Impedance control of redundant manipulators. *Tampereen teknillinen yliopisto. Julkaisu-Tampere University of Technology. Publication; 986*.

Mussa-Ivaldi, F. A., & Hogan, N. (1991). Integrable solutions of kinematic redundancy via impedance control. *The International Journal of Robotics Research*, *10*(5), 481–491.

ODE. (2007). `http://ode.org/`. (Accessed: 2016-04-10)

Orocos. (2016). `http://www.orocos.org/kdl/user-manual`. (Accessed: 2016-04-27)

Pratt, G. A., & Williamson, M. M. (1995). Series elastic actuators. In *Intelligent robots and systems 95.'human robot interaction and cooperative robots', proceedings. 1995 ieee/rsj international conference on* (Vol. 1, pp. 399–406).

Rancourt, D., & Hogan, N. (2001). Stability in force-production tasks. *Journal of motor behavior*, *33*(2), 193–204.

RethinkRobotics. (2016). `http://www.rethinkrobotics.com/`. (Accessed: 2016-04-10)

ROS. (2012). `http://wiki.ros.org/urdf/XML/model`. (Accessed: 2016-04-10)

Roveda, L. (2015). *Model based compliance shaping control of light-weight manipula-*

tor in hard-contact industrial applications (Unpublished doctoral dissertation). Italy.

Schindlbeck, C., & Haddadin, S. (2015). Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks. In *Robotics and automation (icra), 2015 ieee international conference on* (pp. 440–447).

Schneider, S. A., & Cannon Jr, R. H. (1992). Object impedance control for cooperative manipulation: Theory and experimental results. *Robotics and Automation, IEEE Transactions on*, *8*(3), 383–394.

sdk. (2015). `http://sdk.rethinkrobotics.com/wiki/Arms`. (Accessed: 2016-04-10)

Teq. (2016). `http://teq.com/baxter`. (Accessed: 2016-04-10)

Van De Vegte, J. M., Milgram, P., & Kwong, R. H. (1990). Teleoperator control models: effects of time delay and imperfect system knowledge. *Systems, Man and Cybernetics, IEEE Transactions on*, *20*(6), 1258–1272.

Welch, C. L. (2015). *Navigation and manipulation for autonomous underwater dismantling of offshore structures* (Unpublished doctoral dissertation). Massachusetts Institute of Technology and Woods Hole Oceanographic Institution.

Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on man-machine systems*.

Appendix A

Baxter's Kinematic and Dynamic Model

```
function Inertia(q)
%Author: Lucille Hosford, 2015
%This Function Calculated the Cartesian Space Inertia Matrix for a given
%vector,q, containing a given arm configuration

%Names of links

%link 1 "right upper shoulder'
%link 2 "right lower shoulder"
%link 3 "right upper elbow"
%link 4: "right lower elbow"
%link 5: "right upper forearm"
%link 6: "right lower forearm"
%link 7: "right wrist"
%link 8: "right hand"

syms q1 q2 q3 q4 q5 q6 q7 real

%inertial matrix
%masses of each link, kg. link 7 and 8 are combined.
m1=5.700440;
```

```
m1m=[m1 0 0;0 m1 0; 0 0 m1];

m2=3.226980;
m2m=[m2 0 0;0 m2 0; 0 0 m2];

m3=4.312720;
m3m=[m3 0 0;0 m3 0; 0 0 m3];

m4=2.072060;
m4m=[m4 0 0;0 m4 0; 0 0 m4];

m5=2.246650;
m5m=[m5 0 0;0 m5 0; 0 0 m5];

m6=1.609790;
m6m=[m6 0. 0;0 m6 0; 0 0 m6];

m7=0.350930+0.191250;
m7m=[m7 0 0;0 m7 0; 0 0 m7];


%inertias at center of mass of each link, but not aligned with the
    principal axis of inertia

Ia = [0.04709102262   -0.00614870039        0.00012787556 0.0359598847
    -0.00078086899    0.03766976455];
I1=[Ia(1) Ia(2) Ia(3);
    Ia(2) Ia(4) Ia(5);
    Ia(3) Ia(5) Ia(6)];

Ib = [0.0278859752 -0.00018821993 -0.000300963979 0.0207874929
    0.00207675762        0.01175209419];
I2=[Ib(1) Ib(2) Ib(3);
    Ib(2) Ib(4) Ib(5);
    Ib(3) Ib(5) Ib(6)];
```

```
Ic = |0.02661733557   -0.00392189887        0.00029270634  0.0124800832
    -0.0010838933    0.02844355207|;
I3=|Ic(1)  Ic(2)  Ic(3);
    Ic(2)  Ic(4)  Ic(5);
    Ic(3)  Ic(5)  Ic(6)|;


Id = |0.01318227876   -0.00019663418        0.00036036173  0.0092685206
    0.0007459496    0.00711582686|;
I4=|Id(1)  Id(2)  Id(3);
    Id(2)  Id(4)  Id(5);
    Id(3)  Id(5)  Id(6)|;



Ie = |0.01667742825   -0.00018657629        0.00018403705  0.0037463115
    0.00064732352    0.01675457264|;
I5=|Ie(1)  Ie(2)  Ie(3);
    Ie(2)  Ie(4)  Ie(5);
    Ie(3)  Ie(5)  Ie(6)|;



If = |0.00700537914    0.00015348067       -0.00044384784  0.0055275524
    -0.00021115038    0.00387607152|;
I6=|If(1)  If(2)  If(3);
    If(2)  If(4)  If(5);
    If(3)  If(5)  If(6)|;



Ig = |0.00081621358    0.00012844010        0.000189698911        0.00087350127
       0.00010577265    0.00054941487|;
I7=|Ig(1)  Ig(2)  Ig(3);
    Ig(2)  Ig(4)  Ig(5);
    Ig(3)  Ig(5)  Ig(6)|;

Z=zeros(3);

%Uncoupled mass matrix
```

```
M=[m1m Z Z Z Z Z Z Z Z Z Z Z Z Z;
    Z I1 Z Z Z Z Z Z Z Z Z Z Z Z;
    Z Z m2m Z Z Z Z Z Z Z Z Z Z Z;
    Z Z Z I2 Z Z Z Z Z Z Z Z Z Z;
    Z Z Z Z m3m Z Z Z Z Z Z Z Z Z;
    Z Z Z Z Z I3 Z Z Z Z Z Z Z Z;
    Z Z Z Z Z Z m4m Z Z Z Z Z Z Z;
    Z Z Z Z Z Z Z I4 Z Z Z Z Z Z;
    Z Z Z Z Z Z Z Z m5m Z Z Z Z Z;
    Z Z Z Z Z Z Z Z Z I5 Z Z Z Z;
    Z Z Z Z Z Z Z Z Z Z m6m Z Z Z;
    Z Z Z Z Z Z Z Z Z Z Z I6 Z Z;
    Z Z Z Z Z Z Z Z Z Z Z Z m7m Z;
    Z Z Z Z Z Z Z Z Z Z Z Z Z I7];

%Center of Mass with respect to each segment's coordinate frame. [x,y,z]

COM1_C = [ -0.0511700000000000,      0.0790800000000000,
    0.000859999999999956 1];
COM2_C = [  0.00269000000000000,   -0.00529000000000003,
    0.0684499999999999 1];
COM3_C = [ -0.0717600000000000,      0.0814900000000001,
    0.00131999999999994 1];
COM4_C = [  0.00159000000000006,   -0.0111700000000000,
    0.0261799999999999 1];
COM5_C = [ -0.0116799999999999,      0.131110000000000,
    0.00459999999999992 1];
COM6_C = [  0.00697000000000011,    0.00599999999999981,
    0.0604800000000000 1];
COM7_C = [  0.00513704655280540,    0.000957223615773138,
    -0.0668234671142425 1];

%Reference COM from base of shoulder
T1=[0.70711,-0.70711,0,0.064027;
     0.70711,0.70711,0,0.25903;
```

```
      0,0,1,0.09066;
      0,0,0,1];

Tr=[0.70711,0.70711,0,0.064027;
     -0.70711,0.70711,0,-0.25903;
     0,0,1,(0.1185-0.02784);
     0,0,0,1];

%Transformation from Baxter's Base to each joint for right arm. To switch
%to left arm replace Tr with Tl.

T1=Tr*DH(1);
T1_2=Tr*DH(1)*DH(2);
T1_3=Tr*DH(1)*DH(2)*DH(3);
T1_4=Tr*DH(1)*DH(2)*DH(3)*DH(4);
T1_5=Tr*DH(1)*DH(2)*DH(3)*DH(4)*DH(5);
T1_6=Tr*DH(1)*DH(2)*DH(3)*DH(4)*DH(5)*DH(6);
T1_7=Tr*DH(1)*DH(2)*DH(3)*DH(4)*DH(5)*DH(6)*DH(7);

%Position of Center of Mass of each link with respect to Baxter's base

COM1_g=T1*COM1_C';
COM2_g=T1_2*COM2_C';
COM3_g=T1_3*COM3_C';
COM4_g=T1_4*COM4_C';
COM5_g=T1_5*COM5_C';
COM6_g=T1_6*COM6_C';
COM7_g=T1_7*COM7_C';

%Reference rotation from base of shoulder

%Rotation of each segment with respect to each segment's coordinate frame

ROT1_C=[0, 0, q1];
```

```
ROT2_C=[0, 0, q2];
ROT3_C=[0, 0, q3];
ROT4_C=[0, 0, q4];
ROT5_C=[0, 0, q5];
ROT6_C=[0, 0, q6];
ROT7_C=[0, 0, q7];

%Reference rotation from Baxter's Base

ROT1_g=ROT1_C';
ROT2_g=T1(1:3,1:3)*ROT2_C'+ROT1_C';
ROT3_g=T1_2(1:3,1:3)*ROT3_C'+ROT1_C';
ROT4_g=T1_3(1:3,1:3)*ROT4_C'+ROT1_C';
ROT5_g=T1_4(1:3,1:3)*ROT5_C'+ROT1_C';
ROT6_g=T1_5(1:3,1:3)*ROT6_C'+ROT1_C';
ROT7_g=T1_6(1:3,1:3)*ROT7_C'+ROT1_C';

%Build the Coordinate Vector Array

xu=[COM1_g(1:3);ROT1_g;COM2_g(1:3);ROT2_g;COM3_g(1:3);ROT3_g;COM4_g(1:3)
    ;ROT4_g;COM5_g(1:3);ROT5_g;COM6_g(1:3);ROT6_g;COM7_g(1:3);ROT7_g];

th=[q1 q2 q3 q4 q5 q6 q7]';

%Jacobian of coordinate vector array

j_sym=jacobian(xu,th);

%joint space inertial matrix

%Jacobian calculated using DH Parameters
JJ_sym=[cross(Tr(1:3,3),(T1_7(1:3,4)-Tr(1:3,4))) cross(T1(1:3,3),(T1_7
    (1:3,4)-T1(1:3,4))) cross(T1_2(1:3,3),(T1_7(1:3,4)-T1_2(1:3,4)))
    cross(T1_3(1:3,3),(T1_7(1:3,4)-T1_3(1:3,4))) cross(T1_4(1:3,3),(T1_7
    (1:3,4)-T1_4(1:3,4))) cross(T1_5(1:3,3),(T1_7(1:3,4)-T1_5(1:3,4)))
    cross(T1_6(1:3,3),(T1_7(1:3,4)-T1_6(1:3,4)));
    Tr(1:3,3) T1(1:3,3) T1_2(1:3,3) T1_3(1:3,3) T1_4(1:3,3) T1_5(1:3,3)
    T1_6(1:3,3)];

%Replace variables with actual angles
q1=q(1);q2=q(2);q3=q(3);q4=q(4);q5=q(5);q6=q(6);q7=q(7);

%Evaluate Joint Space Inertia Matrix
jj=eval(j_sym);
JSIM=jj'*M*jj; %Inertia due to linkages

%Evaluate Forward Kinematics
fk=eval(T1_7);

%Eavaluate Jacobian
JJ=eval(JJ_sym);

%Include approximation for reflected inertia in calculation of Mobillity
I_r=[0.6 0 0 0 0 0 0;
    0 0.6 0 0 0 0 0;
    0 0 0.6 0 0 0 0;
    0 0 0 0.6 0 0 0;
    0 0 0 0 0.044 0 0;
    0 0 0 0 0 0.044 0;
    0 0 0 0 0 0 0.044];


%Calculate mobility
mobility=(JJ)*inv(JSIM+I_r)*(JJ');

%Calculate inertia
CSIM=inv(mobility);

%Determine Eigen values and Eigen vectors of translational portion of
    CSIM

[V,D]=eig(CSIM(1:3,1:3));
```

```
eigen=D;

%Magnitude and orientation of CSIM's Inertial Ellipsoid
alpha=atan2d(V(2,1),V(1,1));
beta=atan2d(-V(3,1),sqrt(V(3,2)^2+V(3,3)^2));
gamma=atan2d(V(3,2),V(3,3));

a=sqrt(eigen(1,1));
b=sqrt(eigen(2,2));
c=sqrt(eigen(3,3));

%Two differnt visual representations of the Inertia: (1) a bar graph
%showing the magnitude in each direction with a coordinate system
    showing
%the orientation and (2) an inertial ellipsoid

a2=V*|a 0 0|';
b2=V*|0 b 0|';
c2=V*|0 0 c|';
figure
aline=line(|0 a2(1)|,|0 a2(2)|,|0 a2(3)|);
aline.LineWidth=2;
aline.Color='blue';
hold on
bline=line(|0 b2(1)|,|0 b2(2)|,|0 b2(3)|);
bline.LineWidth=2;
bline.Color='red';
cline=line(|0 c2(1)|,|0 c2(2)|,|0 c2(3)|);
cline.LineWidth=2;
cline.Color='green';
grid
xlabel('x')
ylabel('y')
zlabel('z')

figure
```

```
a_b=bar([1],[a]);
a_b.FaceColor='blue';
hold on
b_b=bar([2],[b]);
b_b.FaceColor='red';
c_b=bar([3],[c]);
c_b.FaceColor='green';


x0=0; % x0,y0 ellipse centre coordinates
y0=0;
z0=0;

[x, y, z] = ellipsoid(x0,y0,z0,a,b,c,30);
 figure
 S = surfl(x, y, z);
 colormap copper;
 axis equal
 xlabel('m_x kg')
 ylabel('m_y kg')
 zlabel('m_z kg')
 rotate(S,[1 0 0],gamma)
 rotate(S,[0 1 0],beta)
 rotate(S,[0 0 1],alpha)
```

Inertia.m

```
function T=DH(n);
%Author: Lucille Hosford, 2015
%Baxter's DH Parameters
syms q1 q2 q3 q4 q5 q6 q7 real
q=[q1;q2;q3;q4;q5;q6;q7];
theta=[q(1);q(2)+pi/2;q(3);q(4);q(5);q(6);q(7)];
d=[0.27035;0;0.36435;0;0.37429;0;0.229525];
a=[0.069;0;0.069;0;0.01;0;0];
alpha=[-pi/2;pi/2;-pi/2;pi/2;-pi/2;pi/2;0];

% Tl=[0.70711,-0.70711,0,0.064027;
%       0.70711,0.70711,0,0.25903;
%       0,0,1,0.12963;
%       0,0,0,1];
% Tr=[0.70711,0.70711,0,0.064027;
%       -0.70711,0.70711,0,-0.25903;
%       0,0,1,0.12963;
%       0,0,0,1];


%Stanford model
T=[cos(theta(n)), -cos(alpha(n))*sin(theta(n)), sin(alpha(n))*sin(theta(n)), a(n)*cos(theta(n));
    sin(theta(n)), cos(alpha(n))*cos(theta(n)), -sin(alpha(n))*cos(theta(n)), a(n)*sin(theta(n));
    0, sin(alpha(n)), cos(alpha(n)), d(n);
    0, 0, 0, 1];


end
```

DH.m

Appendix B

Gains Used on Controllers in Scaled Down Tasks

Example gains for scaled down task 1: operating in a poorly mapped environment. The gains listed here are only one set of gains used which successfully completed the task. It should be noted that task 1 was successfully completed multiple times using a wide range of gains.

Table B.1: Stiffness of Virtual Springs at Joints

S0	S1	E0	E1	W0	W1	W2
0.0005	5	5	0.005	0.005	5	5

Table B.2: Stiffness of Virtual Springs at End Effector

	Y	X	Z
Stiffness	300	300	50

Example gains for scaled down task 2: scraping biofouling. The gains listed here are only one set of gains used which successfully completed the task. It should be noted that task 1 was successfully completed multiple times using a wide range of gains.

Table B.3: Stiffness of Virtual Springs at Joints

S0	S1	E0	E1	W0	W1	W2
10	30	0.005	0.005	0.005	20	10

Table B.4: Stiffness of Virtual Springs at End Effector

	Y	X	Z
Stiffness	200	200	75

Appendix C

Calculating Baxter's Net Stiffness

```
function [R,K_je,K_net]=Stiffness(q,x_d,x_act)
%Author: Lucille Hosford, 2016
%Calculates net stiffness at end effector given the joint positions, q,
%the equilibrium position of the virtual springs at the end effector,
    x_d,
%and the actual end effector position, x_act.

syms q1 q2 q3 q4 q5 q6 q7 real

%Reference COM from base of shoulder
Tr=[0.70711,0.70711,0,0.064027;
      -0.70711,0.70711,0,-0.25903;
      0,0,1,(0.1185-0.02784);
      0,0,0,1];

T1=Tr*DH(1);
T1_2=Tr*DH(1)*DH(2);
T1_3=Tr*DH(1)*DH(2)*DH(3);
T1_4=Tr*DH(1)*DH(2)*DH(3)*DH(4);
T1_5=Tr*DH(1)*DH(2)*DH(3)*DH(4)*DH(5);
T1_6=Tr*DH(1)*DH(2)*DH(3)*DH(4)*DH(5)*DH(6);
T1_7=Tr*DH(1)*DH(2)*DH(3)*DH(4)*DH(5)*DH(6)*DH(7);

```

```
%Rotation of each segment with respect to each segment's coordinate frame
%[rx,ry,rz];

ROT1_C=[0, 0, q1];
ROT2_C=[0, 0, q2];
ROT3_C=[0, 0, q3];
ROT4_C=[0, 0, q4];
ROT5_C=[0, 0, q5];
ROT6_C=[0, 0, q6];
ROT7_C=[0, 0, q7];


%Reference rotation from base of shoulder

ROT1_g=ROT1_C';
ROT2_g=T1(1:3,1:3)*ROT2_C'+ROT1_C';
ROT3_g=T1_2(1:3,1:3)*ROT3_C'+ROT1_C';
ROT4_g=T1_3(1:3,1:3)*ROT4_C'+ROT1_C';
ROT5_g=T1_4(1:3,1:3)*ROT5_C'+ROT1_C';
ROT6_g=T1_5(1:3,1:3)*ROT6_C'+ROT1_C';
ROT7_g=T1_6(1:3,1:3)*ROT7_C'+ROT1_C';


%joint angles
th=[q1 q2 q3 q4 q5 q6 q7]';
%forward kinematics
%x_sym=[T1_7(1:3,4);ROT7_g];

%Jacobian

%Jacobian
JJ_sym=[cross(Tr(1:3,3),(T1_7(1:3,4)-Tr(1:3,4))) cross(T1(1:3,3),(T1_7(1:3,4)-T1(1:3,4))) cross(T1_2(1:3,3),(T1_7(1:3,4)-T1_2(1:3,4))) cross(T1_3(1:3,3),(T1_7(1:3,4)-T1_3(1:3,4))) cross(T1_4(1:3,3),(T1_7(1:3,4)-T1_4(1:3,4))) cross(T1_5(1:3,3),(T1_7(1:3,4)-T1_5(1:3,4)))
```

```
    cross(T1_6(1:3,3),(T1_7(1:3,4)-T1_6(1:3,4)));
     Tr(1:3,3) T1(1:3,3) T1_2(1:3,3) T1_3(1:3,3) T1_4(1:3,3) T1_5(1:3,3)
    T1_6(1:3,3)];

%Jacobian Transpose
JJ_sym_T=JJ_sym';


M=7;
N=7;

R_sym=sym('A',[7 7]);
%anti_symmetric
%stiffness of springs at end effector
K_end=[300 0 0 0 0 0;
      0 300 0 0 0 0;
      0 0 50 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0];
%symmetric
% K_end=[300 0 0 0 0 0;
%       0 300 0 0 0 0;
%       0 0 300 0 0 0;
%       0 0 0 0 0 0;
%       0 0 0 0 0 0;
%       0 0 0 0 0 0];



F=-K_end*(x_d'-x_act');

 for i=1:M
     for j=1:N
         R_sym(i,j)=diff(JJ_sym_T(i,:),th(j))*F;
     end
 end

```

```
q1=q(1);q2=q(2);q3=q(3);q4=q(4);q5=q(5);q6=q(6);q7=q(7);
%

%evaluate Jacobian
JJ=eval(JJ_sym);

%evaluate forward kinematics
x=eval(T1_7);

%evaluate R
R=eval(R_sym);

%Stiffness at joints
Kj=[0.005 0 0 0 0 0 0;
      0 5 0 0 0 0 0;
      0 0 5 0 0 0 0;
      0 0 0 0.005 0 0 0;
      0 0 0 0 0.005 0 0;
      0 0 0 0 0 5 0;
      0 0 0 0 0 0 5];

%Stiffness from springs at end effector and force due to R
K_je=inv(JJ*inv(Kj-R)*JJ');

%Net Stiffness
K_net=K_je+K_end;

end

%Example configurations


%anti_symm
% x_act=[0.653 -0.2252 -0.2622 0 0 0]
% x_d=[0.66686 -0.224838 -0.03168-0.08-0.4448 0 0 0]
% q0=[0.9407 0.080917 -0.10316 1.102548 -3.0434 -0.3482 1.451529]
```

```


%sym
% x_act=[0.6566  -0.2296  -0.1002 0 0 0]
% q0=[0.948767  -0.4552  -0.10929 1.6  -3.04456  -0.42631 1.4388];
% x_d=[0.66686  -0.224838  -0.03168-0.03-0.0741 0 0 0];
```

Stiffness.m