Evaluation of Beam Load Cell Use for Base Reaction Force Collision Detection on Industrial Robots

by

Robert M. Williams

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 13, 2014

Keywords: Collision Detection, Load Cells, Parameter Estimation, Base Reaction Force

Approved by

David Bevly, Chair, Professor of Mechanical Engineering
Dan Marghitu, Professor of Mechanical Engineering
George Flowers, Professor of Mechanical Engineering

Abstract

This thesis evaluates the ability of four beam load cells placed under the base of an industrial robot to accurately estimate robot link parameters and detect collision. This setup can be a cheaper option than using six-degree-of-freedom sensors, and is easier to implement than other detection methods. The beam load cells are placed under the base of robotic manipulators, and the reactions at the base of the robot due to its motion can be monitored to detect collision and estimate link parameters. The tradeoff to using only four load cells is a more limited ability to sense the reaction forces and moments.

The results of this research show that the method of using beam load cells, despite the limited ability to measure base reactions, was able to reduce parameter estimation error from 1,475 to 432Nm, or in terms of percentage from 28% to 8%. Furthermore, this method is capable of detecting collisions; however, the accuracy of parameter estimation and collision detection is limited by sensor noise and large amplitude vibrations in the tested robot system.

Acknowledgements

Thanks first to Jesus Christ, who continues to guide me through the unknown, provides peace in all circumstances, and loves me enough to die for my salvation despite my inability to repay. Thanks to my family, who are always supportive of me and have blessed me with the opportunity to pursue an education, and who have always shown love and support. Thanks to my friends, who have both kept me sane and put up with me throughout the years, and pretend to understand when I explain my research. Thanks to the guys in the lab for always being willing to help and provide an opinion; I wish you all the best in your future pursuits. Thanks to Dr. Bevly for providing the opportunity to work in the GAVLAB and the guidance he has provided throughout my studies. Finally, thanks to Siemens Healthcare for providing research funds and the opportunity to work on this project.

Table of Contents

List of Tables

List of Figures

List of Symbols

A_i	Matrix of kinematic equations describing the motion of link i
c_i	A vector describing the center of mass location of link i relative to the local coordinate frame P_i
f_{ii}	The force applied on joint i due to the motion of link i
f_{q_i}	The total force applied on mass i
g	Gravity vector in global coordinate frame
i	The joint and attached link number
$I_{3\times3}$	Three dimensional identity matrix
I_{p_i}	Moment of inertia about joint i
I_{q_i}	Moment of inertia about the center of mass of link i
m_i	The mass of link i
n_{ii}	The net torque at joint i due to the motion of link i
n_{q_i}	The net torque about the center of mass of link i
p_i	A vector describing the location of P_i relative to the global coordinate frame
$\ddot{p}_i$	Acceleration of joint i
P_i	The local coordinate frame of link i attached at joint i
P_{i+1}	The local coordinate frame of link $i + 1$
φ_i	Inertial parameters of link i
q_i	A vector describing the center of mass location of link i relative to the global coordinate frame
$\ddot{q}_i$	The acceleration of the center of mass of link i
Q_i	The coordinate frame attached to the center of mass of link i and aligned with the principle axes of inertia
s_i	A vector from joint i to joint $i + 1$ in the local coordinate frame P_i

T_i A rotation matrix describing the wrench transmission relating $w_{i+1,i+1}$ and $w_{i,i+1}$

U_{ij} A rotation matrix describing the wrench transmission relating w_{jj} and w_{ij}

w_{ii} Wrench at joint i due to the motion of link i

ω_i Angular velocity of link i

$\dot{\omega}_i$ Angular acceleration of link i

Chapter 1
Introduction

The use of robots by medical professionals is becoming more common as technology advances. Some doctors save time, travel, and money by working remotely with the use of Remote Presence (RP) devices like the RP-VITA to patrol the hallways and communicate with patients, other physicians, and healthcare providers. The ability to gather medical information and access data to provide expertise and form a medical decision at a moment's notice from a distance opens up healthcare availability while decreasing costs.

Even in the operating room, robots can be used to advance the field of healthcare. The use of complex surgical robots like the ZEUS Robotic Surgical System assist in the control of blunt dissectors, retractors, graspers, and stabilizers during laparoscopic and thoracoscopic surgeries. Robots like ZEUS can help surgeons make smaller invasive incisions and move with more dexterity and precision when performing micro-surgical tasks during surgery.

Computed tomography (CT) scans and x-rays are commonly used and well-known technological procedures. The Artis Zeego surgical system from Siemens Healthcare takes angiography and CT to a new level, providing scans at angles previously not possible by placing a C-arm at the end of a powerful industrial robot. Typically, the CT scan is confined to rotating in a circle, but attaching the C-arm to an industrial robot allows images to be made in almost any direction due to the multiple degrees of freedom of the robot arm. The ability to create three dimensional images gives surgeons great accuracy in filling aneurisms and placing stents. The system is also capable of guiding needle insertions through the use of a laser light mounted on the C-arm which is maneuvered into the correct display angle, and assists in vascular, cardiac, cardiovascular, neuro, trauma, and orthopedics surgery.

1.1 Motivation

From remote communication to operation room surgery, the increased use of medical robotics typically requires an increase in safety regulations. Whether it is a small patrolling remote communication robot like the RP-VITA or a large scale industrial robot like the Siemens Zeego, collision avoidance and detection are of primary concern. While collision of smaller remote communication robots into walls, people, or furniture is not likely to cause any type of significant harm or damage, the powerful manufacturing robots used in the Zeego system come with a much higher risk of personal injury to surgeons and patients, damage to surrounding structures, and destruction of property to nearby chairs and tables.

Although collision avoidance within a defined workspace has been implemented in the Zeego system to avoid permanent objects with known locations such as the walls and the operating table, the risk of collision with variable parameters including surgeons, patients, and medical carts still remains. It has been widely shown that collision detection is possible [1, 2] and able to stop robot motion before significant harm is caused [3] through the use of six-degree-of-freedom sensors that are installed between joints or through the use of current or voltage measurements to the joint motors to determine the torque applied to a joint. Another method of detection uses piezoelectric material to cover the robot and sense impulses due to collision [4].

While these measurements can be very accurate and informative, in some cases six-degree-of-freedom joint sensing or current and voltage measurements are not feasible or possible, especially since these joint sensors must usually be applied during the manufacturing of the robot. Adding a piezoelectric cover has the advantage of being simple to apply, but has its drawbacks as well due to the added weight to the robot links and aesthetic complications. In the case of the Artis Zeego, an alternative approach is desired in which collision detection is made with the use of four load cells installed underneath the base of the robot due to the ease of installation and relatively low cost. While the goal is to detect 5N collision forces, the acceptable level has been set to a maximum of 50N collision force detection because this can be used as a general definition of the limit of human pain tolerance [5]. This thesis has two goals: first, to present experimental tests of base force reaction measurements and analyze the limits of the use of four load cells underneath a robot; and second, to determine if this approach of collision detection can be considered as a viable option for large medical robots like the Artis Zeego.

1.2 Previous Work

Much research has been invested in robotics modeling, parameter estimation, and collision detection. A brief overview is covered below to provide some background into these areas as well as to present the challenges faced and provide context for the work presented in this thesis.

1.2.1 Modeling History

For a significant amount of research put into modeling the dynamics of robots, a very good estimate of the robot parameters is the end goal without much regard to cost. To reach this goal, expensive six-degree-of-freedom sensors are used, as well as voltage and current measurements in the motors to determine motor torques. In some cases, the manipulator arm is disassembled and each link is measured individually for its inertial properties [6]. Depending on the robot design, this method could be impractical or impossible. Computer-aided design (CAD) based models of robots have also been created to determine the robot parameters. However, accurate CAD models can be difficult to create with large, complex systems where the materials may not be fully known.

Some mechanical models have been created using PD controllers to stabilize the global position of a robot asymptotically around a certain joint configuration. This method requires gravity compensation, and different techniques such as online versus constant linear and nonlinear gravity compensation have been studied [7]. However, the accuracy of these systems is limited to the point about which the system is stabilized, which does not allow for large motions. This thesis uses MATLAB to create the manipulator model, and the model parameters are determined from the parameter estimation of the actual system.

1.2.2 Parameter Estimation History

Estimating manipulator link parameters through reaction measurements typically involves measuring reaction forces and moments at the robot base or between joints. Some difficulties either preventing or requiring much effort to obtain accurate measurement of inertial parameters include unmodeled joint friction, actuator dynamics, and the difficulty of measuring joint accelerations, especially at low speeds, due to noise. Using external force and torque sensors as opposed to joint force and torque sensors removes the internal effects of actuator dynamics and joint friction. For example, Barreto and Muñoz [8] determine the inertial parameters of rigid bodies through the use of a parallel robot, also known as a Stewart Platform, by tilting the body in several different orientations and then measuring the angular velocity as well as the resulting reaction forces on a six axis load cell.

Liu et al. [9] use a six-axis force and torque sensor along with joint velocity measurements during manipulator motion to estimate the inertial parameters. Linear combinations of the parameters are made to reduce the number of estimated parameters. For example, due to motion constraints from the joint between two links, certain parameters of the two links may be related and estimated as a single value. Ridge regression and singular value decomposition methods are then used to solve the resulting vector of parameter combinations. A filter and Laplace transformation eliminate the need for unreliable joint acceleration measurements. The joints are moved through predetermined motions while the base force reactions and joint angular positions and velocities are measured.

Gautier and Khalil [10] eliminate parameters that have no effect on the dynamic system and regroup others by means of closed-form relations in order to create a minimum set of inertial parameters, defined as "the minimum set of constant inertial parameters that do not contain the zero element and are sufficient to calculate the dynamic model of the robot." This regrouping reduces computational costs and simplifies inertial parameter identification.

An, Atkeson, and Hollerbach [11] derive a set of matrix equations from the kinematic Newton-Euler equations to describe the reaction forces at the robot joints. The parameters are estimated with the use of the pseudoinverse of the resulting matrix. Not all parameters are identifiable. The method in this thesis uses the matrix equations described by An et al. for both modeling and to define the expected base reactions to the robot motion.

1.2.3 Collision Avoidance History

Much research has also gone into collision avoidance. Collision avoidance is distinct from collision detection in that collision avoidance seeks to prevent any contact with humans or the environment, as opposed to stopping manipulator motion after contact has already occurred and subsequently been detected. Collision avoidance typically employs vision or proximity sensors [12-14]. However, some drawbacks include requiring computationally heavy image processing, poor distinguishing ability in a busy area, and requiring a clear line of sight.

By combining force control and collision avoidance algorithms, a fuzzy control approach has shown to help with force tracking in highly variable or unknown environmental parameters [15]. An approach similar to the workspace obstacle avoidance method mentioned previously, where the static environment consisting of walls and immovable objects is known, is the position-based force control [16]. Virtual forces are applied to the robot to "push" its links away from a potential collision by monitoring objects in a dynamic work space. This method requires the use of sensors to find and map objects in the planned motion path. Collision avoidance is not implemented on the system in this thesis, but the distinction between collision avoidance and collision detection is important.

1.2.4 Collision Detection History

In rare cases, continuing the motion of the robot after collision may be desired. As mentioned previously, placing sensors on the exterior of the robot is possible. In an environment where humans and robots are working in close proximity, these sensors can be used to slow the robot after collision just enough to allow the human reflex enough time to move out of the way, given that the collision was within a certain tolerance limit [5]. Setting a tolerance limit requires a somewhat ambiguous definition of what magnitude of force is within a tolerable range of pain for humans, as well as modification to the existing robot design through the addition of coverings.

In other cases, damage will very likely be done if the robot continues its motion after collision. Therefore, it is necessary that the robot stops its motion immediately. An example would be a powerful industrial robot colliding with an object that has been left nearby, like a chair which could be crushed and broken. Robots manipulating sharp utensils, such as those possibly performing surgeries, should stop their motion when a collision is sensed with an unknown object in order to prevent harm to patients and doctors. It has been shown that motion can be stopped in time to avoid damage to biological tissue, even when the contacting edge is a sharp knife [17]. The method presented by Lu and Chung [17] utilizes joint torque sensing to measure contact forces.

A method using weighted path planning has been suggested in which a combination of a base reaction sensor and a wrist sensor are used to detect collisions [18]. This design has the benefit of continuing motion by limiting the force applied in collision but has the drawback of expensive six-degree-of-freedom sensors - one of which is between robot joints that would require complicated installation. The method presented in this thesis implements the use of load cells placed under the base of the robot.

1.3 Contributions

This research involves the use of two separate robotic manipulators: the CRS A255 and the Artis Zeego. The CRS A255, a small 17kg robot, is used for initial algorithm verification and tests the sensor setup before being implemented on the Artis Zeego system, which is considerably larger. Through modeling and experimentation using both robots, this thesis provides the following contributions:

- An evaluation of estimating robot manipulator parameters with limited base force and moment reaction measurement capabilities.

- An evaluation of collision detection sensing in a manipulator with limited base reaction measurement capabilities.
- Identification of shortcomings and possible improvements to parameter estimation and collision detection capabilities.

1.4 Thesis Outline

This opening chapter has provided a brief overview of the current state of robotics in the medical field as well as background and context for the work presented in this thesis. In Chapter 2, the basic idea behind collision detection is discussed, and the method used in this thesis for parameter estimation is presented. The kinematic equations relating to this work are also explained.

A description of the model creation is covered in Chapter 3. MATLAB is used to create models for two different robots. The first robot, the CRS A255, is used for model verification and algorithm testing to be used on the second robot, an industrial robot provided by Siemens. The methods used to create the models are explained. Chapter 4 then provides details on the experimental setup of the two systems, describing the equipment and the procedures for equipment setup.

Finally, the processes used to collect data and the experimental results from the data collection are presented in Chapters 5 and 6 for the CRS A255 and the Zeego, respectively. Also, the robot models created in Chapter 3 are compared with the experimental results, and the similarities and differences of the predicted and measured base reactions are discussed. Based on these comparisons, conclusions are drawn and suggestions for future improvements are made in Chapter 7.

Chapter 2

Collision Detection and Parameter Estimation Theory and Equation Formation

In order to detect collision, it is necessary to have a model of the robot manipulator. To create an accurate model of the manipulator, the link parameters need to be estimated in order to be applied in the model. The details of and ideas behind this process are explained in this chapter.

2.1 Collision Detection

Collision detection on a robotic arm requires the use of sensors to measure unexpected disturbances to the robot. Therefore, two main things must be considered: an expected measurement and a current measurement. There must be an expected measurement with which to compare the current measurement; the current measurements would be made as the robot is in motion or perhaps in some cases, standing still. This is the live feedback of what the system is currently doing. If a current measurement is made that conflicts with or in some way varies from what is expected to be measured, it can be assumed that a collision has occurred. In order to have an expected measurement, it is necessary to have some method to determine what the expected measurements should be.

As an example, imagine the previously mentioned work that suggests placing a piezoelectric cover over the outside surface of the robot to detect collision [4]. In this case, the expected sensor measurements would be zero, meaning there is no collision to cause a signal. A signal would be detected in two cases: 1) a temperature change has occurred in the piezoelectric film, or 2) an impulse has been applied to the film. In a hospital operating room environment, it is expected that the temperature would vary minimally, thus the only expected measurements would be from an impact. If the measurement at any point varies from zero, it is assumed that a collision has been detected and the motion of the robot is stopped.

This thesis studies the limited reaction measurement capability of four beam load cells used at the base of the robot. These measurements are compared to the expected reactions, determined from a model of the robot. If the parameters of each link are known or can be estimated, along with the angular positions and accelerations of each joint, then the expected reactions at the base of the robot can be modeled using Newton-Euler equations. Thus, the expected measurements would be the expected forces and moments at the base of the robot determined by the model, and the current measurements would be the actual forces and moments being measured at the base of the robot by the load cell sensors. The implementation of the four beam load cell sensors is described in Chapter 4.

2.1.1 Base Reaction Equation Formation

This section describes the formation of the kinematic equations used to create a model of the robot. This model is then used for both parameter estimation and collision detection by calculating the expected base reactions, since the set of equations provides a relationship between the link parameters, link motion, and base reaction forces and moments. These equations have been previously presented and shown to work [11, 19].

The system must first be defined. Figure 2.1 shows an illustration of the following coordinate frame and location vector definitions. The List of Symbols page lists a summary of all the nomenclature used in this section. Bold terms denote matrices or vectors. For a multiple link system, each link has a local coordinate frame P_i at the link joint i. The frame P_i at joint i is located with respect to the global coordinate frame at the robot base by vector $\boldsymbol{p_i}$. The i link's center of mass, m_i, is located with respect to joint i by the vector $\boldsymbol{c_i}$ in the local coordinate frame and with respect to the global coordinate frame by vector $\boldsymbol{q_i}$. Another coordinate frame Q_i, attached at the center of mass, is aligned with the principle axes of inertia. The location of the joint where the next link and its local coordinate frame P_{i+1} are attached is located with respect to P_i by $\boldsymbol{s_i}$.

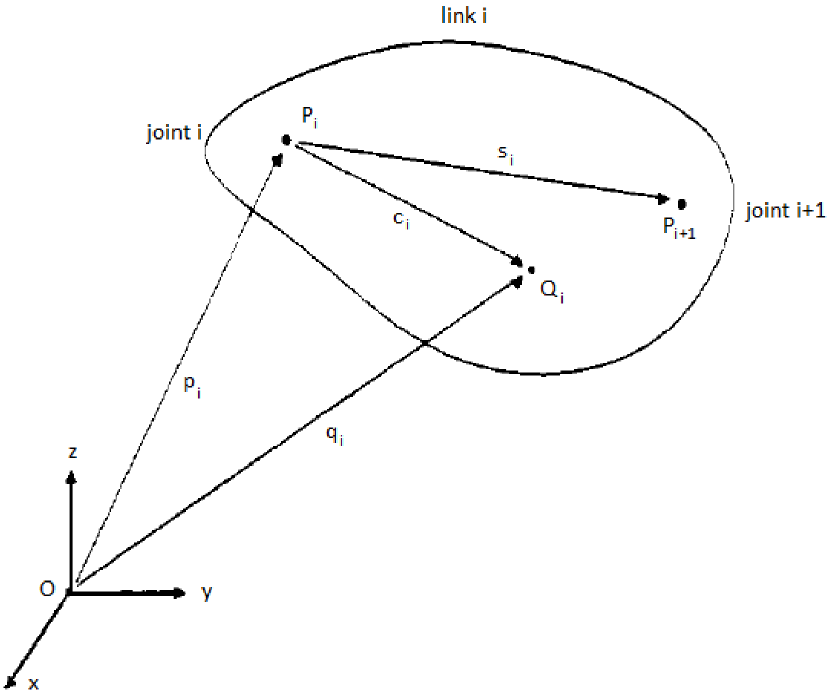

Figure 2.1: Illustration of link joints and location vectors definitions [11].

With the system definition in mind, the following terms can be applied to the Newton-Euler equations describing force and moment:

$$\boldsymbol{f}_{\boldsymbol{q}_i} = \boldsymbol{f}_{\boldsymbol{ii}} + m_i\boldsymbol{g} = m_i\ddot{\boldsymbol{q}}_{\boldsymbol{i}} \tag{2.1}$$

$$\boldsymbol{n}_{\boldsymbol{q}_i} = \boldsymbol{n}_{\boldsymbol{ii}} - \boldsymbol{c}_{\boldsymbol{i}} \times \boldsymbol{f}_{\boldsymbol{ii}} = \boldsymbol{I}_{\boldsymbol{q}_i}\dot{\boldsymbol{\omega}}_{\boldsymbol{i}} + \boldsymbol{\omega}_{\boldsymbol{i}} \times \left(\boldsymbol{I}_{\boldsymbol{q}_i}\boldsymbol{\omega}_{\boldsymbol{i}}\right) \tag{2.2}$$

Looking at Equation (2.1), $\boldsymbol{f}_{\boldsymbol{q}_i}$ defines the total force acting on the link mass at $\boldsymbol{q}_{\boldsymbol{i}}$ and is a three-by-one vector of the three force components. This force is equivalent to $\boldsymbol{f}_{\boldsymbol{ii}}$, the forces acting at joint i due to the motion of link i, summed with the gravity force $m_i\boldsymbol{g}$. This force is also equivalent to the mass m_i of link i multiplied by the acceleration of the location of the center of mass $\ddot{\boldsymbol{q}}_{\boldsymbol{i}}$.

In Equation (2.2), $\boldsymbol{n}_{\boldsymbol{q}_i}$ defines the net torque acting on the mass about $\boldsymbol{q}_{\boldsymbol{i}}$ and is a three-by-one vector of the three moment components. This torque is equivalent to the torque at joint i due to the motion of link i, minus the effects of the forces acting at joint i due to the motion of link i which are causing a torque about the center of mass. This torque is also equivalent to the effects of inertia $\boldsymbol{I}_{\boldsymbol{q}_i}$ about $\boldsymbol{q}_{\boldsymbol{i}}$, along with the angular acceleration $\dot{\boldsymbol{\omega}}_{\boldsymbol{i}}$ and angular velocity $\boldsymbol{\omega}_{\boldsymbol{i}}$ of the link.

These previous two equations relate all the inertial parameters of mass, location of center of mass, and moments of inertia. Next, the relationship of $\ddot{\boldsymbol{q}}_{\boldsymbol{i}}$ and $\ddot{\boldsymbol{p}}_{\boldsymbol{i}}$ is defined by the following kinematic equation:

$$\ddot{\boldsymbol{q}}_{\boldsymbol{i}} = \ddot{\boldsymbol{p}}_{\boldsymbol{i}} + \dot{\boldsymbol{\omega}}_{\boldsymbol{i}} \times \boldsymbol{c}_{\boldsymbol{i}} + \boldsymbol{\omega}_{\boldsymbol{i}} \times (\boldsymbol{\omega}_{\boldsymbol{i}} \times \boldsymbol{c}_{\boldsymbol{i}}) \tag{2.3}$$

The acceleration of the center of mass of a link is related to the acceleration of the link joint, the angular acceleration and angular velocity of the link, and the distance between the two points $\boldsymbol{p}_{\boldsymbol{i}}$ and $\boldsymbol{q}_{\boldsymbol{i}}$, defined by $\boldsymbol{c}_{\boldsymbol{i}}$.

Equation (2.3) can then be plugged into Equation (2.1) to solve for the forces $\boldsymbol{f}_{\boldsymbol{ii}}$ acting on joint i due to the motion of link i:

$$\begin{aligned}
\boldsymbol{f}_{\boldsymbol{ii}} + m_i\boldsymbol{g} &= m_i\ddot{\boldsymbol{q}}_{\boldsymbol{i}} \\
\boldsymbol{f}_{\boldsymbol{ii}} &= m_i\ddot{\boldsymbol{q}}_{\boldsymbol{i}} - m_i\boldsymbol{g} \\
\boldsymbol{f}_{\boldsymbol{ii}} &= m_i\big(\ddot{\boldsymbol{p}}_{\boldsymbol{i}} + \dot{\boldsymbol{\omega}}_{\boldsymbol{i}} \times \boldsymbol{c}_{\boldsymbol{i}} + \boldsymbol{\omega}_{\boldsymbol{i}} \times (\boldsymbol{\omega}_{\boldsymbol{i}} \times \boldsymbol{c}_{\boldsymbol{i}})\big) - m_i\boldsymbol{g} \\
\boldsymbol{f}_{\boldsymbol{ii}} &= m_i(\ddot{\boldsymbol{p}}_{\boldsymbol{i}} - \boldsymbol{g}) + \dot{\boldsymbol{\omega}}_{\boldsymbol{i}} \times m_i\boldsymbol{c}_{\boldsymbol{i}} + \boldsymbol{\omega}_{\boldsymbol{i}} \times (\boldsymbol{\omega}_{\boldsymbol{i}} \times m_i\boldsymbol{c}_{\boldsymbol{i}}) \qquad (2.4)
\end{aligned}$$

Next, Equation (2.4) is used to rewrite Equation (2.2) to solve for the moments $\boldsymbol{n}_{\boldsymbol{ii}}$ acting on joint i due to the motion of link i:

$$\begin{aligned}
\boldsymbol{n}_{\boldsymbol{ii}} - \boldsymbol{c}_{\boldsymbol{i}} \times \boldsymbol{f}_{\boldsymbol{ii}} &= \boldsymbol{I}_{\boldsymbol{q}_i}\dot{\boldsymbol{\omega}}_{\boldsymbol{i}} + \boldsymbol{\omega}_{\boldsymbol{i}} \times \big(\boldsymbol{I}_{\boldsymbol{q}_i}\boldsymbol{\omega}_{\boldsymbol{i}}\big) \\
\boldsymbol{n}_{\boldsymbol{ii}} &= \boldsymbol{c}_{\boldsymbol{i}} \times \boldsymbol{f}_{\boldsymbol{ii}} + \boldsymbol{I}_{\boldsymbol{q}_i}\dot{\boldsymbol{\omega}}_{\boldsymbol{i}} + \boldsymbol{\omega}_{\boldsymbol{i}} \times \big(\boldsymbol{I}_{\boldsymbol{q}_i}\boldsymbol{\omega}_{\boldsymbol{i}}\big) \\
\boldsymbol{n}_{\boldsymbol{ii}} &= \boldsymbol{c}_{\boldsymbol{i}} \times \big(m_i(\ddot{\boldsymbol{p}}_{\boldsymbol{i}} - \boldsymbol{g}) + \dot{\boldsymbol{\omega}}_{\boldsymbol{i}} \times m_i\boldsymbol{c}_{\boldsymbol{i}} + \boldsymbol{\omega}_{\boldsymbol{i}} \times (\boldsymbol{\omega}_{\boldsymbol{i}} \times m_i\boldsymbol{c}_{\boldsymbol{i}})\big) \\
&\quad + \boldsymbol{I}_{\boldsymbol{q}_i}\dot{\boldsymbol{\omega}}_{\boldsymbol{i}} + \boldsymbol{\omega}_{\boldsymbol{i}} \times \big(\boldsymbol{I}_{\boldsymbol{q}_i}\boldsymbol{\omega}_{\boldsymbol{i}}\big)
\end{aligned}$$

$$\begin{aligned}\boldsymbol{n_{ii}} = (\boldsymbol{g} - \ddot{\boldsymbol{p}}_i) \times m_i\boldsymbol{c_i} + \boldsymbol{I}_{\boldsymbol{q}_i}\dot{\boldsymbol{\omega}}_i + m_i\boldsymbol{c_i} \times (\dot{\boldsymbol{\omega}}_i \times \boldsymbol{c_i}) \\ +\boldsymbol{\omega_i} \times (\boldsymbol{I}_{\boldsymbol{q}_i}\boldsymbol{\omega_i}) + m_i\boldsymbol{c_i} \times (\boldsymbol{\omega_i} \times (\boldsymbol{\omega_i} \times \boldsymbol{c}_i))\end{aligned} \tag{2.5}$$

Note that $m_i\boldsymbol{c_i} \times (\ddot{\boldsymbol{p}}_i - \boldsymbol{g})$ switches to $(\boldsymbol{g} - \ddot{\boldsymbol{p}}_i) \times m_i\boldsymbol{c_i}$ due to the anticommutative property of cross products. This formulation is more convenient for later calculations. Also, the terms $m_i\boldsymbol{c_i} \times (\dot{\boldsymbol{\omega}}_i \times \boldsymbol{c}_i)$ and $m_i\boldsymbol{c_i} \times (\boldsymbol{\omega_i} \times (\boldsymbol{\omega_i} \times \boldsymbol{c}_i))$ cause the center of mass location to be a quadratic solution, which increases calculation difficulty, particularly when solving for the unknown location values. This issue is simplified with the generalized version of the parallel axis theorem:

$$\boldsymbol{I}_{\boldsymbol{p}_i} = \boldsymbol{I}_{\boldsymbol{q}_i} + m_i[(\boldsymbol{c}_i^T\boldsymbol{c}_i)\boldsymbol{I}_{3\times3} - (\boldsymbol{c}_i\boldsymbol{c}_i^T)] \tag{2.6}$$

$\boldsymbol{I}_{\boldsymbol{p}_i}$ is the inertia about the parallel axis. The term $\boldsymbol{I}_{3\times3}$ represents the three dimensional identity matrix. Since $m_i\boldsymbol{c_i} \times (\dot{\boldsymbol{\omega}}_i \times \boldsymbol{c}_i) = m_i[(\boldsymbol{c}_i^T\boldsymbol{c}_i)\boldsymbol{I}_{3\times3} - (\boldsymbol{c}_i\boldsymbol{c}_i^T)]\dot{\boldsymbol{\omega}}_i$ and $m_i\boldsymbol{c_i} \times (\boldsymbol{\omega_i} \times (\boldsymbol{\omega_i} \times \boldsymbol{c}_i)) = \boldsymbol{\omega_i} \times (m_i[(\boldsymbol{c}_i^T\boldsymbol{c}_i)\boldsymbol{I}_{3\times3} - (\boldsymbol{c}_i\boldsymbol{c}_i^T)]\boldsymbol{\omega}_i)$ from the skew matrix identity, Equation (2.5) can then be rewritten:

$$\begin{aligned}\boldsymbol{n_{ii}} = (\boldsymbol{g} - \ddot{\boldsymbol{p}}_i) \times m_i\boldsymbol{c_i} + \boldsymbol{I}_{\boldsymbol{q}_i}\dot{\boldsymbol{\omega}}_i + m_i[(\boldsymbol{c}_i^T\boldsymbol{c}_i)\boldsymbol{I}_{3\times3} - (\boldsymbol{c}_i\boldsymbol{c}_i^T)]\dot{\boldsymbol{\omega}}_i \\ +\boldsymbol{\omega_i} \times (\boldsymbol{I}_{\boldsymbol{q}_i}\boldsymbol{\omega_i}) + \boldsymbol{\omega_i} \times (m_i[(\boldsymbol{c}_i^T\boldsymbol{c}_i)\boldsymbol{I}_{3\times3} - (\boldsymbol{c}_i\boldsymbol{c}_i^T)]\boldsymbol{\omega}_i)\end{aligned} \tag{2.7}$$

Finally, Equation (2.6) is applied to Equation (2.7):

$$\boldsymbol{n_{ii}} = (\boldsymbol{g} - \ddot{\boldsymbol{p}}_i) \times m_i\boldsymbol{c_i} + \boldsymbol{I}_{\boldsymbol{p}_i}\dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega_i} \times (\boldsymbol{I}_{\boldsymbol{p}_i}\boldsymbol{\omega}_i) \tag{2.8}$$

The force and moment equations are now established. Ideally, the force and moment equations can be combined into one kinematic matrix expression that can be multiplied with a vector of the parameters. In order to reach this formulation, the equations need to be rewritten further. Since a cross product can be expressed as a product of a skew-symmetric matrix and a vector, the following definition is used:

$$\boldsymbol{\omega} \times \boldsymbol{c} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \triangleq [\boldsymbol{\omega} \times]\boldsymbol{c} \tag{2.9}$$

Similarly, given that $\boldsymbol{I}$ is the three by three inertia matrix,

$$\boldsymbol{I\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z & 0 & 0 & 0 \\ 0 & \omega_x & 0 & \omega_y & \omega_z & 0 \\ 0 & 0 & \omega_x & 0 & \omega_y & \omega_z \end{bmatrix} \begin{bmatrix} I_{xx} \\ I_{xy} \\ I_{xz} \\ I_{yy} \\ I_{yz} \\ I_{zz} \end{bmatrix} \triangleq [\cdot\, \boldsymbol{\omega}] \begin{bmatrix} I_{xx} \\ I_{xy} \\ I_{xz} \\ I_{yy} \\ I_{yz} \\ I_{zz} \end{bmatrix} \tag{2.10}$$

Finally, a compact matrix equation can be written by applying Equations (2.9) and (2.10) to Equations (2.4) and (2.8):

$$\begin{bmatrix} \boldsymbol{f_{ii}} \\ \boldsymbol{n_{ii}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\ddot{p}_i} - \boldsymbol{g} & [\boldsymbol{\dot{\omega}_i} \times] + [\boldsymbol{\omega_i} \times][\boldsymbol{\omega_i} \times] & \boldsymbol{0} \\ \boldsymbol{0} & [(\boldsymbol{g} - \boldsymbol{\ddot{p}_i}) \times] & [\cdot\, \boldsymbol{\dot{\omega}_i}] + [\boldsymbol{\omega_i} \times][\cdot\, \boldsymbol{\omega_i}] \end{bmatrix} \begin{bmatrix} m_i \\ m_i c_{x_i} \\ m_i c_{y_i} \\ m_i c_{z_i} \\ I_{xx_i} \\ I_{xy_i} \\ I_{xz_i} \\ I_{yy_i} \\ I_{yz_i} \\ I_{zz_i} \end{bmatrix} \tag{2.11}$$

Equation (2.11) now relates the force and moment reactions at a joint i to the inertial parameters and motion of link i. The forces and moments are combined into one six-by-one vector. This equation can be written more simply as

$$\boldsymbol{w_{ii}} = \boldsymbol{A_i \varphi_i} \tag{2.12}$$

In Equation (2.12), $\boldsymbol{w_{ii}}$, the wrench, is the six-by-one vector of forces and torques at joint i due to the motion of link i. The motion of link i is described by the kinematic matrix $\boldsymbol{A_i}$, which is a six-by-ten matrix multiplied by the ten-by-one vector of unknown inertial link parameters $\boldsymbol{\varphi_i}$. To define the total wrench at a joint i,

$$\boldsymbol{w_i} = \sum_{j=i}^{N} \boldsymbol{w_{ij}} \tag{2.13}$$

The wrench at any joint i is the sum of each wrench on the links from i to the last link N. However, each successive wrench $\boldsymbol{w_{ij}}$ needs to be rotated from its local coordinate frame into the link i coordinate frame P_i. The rotation matrix between the P_i coordinate frame and the P_{i+1} coordinate frame is defined as $\boldsymbol{R_i}$. Therefore:

$$\begin{bmatrix} \boldsymbol{f_{i,i+1}} \\ \boldsymbol{n_{i,i+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R_i} & \boldsymbol{0} \\ [\boldsymbol{s_i} \times] \cdot \boldsymbol{R_i} & \boldsymbol{R_i} \end{bmatrix} \begin{bmatrix} \boldsymbol{f_{i+1,i+1}} \\ \boldsymbol{n_{i+1,i+1}} \end{bmatrix} \tag{2.14}$$

Equation (2.14) is expressed more simply as:

$$\boldsymbol{w_{i,i+1}} = \boldsymbol{T_i w_{i+1,i+1}} \tag{2.15}$$

This relates the wrench on one joint due to the wrench acting upon the next joint, and $\boldsymbol{T_i}$ is a six-by-six matrix called the wrench transmission matrix from joint $i+1$ to joint i. The wrench at joint i is therefore related to the wrench at a distant joint j by cascading the wrench transmission matrices:

$$\begin{aligned} \boldsymbol{w_{ij}} &= \boldsymbol{T_i T_{i+1} \dots T_{j-1} w_{jj}} \\ &= \boldsymbol{T_i T_{i+1} \dots T_{j-1} A_j \varphi_j} \\ &= \boldsymbol{U_{ij} \varphi_j} \end{aligned} \tag{2.16}$$

Finally, by combining Equations (2.16) and (2.13), a matrix expression is derived to represent the wrench at each joint of a serial kinematic chain:

$$\begin{bmatrix} \boldsymbol{w_1} \\ \boldsymbol{w_2} \\ \boldsymbol{w_3} \\ \vdots \\ \boldsymbol{w_N} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U_{11}} & \boldsymbol{U_{12}} & \boldsymbol{U_{13}} & \cdots & \boldsymbol{U_{1N}} \\ \mathbf{0} & \boldsymbol{U_{22}} & \boldsymbol{U_{23}} & \cdots & \boldsymbol{U_{2N}} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{U_{33}} & \cdots & \boldsymbol{U_{3N}} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & \cdots & \boldsymbol{U_{NN}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varphi_1} \\ \boldsymbol{\varphi_2} \\ \boldsymbol{\varphi_3} \\ \vdots \\ \boldsymbol{\varphi_N} \end{bmatrix} \tag{2.17}$$

This equation relates the reaction forces and moments at each joint to the inertial parameters of each link due to the collective motion of all the links in the kinematic chain of the manipulator. Ultimately, the only wrench of interest in this thesis is $\boldsymbol{w_1}$ since it is the wrench at the base of the robot, where the sensors are placed.

2.2 Parameter Estimation

In order to create an accurate model for collision detection measurements, the parameters of the robot need to be determined. Generally, the link parameters of manipulator arms are unknown, because in most cases manipulators are used for “pick and place” operations, where the manipulator end is manually controlled into certain positions. Typically, each position is saved to memory, to be recalled as the manipulator moves between the two or more positions automatically. Therefore, the position locations are based solely on the axis angle information, and joint encoders are usually the only feedback required for manipulator operation. Usually no collision detection or sensing of any kind is implemented in industrial robots besides the encoder measurements, and it is not necessary to have knowledge of the link parameter values for most applications. This lack of information in industrial robots is why parameter estimation techniques have been developed and employed on robotic manipulators through the use of force and torque sensors.

Two main types of parameter identification techniques exist using force and torque sensors. The first type, internal models, use force and torque sensors between joints to model robot dynamics but must model the frictional forces between the joints being measured as well. However, the advantage is the ability to more precisely locate the external forces acting on the robot. For example, if a disturbance is measured by one joint sensor but not by the next, the disturbing force must have occurred on the link between the two.

The second type, external models, use force and torque sensors at the base of the robot to measure the reactions with the advantage of requiring no model of the internal friction forces, but more computationally intensive calculations are necessary to determine manipulator parameters. Techniques simultaneously using internal and external parameter identification techniques have been shown to have low model errors [18]. The method studied in this thesis is solely an external technique, and the estimation method is shown in the following section.

2.2.1 Robot Parameter Estimation Technique

The estimation technique used for this thesis is similar to the technique used by An, Atkeson, and Hollerbach [11]. Since the only wrench of interest is $\boldsymbol{w_1}$, Equation (2.17) can be modified in the following way:

$$\boldsymbol{w_1} = [\boldsymbol{U_{11}} \quad \boldsymbol{U_{12}} \quad \cdots \quad \boldsymbol{U_{1N}}] \begin{bmatrix} \boldsymbol{\varphi_1} \\ \boldsymbol{\varphi_2} \\ \vdots \\ \boldsymbol{\varphi_N} \end{bmatrix} \tag{2.18}$$

Equation (2.18) is defined more simply in the following form:

$$\boldsymbol{w} = \boldsymbol{U\varphi} \tag{2.19}$$

Equation (2.19) defines the actual wrench at the base due to the motion of the links with certain parameters. $\boldsymbol{w}$ is a six-by-one vector of wrench at the base, $\boldsymbol{U}$ is a six-by-$10N$ matrix, and $\boldsymbol{\varphi}$ is a $10N$-by-one vector of parameters. The estimated wrench due to the motion of the links with the estimated parameters is defined as follows:

$$\boldsymbol{\widehat{w}} = \boldsymbol{U\widehat{\varphi}} \tag{2.20}$$

Now, since the link motions are known through axis encoder measurements, the parameters can be estimated. First, if the difference is taken between the estimated wrench calculated from the model and the actual wrench measured by the load cells, the following equation results:

$$\begin{gathered}\boldsymbol{w} - \boldsymbol{\widehat{w}} = \boldsymbol{U\varphi} - \boldsymbol{U\widehat{\varphi}} = \boldsymbol{U} * (\boldsymbol{\varphi} - \boldsymbol{\widehat{\varphi}}) \\ \Delta\boldsymbol{w} = \boldsymbol{U} * \Delta\boldsymbol{\varphi} \qquad (2.21)\end{gathered}$$

Next, to determine the error difference in the vector of estimated parameters, the inverse is taken. Note that a least squares estimate cannot be performed because $\boldsymbol{U^TU}$ is not invertible due to the loss of rank from restricted degrees of freedom at the joints and the lack of full force and moment sensing [11]:

$$\Delta\boldsymbol{\varphi} = (\boldsymbol{U^TU})^{-1}\boldsymbol{U^T}\Delta\boldsymbol{w}$$

However, the pseudo inverse of $\boldsymbol{U}$ can be taken instead because the singular value decomposition of $\boldsymbol{U}$ is performed [20]. At each time step the parameter error difference is used to update the estimated parameters to be used in the next time step:

$$\boldsymbol{\widehat{\varphi}_{k+1}} = \boldsymbol{\widehat{\varphi}_k} - \Delta\boldsymbol{\varphi} \tag{2.22}$$

The application and results of this estimation technique are discussed in Chapter 5.

Chapter 3

CRS A255

This chapter covers the model creation, experimental setup, and data collection of the CRS A255 robot used for initial development and experimentation. Section 3.1 presents the model creation and results, Section 3.2 explains the experimental setup, and Section 3.3 covers the data collection and analysis.

A major difference between this thesis and most previous studies into force control and collision detection is that the previous works are not typically measuring high dynamics, which usually contain noise and vibrations. Instead, the end goal of these previous works is either precision control or force control for high-torque systems, which usually have poor precision due to friction effects at low manipulator speeds [21,22]. Precision control seeks to make the position of the end effector very precise in small amplitude motions, and force control seeks to make the applied force at the end effector very precise.

This thesis seeks to provide the results of high amplitude motion measurements as an evaluation of beam load cell use for base reaction force collision detection on industrial robots. The CRS A255 manipulator was used to verify the MATLAB model, the kinematic equations presented in Chapter 2, and the ability to measure reaction forces with the four beam load cell setup.

3.1 CRS A255 Model Creation

Based off of the equations presented in Section 2.1.1, kinematic models of the robotic manipulator is created using MATLAB. First, the global coordinate frame is defined. The x-axis originates at the center of the base of the robot and is directed forward relative to the robot’s home position. The y-axis is to the left of the direction the robot faces in its home position and the z-axis points directly up. The robot's first axis of rotation is about the global z-axis. In each link's local coordinate frame, the x-axis is along the direction of the link from joint i to joint $i + 1$.

Figure 3.1 shows the CRS A255 robot used for initial development at Auburn. In the figure, the x-axis would be coming out of the page, the y-axis to the right, and the z-axis straight up. This system can be simplified in the MATLAB model and defined as a two-link system. Joint 1 has two rotation axes: first, about the global z-axis, and second, about the local y-axis. The second joint rotates about its local y-axis.

To run a model simulation and obtain reaction measurements, the parameters must be known. Since the parameters are not yet known, however, for now it is sufficient to make some educated guesses of the parameter values simply to test the model and check that the results seem reasonable. Once the parameter estimation tests are run, those estimated values will be used.

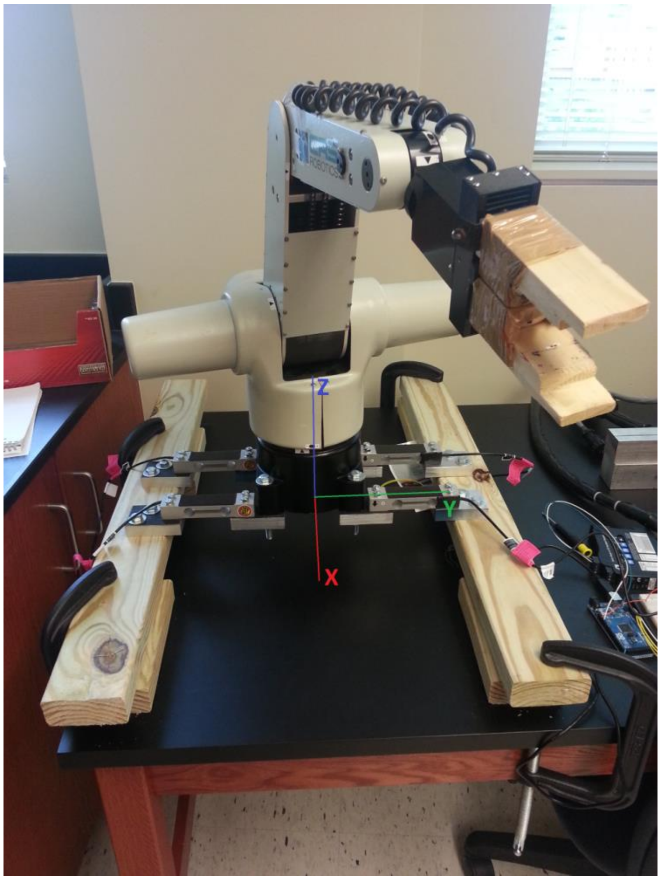

Figure 3.1: The five axis CRS A255 robot used for testing at Auburn.

Table 3.1 shows the values used for the model parameters. The mass of each link is estimated to be 3kg since the total mass of the manipulator is 17kg and the base is quite heavy. The links are each 0.254m long and fairly symmetric, so the center of mass is assumed to be almost halfway down the length of the links at 0.1m. To determine inertia, the links are assumed to be rods, such that the inertia about the y- and z-axes is defined by $I = \frac{1}{3} m * L^2$.

Table 3.1: Initial parameters used for CRS A255 model creation.

Parameter	**Link 1**	**Link 2**	**Units**
m_i	3	3	kg
c_{x_i}	0.1	0.1	m
c_{y_i}	0	0	m
c_{z_i}	0	0	m
I_{xx_i}	0	0	kg*m^2
I_{xy_i}	0	0	kg*m^2
I_{xz_i}	0	0	kg*m^2
I_{yy_i}	0.0645	0.0645	kg*m^2
I_{yz_i}	0	0	kg*m^2
I_{zz_i}	0.0645	0.0645	kg*m^2

To test the dynamic model, a simple motion is simulated. The first link is in a vertical position, and the second link is horizontal, 90 degrees from the first. The robot is then rotated around the first axis, which is about the vertical z-axis of the global coordinate frame. An important aspect of this model is that it requires the axis position, velocity, and acceleration measurements. Ultimately, these values will come from measurements made on the system, but for now they are simulated. A velocity limit is imposed at 1 rad/s, and the acceleration is limited to 0.5 rad/s^2.

The simulation results are shown in Figure 3.2. In Figure 3.2a, the axis positions, velocities, and accelerations are shown. The second and third axes are held constant at 90 degree angles, while the first axis rotates from 0 degrees to 90 degrees. The velocities and accelerations of the second two axes are at zero. The acceleration limit of 0.5 rad/s^2 is seen on the first axis, but the velocity limit of 1 rad/s is not quite reached before the rotation begins to slow.

Figure 3.2b shows the resulting reaction forces at the base. The force in the z-direction, not shown so the x- and y-direction forces can be seen, remains constant near 167N. This force is equivalent to the mass times the acceleration of gravity. As the robot arm swings around, forces are applied in the x- and y-directions. Because the simulation starts immediately at a positive acceleration and ends at a negative acceleration, as seen in Figure 3.2a, the reaction forces also start and end at positive values, rather than at zero. The jump from positive to negative acceleration is also seen in Figure 3.2b near 1.8 seconds.

In Figure 3.2c, the moment about the z-axis begins as a positive value, and then switches to a negative value. This switch makes sense because it reflects the change of the acceleration about that axis. Similar to the forces, and for the same reason of immediate positive acceleration and an ending negative acceleration, the moment about the x-axis does not start at zero and the moment about the y-axis does not end at zero. However, it does make sense that the moment about the x-axis would move away from zero, while the moment about the y-axis would move towards zero as the second link moves from being parallel to the x-axis to being parallel to the y-axis.

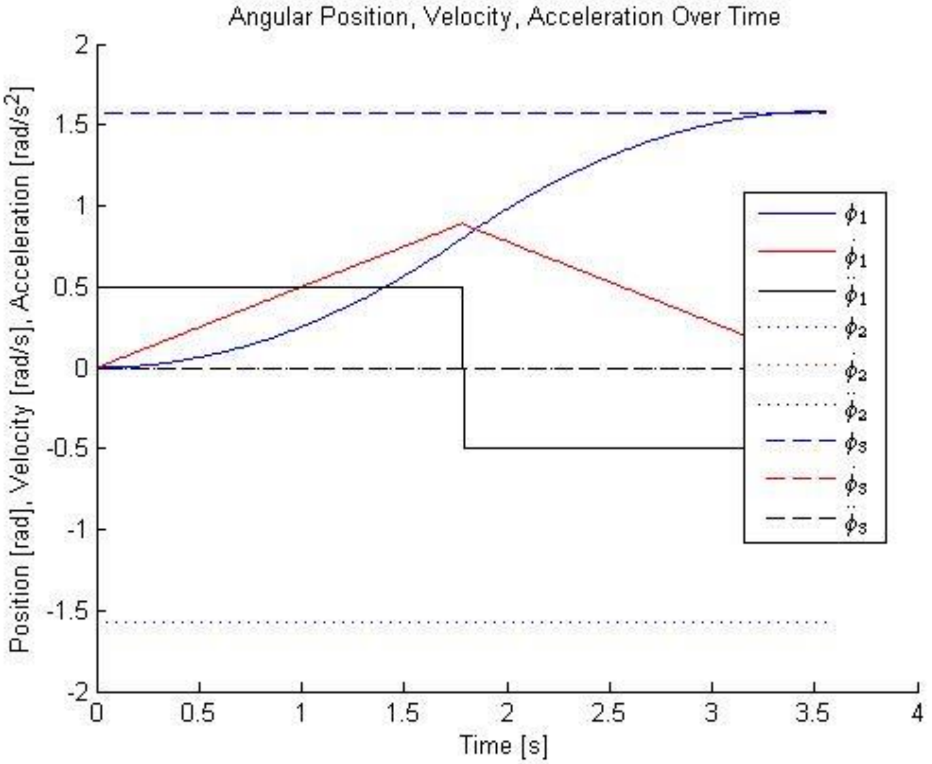

Figure 3.2a: Simulated joint angles of the Auburn robot model.

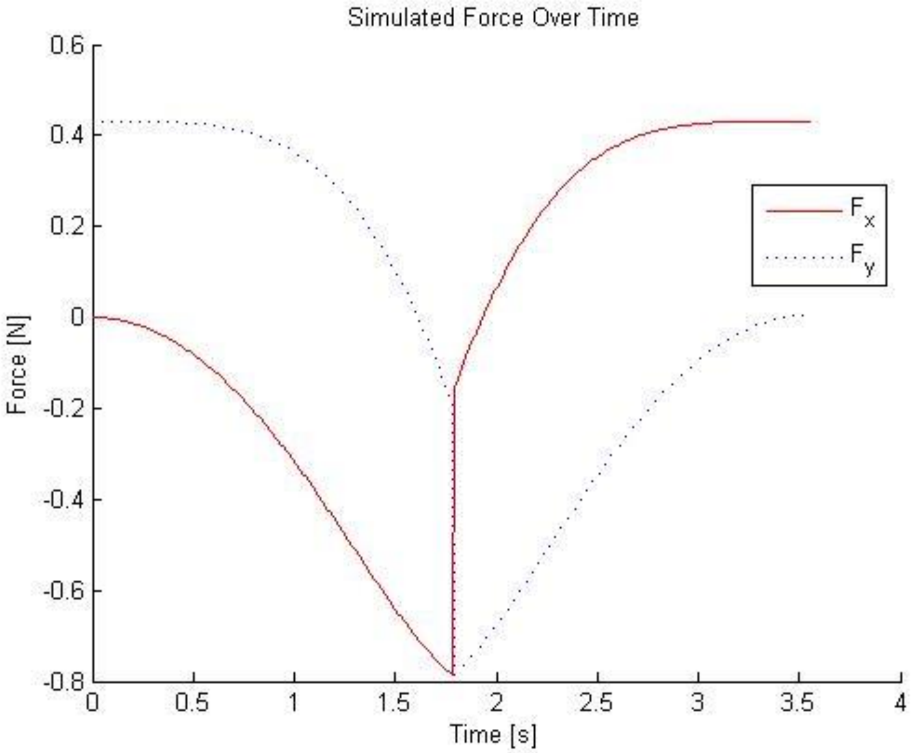

Figure 3.2b: Resulting base reaction forces due to the inputs of Figure 3.2a.

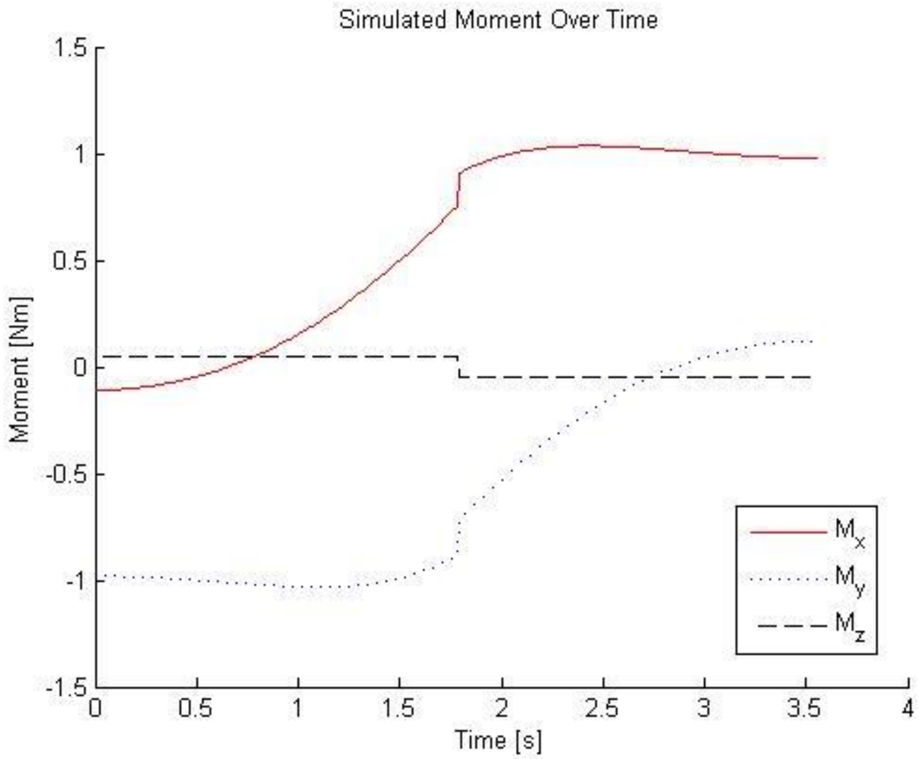

Figure 3.2c: Resulting base reaction moments due to the inputs of Figure 3.2a.

3.2 CRS A255 Equipment Setup

Below is a list of the equipment used for testing, evaluation, and data collection in Auburn:

- The robot used was a CRS A255, weighing 17kg. The robot has five axes: the waist (axis 1), shoulder (axis 2), elbow (axis 3), wrist pitch (axis 4), and wrist roll (axis 5). Two cables from the robot, one for feedback and one for power supply, are connected to the controller, a C500.
- The four load cells chosen to measure the base reactions of the robot were VPG-Celtron LPS Single Point load cells, with a load rating each of 20kg to ensure that any single load cell would not be overloaded by the weight of the robot. In theory, each load cell carries about one-quarter of the weight of the robot. The full scale output of the load cells is 2mV/V with a rated excitation of 10V DC.

- An Arduino Mega 2560 version 3 read the analog output from the load cells and converted the signals from analog to digital.
- The amplifier chosen was a DMD-4059 DC supply low voltage signal conditioner. It is powered with 12V DC from a power supply. The amplifier also supplies the 10V excitation voltage to the load cells.
- An analog 8 signal differential multiplexer was used to switch among the load cell signals.
- A QSB-S Quadrature to USB Adaptor was used to interpret the encoder measurements.

The two types of measurements required for collision detection and parameter estimation are the base reaction force and moment measurements through the use of beam load cells, and the joint angle information from the joint encoders. The encoders are installed by default by the manufacturer of the robot but must still be accessed. The methods used to access the encoders and install the load cells are described below.

In order to read the encoders, a direct connection to the wires carrying the encoder information is required. To access the internal wiring, a round 57 pin plastic connector, the same type on the feedback cable, was outfitted with pins and sockets in the appropriate male and female connectors. These exposed wires were then labeled and accessed according to the provided wiring documentation. Through the direct connection to the wiring, the encoders are read with the use of the QSB-S USB connectors, which are connected to a laptop on which the data is logged and saved.

The beam load cells are fastened to wooden “two-by-fours” for stability and the robot is placed on top of the load cells. Since the full scale output is 2mV/V and the required supply is 10V, the maximum signal from the load cells is only 20mV, meaning an amplifier is required to move the output range from 0-20mV to 0-5V. Besides conditioning and amplifying the signal, the amplifier also supplies the 10V to the load cells and is itself supplied with 12V from a power supply. Since there are four load cells and only one amplifier, a differential multiplexer switches between the four different signals. The multiplexer is controlled by the Arduino microcontroller. The Arduino receives the output from the amplifier and connects to a laptop, which records the output signals. A wiring diagram is shown in Figure 3.3 and a picture of the setup is shown in Figure 3.4.

Before the equipment was installed and connected, the load cell response was tested for linearity and hysteresis. Objects of known mass were placed one at a time on a sensor, and the resulting sensor output was recorded to check the linearity of the signal response to load. The masses were then incrementally removed in reverse order to check for hysteresis. A line of best fit was applied to the resulting data. Since the maximum rated load is 20kg, and the Arduino output ranges from 0 to 1022, it is expected that the line of best fit would be near 1022/20 = 51.1. The resulting slope of 52.3 is very close to this value, as seen in Figure 3.5.

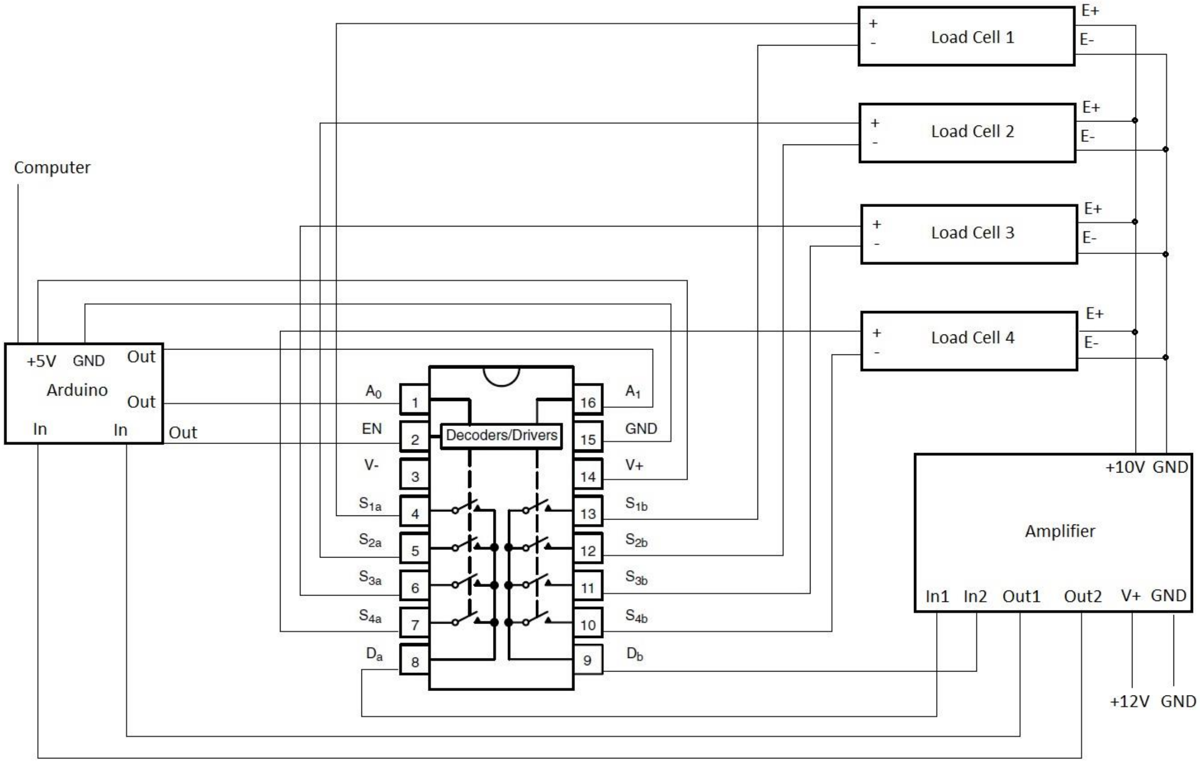

Figure 3.3: Wiring diagram for the CRS A255 measurement system setup.

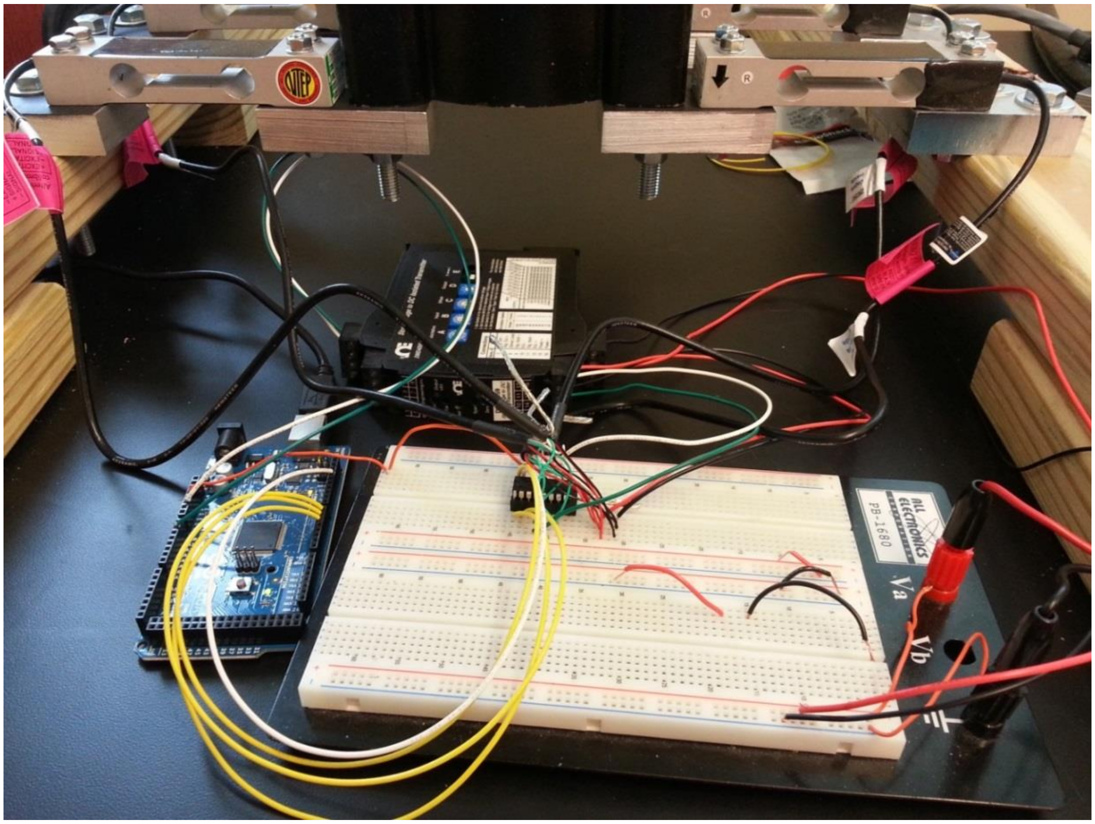

Figure 3.4: Wiring for the CRS A255 measurement system setup.

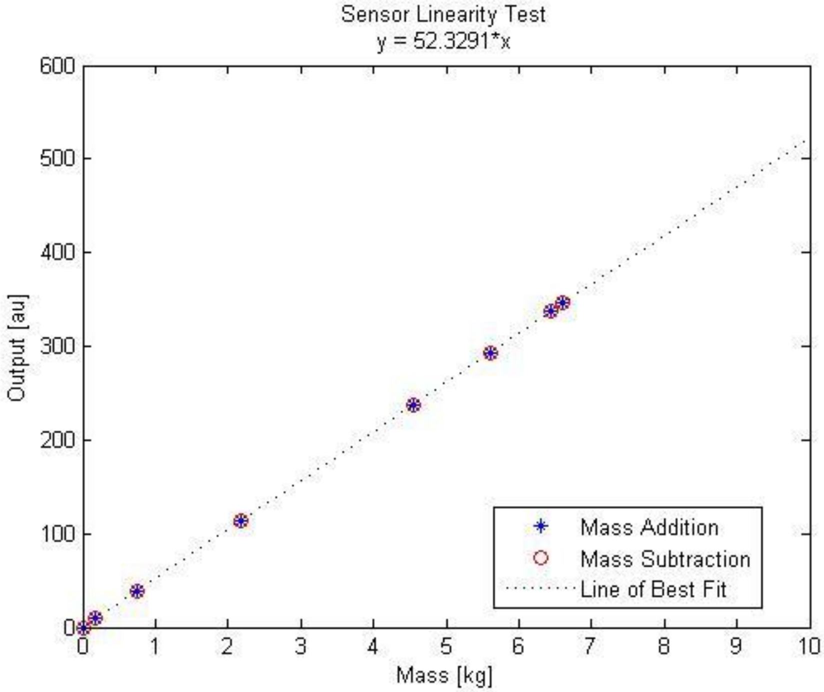

Figure 3.5: Beam load cell sensor linearity test.

One concern which needs to be addressed is the response time of the system. Following the instant a multiplexer selects the signal of one of the four sensors to send to the amplifier, and further to the Arduino, there is a period of time which the signal requires before reaching its full output value. The two components which must be considered for their time response characteristics are the multiplexer and the amplifier. The multiplexer has a rated time response of 160ns, and the amplifier has a rated response time of 70ms. Therefore, the limiting factor is the amplifier; the signal readings cannot be switched faster than 70ms, or the full signal values will not be obtained.

This 70ms value is still only an estimate. To get a better idea of the actual response time, a simple test is performed. The robot is mounted on the load cells, and this static position is recorded two different ways. First, the direct outputs of each of the sensors are individually recorded. Second, the outputs of the sensors are measured while switching between the signals with the multiplexer. By making several different recordings of the multiplexer outputs and varying the switch time between signals, the proper response time can be determined. Figure 3.6 shows the direct measurements compared with the multiplexer outputs. A multiplexer switch time of 0.5s is decidedly the most accurate, meaning it allows the signals enough response time to reach full output while switching between signals quickly in order to collect readings from all four sensors during robot motion.

During initial operation, it was observed that one of the load cell measurements was exceeding maximum output due to the load cell being bent into the plane of the other three load cells. Since three points create a flat plane, three load cells are all that is necessary to create a plane. If one load cell is not perfectly mounted in the plane of the other three, they are all forced to bend into the same plane, defined by the base of the robot, when bolted down. The bias of the applied “load” is a source of incorrect measurements. How much of a biased measurement results per the distance the load cells are deflected in order to be in the same plane is not known. To solve this problem a washer was inserted into the space between the base of the robot and the sensor, allowing the robot to rest evenly upon all four load cells.

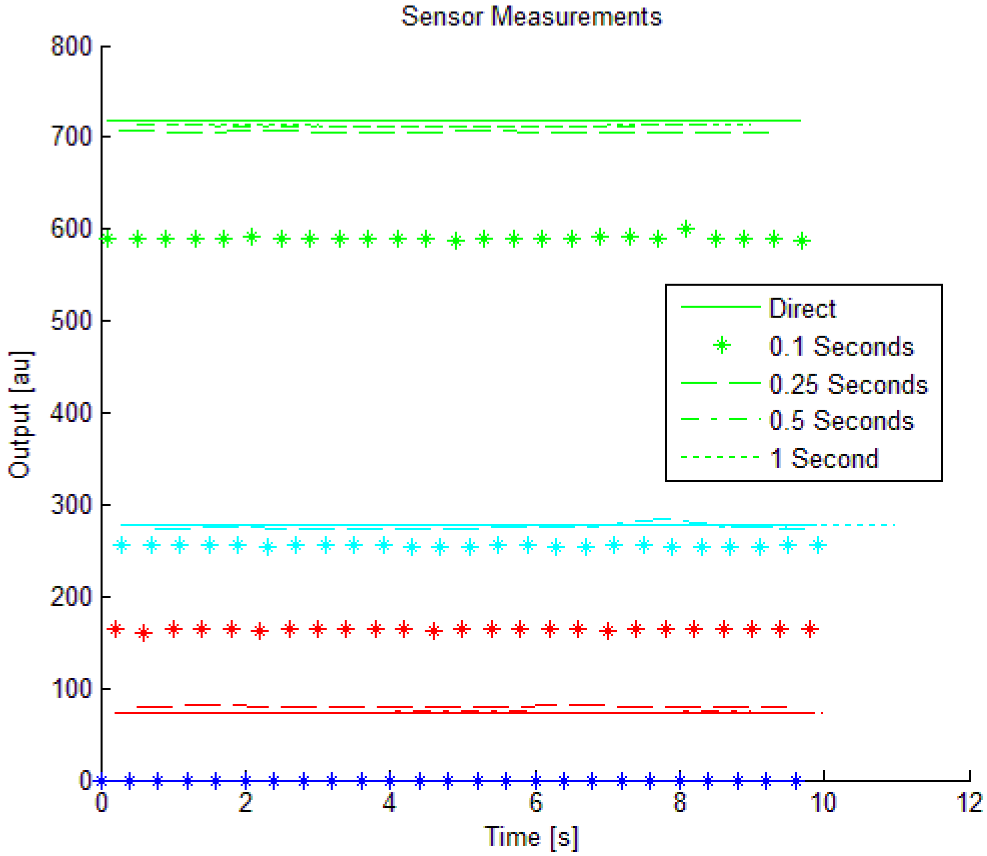

Figure 3.6: Response time tests for the CRS A255 setup.

3.3 CRS A255 Data Collection and Experimental Results

This section outlines the method used to make measurements on the CRS A255 and presents some measurement examples and analysis. End effector positions can be set by using the teach pendant to move the robot into a certain position and save that position. By setting two different positions, the robot can then be commanded to move between those two positions. The angle information is read with the use of the QSB-S Adaptor, and the base reaction forces are measured with the load cells. The collected encoder information is then used in the model to estimate the parameters.

Figure 3.7 shows the load cell sensor layout under the CRS A255. Based on this layout, the moments about the x- and y-axes can be described by the following equations, where d_1 is the y-distance from the origin to the sensors (0.0935m), d_2 is the x-distance from the origin to the sensors (0.0635m), g is gravity, and the sensor values s_n are provided in kilograms:

$$M_x = (s_1 + s_2 - s_3 - s_4) * d_1 * g \tag{3.1}$$
$$M_y = (s_1 + s_4 - s_2 - s_3) * d_2 * g \tag{3.2}$$

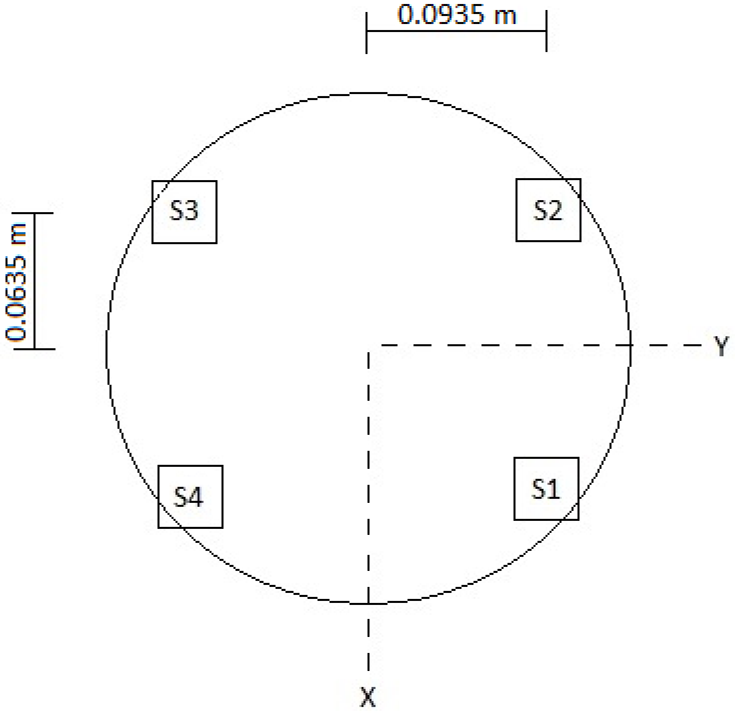

Figure 3.7: Load cell sensor layout under the CRS A255.

To test the load cell sensor layout's ability to detect disturbances, the CRS A255 was held in a static position while known masses were applied at a known location. The expected base reactions were then compared to the measured base reactions. Each new mass was added at ten second intervals, to allow the system measurements to settle. Figure 3.8 shows the reaction measurement results, and Table 3.2 summarizes the difference between the expected and measured reactions. Since the masses were applied along the global x-axis and therefore will not cause a significant change about the x-axis, the changes in the x-moment are not listed in the table. As can be seen, there is little difference between the predicted and observed reactions, confirming the model accuracy and the ability of the sensor setup to take accurate measurements.

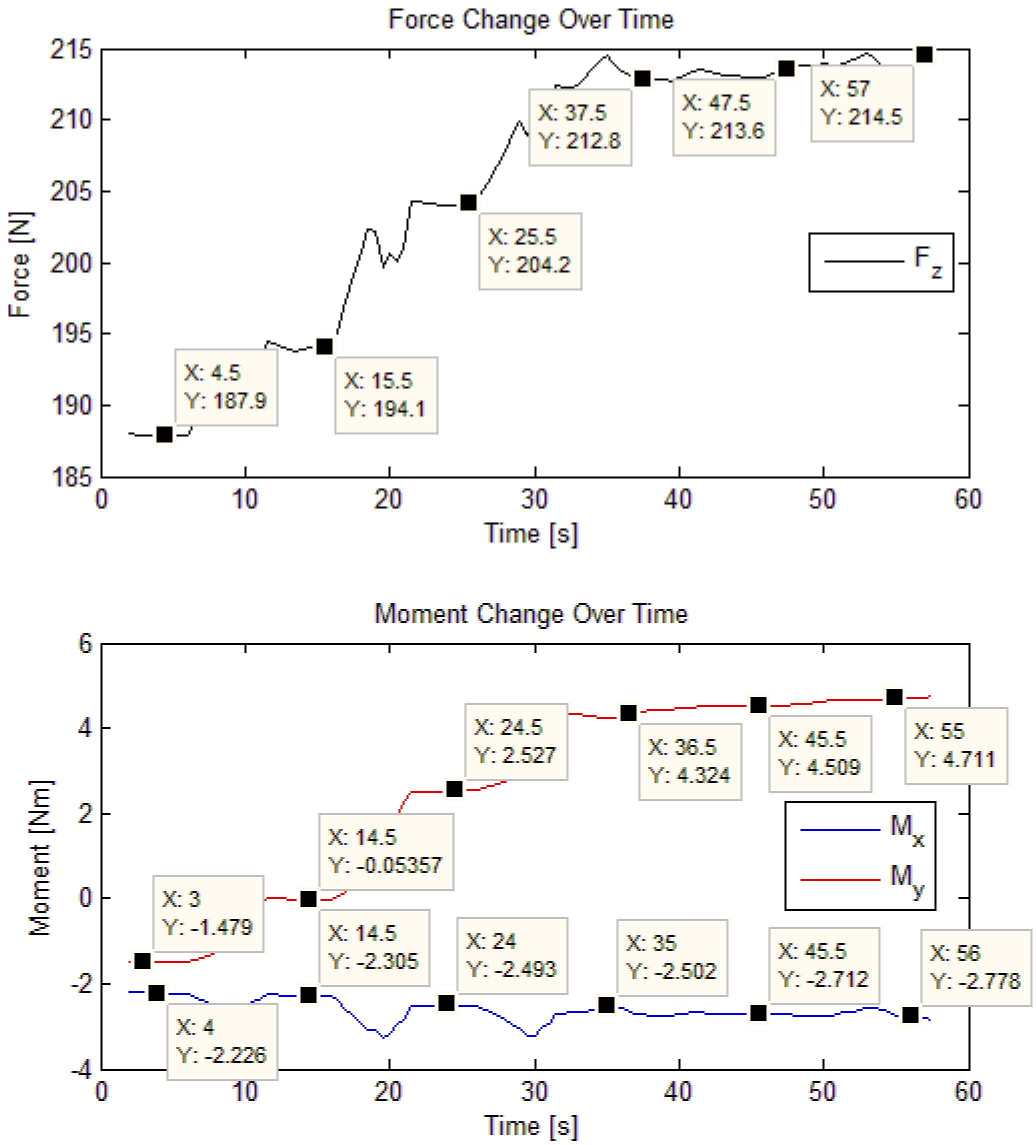

Figure 3.8: Base reactions due to mass addition at a known location.

Table 3.2: Predicted and observed base reactions for a static test.

Force [N]	**Z-Force [N]**			**Y-Moment [Nm]**		
	Predicted	**Observed**	**Diff. (%)**	**Predicted**	**Observed**	**Diff. (%)**
5.6	5.6	6.2	0.6 (10%)	1.4	1.4	0 (0%)
16.0	16.0	16.3	0.3 (2%)	4.1	4.0	-0.1 (3%)
24.2	24.2	24.9	0.7 (3%)	6.1	5.8	-0.3 (5%)
25.1	25.1	25.7	0.6 (2%)	6.4	6.0	-0.4 (7%)
25.9	25.9	26.6	0.7 (3%)	6.6	6.2	-0.4 (6%)

A sample run in which sensor data was collected is shown in Figure 3.9. Since the data collection rate is slow due to the multiplexer, the sensor measurements are not able to show any reactions caused by high dynamics of the manipulator. Each sensor is read at two-second intervals in order to capture the full output of the sensor and avoid the limitations of the amplifier time responses. In order to further limit high dynamics and improve measurement accuracy, the manipulator was run slowly at ten percent velocity.

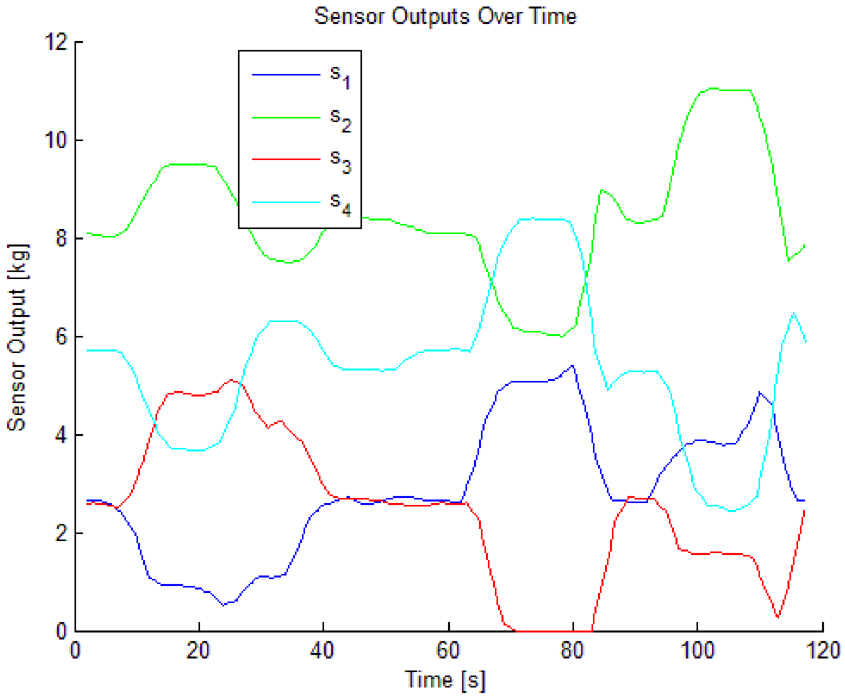

Figure 3.9: Sensor measurements logged during motion of the CRS A255.

The resulting base reaction force calculations are shown in Figure 3.10. Since the robot alone has a mass of 17kg and the attached cables add some extra mass, the expected mass is somewhere between 17kg and 20kg, which translates to an expected z-force between 167N and 196N. On average, the z-force is measured to be 187N, suggesting the measurements are accurate. The variation in the z-force measurement is from either the motion of the robot arm or from the position of the robot causing the cables to pull downward with more force.

Since the encoder measurements over the course of a run are accessed and recorded, as shown in Figure 3.11, the expected base reactions can be determined. To help create a more accurate model, two links were added than previously shown in Table 3.1. Using these new initial parameter estimates, shown in Table 3.3 in parentheses, along with the recorded angles, the expected base reactions are calculated by using the model. In Figure 3.12, a comparison of the measured reaction is shown with two different expected reaction measurements. First, the parameters are held constant over the course of the run. Second, the parameters are varied by being updated every quarter of a second.

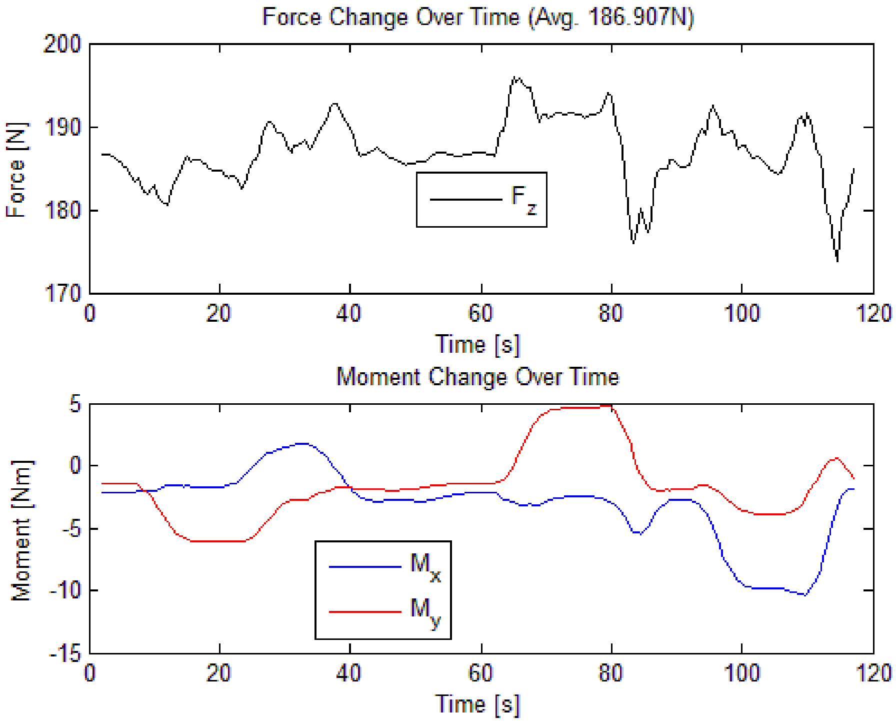

Figure 3.10: Base reaction force and moment measurements.

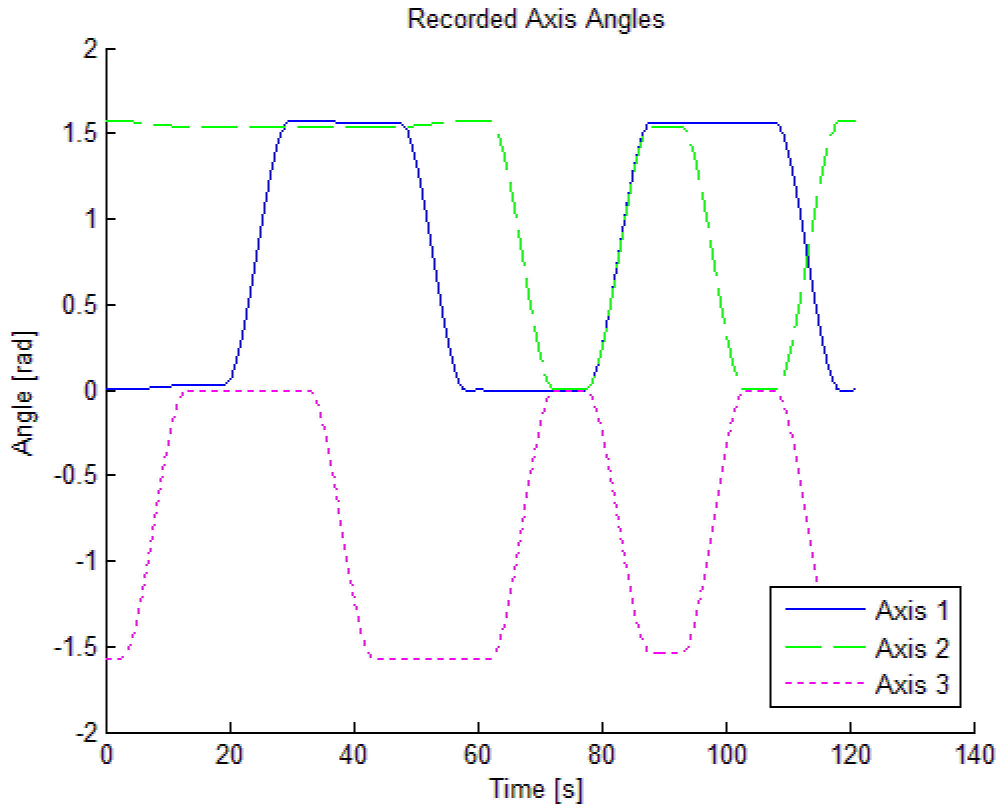

Figure 3.11: Measured axis angles from CRS A255 encoders.

Table 3.3: New CRS A255 parameter estimates. If there was a change after the estimate update, the values in parentheses show estimates before updating.

Parameter	Base	Link 1	Link 2	Link 3	Units
m_i	7	6	3	3	kg
c_{x_i}	-0.04	-0.05	0.06 (0.12)	0.07 (0.12)	m
c_{y_i}	-0.025	0.02 (0)	0.03 (0)	-0.03 (0)	m
c_{z_i}	0	0	0.04 (0)	0.05 (0)	m
I_{xx_i}	0	0	0	0	kg*m^2
I_{xy_i}	0	0	0	0	kg*m^2
I_{xz_i}	0	0	0	0	kg*m^2
I_{yy_i}	0	0	0.0805	0.0645	kg*m^2
I_{yz_i}	0	0	0	0	kg*m^2
I_{zz_i}	0	0	0.064	0.0645	kg*m^2

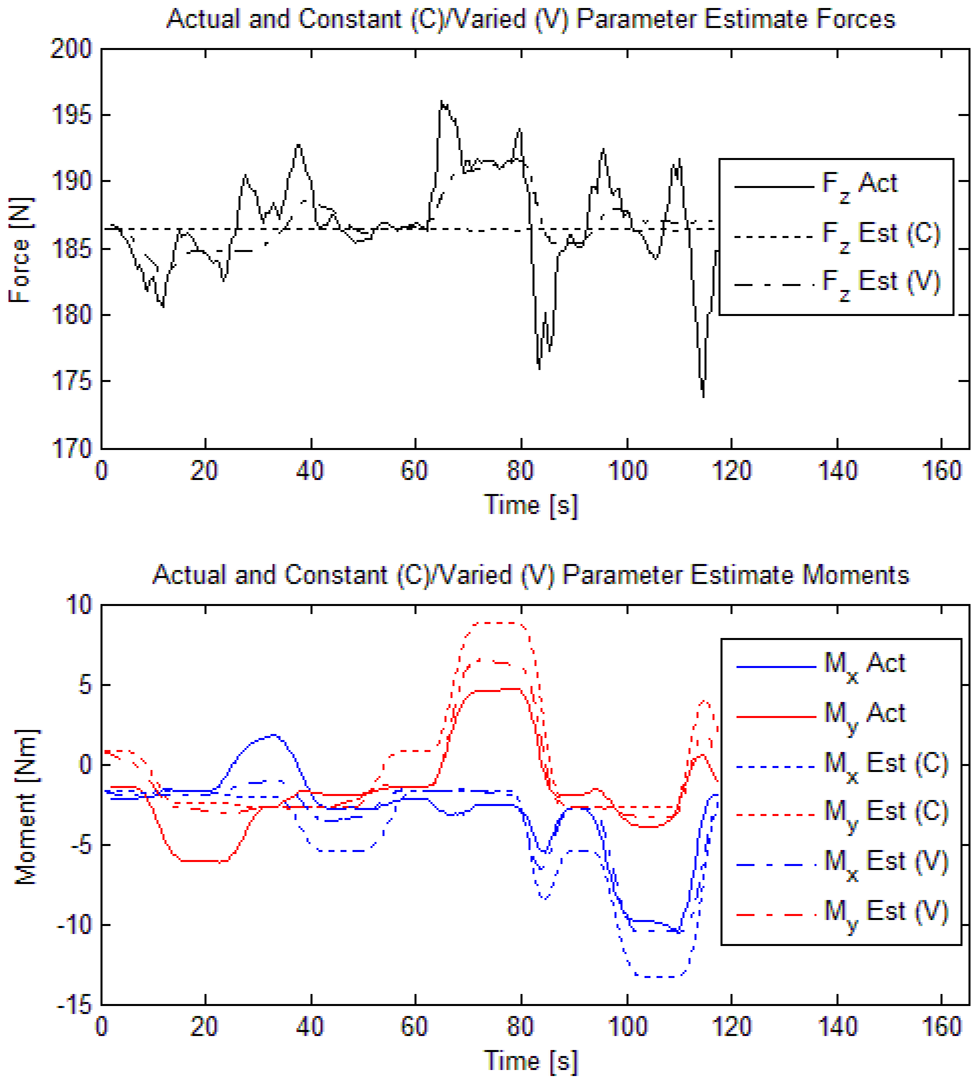

Figure 3.12: Expected base reactions with initial parameter estimates, held constant (C) and updated every quarter of a second (V), compared with the actual base reactions.

The new initial parameter estimates create far from accurate base reaction estimates. The varied parameters - those that are updated periodically over the course of the run - show a better ability to track the estimated reactions. Since the initial parameter estimates are not accurate, these updated estimates should be used to provide better accuracy in the base reaction estimates. Table 3.3 shows the updated parameter estimates, along with the previous initial estimates in parentheses.

Figure 3.13 shows the results of the more accurate, updated parameter estimates being applied to the model to determine the expected base reactions. On the moment graph in Figure 3.13, the expected reactions show a much better prediction than with the old parameter estimates. Not much difference is seen between the constant and varied parameter estimate reaction predictions on the moment graph, but a more significant difference on the force graph is apparent. These large variations in the z-direction force may be due to the measurement method. As the multiplexer switches among the four sensor signals every half second, the signal of one load cell is effectively held constant for two seconds until its signal is read again.

To further test the collision detection capabilities, the robot was run twice through a series of motions while making parameter estimate updates. On the second run, an object was placed in the path, which was pushed away by the motion of the robot. The resulting base reaction measurements and estimated reactions are shown in Figure 3.14. The arrows on the graph mark the two collisions. For this low-level collision, the parameter estimates are not accurate enough to detect the error. However, when comparing the second run to the first run, it is much more obvious there is a difference between the measurements at 8 and 39 seconds, and 25 and 58 seconds.

Overall the setup shows that decent parameter estimation is possible with the four beam load cell setup, and in this case, allows for around 2Nm of collision detection. The detection ability is based on the difference between the estimated reaction and the actual reaction. Since the method shows promise on the CRS A255, tests were further continued with the use of a much larger industrial robot at Siemens in Germany. The parameter estimation concept is implemented on the Siemens Artis Zeego system and presented in Chapter 4.

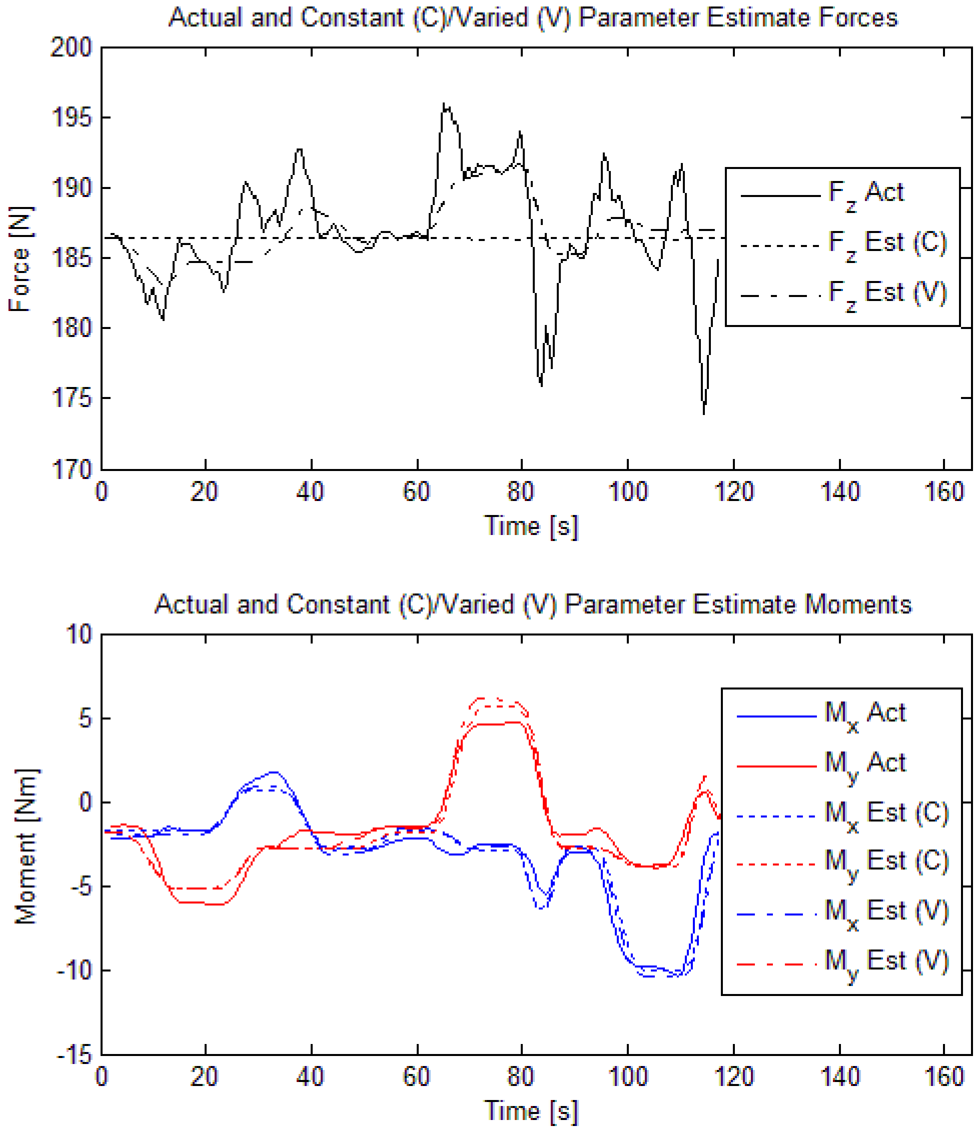

Figure 3.13: Expected base reactions with updated initial parameter estimates, held constant (C) and updated every quarter of a second (V), compared with the actual base reactions.

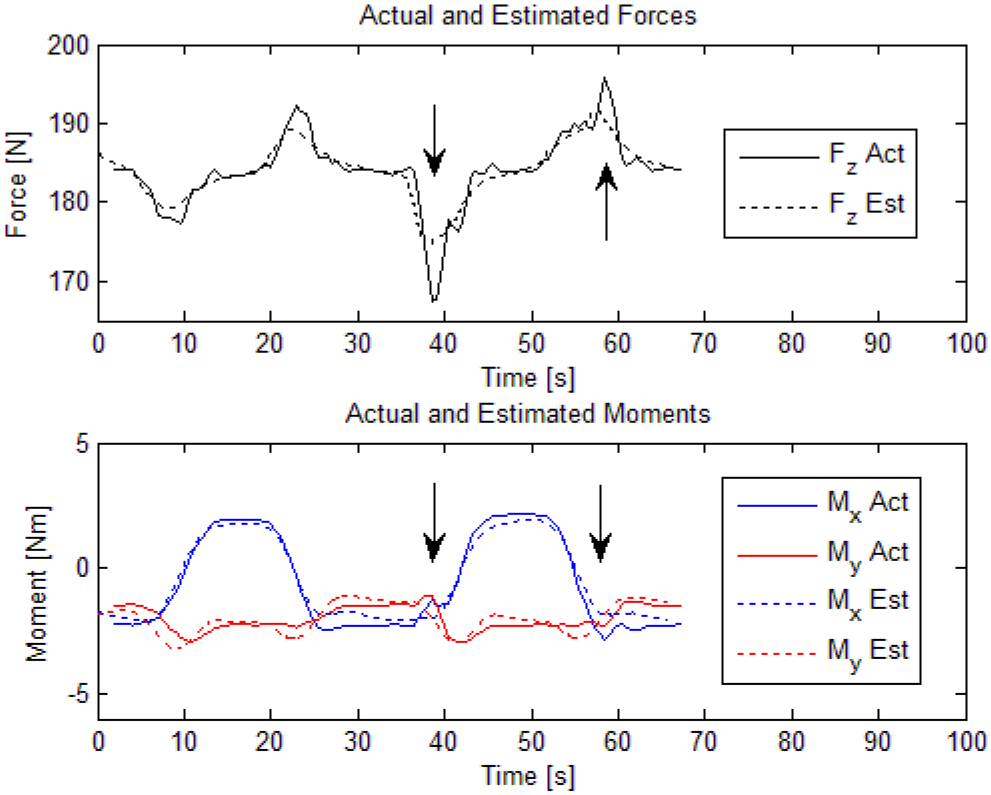

Figure 3.14: Collision detection test.

Chapter 4
Artis Zeego

This chapter covers the model creation, experimental setup, and data collection of the Artis Zeego robot. Section 4.1 presents the model creation and results, Section 4.2 explains the experimental setup, and Section 4.3 covers the data collection and analysis.

4.1 Artis Zeego Model Creation

An example of the Siemens Artis Zeego robot used in the medical industry can be seen in Figure 4.1. The six axes are labeled, as well as the global coordinate frame. This system can be simplified in the MATLAB model and defined as a four-link system. Joint 1 rotates about the global z-axis. Joints 2 and 3 rotate about their local y-axes. Joint 4 is more complicated, as it has three rotations; it rotates first about the x-axis, then the y-axis, then the x-axis again. An example of this modeled in MATLAB is shown in Figure 4.7.

An educated guess must be made in order to have initial parameter estimates. Better estimates will be made and added through the parameter estimation measurements. Table 4.1 shows the values used for the model parameters. The link masses are known. The second link is slightly to the side of the first link, so its center of mass is assumed to be a short distance in the negative y-direction. The fourth link, which is the C-arm and can be detached from the robot, has been measured to determine its parameters.

A simple motion similar to the run on the CRS A255 model can be run to test the validity of the Zeego model. All axes are held constant at angles shown in Figure 4.1 except the first axis, which rotates about the vertical z-axis of the global coordinate frame. Only the first three axes are shown in Figure 4.2a, since the remaining three are simply flat lines at 0 degrees.

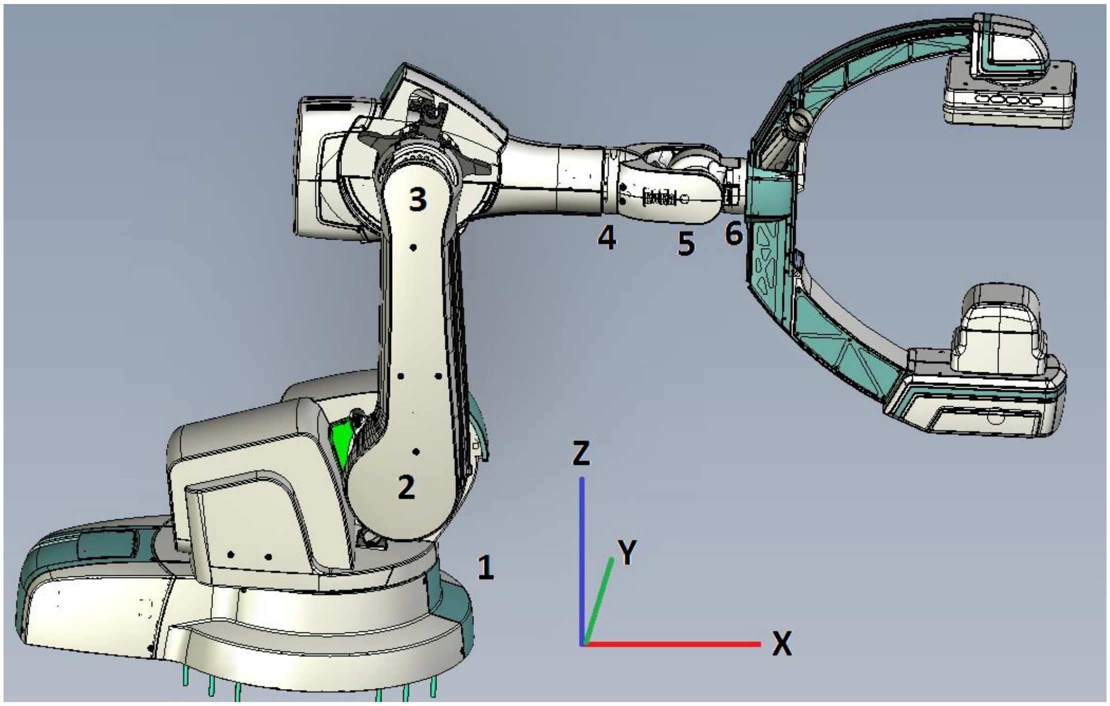

Figure 4.1: The six-axis Zeego medical robot produced by Siemens.

Table 4.1: Parameters used for the Zeego model creation.

Parameter	**Link 1**	**Link 2**	**Link 3**	**Link 4**	**Units**
m_i	635	240	275	187	kg
c_{x_i}	0	0.5	0.5	0.681	m
c_{y_i}	0	-0.3	0	0.254	m
c_{z_i}	0.3	0	0	0.018	m
I_{xx_i}	0	0	0	100	kg*m^2
I_{xy_i}	0	0	0	0	kg*m^2
I_{xz_i}	0	0	0	0	kg*m^2
I_{yy_i}	122.37	105.80	165.89	66	kg*m^2
I_{yz_i}	0	0	0	0	kg*m^2
I_{zz_i}	122.37	105.80	165.89	56	kg*m^2

The base reaction forces in Figure 4.2b show similar results as before, which is expected. The graphed x- and y-forces have the same shape as seen in Figure 3.2, although different magnitudes are seen due to the heavier system. Once again, the y-force does not start at zero and the x-force does not end at zero because the acceleration is never zero.

The z-force is shown as well, which appears to be near 13,500N. The total mass of the robot is near 1337kg, leading to a total z-force of 13606N.

Figure 4.2c shows that the x-moment is minimal when the links are along the x-axis and the y-moment is minimal when the links are along the y-axis, which is expected. The moment about the z-axis corresponds to the positive acceleration and negative acceleration of the robot turning about the z-axis.

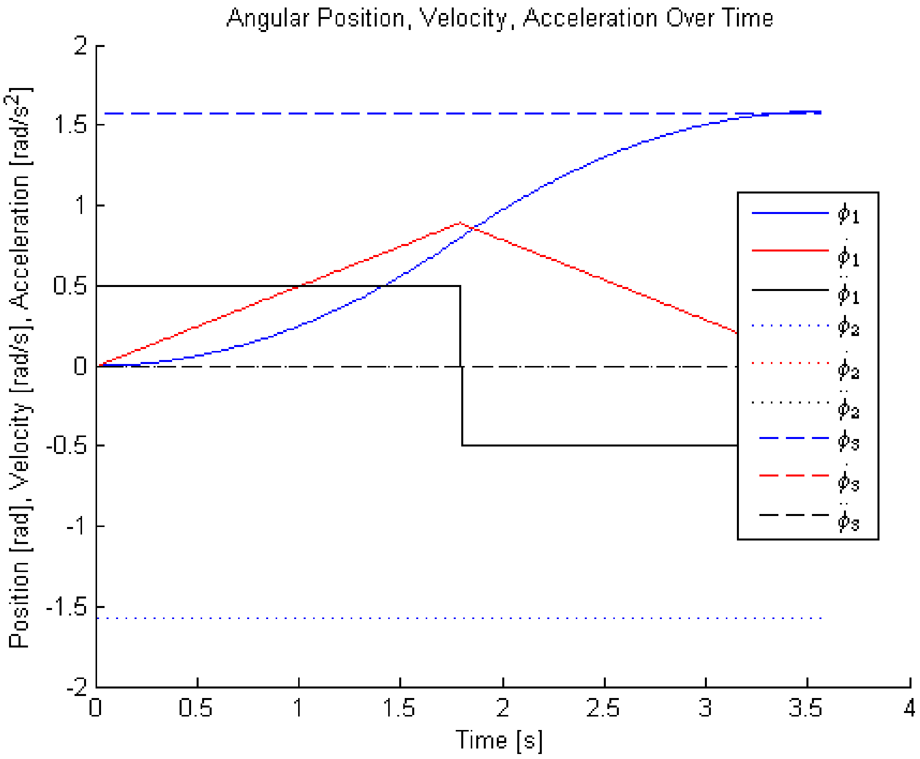

Figure 4.2a: Simulated joint angles of the Siemens robot model.

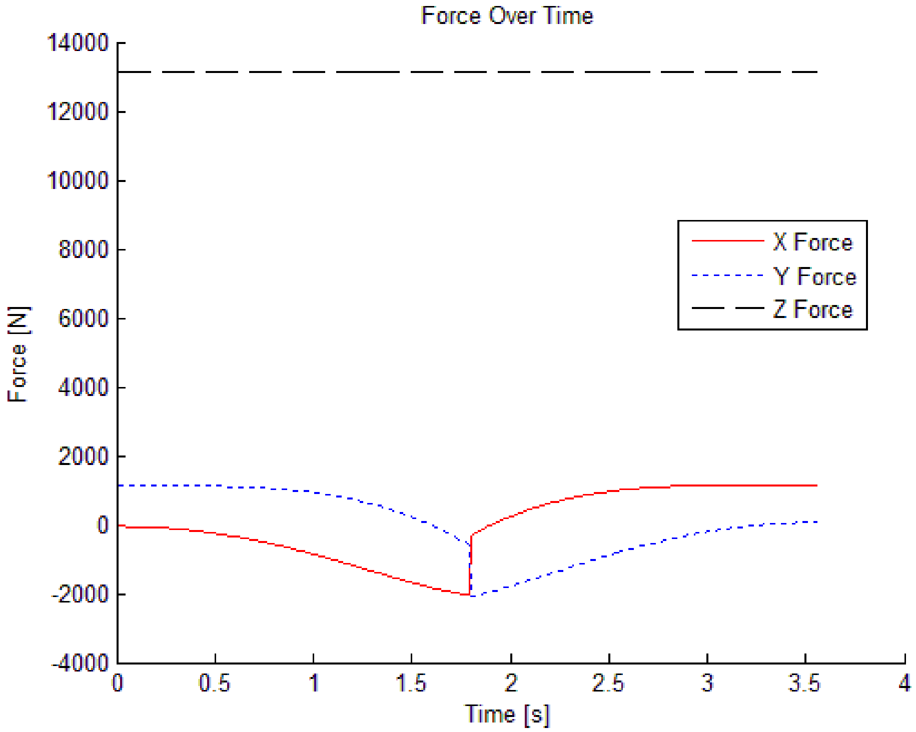

Figure 4.2b: Resulting base reaction forces due to the inputs of Figure 3.4a.

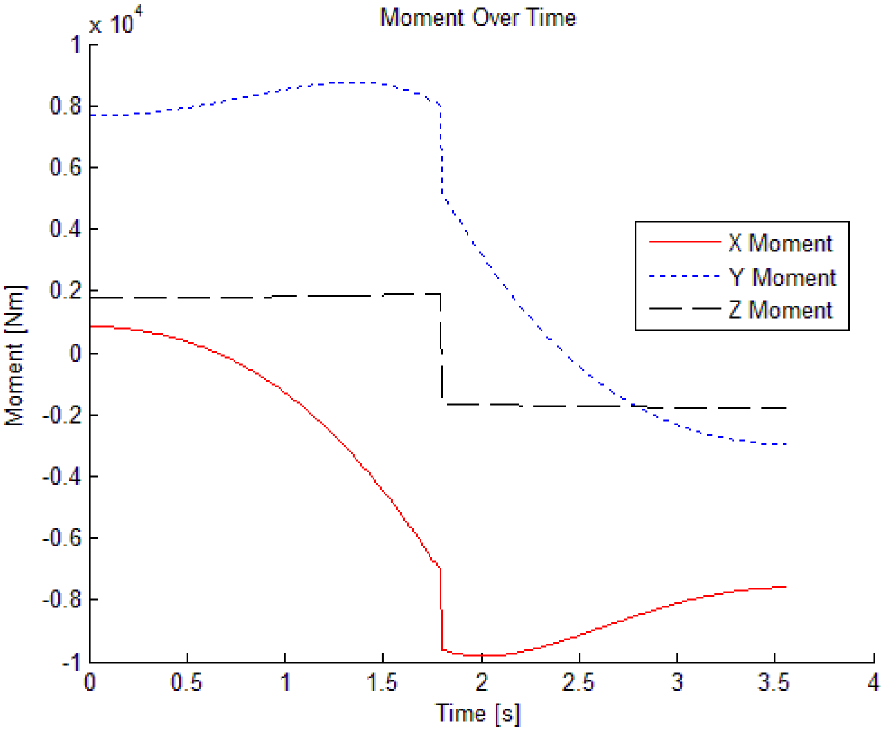

Figure 4.2c: Resulting base reaction moments due to the inputs of Figure 3.4a.

Lastly, it should be noted that the previous force and moment graphs show some forces and moments that cannot be detected through the method implemented in this thesis. The beam load cells installed underneath both the small CRS A255 manipulator and the large Artis Zeego robot can sense load applications only in the vertical direction. Therefore, only one force and two moments can be detected: the vertical z-force, and the moments about the x- and y-axes. Any forces applied parallel to and within the x-y plane cannot be measured or sensed. Conversely, forces that are applied parallel to the x-y plane but outside the plane can be detected, not due to the moment they cause about the z-axis, but due to the moment they cause about the x- and y-axes. Therefore, forces applied further away from the origin, both in vertical height and horizontal distance, are easier to detect because they cause a greater moment, since moment is a function of both force magnitude and distance from the axis.

Figure 4.3 represents this concept with three dimensional plots of sensitivity to applied unit forces. An area in front of the robot is represented, with a width of ten meters and a height of five meters, as well as a forward range from one to five meters. This range is well within the possible operation range of motion for the Siemens Zeego system in the operating room setting. The dark areas represent where the ability to detect collision to a particular force is poor, and the lighter areas represent where the ability to detect collision is greatest, relative to the other positions plotted in the graph. Points are plotted at half meter intervals in each direction. The plotted results are the expected base reactions calculated per unit of force compared to all the other points shown. Thus, a point of high sensitivity is a point at which, when a unit force is applied, produces a larger measurable base reaction than is produced at any other point. Conversely, a point of low sensitivity produces a smaller measureable base reaction compared to the other plotted points per applied unit force.

Figure 4.3a shows the ability to measure forces that are applied in either the x-direction or the y-direction. As mentioned previously, the sensors under the robot cannot directly measure x- and y-force applications, but they can measure the moments these forces cause. Since moment is a function of the distance a force is applied from the axis about which the moment is calculated, it would make sense that forces applied further away from the axis would be easier to measure, and this is seen in the figure.

Figure 4.3b shows the ability to measure forces that are applied in the z-direction. Due to the sensor setup, all z-direction forces are able to be measured. However, z-direction forces that are applied at a distance from but still parallel to the z-axis are easier to measure, since the force will cause a moment about either or both the x- and y-axes. As seen in the figure, the application of a z-direction force becomes easier to detect as its point of application moves further away from the z-axis.

Finally, Figure 4.3c shows the results of combining the previous sensitivity maps, and shows ultimately the areas where the ability to detect collision forces is best. These are areas where there is some distance from the x- and y-axes. In a related sense, this would also show where the C-arm should perform the majority of its motions, because if it happens to collide with anything in the environment in these areas, it will be easier to detect. In other words, Zeego operations performed at a 45 degree angle from the x- and y-axes allow the sensor setup to best detect collision.

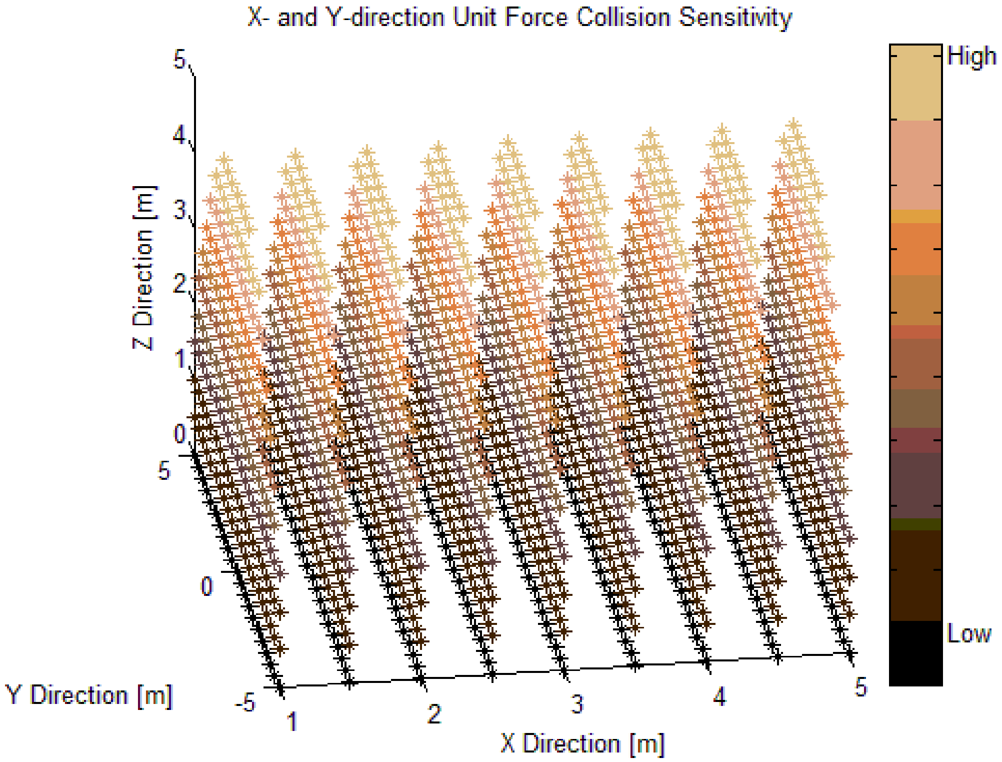

Figure 4.3a: Sensor sensitivity to unit forces applied in either the global x- or y-direction.

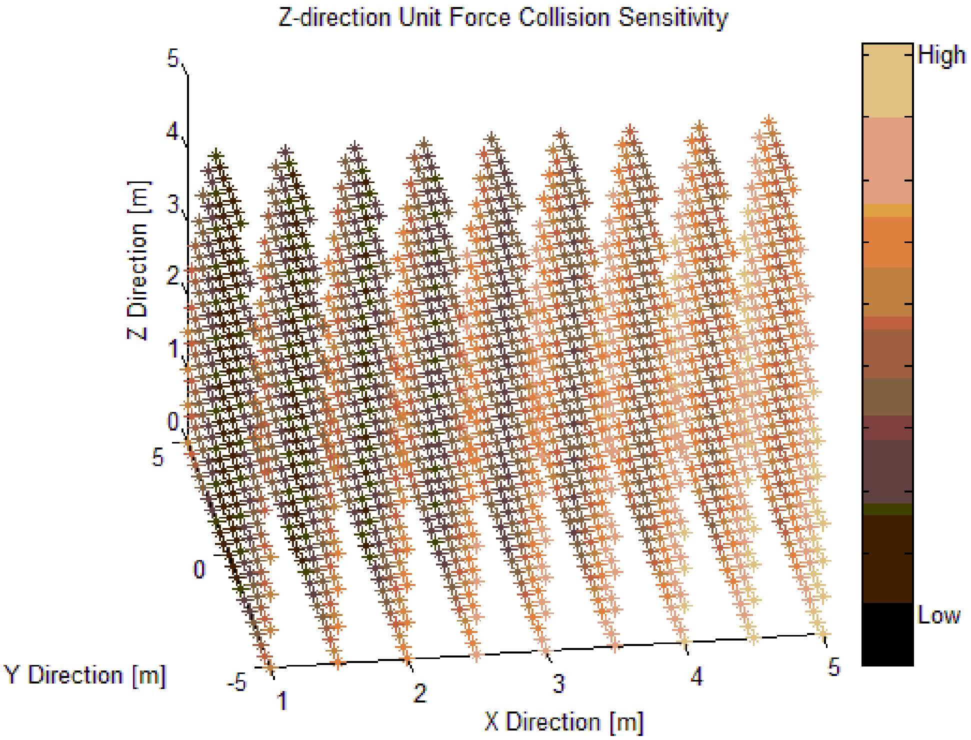

Figure 4.3b: Sensor sensitivity to unit forces applied in the global z-direction.

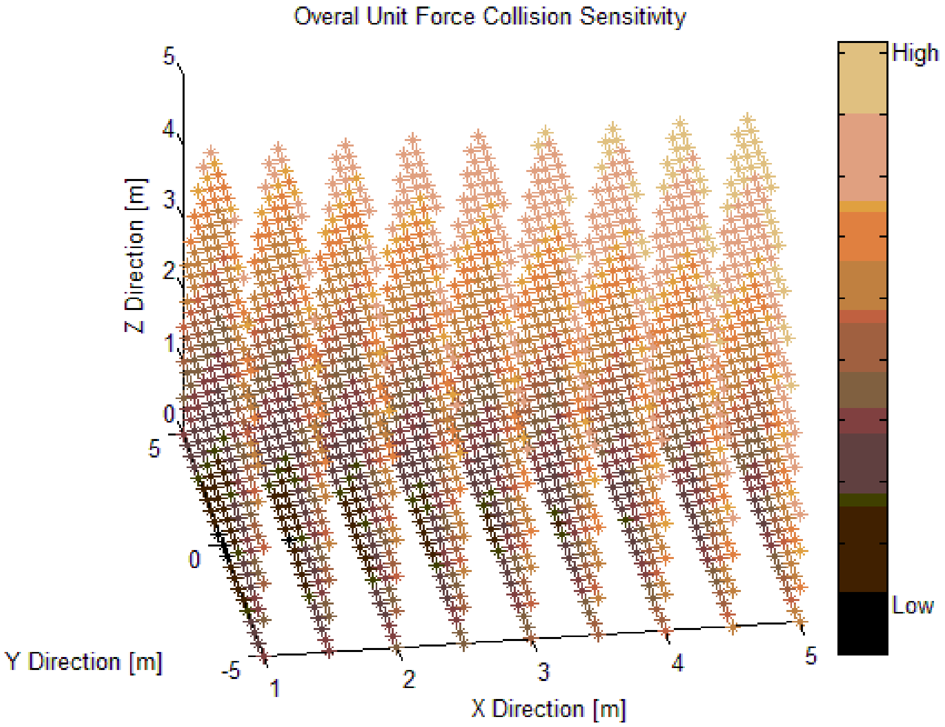

Figure 4.3c: Overall sensor sensitivity to unit force application.

4.2 Artis Zeego Equipment Setup

The following equipment was provided by Siemens AG for performing tests and collecting data:

- Four HBM HLC Legal for Trade Load Cells, nominal load 2.2t, accuracy class C3 were used for base reaction measurements.
- An industrial robot was provided of mass 1129kg. The robot has six axes: the waist (axis 1), shoulder (axis 2), elbow (axis 3), wrist roll (axis 4), wrist pitch (axis 5), and another wrist roll (axis 6).
- A C-arm of mass 187kg was attached at the end of the robot. A cable was attached to the C-arm, and runs along the robot links to the ground nearby. This prototype setup was very similar to, but not an exact representation of, the finished medical robot.
- A Smartpad was used to control the robot. Preprogrammed runs can be assigned, as well as manual manipulation, and axis angle information can be stored.
- The four load cells were connected to an HBM amplifier, which was connected to a laptop through an Ethernet cable. The laptop collected and stored the load cell measurements.
- The laptop contained software, QuantumX Assistant, which translated the load cell signals and logged the data measurements. The software also converted the signals into mass measurements.

The four load cells were placed under the industrial robot and fastened to a base plate fixed on the floor. A washer was placed between the load cell and a metal block adapter, which was fastened to the load cell with the robot fastened to the top of the block. The washer ensured that the load cell was measuring the point load, applied at the washer, and prevented the load from being spread along the load cell sensor through the surface contact between load cell and metal block. Figure 4.4 shows an image of the load cell attachment setup. The wiring from the load cells was connected to the amplifier, which in turn was connected to a laptop by an Ethernet cable. The angle measurements were made by the Smartpad and were stored to a USB stick.

Whether or not the same issues with load cell bias initially seen in the CRS robot setup were also present in this system is unknown. The load cells are assumed to be mounted on a much more even surface than was available for the small CRS robot. It is safe to assume that there is minimal bias, if any, since the mass measurements of the load cells is very close to the expected mass of the system. The exact mass of the system is unknown because the mass of the attached cable is unknown and changes depending on the robot position and how much of the cable is resting on the floor.

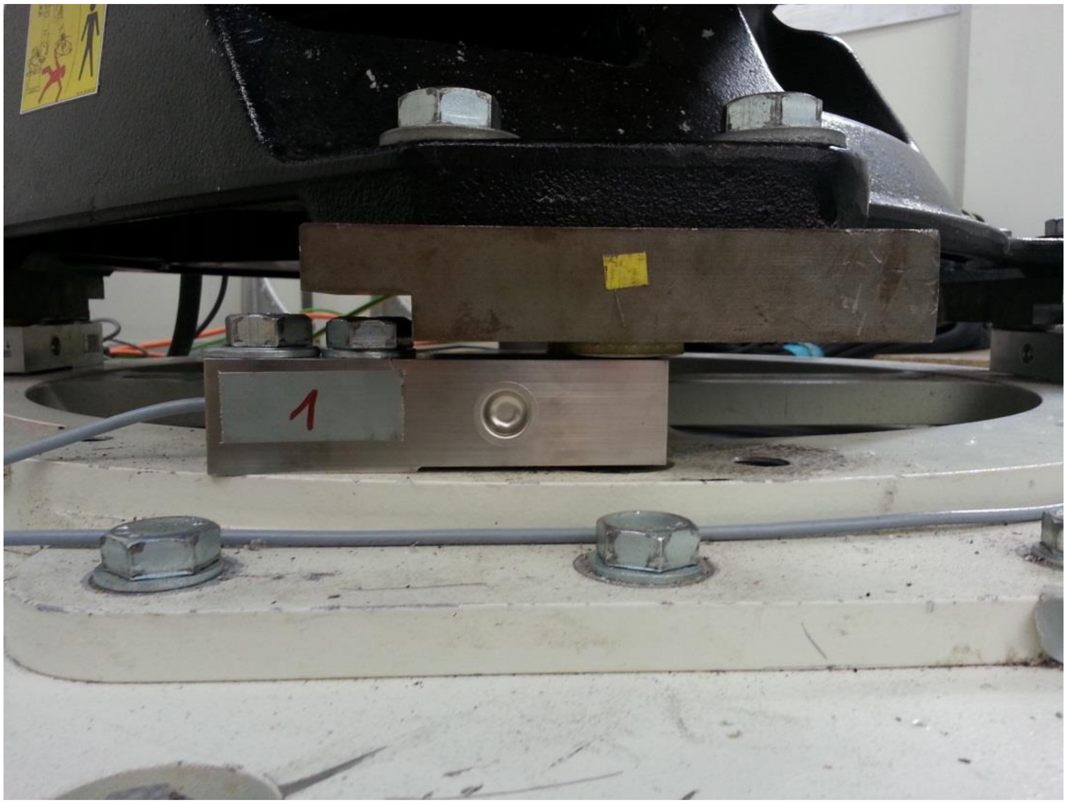

Figure 4.4: Zeego robot attachment to load cell sensor.

4.3 Artis Zeego Data Collection and Experimental Results

This section outlines the method used to make measurements on the industrial robot and presents some measurement examples and analysis. First, sensor data was collected with the robot in a static position. Then, known magnitudes of force were applied in known locations and in known directions to test force and moment sensing ability. These static tests are presented in Section 4.3.1. Next, both the axis angle data and the load cell measurements were captured as the robot executed some preprogrammed motions. The axis angle measurements are used in the MATLAB model to predict the reaction forces and moments, and the load cell measurements are used to calculate the actual reaction forces and moments. These measurements are presented in Section 4.3.2. The model and experimental result differences, similarities, and potential issues are discussed in Section 4.3.3. The parameter estimation results are presented in Section 4.3.4.

Collision detection using the base load cell method on the Zeego system was initially believed to be possible due to the test run results shown in Figure 4.5, which was one of the first indicators of the collision sensing abilities of this setup. The robot was in a position very similar to that shown in Figure 4.7, and the C-arm was rotated from a horizontal position to a vertical position, then back to a horizontal position. The rate of rotation was relatively slow; approximately 10 seconds were required to rotate 90 degrees. Figure 4.5 shows the output for one of the sensors during this motion. The large V-shaped dip represents the sensor measurement changing due to the rotation of the C-arm. The bottom point of the V is where the rotation was stopped and then reversed. Some points of interest on this graph are marked with arrows. The two outside arrows represent points where the C-arm, while in motion, was pushed slightly by hand. The middle arrow marks an impulse when the C-arm was bumped with a fist, which caused vibration with amplitudes larger than the standard noise and vibration seen throughout the overall measurement. The forces applied are unknown but obviously well within pain tolerance, which is promising and suggests that collision detection is indeed possible. These results led to the further studies into collision detection possibilities presented in this thesis.

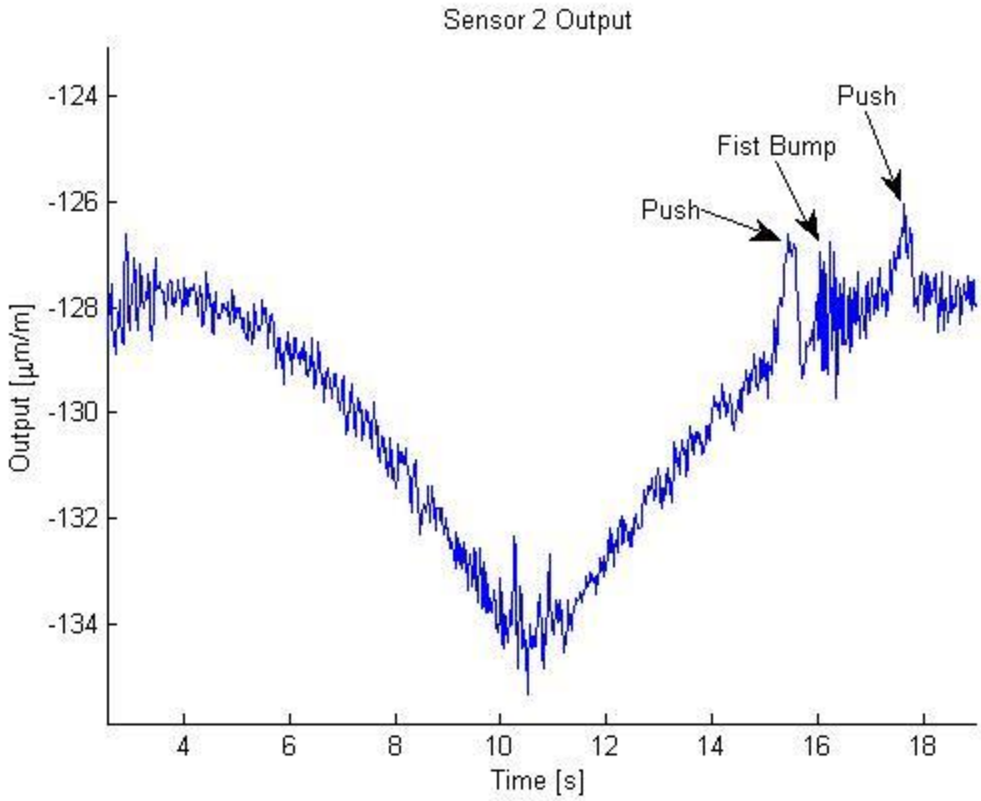

Figure 4.5: Output of a sensor during an experimental run in which the C-arm was hit by hand while in motion.

4.3.1 Static measurements

Figure 4.6 shows the sensor layout underneath the industrial robot with respect to the global axes. Based on this layout, the moments about the x- and y-axes can be described, respectively, as:

$$M_x = (s_1 + s_4 - s_2 - s_3) * d * g \tag{4.1}$$

$$M_y = (s_1 + s_2 - s_3 - s_4) * d * g \tag{4.2}$$

The sensor values s_n are provided in kilograms. The distance d is half the distance between two adjacent sensors and g is gravity. Figure 4.7 shows a model diagram of the robot position for the static tests performed in this section.

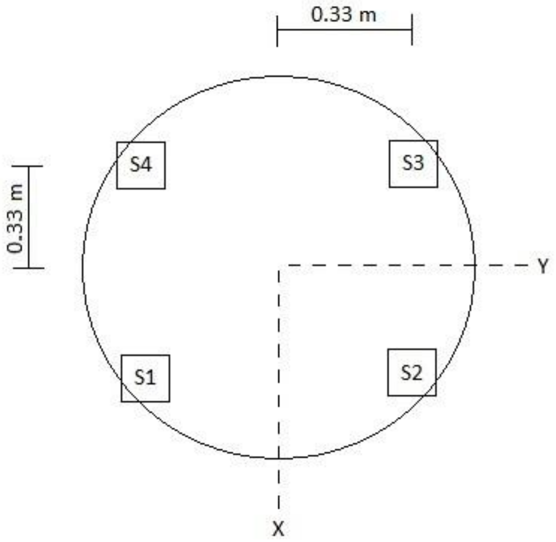

Figure 4.6: Sensor layout under the industrial robot.

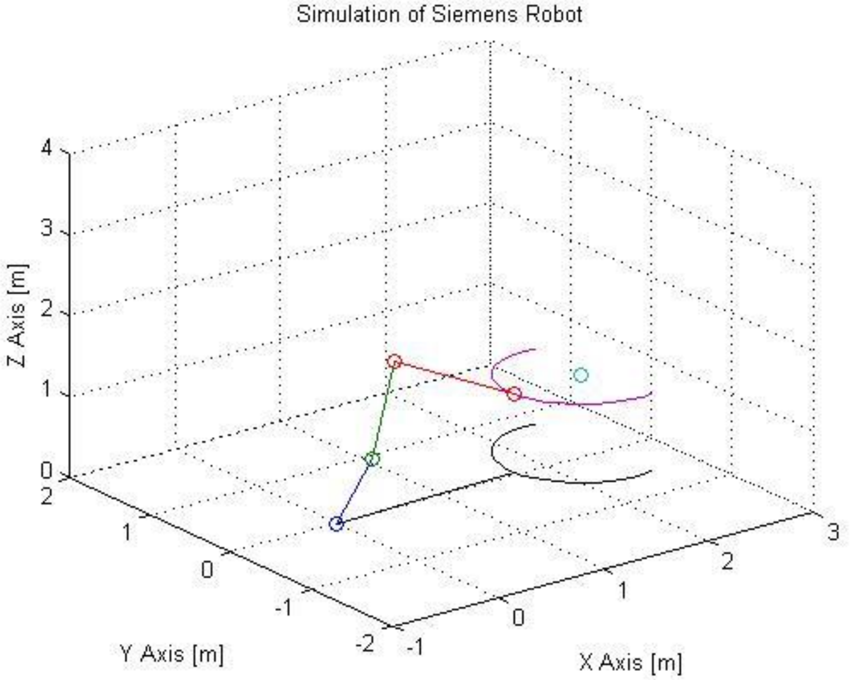

Figure 4.7: Position for static tests. This position also relates closely to the Figure 4.5 results.

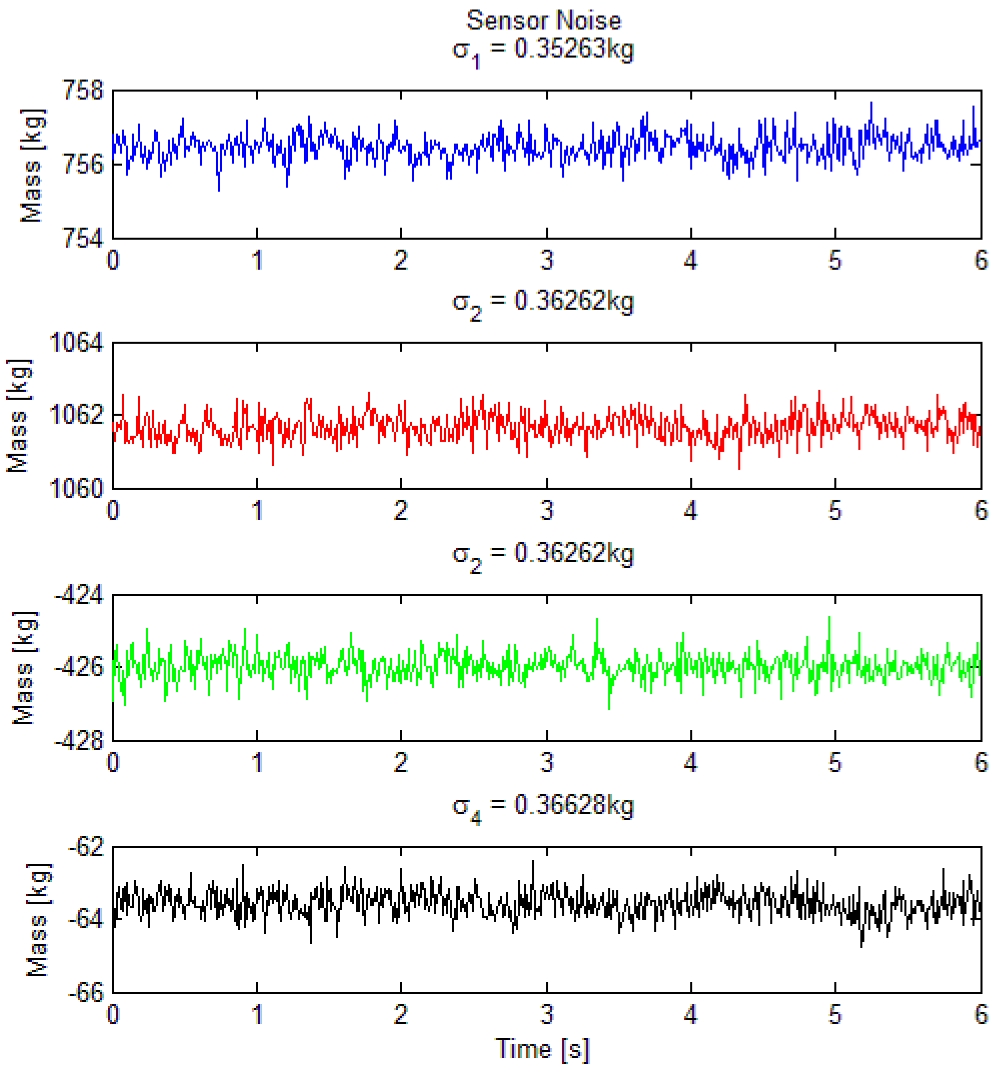

Figure 4.8: Load cell sensor mass measurements in a static position test.

The first factor affecting the ability to detect collision in the Siemens system is sensor noise. Changes in the sensor measurements can be masked by the random characteristics of the sensor noise. Figure 4.8 shows the sensor mass measurements of the robot in a static position. The load cell sensor noise is evident, with a standard deviation of around 0.36kg. Figure 4.9 shows a sample of how the noise affects the moment measurement in the same static position. The top plot in the figure clearly shows the noise in the measurement. The bottom plot shows that the noise has a Gaussian distribution, and the standard deviation in the static measurement is about 2.8Nm. Based on this information, it would probably be difficult to detect an impulse moment of less than 5Nm, although a constant disturbance moment applied over some period of time could possibly be detected. To prove this, another test is performed.

In order to test the ability of the sensors to detect external forces acting on the robot over a period of time, a known magnitude of force is applied at the fourth joint in a known direction, and the resulting sensor measurements are recorded and analyzed. The fourth joint is between the third link and the C-arm, and its location with respect to global coordinates is [1.732 0.055 0.966]m. The forces were applied at this location in the following manner: a 10N force downward in the negative z-direction, a 10N force to the side in the positive y-direction, a 20N force downward and to the side, a 30N force downward and to the side, and a 40N force downward and to the side. These values are within ±1.5N.

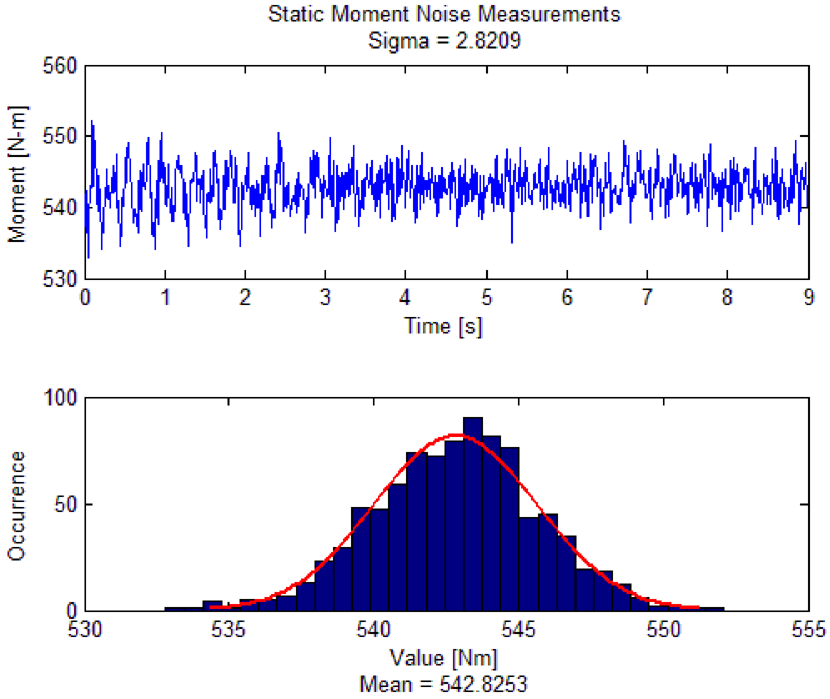

Figure 4.9: Analysis of the effect of sensor noise on moment measurements.

Theoretically, the forces applied downward should only cause a moment about the y-axis, and the forces applied in the horizontal y-direction should only cause a moment about the x-axis. Furthermore, since the links are extended along the x-axis, during the downward force application there should be very little moment measured about the x-axis, while a much larger moment should be measured about the y-axis. Figure 4.10 shows that this is indeed the case. The moment is large about the y-axis, near 7975Nm, while the moment about the x-axis is much smaller, near 542Nm. The increasing magnitude of applied forces is reflected in the increasing displacement of the different moment measurements. Another noticeable aspect of these graphs is the noise in the sensor measurements, previously discussed.

Now, the theoretical moments can be calculated to determine the accuracy of the moment measurements. If the distance to the joint is 1.732m in the x-direction, then the resulting moments about the y-axis due to the downward forces should be near 17.32Nm, 34.64Nm, 51.96Nm, and 69.28Nm. According to the graph of the y-moments, the measured moment applications are near 14Nm, 33Nm, 49Nm, and 62Nm. If the distance to the joint is 0.966m in the z-direction, the resulting moments about the x-axis due to the horizontal y-direction force should be near 9.66Nm, 19.32Nm, 28.98Nm, and 38.64Nm. According to the graph of the x-moment, the measured moment applications are near 12.0Nm, 23.0Nm, 34.8Nm, and 45.5Nm. These results are summarized in Table 4.2. The differences between the predicted and observed moments are due partially to the inaccuracy of the force applications, and perhaps due to inaccurate distance measurements.

The results are promising, showing that the ability to detect collision is sensitive down to 10Nm before sensor noise begins to hinder accurate detection. However, this is a static case; the next section presents the results of tests performed with the robot in motion. The high amplitude motions are expected to present more difficulties in collision detection ability.

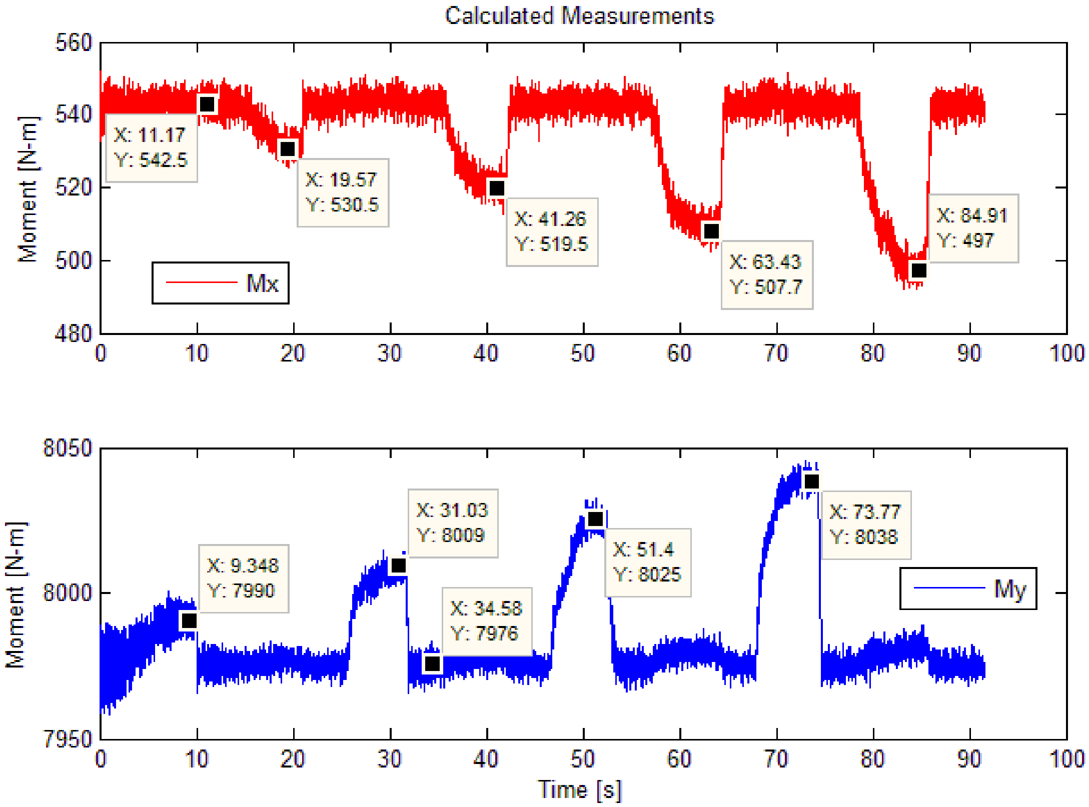

Figure 4.10: Moment measurements due to force application on the static system.

Table 4.2: Predicted and observed moment measurements due to force application.

Force [N]	**X-Moment [Nm]**			**Y-Moment [Nm]**		
	Predicted	**Observed**	**Difference (%)**	**Predicted**	**Observed**	**Difference (%)**
10	9.7	12.0	2.3 (19%)	17.3	14	3.3 (24%)
20	19.3	23.0	3.7 (16%)	34.6	33	1.6 (5%)
30	29.0	34.8	5.8 (17%)	52.0	49	3.0 (6%)
40	38.6	45.5	6.9 (15%)	69.3	62	7.3 (12%)

4.3.2 Dynamic Measurements

To study the collision detection limitations due to the robot being in motion, the robot was preprogrammed to run through some various motions. During the first 55 seconds, the C-arm rotated as it would normally around the patient table at three different approach angles and then moved through some other various motions with complex angles and linear motions. It should be noted that the motions beyond the initial 55 seconds are not typical in normal application and produce some extreme reactions and vibrations. As the robot moved through these motions, the angle encoder information was stored, as well as the load cell measurements. The six axis angles over time are shown in Figure 4.11, and the associated load cell measurements are shown in Figure 4.12. Figure 4.13 shows the moments calculated using Equations 4.1 and 4.2 and the load cell measurements.

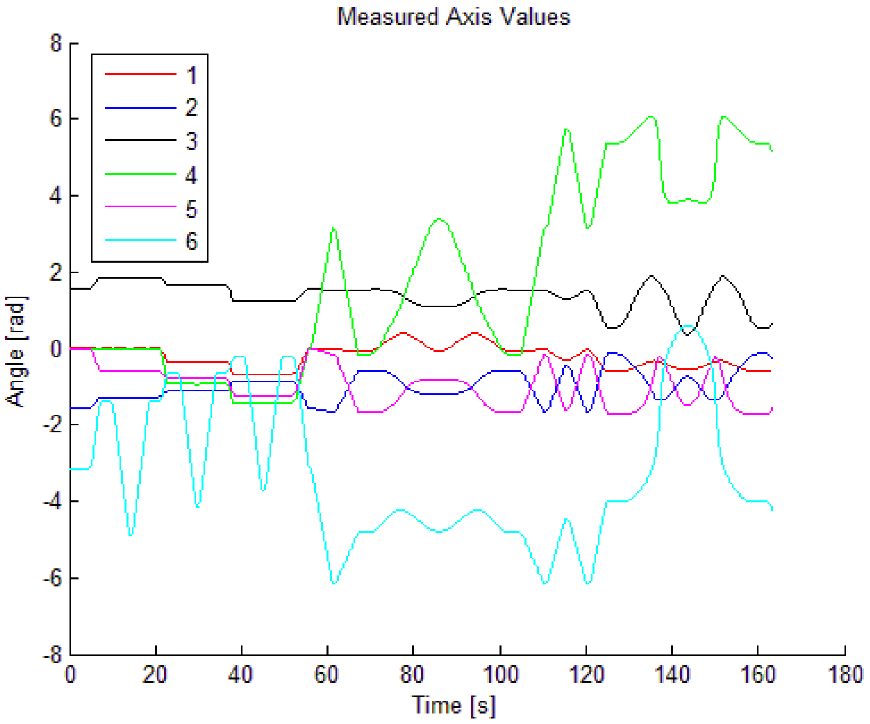

Figure 4.11: Recorded axis angle data for each of the six axes on the industrial robot.

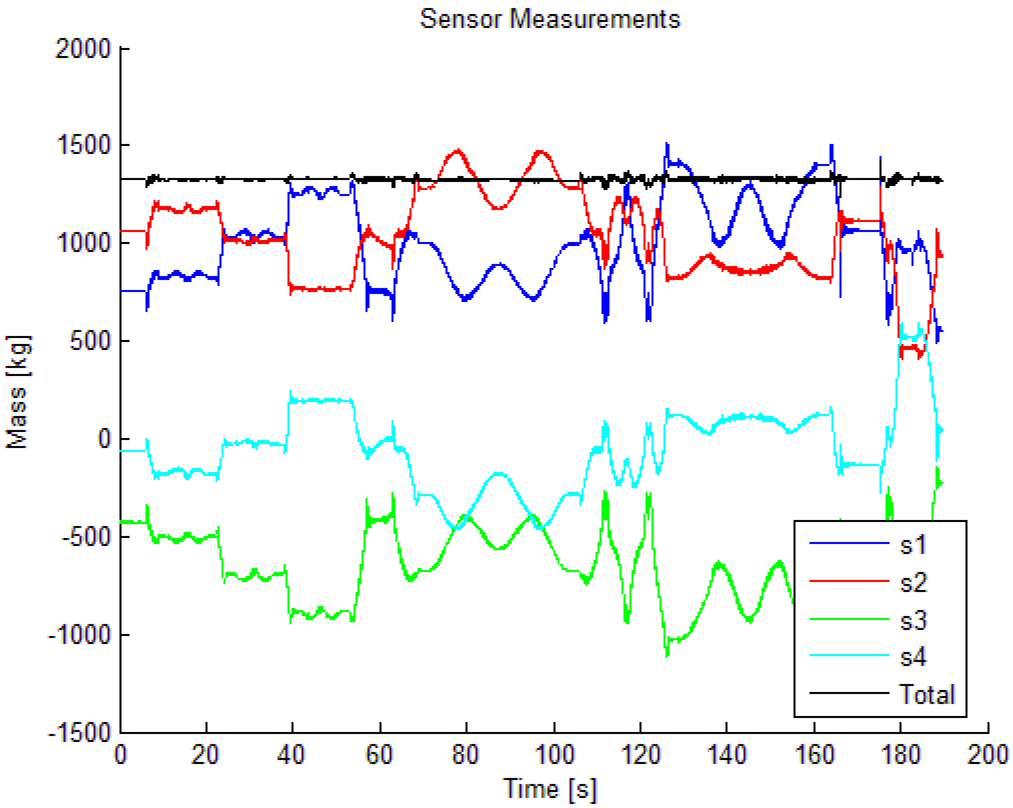

Figure 4.12: Recorded load cell data. The four sensor outputs and the sum are shown.

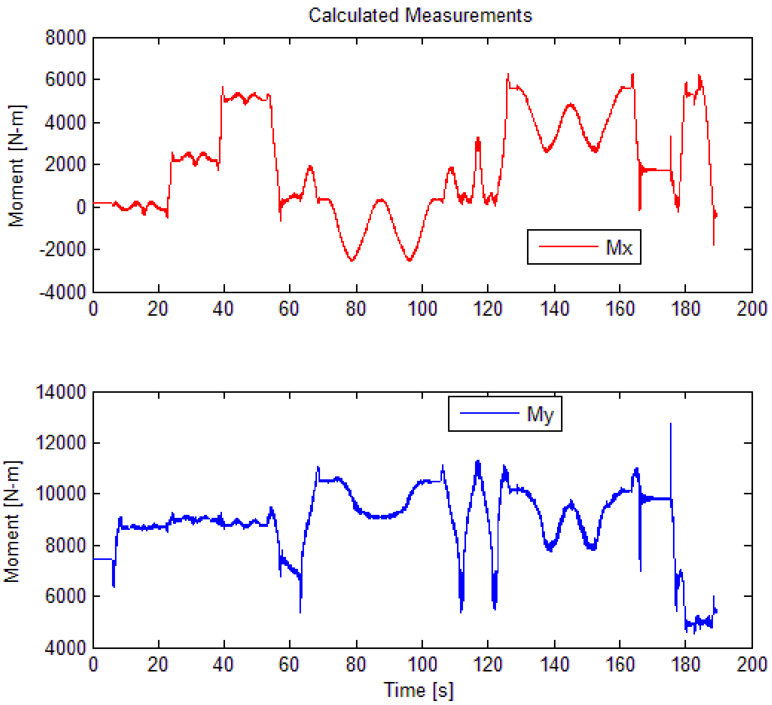

Figure 4.13: Calculated moment measurements based on the recorded load cell data.

The angle measurements from the robot motion are used in the MATLAB model to predict the reaction forces and moments at the base of the robot. The equations used in the model were discussed previously in Chapter 2. Figure 4.14 shows the expected base reaction results from using the angle measurements in the model, and these predicted measurements can be compared against the actual measurements, shown in the following section.

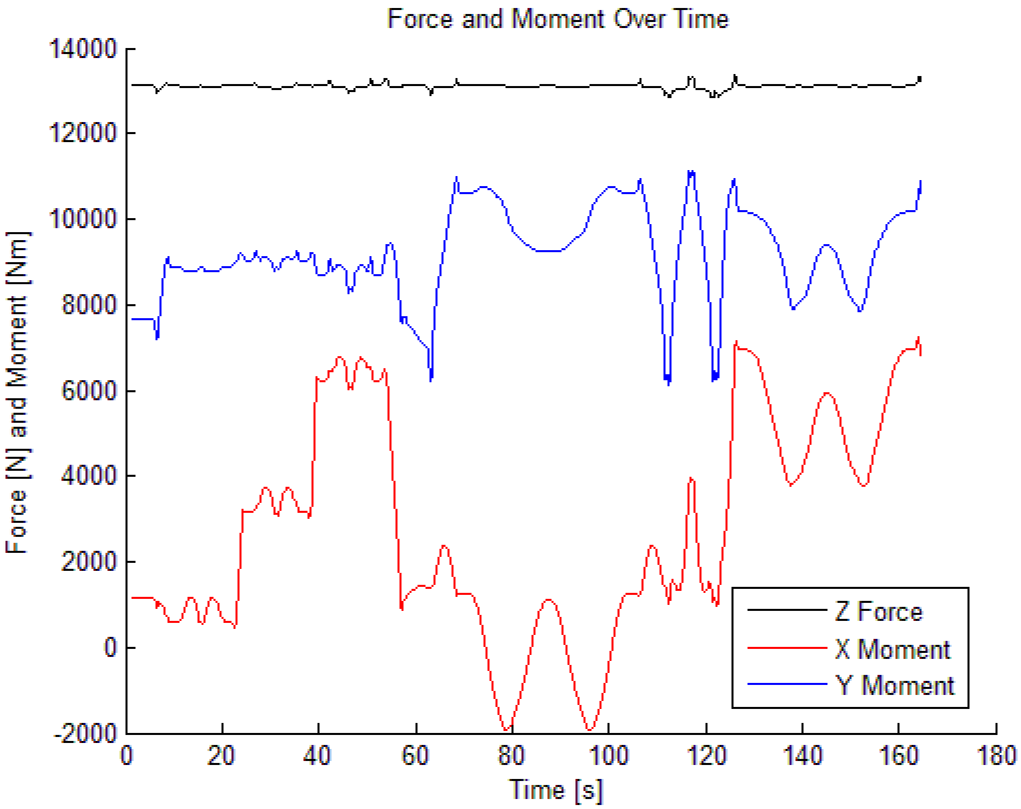

Figure 4.14: Predicted base reactions based on the axis angle data.

4.3.3 Result Comparison and Discussion

Now the expected base reaction and the actual base reaction can be compared. Figure 4.15 shows the results on the same plot, where the dotted lines represent the predicted measurements and the solid lines represent the actual measurements. Two major differences are immediately noticed in terms of how the plots match. First, the x-moment seems to be shifted vertically between the expected and predicted values, and second, all of the predicted measurements seem to be shifted to the left of the actual. The second difference is due to the time difference between when the data collection measurements were started, and can easily be corrected by adding in a time shift. Figure 4.16 shows the same measurements, but with a time shift correction of 1.54 seconds. Note that this is not live feedback but simply a factor of the data collection process used, where the angle data and sensor data were collected separately. In practice, there would be no delay between the axis measurements and the sensor measurements, as these two would be collected simultaneously.

The difference in the vertical position between the two plots is most likely due to inaccurate estimates of the parameter values. Recall that very few of the parameters are known initially. The center of mass locations were guessed, and even the C-arm parameters, which have been measured, may be inaccurate because some parts of the C-arm had been removed before the tests were run. With better parameter estimates, the differences between expected and measured results should be minimized.

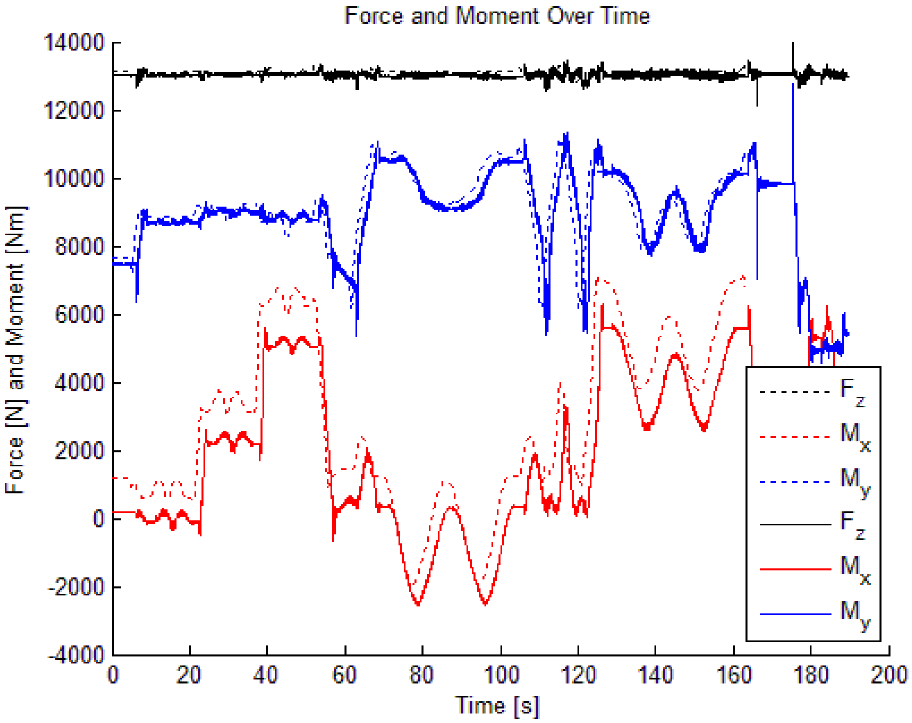

Figure 4.15: Comparison of predicted (dotted) and measured (solid) base reactions.

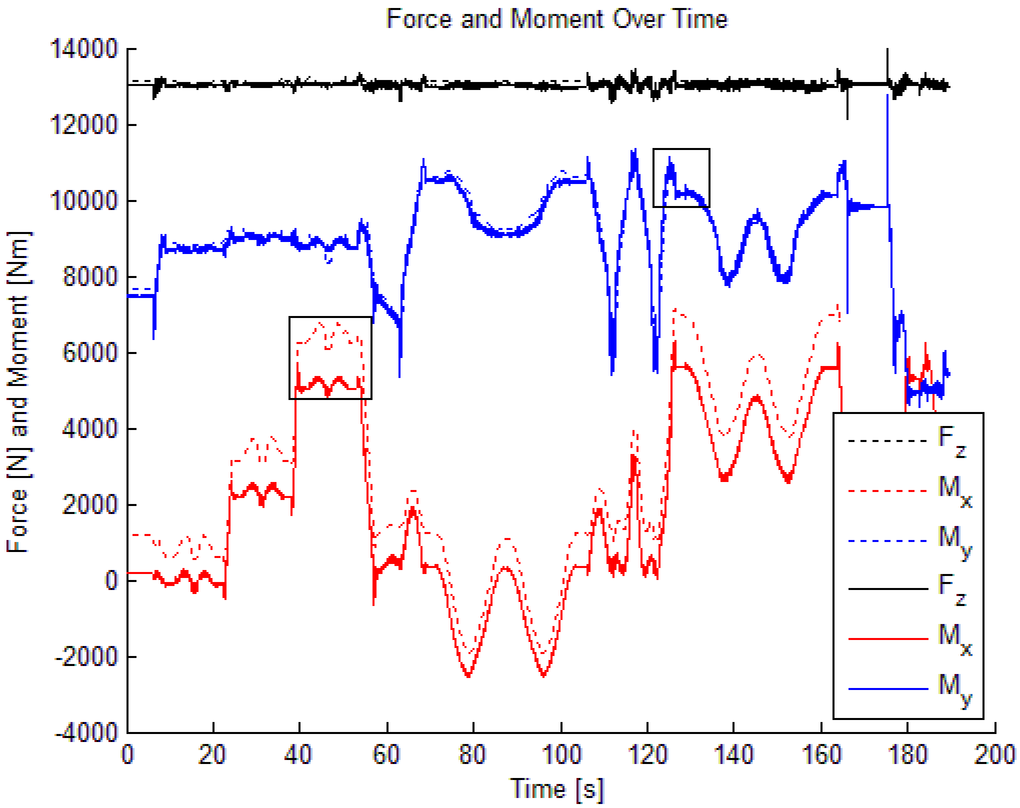

Figure 4.16: Time-shift corrected comparison of measured (solid) and predicted (dashed) base reactions.

Two portions of Figure 4.16 marked by boxes are shown in Figures 4.17 and 4.18 for a closer look at the similarities and differences between the results. Although the shapes are similar, the vibrations in the system become more apparent, especially in Figure 4.18. Recall that this figure, however, is not representative of typical motions, which are seen in the first 55 seconds. A normal C-arm rotation measurement is shown in Figure 4.19 along with a closer view, to better see a typical motion measurement with vibration.

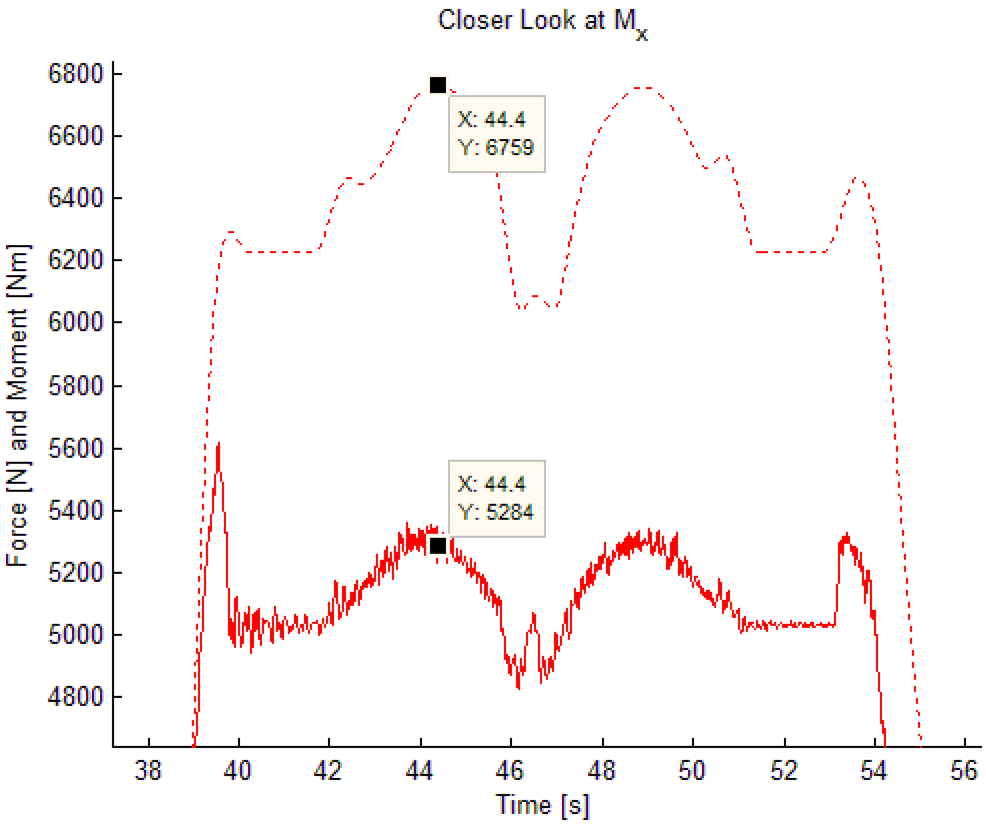

Figure 4.17: A closer look at the x-moment difference between measured (solid line) and predicted (dashed line) base reactions from Figure 4.16. The difference is around 1,475Nm, or 28%.

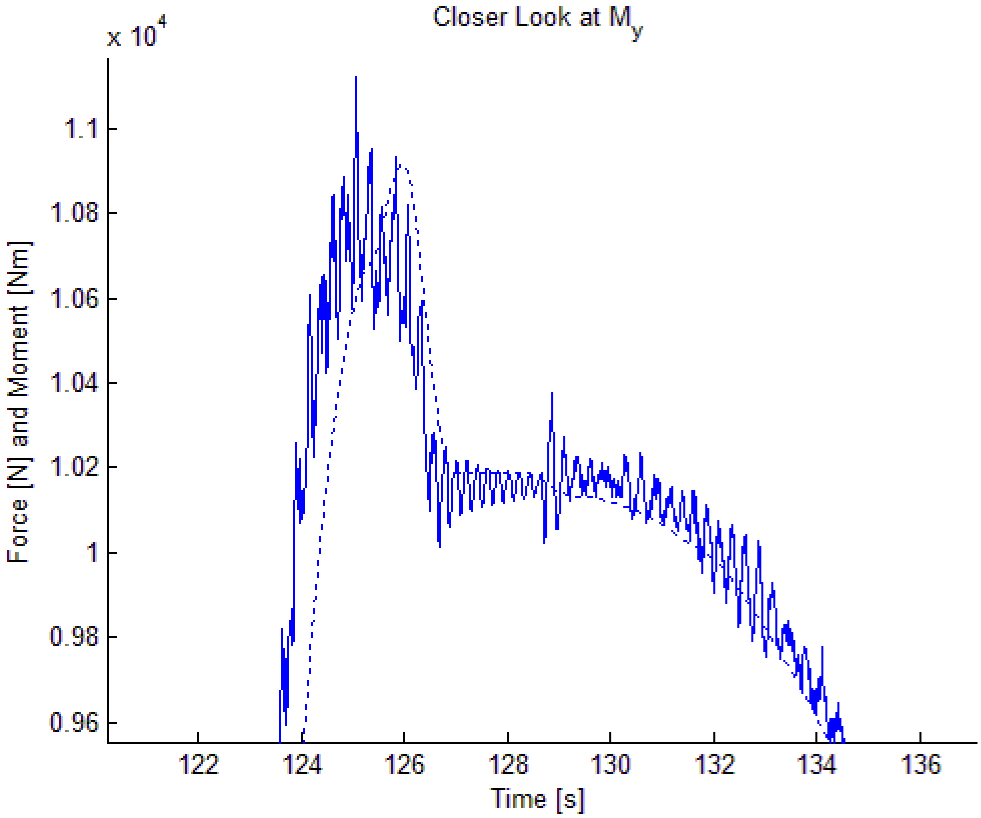

Figure 4.18: A closer look at the y-moment measured and predicted base reactions, from Figure 4.16.

Looking at Figure 4.19, it is noticed that the vibrations in the system cause moment measurements that can vary within a very wide range, here shown to be near a 125Nm peak to peak difference. This difference limits collision detection to half of that difference, perhaps 64Nm at the minimum, and that is only if a model is created that perfectly describes the expected reactions. These measurements are the results of the C-arm making a 180 degree rotation, then reversing direction and turning back the full 180 degrees. During other robot motions the vibrations may potentially be even larger, since the motions shown here are relatively low-amplitude compared to the abilities of the industrial robot system.

These vibrations in the system are a major factor affecting the reliability of collision detection, much more so than the presence of noise in the sensors. When the robot performs large amplitude motions, it causes these vibrations, which in turn cause large moment measurements. Any collision forces are effectively lost in the vibration amplitude of the robot. As seen previously in Figure 4.10 and Table 4.2, the noise from the sensors is small enough that even 10Nm disturbance moments are detectable, so these vibrations greatly overshadow the sensor noise.

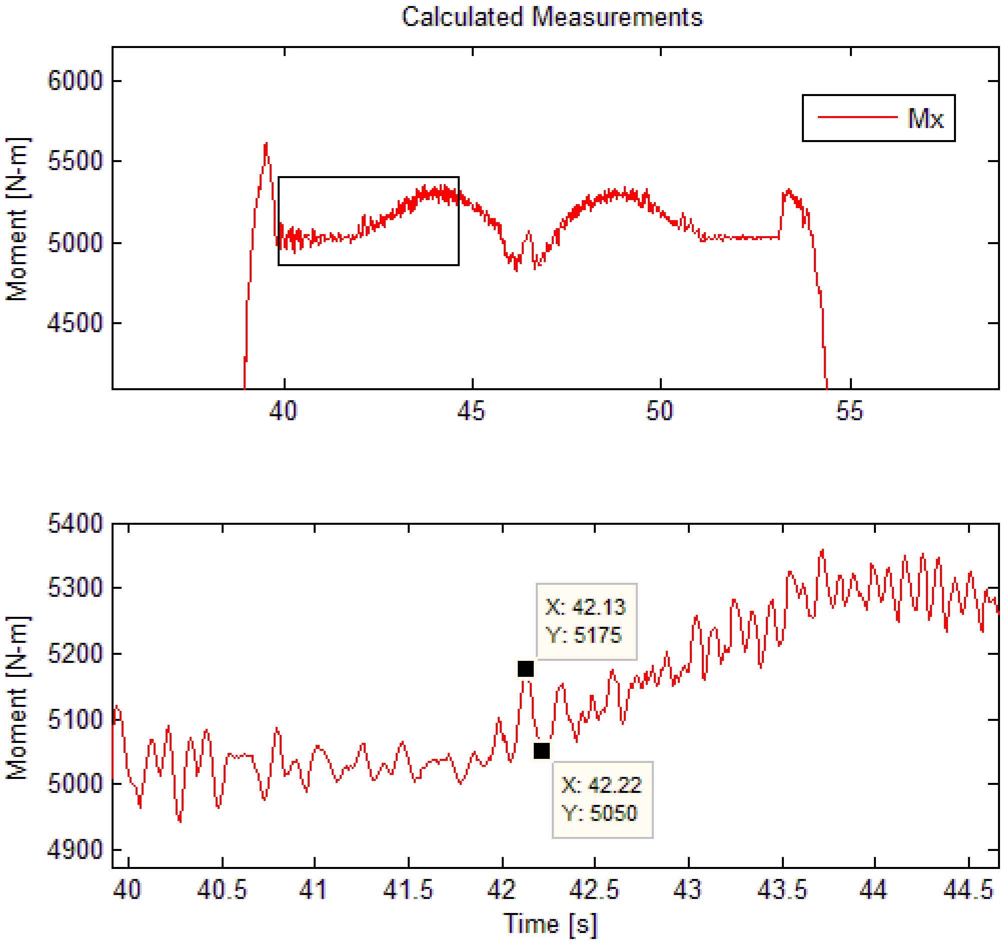

Figure 4.19: TOP: Typical C-arm rotation. BOTTOM: Vibration amplitude measurement.

As mentioned previously, the vibrations in the system limit collision detection to about 64Nm. This would be equivalent to a 64N vertical force at a distance of 1m, or a 50N vertical force at a distance of 1.3m. Since the C-arm center would typically be about 2m away from the origin give or take perhaps 0.3m, the detection of a 32N vertical force on the C-arm is theoretically possible. However, it is only possible to detect collision if there is either an accurate model or a-priori knowledge of the forces from a calibration run with which to compare the measurements. Therefore, the model accuracy is a limiting factor. If the parameters can be determined accurately to increase the model accuracy, then collision forces of 32N can be detected. If the vibrations can be accounted for in the model, then it would become possible to detect collision forces with even greater sensitivity.

Referring back to Figure 4.16, the y-moment measurements matched the predicted values much better than the x-moment measurements. By changing the initial parameter values, perhaps a more accurate fit can be obtained. Since throughout the range of motions the x-moment measurements were consistently higher than predicted, two assumptions can be made. First, that there is more mass located somewhere in the y-axis direction than initially estimated, which would reduce the x-moment measurements. Second, this mass is not in the x-axis direction since the y-moment measurements seem more accurate.

The best fit to these assumptions is that the base link center of mass, which was initially estimated to be at a location of [0 0 0.3]m, actually has some y-direction value, so the value is updated to [0 0.2 0.3]m. The results are shown in Figure 4.20. As can be seen, the predicted and measured reactions overall have a closer match, though they are still not quite perfect.

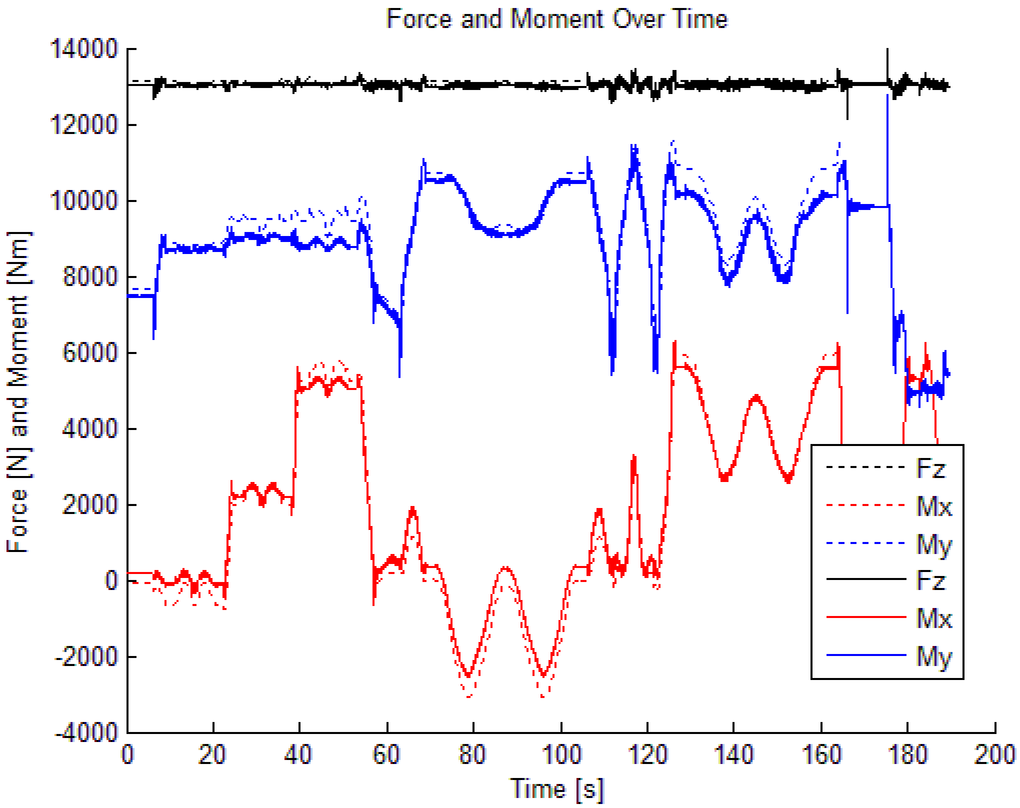

Figure 4.20: Updated center of mass location for link 1.

4.3.4 Parameter Estimation Results

To help estimate the robot parameters, several assumptions are made. In order to simplify estimations, the number of unknown parameters should be narrowed down to as few as possible. Since the first link of the four link system rotates only about the z-axis, the only measurable parameters are the mass and the x- and y-locations of the center of mass. The remaining parameters can be set to zero. Furthermore, the mass of each link is held constant since these are known with a good amount of confidence in their accuracy, with exception to the large cable hanging from the third link. This cable is ignored for now, but in future attempts will need to be included for better estimation and model accuracy. It is assumed that there is no z-component to the z-direction location of the center of mass of the second link. Likewise, it is assumed there is no y- or z-direction distance to the center of mass of the third link. Beside the mass, all of the parameters of the fourth link are limited to a one percent change in difference from their last estimated value to prevent large swings in the estimated parameter values and thus increase accuracy. This limitation is applied on various other parameters as well, and application is determined by experimentation and inspecting which values seem to be changing at high rates.

This method of narrowing down the number of parameters to estimate is combined with the estimation technique presented in Section 2.2.1. The results are shown in Figure 4.21. Figure 4.22 shows a zoomed view of the area marked in Figure 4.21, where the difference between the estimated and measured values is marked. The estimation results are clearly better than those seen in Figure 4.20 with the difference improved to 432Nm, though this difference is still not quite small enough to reach the goal of 50N collision force detection. However, these results show good promise that better initial estimates will close the gap between expected and measured reactions to a reasonable distance.

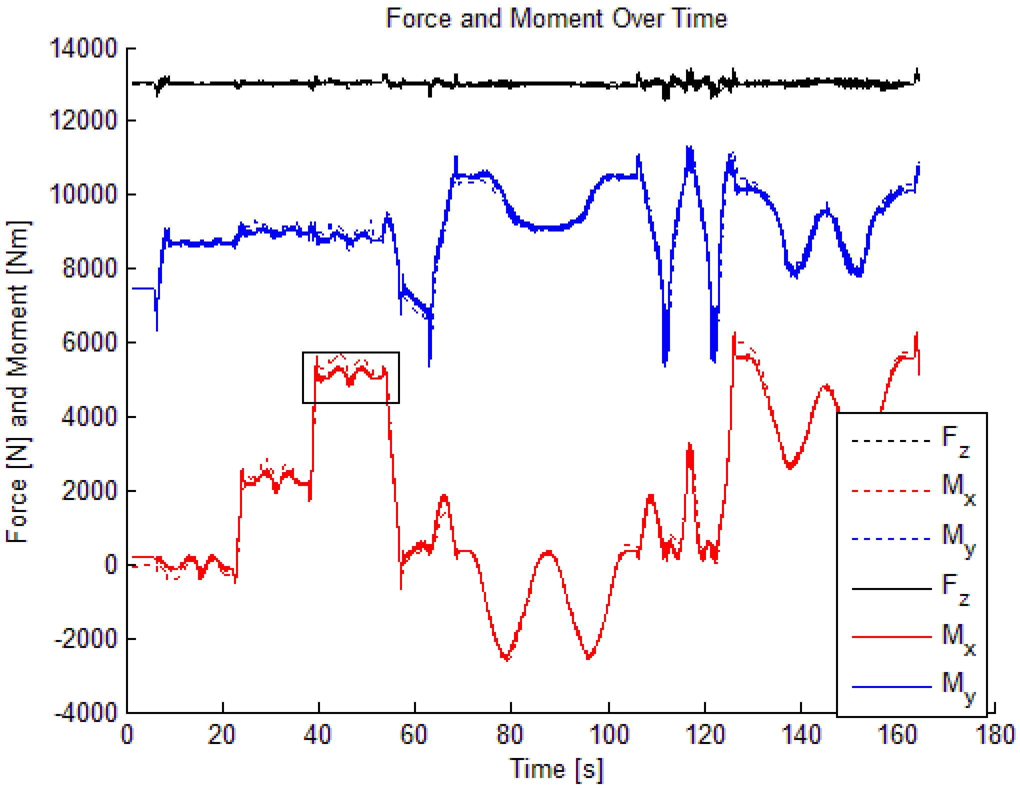

Figure 4.21: Difference between measured base reactions and predicted base reactions made with updated parameter estimates.

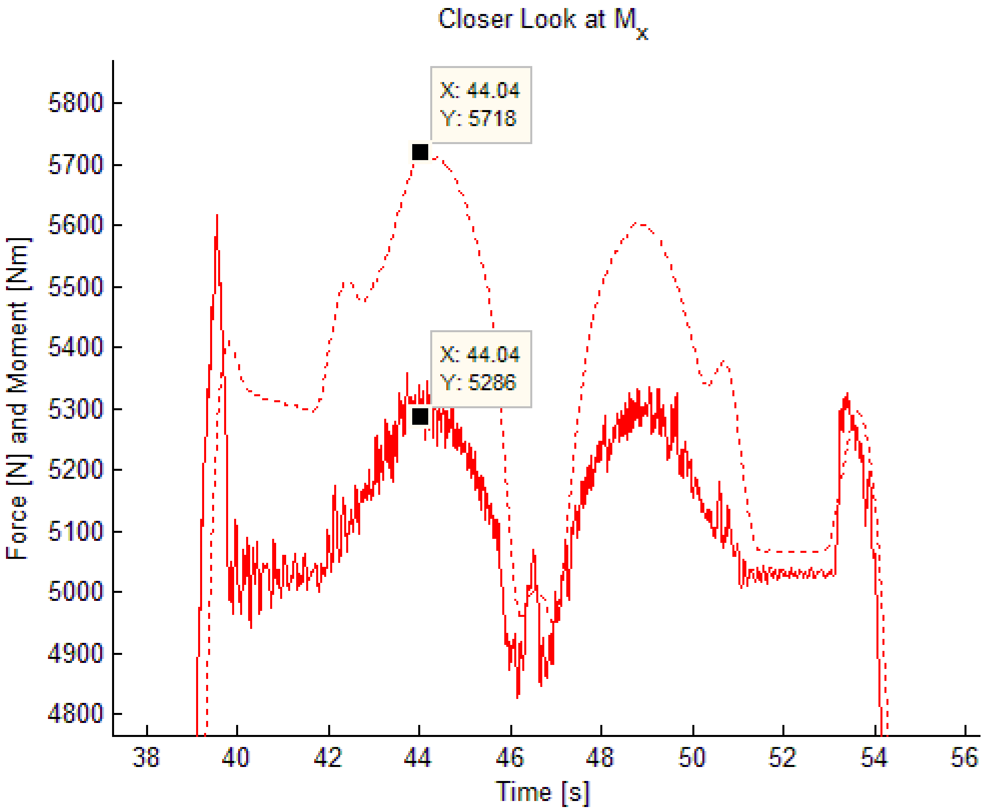

Figure 4.22: A closer look at the x-moment difference between measured (solid line) and predicted (dashed line) base reactions from Figure 4.19. The difference is around 432Nm, or 8%.

Chapter 5
Conclusions

Based on the results of this research, the method of using beam load cells to estimate link parameters and measure collision detection in high dynamics industrial robots shows promise. The parameter estimation method shows a good ability to reduce the errors of the expected base reaction measurements, despite the limited sensing capabilities of the setup of the four beam load cells. The largest error was reduced from 1,475 to 432Nm, or in terms of percentage from 28% to 8%. In research by Iagnemma et al. [9], the largest error between the predicted reaction moment and measured reaction moment was near 25%.

If the parameters can be estimated accurately, collision forces of less than 50N can be detected despite vibrations in the system. Good parameter estimates without modeled vibration could detect down to 32N of vertical collision force at a distance of 2m from the origin, which is within the operating area of the C-arm. The unmodeled vibrations are causing the 32N limit; a large part of the estimated parameter inaccuracy is most likely due to these unmodeled vibrations in the system.

The large number of unknown parameters increases the difficulty of parameter estimation as well. Perhaps if the number of unknown parameters was minimized by combining the linearly dependent columns of the U matrix from Equation 2.19 as is discussed by Iagnemma, Gautier, and An [9,10,11], the parameter estimations would be more accurate. For example, Iagnemma [9] minimizes a two link system from twenty parameters down to twelve. As An [11] notes,

> “Some parameters can only be identified in linear combinations, because proximal joints must provide the torque sensing to identify fully the parameters of each link. Certain parameters from distal links are carried

> down to proximal links until a link appears with a rotation axis oriented appropriately for completing the identification. In between, these parameters appear in linear combinations with other parameters. This partial identifiability and the difficulty of analysis become worse as the number of links is increased."

Another method to more accurately estimate the parameters would be the selection of efficient exciting trajectories. Simpler motions can be performed to help estimate parameters; as it is currently, the motions are complex. If simple motions are performed, like moving one link at a time about axes parallel to the global coordinate axes, it would limit the measureable parameters for that certain motion and allow the number of parameters being estimated to be narrowed down. With fewer parameters to estimate, the more accurately the parameters can be estimated. Then, these parameters can be used as more accurate initial estimates for the more complex motions in which other unknown parameters are estimated.

The unmodeled hose hanging from the industrial test robot is another probable significant factor affecting the reaction force prediction measurement errors. Besides adding an unknown amount of mass to the system, it also adds nonlinear dynamics as it is dragged along the ground and twisted around or as it slides off of the arm as the system is moving. The end product will not have this issue; the hose will be nicely contained, unlike on the prototype, which should allow for more accurate parameter estimations. If Siemens has a CAD model of the robot system, this could also be used to make initial estimates of the parameters.

Perhaps another reasonable way to detect collision would be to create a database of the base reaction measurements for preprogrammed runs. Then, when the system is commanded to run through these preprogrammed motions, it can compare the base reaction measurements with the database to check for significant differences.

The accuracy of these collision detection methods can be increased by limiting the vibrations in the system by adjusting the motions of the robot. For example, before certain motions, the system can be stopped for a certain amount of time while the vibrations of the system minimize due to damping. In this manner, the size of collision forces which are masked due to the vibrations in the system are minimized. The accuracy of collision detection can also be increased by installing the sensors so the operating space of the C-arm is about 45 degrees from the x- and y-axes where sensitivity to collision is greatest, as presented in Section 4.1.

A practical design concern is the reaction speed of the robot, which needs to be considered in application. Upon sensing a collision, the robot should stop its motion and perhaps retract. Reaction speed is a function of the controls capabilities of the robotic system and outside the scope of this work to make any assumptions as to the feasibility. Retraction from a collision point is possible since the robot can keep track of its motion and, when a collision is detected, can reverse its motion back along the path it was traveling. If an object is hit and pinned down by the robot, the object can be released if the robot retracts. On the other hand, an object or person may be pinned down if the robot does not retract and simply stops moving when the collision is detected.

Bibliography

[1] Alessandro De Luca, Alin Albu-Schaffer, Sami Haddadin, and Gerd Hirzinger. Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 1623-1630. IEEE, 2006.

[2] L Gonchar, D Engel, J Raczkowsky, and H Worn. Virtual simulation system for collision avoidance for medical robot. *Studies in health technology and informatics*, pages 168-170, 2001.

[3] Sami Haddadin, Alin Albu-Schäffer, Alessandro De Luca, and Gerd Hirzinger. Evaluation of collision detection and reaction for a human-friendly robot on biological tissues. In *IARP International Workshop on Technical challenges and for dependable robots in Human environments (IARP2008), Pasadena, USA*, 2008.

[4] James Michael Wooten. High-dynamic range collision detection using piezoelectric polymer films for planar and non-planar applications. Thesis, Auburn University, 2013.

[5] Yoji Yamada, Yasuhiro Hirasawa, Shengyang Huang, Yoji Umetani, and Kazustugu Suita. Human-robot contact in the safeguarding space. *Mechatronics, IEEE/ASME Transactions on*, 2(4):230-236, 1997.

[6] Brian Armstrong, Oussama Khatib, and Joel Burdick. The explicit dynamic model and inertial parameters of the puma 560 arm. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 510-518. IEEE, 1986.

[7] Hee-Jun Kang, Young-Shick Ro, et al. Robot manipulator modeling in matlab-simmechanics with pd control and online gravity compensation. In *Strategic Technology (IFOST), 2010 International Forum on*, pages 446-449. IEEE, 2010.

[8] Juan Barreto and Luis E Muñoz. Inertia parameter identification using a stewart platform. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1-8. VDE, 2010.

[9] Guangjun Liu, Karl Iagnemma, Steven Dubowsky, and Guillaume Morel. A base force/torque sensor approach to robot manipulator inertial parameter estimation. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 4, pages 3316-3321. IEEE. IEEE, 1998.

[10] Maxime Gautier and Wisama Khalil. Direct calculation of minimum set of inertial parameters of serial robots. *Robotics and Automation, IEEE Transactions on*, 6(3):368-373, 1990.

[11] Chae H An, Christopher G Atkeson, and John M Hollerbach. Estimation of inertial parameters of rigid body links of manipulators. In *Decision and Control, 1985 24th IEEE Conference on*, volume 24, pages 990-995. IEEE, 1985.

[12] James H Graham, John F Meagher, and Stephen J Derby. A safety and collision avoidance system for industrial robots. *Industry Applications, IEEE Transactions on*, (1):195-203, 1986.

[13] Vladimir J Lumelsky and Edward Cheung. Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators. *Systems, Man and Cybernetics, IEEE Transactions on*, 23(1):194-203, 1993.

[14] Jim L Novak and John T Feddema A capacitance-based proximity sensor for whole arm obstacle avoidance. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 1307-1314. IEEE, 1992.

[15] Peitao Shi, Min Tan, and Xiaojun Ma. Open fuzzy force controller of manipulators with unknown environment parameters. In *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 2, pages 1056-1061. IEEE, 2000.

[16] Homayoun Seraji and Bruce Bon. Real-time collision avoidance for position-controlled manipulators. *Robotics and Automation, IEEE Transactions on*, 15(4):670-677, 1999.

[17] Shujun Lu and Jae H Chung. Collision detection enabled weighted path planning: a wrist and base force/torque wensors approach. In *Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on*, pages 165-170. IEEE, 2005.

[18] Walter Verdonck, Jan Swevers, Xavier Chenut, and Jean-Claude Samin. Combining internal and external robot models to improve model parameter estimation. In

Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, volume 3, pages 2846-2851. IEEE, 2001.

[19] KR Symon. Mechanics addison-wesley. *Reading, MA*, 1971.

[20] Chapter 11: Least Squares, Pseudo-Inverses, PCA & SVD [last verified date: 5 Aug 14], sci.utah.edu, http://www.sci.utah.edu/~gerig/CS6640-F2012/Materials/pseudoinverse-cis61009sl10.pdf

[21] Tsuneo Yoshikawa. Force control of robot manipulators. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 1, pages 220 226. IEEE, 2000.

[22] K Iagnemma, G Morel, and S Dubowski. A model-free fine position control system using the base-sensor: With application to a hydraulic manipulator. In *Symposium on Robot Control, SYROCO*, volume 97, pages 359 365, 1997.