Identification of the Inertial Parameters of Manipulator Payloads

Ryan-David Reyes
IDENTIFICATION OF THE INERTIAL PARAMETERS OF MANIPULATOR PAYLOADS

By

RYAN - DAVID REYES

An Honors Thesis submitted to the Department of Electrical and Computer Engineering in partial fulfillment of the requirements for graduation with Honors in the Major

Degree Awarded:
Spring Semester, 2015
The members of the committee approve the Honors Thesis of Ryan - David Reyes defended on August 5th, 2014.

Dr. Oscar Chuy  
Professor Directing Honors Thesis

Dr. Rodney Roberts  
Committee Member

Dr. Emmanuel Collins  
Committee Member
TABLE OF CONTENTS

List of Tables ............................................................... iv
List of Figures ................................................................. v
Abstract .............................................................................. vi

1. Introduction ................................................................. 1

2. Related Work ............................................................... 2
   2.1 Least Squares Identification Methods ......................... 2
   2.2 Kalman Filter Identification Methods ......................... 3
   2.3 Exciting Trajectory Determination ............................... 3

3. Dynamic Model of a 2 DOF Manipulator ......................... 5

4. Methodology ............................................................... 10
   4.1 Method of Least Squares Estimation ......................... 10
   4.2 Exciting Trajectories .............................................. 12
   4.3 Implementation in Simulation ................................. 13

5. Experimental Results and Discussion ............................ 15
   5.1 Experimental Setup ............................................... 15
   5.2 Exciting Trajectories .............................................. 16

6. Conclusion ................................................................. 30
   6.1 Future Work ......................................................... 30

Appendices ........................................................................ 31

A. .................................................................................... 32

REFERENCES .................................................................... 38
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Base Inertial Parameters of the Pelican and CISCOR Manipulators</td>
<td>9</td>
</tr>
<tr>
<td>5.1</td>
<td>Loaded Inertial Parameters of the CISCOR Manipulator</td>
<td>16</td>
</tr>
<tr>
<td>5.2</td>
<td>Estimation Results for Trajectory $-0.674 \leq y \leq -0.475$ with a 5 lb Load.</td>
<td>17</td>
</tr>
<tr>
<td>5.3</td>
<td>Estimation Results for Trajectory $-0.674 \leq y \leq -0.475$ with a 10 lb Load.</td>
<td>18</td>
</tr>
<tr>
<td>5.4</td>
<td>Estimation Results for Trajectory $-0.674 \leq y \leq -0.575$ with a 5 lb Load.</td>
<td>19</td>
</tr>
<tr>
<td>5.5</td>
<td>Estimation Results for Trajectory $-0.674 \leq y \leq -0.575$ with a 10 lb Load.</td>
<td>19</td>
</tr>
<tr>
<td>5.6</td>
<td>Estimation Results for Trajectory $-0.1 \leq x \leq 0.1, y = -0.575$ with a 5 lb Load.</td>
<td>20</td>
</tr>
<tr>
<td>5.7</td>
<td>Estimation Results for Trajectory $-0.1 \leq x \leq 0.1, y = -0.575$ with a 10 lb Load.</td>
<td>21</td>
</tr>
<tr>
<td>5.8</td>
<td>Estimation Results for Trajectory $-0.2 \leq x \leq 0.2, y = -0.575$ with a 5 lb Load.</td>
<td>21</td>
</tr>
<tr>
<td>5.9</td>
<td>Estimation Results for Trajectory $-0.2 \leq x \leq 0.2, y = -0.575$ with a 10 lb Load.</td>
<td>22</td>
</tr>
<tr>
<td>5.10</td>
<td>Estimation Results for Trajectory $0 \leq q_2 \leq \frac{\pi}{6}$ with a 5 lb Load.</td>
<td>24</td>
</tr>
<tr>
<td>5.11</td>
<td>Estimation Results for Trajectory $0 \leq q_2 \leq \frac{\pi}{6}$ with a 10 lb Load.</td>
<td>24</td>
</tr>
<tr>
<td>5.12</td>
<td>Inertia Estimation Results for Trajectory $0 \leq q_2 \leq \frac{\pi}{6}$ using reduced regressor.</td>
<td>25</td>
</tr>
<tr>
<td>5.13</td>
<td>Estimation Results for Oscillating Trajectory $-\frac{\pi}{6} \leq q_2 \leq \frac{\pi}{6}$ with a 5 lb Load.</td>
<td>26</td>
</tr>
<tr>
<td>5.14</td>
<td>Estimation Results for Trajectory $-\frac{\pi}{6} \leq q_2 \leq \frac{\pi}{6}$ with a 10 lb Load.</td>
<td>26</td>
</tr>
<tr>
<td>5.15</td>
<td>Estimation Results for Oscillating Trajectory $-\frac{\pi}{12} \leq q_2 \leq \frac{\pi}{12}$ with a 5 lb Load.</td>
<td>27</td>
</tr>
<tr>
<td>5.16</td>
<td>Estimation Results for Trajectory $-\frac{\pi}{12} \leq q_2 \leq \frac{\pi}{12}$ with a 10 lb Load.</td>
<td>27</td>
</tr>
<tr>
<td>5.17</td>
<td>Inertia Estimation Results for Trajectory $-\frac{\pi}{12} \leq q_2 \leq \frac{\pi}{12}$ using reduced regressor.</td>
<td>28</td>
</tr>
<tr>
<td>5.18</td>
<td>Inertia Estimation Results for Trajectory $-\frac{\pi}{6} \leq q_2 \leq \frac{\pi}{6}$ using reduced regressor.</td>
<td>28</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

3.1 Diagram of the Pelican Prototype Robot [1]. ............................................. 5
3.2 The 2 DOF manipulator used at CISCOR, Robotics Lab. .............................. 7
5.1 An example of a vertical trajectory which excites the gravity effect. ............. 17
5.2 Joint positions, velocities, and accelerations for trajectory $-0.674 \leq y \leq -0.475$. 18
5.3 Joint positions, velocities, and accelerations for trajectory $-0.674 \leq y \leq -0.575$. 19
5.4 An example of a horizontal trajectory which excites the inertia effect. ......... 20
5.5 Joint positions, velocities, and accelerations for trajectory $-1 \leq x \leq 0.1$, $y = -0.575$. ................................................................. 21
5.6 Joint positions, velocities, and accelerations for trajectory $-0.2 \leq x \leq 0.2$, $y = -0.575$. ................................................................. 22
5.7 An example of a trajectory which only moves the second link. ................. 23
5.8 Joint positions, velocities, and accelerations for trajectory $0 \leq q_2 \leq \frac{\pi}{6}$. ....... 23
5.9 Joint positions, velocities, and accelerations for oscillating trajectory $-\frac{\pi}{12} \leq q_2 \leq \frac{\pi}{12}$. ................................................................. 26
5.10 Joint positions, velocities, and accelerations for oscillating trajectory $-\frac{\pi}{6} \leq q_2 \leq \frac{\pi}{6}$. ................................................................. 27
Momentum based motion planning allows small and lightweight manipulators to lift loads that exceed their rated load capacity. One such planner, *Sampling Based Model Predictive Optimization (SBMPO)* developed at the *Center for Intelligent Systems, Control, and Robotics (CISCOR)*, uses dynamic and kinematic models to produce trajectories that take advantage of momentum. However, the inertial parameters of the payload must be known before the trajectory can be generated.

This research utilizes a method based on least squares techniques for determining the inertial parameters of a manipulator payload. It is applied specifically to a two degree of freedom manipulator. A set of exciting trajectories, i.e., trajectories that sufficiently excite the manipulator dynamics, in task space will be commanded to the system. Inverse kinematics are then used to determine the desired angle, angular velocity, and angular acceleration for the manipulator joints. Using the sampled torque, joint position, velocity, and acceleration data, the least squares technique produces an estimate of the inertial parameters of the payload. This paper focuses on determining which trajectories produce sufficient excitation so that an adequate estimate can be obtained.

Keywords: Manipulator, Payload, Inertia, Identification
CHAPTER 1

Introduction

Robotic manipulators are used heavily in the industrial field, performing tasks such as lifting, cutting, and material handling. Large manipulators are used to carry heavy loads, but they are very expensive to manufacture. By reducing the size of these manipulators, much of the manufacturing cost can be eliminated, and the range of applications is increased. For example, with their smaller size, manipulators are fit for use in the home as a personal robot, whereas a larger manipulator would be intrusive and unwieldy. The problem arises in that these smaller manipulators are incapable of lifting heavier loads, and are not suitable to replace larger manipulators.

Recent research has been done to circumvent this problem, via exploitation of momentum. The intelligent motion planner presented in [2], known as Sampling Based Model Predictive Optimization, is developed at the Center for Intelligent Systems, Control, and Robotics (CISCOR). This motion planner creates a trajectory that utilizes momentum, which allows smaller manipulators to lift loads beyond their specification to the desired target location. The motion planner requires a priori knowledge of the inertial parameters of the payload attached. If unknown, the planner may generate a trajectory that violates the torque constraint of the manipulator joints. In application, a priori knowledge cannot be guaranteed, since the inertial parameters vary according to the load.

The research described by this thesis will enable fast identification of the inertial parameters of the load, specifically for a two degree of freedom manipulator. It focuses on determining adequate exciting trajectories, so that the identification technique can produce a good estimate of the payload parameters. This thesis contributes a set of feasible trajectories for a load bearing two degree of freedom manipulator that produces good estimates of the mass, center of mass, and inertia.
CHAPTER 2

Related Work

The dynamics of a manipulator are a set of nonlinear differential equations, which makes
the determination of the inertial parameters challenging. There are two major numerical
optimization approaches used in the identification algorithm that determines the inertial
parameters [3]: least squares techniques and Kalman filters. This section will introduce
and compare these two techniques, and conclude with the problem of generating exciting
trajectories.

2.1 Least Squares Identification Methods

The least squares estimation method is a well tested algorithm for robot parameter identi-
fication [4–9]. These methods are based upon the Newton-Euler or Lagrangian formulation
of dynamics, and try to modify the dynamic equation to be expressed as a set of linear
equations with respect to the inertial parameters. This formulation allows the least squares
 technique to estimate the inertial parameters of the payload.

There are many different approaches to implementing this method for parameter iden-
tification. In [4], a force sensor was used on the end effector to simplify the parameter
identification of the load, whereas [5] did not require one in its implementation. In [6], the
reflected inertias of the actuators and the paths in joint space were used as the measurement
data, as opposed to the more usual method of only joint measurements. The method
presented in [7] utilizes the total least squares field to identify all the inertial and drive
gain parameters, which is demonstrated to be more efficient than ordinary least squares
techniques. In [8], a method was proposed that was capable of determining the inertial
parameters from using only the torque data of a robot, by using non-linear least squares
estimation to match the parameters of a simulation to actual sampled data.
2.2 Kalman Filter Identification Methods

The second class of parameter identification methods use Kalman filters. Kalman filters provide good estimates of the system state when used in conjunction with the dynamic model [10–12], by acting recursively on noisy and inaccurate data. They determine the best averaging factor for subsequent states based upon past measurements of the system. In [11], it is stated that the main drawback of using a Kalman filter is the requirement of measured joint accelerations. These accelerations are commonly derived from digital differentiation of the velocity, which introduces noise. This degrades the accuracy of the estimated inertial parameters.

A comparison between Kalman filters and the weighted least squares techniques was made in [12]. Both implementations were able to estimate the parameters with good accuracy, but the Kalman filter was very sensitive to initial conditions, and took a longer time to converge. The conclusion of [12] was that the weighted least squares implementation was superior to the Kalman filter for off-line computation. Thus, the methodology presented in this work implements the least squares technique for estimating the inertial parameters.

2.3 Exciting Trajectory Determination

An important sub-problem that this work will focus on is determining an exciting trajectory that produces the best estimation of inertial parameters [13–16]. The identification methods outlined above can work with any trajectory, but produce better estimates when the inertial, Coriolis, and gravity terms in the dynamic equation are sufficiently excited. It is shown that the condition number of the excitation matrix directly influences the convergence rate and noise sensitivity of the identification algorithm [13]. It is suggested in [13] that the condition number must be optimized close to unity to produce the best identification results. In [14], this idea is extended in that a trajectory is generated using nonlinear optimization techniques to minimize a cost function involving the condition number. [15] proposes a method to generate an exciting trajectory made using mutual productwise odd functions.

The above trajectory generation algorithms are not guaranteed to produce a feasible trajectory for a load bearing system. The motors may be incapable of tracking the trajectory since it violates the motor torque constraints. The motors may become saturated and unable to move in this situation. A sequential identification paradigm is presented in [16], where a
set of different trajectories are sampled, with each one exciting different parameters. Each
trajectory is also designed to minimize the condition number of the excitation matrix. It
is easier to assure feasible trajectories for the manipulator when using this paradigm, and,
as such, will be the method implemented in this work. As such, this paradigm will be used
when developing exciting trajectories for identification of the payload.
CHAPTER 3

Dynamic Model of a 2 DOF Manipulator

The dynamic model of the two degree of freedom manipulator is identical to the one used to describe the Pelican Prototype Robot used in [1]. Figure 3.1 shows how the parameters of the robot arm are defined:

- \( l_1 \) is the length of link 1.
- \( l_2 \) is the length of link 2.
- \( l_{c1} \) is the distance between the first joint and the center of mass of the first link.
- \( l_{c2} \) is the distance between the second joint and the center of mass of the second link.

![Diagram of the Pelican Prototype Robot](image)

Figure 3.1: Diagram of the Pelican Prototype Robot [1].
\begin{itemize}
    \item $q_1$ is the angle of the first joint relative to the vertical.
    \item $q_2$ is the angle of the second joint measured relative to angle $q_1$.
    \item $I_1$ is the moment of inertia of the first link w.r.t. the axis that passes through its center of mass and is parallel to the $x$ axis.
    \item $I_2$ is the moment of inertia of the second link w.r.t. the axis that passes through its center of mass and is parallel to the $x$ axis.
    \item $m_1$ is the mass of link 1.
    \item $m_2$ is the mass of link 2.
\end{itemize}

These parameters are defined the same way for the manipulator available in the CISCOR robotics lab, as shown in Fig. 3.2. Table 3.1 details the base parameters of both manipulators.

The dynamics of the two DOF manipulator are derived from the Lagrangian [1], given by

\begin{equation}
\mathcal{L} = \mathcal{K}(q, \dot{q}) - \mathcal{U}(q),
\end{equation}

where $\mathcal{K}$ and $\mathcal{U}$ are the sum of the kinetic and potential energy, respectively, of the two links.

The total Lagrangian of the system is found to be

\begin{equation}
\mathcal{L} = \frac{1}{2} [m_1 l_1^2 + m_2 l_1^2] \dot{q}_1^2 \\
+ \frac{1}{2} m_2 l_2^2 [\dot{q}_1^2 + 2 \dot{q}_1 \dot{q}_2 + \dot{q}_2^2] \\
+ m_2 l_1 l_2 \cos(q_2) [\dot{q}_1^2 + \dot{q}_1 \dot{q}_2] \\
+ [m_1 l_1 + m_2 l_1] g \cos(q_1) \\
+ m_2 g l_2 \cos(q_1 + q_2) \\
+ \frac{1}{2} I_1 \dot{q}_1^2 + \frac{1}{2} I_2 [\dot{q}_1 + \dot{q}_2]^2.
\end{equation}
The dynamic equations are then derived via Lagrange’s Equations of motion:

\[
\tau_1 = \left[ m_1 l_1^2 + m_2 l_2^2 \right] \ddot{q}_1 + \left[ m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2) + I_2 \right] \ddot{q}_2 \\
+ 2 m_2 l_1 l_2 \cos(q_2) \dot{q}_1 \dot{q}_2 \\
+ \left[ m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2) + I_2 \right] \ddot{q}_2 \\
- 2 m_2 l_1 l_2 \sin(q_2) \dot{q}_1 \dot{q}_2 \\
- m_2 l_1 l_2 \sin(q_2) \dot{q}_2^2 \\
+ [m_1 l_1 + m_2 l_1] g \sin(q_1) \\
+ m_2 g l_2 \sin(q_1 + q_2)
\] (3.3)
and
\[
\tau_2 = \left[ m_2 \ell_2^2 + m_2 \ell_1 \ell_2 \cos(q_2) + I_2 \right] \ddot{q}_1 \\
+ \left[ m_2 \ell_2^2 + I_2 \right] \ddot{\bar{q}}_2 \\
+ m_2 \ell_1 \ell_2 \sin(q_2) \dot{q}_1^2 \\
+ m_2 g \ell_2 \sin(q_1 + q_2),
\]
where \( \tau_1 \) and \( \tau_2 \) are the torques at each of the joints, given \( q_1, \dot{q}_1, \ddot{q}_1, q_2, \dot{q}_2, \) and \( \bar{q}_2 \). These equations can be rewritten in compact form as
\[
\tau = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q),
\]
which represents the inverse dynamics of the system. The inertia matrix, \( M(q) \), contains all the inertia terms that affect the torque. It is defined as
\[
M(q) = \begin{bmatrix}
M_{11}(q) & M_{12}(q) \\
M_{21}(q) & M_{22}(q)
\end{bmatrix},
\]
where the elements of \( M \) are given as
\[
M_{11} = m_1 \ell_1^2 + m_2 \left[ \ell_1^2 + \ell_2^2 + 2 \ell_1 \ell_2 \cos(q_2) \right] + I_1 + I_2, \\
M_{12} = M_{21} = m_2 \left[ \ell_2^2 + \ell_1 \ell_2 \cos(q_2) \right] + I_2, \\
M_{22} = m_2 \ell_2^2 + I_2.
\]
The matrix of Coriolis terms, \( C(q, \dot{q}) \), is defined as
\[
C(q, \dot{q}) = \begin{bmatrix}
C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\
C_{21}(q, \dot{q}) & C_{22}(q, \dot{q})
\end{bmatrix},
\]
The elements of \( C \) are given as
\[
C_{11} = -m_2 \ell_1 \ell_2 \sin(q_2) \dot{q}_2, \\
C_{12} = -m_2 \ell_1 \ell_2 \sin(q_2) [\dot{q}_1 + \dot{q}_2], \\
C_{21} = m_2 \ell_1 \ell_2 \sin(q_2) \dot{q}_1 \\
C_{22} = 0.
\]
Finally \( G(q) \) is the matrix of gravity terms. This matrix is defined as
\[
G(q) = \begin{bmatrix}
G_1(q) \\
G_2(q)
\end{bmatrix},
\]
Table 3.1: Base Inertial Parameters of the Pelican and CISCOR Manipulators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Pelican Manipulator</th>
<th>CISCOR Manipulator</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_1$</td>
<td>0.26</td>
<td>0.375</td>
<td>m</td>
</tr>
<tr>
<td>$l_2$</td>
<td>0.26</td>
<td>0.3</td>
<td>m</td>
</tr>
<tr>
<td>$l_{c1}$</td>
<td>0.0983</td>
<td>0.195</td>
<td>m</td>
</tr>
<tr>
<td>$l_{c2}$</td>
<td>0.0229</td>
<td>0.22</td>
<td>m</td>
</tr>
<tr>
<td>$m_1$</td>
<td>6.5225</td>
<td>2.883</td>
<td>kg</td>
</tr>
<tr>
<td>$m_2$</td>
<td>2.0458</td>
<td>1.085</td>
<td>kg</td>
</tr>
<tr>
<td>$I_1$</td>
<td>0.1213</td>
<td>0.0345</td>
<td>kg m$^2$</td>
</tr>
<tr>
<td>$I_2$</td>
<td>0.0116</td>
<td>0.013</td>
<td>kg m$^2$</td>
</tr>
</tbody>
</table>

and the elements of $G$ are

\[
G_1 = [m_1 l_{c1} + m_2 l_1]g \sin(q_1) + m_2 g l_{c2} \sin(q_1 + q_2),
\]

\[
G_2 = m_2 l_{c2} \sin(q_1 + q_2).
\]

The compact form equation, (3.5), provides the foundation upon which our methodology is developed.
CHAPTER 4

Methodology

The main focus of this research is to develop a set of exciting trajectories that will enable the identification algorithm to estimate the inertial parameters of a payload connected to the end effector of a manipulator. This will be developed, simulated, and tested using a two degree of freedom manipulator. The following describes the basic steps of this methodology. Ideally, this algorithm can be extended for n-DOF manipulators.

A preplanned trajectory is commanded to the unloaded system, and the torque data is collected. Using the least squares technique described below, the inertial parameters of the manipulator are estimated from the collected data. This step only needs to be run once to provide the base measurements of the system. The payload is attached to the end effector, and the same trajectory is commanded to the system. The torque data of the loaded system is collected, and the inertial parameters of the end effector are estimated using the same technique. The difference between the estimated values of the loaded and unloaded system is used to calculate the inertial parameters of the payload itself.

To create the exciting trajectory, a fifth order trajectory, \((x_d, y_d)\), is created in task space. The trajectory is then mapped to joint space using the inverse kinematics. Given \((x_d, y_d)\), \(q_{d1}\) and \(q_{d2}\) are

\[
q_{d2} = \cos^{-1} \left( \frac{x_d^2 + y_d^2 - l_1^2 - l_2^2}{2l_1 l_2} \right) \tag{4.1}
\]

\[
q_{d1} = \tan^{-1} \left( \frac{x_d}{y_d} \right) - \tan^{-1} \left( \frac{l_2 \sin(q_{d2})}{l_1 + l_2 \cos(q_{d2})} \right), \tag{4.2}
\]

The joint velocities and accelerations are calculated using differentiation.

4.1 Method of Least Squares Estimation

The dynamic equation (3.5) can be written in a form that is linear in combinations of
the inertial parameters - mass, lengths, and moments of inertia [17]. It is this fact that is used to determine the inertial parameters of the load. The matrices \( M(q) \), \( C(q, \dot{q}) \), and \( G(q) \) in Equation (3.5) are written as
\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Y(q, \dot{q}, \ddot{q})\chi, \tag{4.3}
\]
where \( Y(q, \dot{q}, \ddot{q}) \) is known as the regressor, and \( \chi \) is the vector of inertial parameters. The elements of \( \chi \) are
\[
\chi = \begin{bmatrix}
\chi_1 \\
\chi_2 \\
\chi_3 \\
\chi_4 \\
\chi_5
\end{bmatrix} = \begin{bmatrix}
m_1l_{c1}^2 + m_2(l_1^2 + l_{c2}^2) + I_1 + I_2 \\
m_2l_{c1}l_{c2} \\
m_2l_{c2}^2 + I_2 \\
m_1l_{c1} + m_2l_1 \\
m_2l_{c2}
\end{bmatrix}, \tag{4.4}
\]
for the two DOF manipulator [17]. To derive \( Y(q, \dot{q}, \ddot{q}) \), the entries of the three matrices, \( M(q) \), \( C(q, \dot{q}) \), and \( G(q) \) are rewritten in terms of elements of \( \chi \), and given as follows:
\[
M_{11} = \chi_1 + 2\chi_2 \cos(q_2), \\
M_{12} = M_{21} = \chi_3 + \chi_2 \cos(q_2), \\
M_{22} = \chi_3, \\
C_{11} = -\chi_2 \sin(q_2)\dot{q}_2, \\
C_{12} = -\chi_2 \sin(q_2)[\dot{q}_1 + \dot{q}_2], \\
C_{21} = \chi_2 \sin(q_2)\dot{q}_1, \\
G_1 = \chi_4 \sin(q_1) + \chi_5 g \sin(q_1 + q_2), \\
G_2 = \chi_5 g \sin(q_1 + q_2). \tag{4.5}
\]
\( Y(q, \dot{q}, \ddot{q}) \) is derived from these equations as
\[
\begin{bmatrix}
\dot{q}_1 \\
Y_{12} \\
\dot{q}_2 \\
Y_{22}
\end{bmatrix} = \begin{bmatrix}
g \sin(q_1) & g \sin(q_1 + q_2) \\
0 & \dot{q}_1 + \dot{q}_2 & 0 \\
g \sin(q_1 + q_2)
\end{bmatrix},
\]
where \( Y_{12} \) and \( Y_{22} \) are respectively given as
\[
Y_{12} = \cos(q_2)(2\ddot{q}_1 + \ddot{q}_2) - \sin(q_2)(\dot{q}_1^2 + 2\dot{q}_1\dot{q}_2), \\
Y_{22} = \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2.
\]

When matrices \( Y \) and \( \chi \) are substituted into (3.5), the resulting equation is similar to the linear system \( Ax = b \) [17]:
\[
Y\chi = \tau. \tag{4.6}
\]
This produces an underdetermined system. To account for this, samples are taken and are augmented into $Y(q, \dot{q}, \ddot{q})$ and $\tau$ as follows

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_N \end{bmatrix}$$

where $N$ is the number of samples taken [4]. Every sample adds two equations to the matrix, and this continues until the system is overdetermined. It follows that $Y$ is a full rank matrix [4]. A least squares approximation is done on the data, using the left pseudoinverse of $Y$, shown in (4.8):

$$\hat{\chi} = (Y^T Y)^{-1} Y^T \tau.$$  

(4.8)

Using the computed entries of $\hat{\chi}$, the inertial parameters - $l_2$, $m_2$, and $I_2$, - are solved for by (4.4). The difference between the values computed for the loaded versus the unloaded system are indicative of the inertial parameters of the payload itself.

Using the known base parameters of the system, the updated inertial parameters of the end effector are solved as follows:

$$m_2 = \frac{\hat{\chi}_4 - m_1 l_1}{l_1}$$

$$l_2 = \frac{\hat{\chi}_5}{m_2}$$

$$I_2 = \hat{\chi}_3 - m_2 l_2^2$$

### 4.2 Exciting Trajectories

As indicated by [13], the sensitivity of the least squares solution to noise error is dependent on the condition number of the excitation matrix. Given the linear system $Ax = b$, the condition number is defined as follows:

$$\text{cond}(A) = \frac{\|A^{-1}e\|/\|A^{-1}b\|}{\|e\|/\|b\|},$$

(4.9)

where $e$ is the relative error in $\tau$. The condition number gives an idea of how sensitive the solution $\hat{x}$ will be to perturbations in $b$. The ideal situation is a condition number of 1, which indicates that a solution to the linear system can be found to an arbitrary precision.
If the operator norm in (4.9) is defined as the 2-norm, then the condition number reduces to the following:
\[
\text{cond}(A) = \frac{\sigma_{\text{max}}(A)}{\sigma_{\text{min}}(A)},
\]
where \(\sigma_{\text{max}}(A)\) is the largest singular value of \(A\), and \(\sigma_{\text{min}}(A)\) is the smallest. In [13], the condition number of the regressor matrix \(Y\) is examined:
\[
\text{cond}(Y) = \frac{\sigma_{\text{max}}(Y)}{\sigma_{\text{min}}(Y)}.
\]
It was demonstrated in [13] that minimizing the condition number of \(Y\) minimizes the noise sensitivity of the estimated parameters \(\hat{\chi}\).

Using a sequential set of trajectories, as proposed in [16], is the method that this work will implement. This paradigm was chosen because trajectory generators, such as the one in [14], are not guaranteed to create a trajectory that the overloaded manipulator can track. The trajectory may contain points in which the torque constraints of the manipulator are violated. By having this set of manually defined trajectories, it is easier to ensure they do not exceed the torque and position constraints.

In [16], Vandanjon defines four trajectories that excite different physical phenomena. These are the inertial effect, centrifugal coupling, inertial coupling, and gravity effect. In each trajectory, a limited number of joints will move at a time, to simplify the dynamic equations. The first three trajectories consist of periodic motions, but the fourth trajectory does not require periodicity. It is shown that these trajectories produce a very low condition number, while providing sufficient excitation of the regressor. Only the trajectories that excite the gravity and inertial effect are required to determine the inertial parameters of the end effector.

### 4.3 Implementation in Simulation

In this work, a simulation of the system was developed in the MATLAB computing environment, to facilitate the identification of an adequate set of exciting trajectories. The source code for the simulation can be found in Appendix A. For the purpose of simulation, the base parameters and the load parameters of the manipulator are considered known. This allows the program to determine the error, and thus the effectiveness of a particular trajectory. The theoretical load is added to the end effector using the parallel axis theorem, and the assumption is made that the load is cylindrical.
The first step in the MATLAB simulation is to generate a fifth order task space trajectory for the manipulator. This trajectory is then converted to joint space using inverse kinematics. The joint space trajectory is fed to the inverse dynamic equation, which will calculate torques for all sample time \( t \). The elements of the regressor \( Y \) are calculated from the trajectory, then “sampled” by augmenting newer samples in time to the bottom of the matrix. The same is done with the torque values.

At this point, the simulation can add a normally distributed, zero mean random value to the torque and position data to act as noise to approximate a real system. Once the sampling routine is finished, the left pseudoinverse of the regressor is found, and used to compute \( \hat{\chi} \), as per (4.8). The inertial parameters of the second link are then derived from \( \hat{\chi} \).
CHAPTER 5

Experimental Results and Discussion

5.1 Experimental Setup

The methodology proposed was experimentally verified on a 2 DOF manipulator driven by two Maxon RE40 150W motors. These are coupled with a Maxon GP52 gearing system with a gear ratio of 66:1. An encoder is directly attached to the motor, allowing direct measurement of the angular position and is differentiated to obtain angular velocity and acceleration. This encoder registers 500 pulses per motor revolution. The motor driver used in this system is configured in torque/current mode, and has the ability to output the actual current of the system.

A Pentium III 900 MHz computer running the QNX real-time operating system is used to perform the tracking controller implementation (computed torque control), angular position sensing, and velocity calculation from the joint positions. This system is capable of running with a 1 kHz sampling rate.

A C++ implementation of the identification algorithm was developed. This was done so that the identification process can be carried out on the system natively on QNX, as opposed to offloading the data to a different system for analysis. The source code for this program can be found in Appendix A.

The system was loaded with two different weights to identify: a 5 lb weight and 10 lb weight. Table 5.1 shows the inertial parameters of the system, which were calculated using CAD, measurement, and numerical techniques. These are the control values which we will compare the estimates against.
Table 5.1: Loaded Inertial Parameters of the CISCOR Manipulator

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unloaded</th>
<th>5 lb (2.268 kg) Load</th>
<th>10 lb (4.536 kg) Load</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_1 )</td>
<td>2.883</td>
<td>2.883</td>
<td>2.883</td>
<td>kg</td>
</tr>
<tr>
<td>( l_{c1} )</td>
<td>0.195</td>
<td>0.195</td>
<td>0.195</td>
<td>m</td>
</tr>
<tr>
<td>( I_1 )</td>
<td>0.0345</td>
<td>0.0345</td>
<td>0.0345</td>
<td>kg m^2</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>1.085</td>
<td>3.401</td>
<td>5.6209</td>
<td>kg</td>
</tr>
<tr>
<td>( l_{c2} )</td>
<td>0.22</td>
<td>0.2744</td>
<td>0.2849</td>
<td>m</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>0.013</td>
<td>0.0296</td>
<td>0.0435</td>
<td>kg m^2</td>
</tr>
</tbody>
</table>

5.1.1 Data Processing

To mitigate experimental noise, all sampled data is run through a low pass filter with a cutoff frequency of 50Hz. This eliminates the majority of the noise which occurs at frequencies 100Hz and above.

The efficiency of the motors and the gearing system is about 73%. This skews the torque data and introduces a large systematic error in the estimation. The torque data is scaled by a factor \( \alpha \), to reduce this error. This scaling also helps to mitigate error due to frictional effects. A baseline \( \alpha \) is found by minimizing the error when the system is loaded with a 5 lb weight. The parameter \( \alpha \) is then kept constant in all subsequent trajectories and load trials. This factor was found to be \( \alpha = 1.36 \), and application of this factor to the torque always produces better estimations from the least squares technique. All tabulated experimental results are the average of three runs.

5.2 Exciting Trajectories

5.2.1 Vertical Trajectory

The simplest trajectory is the one that excites the gravity effect [16]. This trajectory is required to move the load directly against gravity, and contains no periodic movement. A vertical trajectory was chosen as the most intuitive solution to this requirement, which moves the link along the Y axis, from fully extended to a destination above the initial point. An example of this trajectory can be seen in Figure 5.1.

To reduce this trajectory’s sensitivity to noise, the condition number must be minimized. The condition number was found to decrease steadily as the load traveled further and
higher. The lowest condition number feasible was 48.887, with the trajectory being \(-0.674 \leq y \leq -0.475\). This was the highest the manipulator can lift due to joint constraints, in that \(q_2\) cannot physically be much more than 90 degrees. Figure 5.2 details the specifics of the trajectory. Table 5.2 details the results of the identification algorithm. The experiment evinces that this trajectory produces an acceptable estimate for the mass and center of mass, but not for inertia.

The above trajectory is not feasible for the manipulator to complete with a 10 lb load. It should be noted that this identification technique should work up until the motors become saturated, where the commanded joint positions deviate from the actual joint positions. In this case, it is sufficient to truncate the sampled data once the motor saturation is evident. With this done, the results of the 10 lb trial are seen in Table 5.3.

Since the manipulator could not complete the above trajectory with the 10 lb load, a smaller trajectory was designed where \(-0.674 \leq y \leq -0.575\). Figure 5.3 details the specifics
of this trajectory. Since the manipulator travels less, the condition number is higher: 94.25, thus a less accurate estimate is expected. The results of this run with both 5 lb and 10 lb loads are found in Tables 5.4 and 5.5. With a heavy load bearing system, the ideal trajectory would be one where there would be minimal manipulator movement, while still accurately determining the inertial parameters. However, the opposite is indicated by the condition number; better results are achieved with more movement. Interestingly, comparable results are achieved with this trajectory.
Figure 5.3: Joint positions, velocities, and accelerations for trajectory $-0.674 \leq y \leq -0.575$.

Table 5.4: Estimation Results for Trajectory $-0.674 \leq y \leq -0.575$ with a 5 lb Load.

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>Estimate</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_2$ (kg)</td>
<td>3.401</td>
<td>3.6509</td>
<td>7.349</td>
</tr>
<tr>
<td>$l_{c2}$ (m)</td>
<td>0.2744</td>
<td>0.2849</td>
<td>3.8299</td>
</tr>
<tr>
<td>$I_2$ (kg m$^2$)</td>
<td>0.0296</td>
<td>-0.1777</td>
<td>700.6248</td>
</tr>
</tbody>
</table>

Table 5.5: Estimation Results for Trajectory $-0.674 \leq y \leq -0.575$ with a 10 lb Load.

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>Estimate</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_2$ (kg)</td>
<td>5.6209</td>
<td>5.8717</td>
<td>4.5036</td>
</tr>
<tr>
<td>$l_{c2}$ (m)</td>
<td>0.2849</td>
<td>0.2725</td>
<td>4.3258</td>
</tr>
<tr>
<td>$I_2$ (kg m$^2$)</td>
<td>0.0435</td>
<td>-0.3116</td>
<td>816.5</td>
</tr>
</tbody>
</table>
Figure 5.4: An example of a horizontal trajectory which excites the inertia effect.

Table 5.6: Estimation Results for Trajectory $-1 \leq x \leq 1, y = -0.575$ with a 5 lb Load.

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>Estimate</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_2$ (kg)</td>
<td>3.401</td>
<td>1.3471</td>
<td>60.3917</td>
</tr>
<tr>
<td>$l_{c2}$ (m)</td>
<td>0.2744</td>
<td>0.4561</td>
<td>66.2033</td>
</tr>
<tr>
<td>$I_2$ (kg m$^2$)</td>
<td>0.0296</td>
<td>-0.2878</td>
<td>1072.2378</td>
</tr>
</tbody>
</table>

5.2.2 Horizontal Trajectory

A horizontal trajectory was developed and tested. The first trajectory fixed the Y axis at $y = -0.575$, and moved the load from $-1 \leq x \leq 1$. The second trajectory held the same Y value, but moved the load from $-2 \leq x \leq 2$. Figures 5.5 and 5.6 detail the joint positions, velocities, and accelerations for these trajectories respectively. An example of this trajectory can be seen in Figure 5.4. The same trend with the condition number occurs here also: the further the manipulator moves, the lower the condition number. The condition number of the first trajectory is 94.9405, and for the second trajectory it is 22.3383. Since the second condition number is much lower than the condition numbers for the vertical trajectory, it is expected to perform much better in experiment. Indeed, this trajectory produces the best results in simulation. However, this is not the case, as seen in Tables 5.6 - 5.9.
Figure 5.5: Joint positions, velocities, and accelerations for trajectory $-1 \leq x \leq 1$, $y = -0.575$.

Table 5.7: Estimation Results for Trajectory $-1 \leq x \leq 1$, $y = -0.575$ with a 10 lb Load.

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>Estimate</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_2$ (kg)</td>
<td>5.6209</td>
<td>2.8254</td>
<td>49.7332</td>
</tr>
<tr>
<td>$l_{c2}$ (m)</td>
<td>0.2849</td>
<td>0.3601</td>
<td>26.3964</td>
</tr>
<tr>
<td>$I_2$ (kg m²)</td>
<td>0.0435</td>
<td>-0.4229</td>
<td>1072.2913</td>
</tr>
</tbody>
</table>

Table 5.8: Estimation Results for Trajectory $-2 \leq x \leq 2$, $y = -0.575$ with a 5 lb Load.

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>Estimate</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_2$ (kg)</td>
<td>3.401</td>
<td>0.7481</td>
<td>76.3147</td>
</tr>
<tr>
<td>$l_{c2}$ (m)</td>
<td>0.2744</td>
<td>0.6383</td>
<td>132.6110</td>
</tr>
<tr>
<td>$I_2$ (kg m²)</td>
<td>0.0296</td>
<td>-0.1292</td>
<td>536.5363</td>
</tr>
</tbody>
</table>
Figure 5.6: Joint positions, velocities, and accelerations for trajectory \(-0.2 \leq x \leq 0.2, y = -0.575\).

Table 5.9: Estimation Results for Trajectory \(-0.2 \leq x \leq 0.2, y = -0.575\) with a 10 lb Load.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Control</th>
<th>Estimate</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(m_2) (kg)</td>
<td>5.6209</td>
<td>1.4776</td>
<td>65.1711</td>
</tr>
<tr>
<td>(l_2) (m)</td>
<td>0.2849</td>
<td>0.4255</td>
<td>49.3494</td>
</tr>
<tr>
<td>(I_2) (kg m(^2))</td>
<td>0.0435</td>
<td>-0.2200</td>
<td>605.6410</td>
</tr>
</tbody>
</table>

5.2.3 Single Link Trajectory

To excite the inertia of the payload, a trajectory was developed which isolates and moves the second link only, from \(0 \leq q_2 \leq \frac{\pi}{6}\). An example can be seen in Figure 5.7, with the trajectory specifics in Figure 5.8. However, this causes the regressor matrix, \(Y\), to become rank deficient. This means that not all columns of \(Y\) are linearly independent, which makes calculation of the pseudoinverse impossible.

This can be shown by taking the singular value decomposition of the augmented observer
Figure 5.7: An example of a trajectory which only moves the second link.

Figure 5.8: Joint positions, velocities, and accelerations for trajectory $0 \leq q_2 \leq \frac{\pi}{6}$. 
matrix. When the regressor was filled with a simulated trajectory with no noise, there are two zero singular values. This indicates that the condition number is infinite, and that only three of the five columns are linearly independent. With a sampled trajectory from experimental data, it can be shown numerically that the pseudoinverse will produce poor results, due to $\sigma_{\text{min}}$ being close to zero, and $\sigma_{\text{max}}$ being much larger. This results in a very large condition number. All of these results indicate that this trajectory is non-ideal for identifying the parameters when used by itself. This trajectory produces some of the best measurements for the inertia $I_2$, albeit with more than 100% error. These results are tabulated in Tables 5.10 and 5.11.

It is possible to take a sub-matrix of the regressor that corresponds to the linearly independent columns, specifically columns 2, 3, and 5. First, the columns are rearranged as to place the two zero columns in the front of the matrix $Y$, and to consider the nonzero columns as another matrix:

$$
\begin{bmatrix}
0 & 0 \\
\vdots & \vdots \\
\end{bmatrix}
\begin{bmatrix}
\chi_1 \\
\chi_4 \\
\chi_2 \\
\chi_3 \\
\chi_5
\end{bmatrix}
= \tau
$$

(5.1)

This sub-matrix will be denoted as $W$, which can now be used to solve for $\chi_2, \chi_3$, and $\chi_5$ using the same least squares technique as outlined above:
Table 5.12: Inertia Estimation Results for Trajectory $0 \leq q_2 \leq \frac{\pi}{6}$ using reduced regressor.

<table>
<thead>
<tr>
<th>Load (lb)</th>
<th>Estimated I (kg m$^2$)</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-0.3053</td>
<td>1131.2667</td>
</tr>
<tr>
<td>10</td>
<td>-0.5225</td>
<td>1301.2333</td>
</tr>
</tbody>
</table>

$$W \begin{bmatrix} \hat{\chi}_2 \\ \hat{\chi}_3 \\ \hat{\chi}_5 \end{bmatrix} = \tau,$$

$$\begin{bmatrix} \hat{\chi}_2 \\ \hat{\chi}_3 \\ \hat{\chi}_5 \end{bmatrix} = (W^TW)^{-1}W^T\tau. \quad (5.2)$$

Due to the elements of $\chi$ being nonlinear combinations of the inertial parameters as seen in (4.4), the values of $m_2$, $l_{c2}$, and $I_2$ cannot be determined individually. However, if the mass $m_2$ and $l_{c2}$ are estimated using a different trajectory, then it is possible to use these results to solve for the inertia. These results are tabulated in Table 5.12. When analyzing the condition number for $W$, there was a very small increase as a function of the total angle traveled. However, this difference was only 0.12.

### 5.2.4 Oscillating Trajectory

An oscillating trajectory, that oscillates the second link between $-\frac{\pi}{6} \leq q_2 \leq \frac{\pi}{6}$ for two periods over 6 seconds was developed. A second trajectory where $-\frac{\pi}{12} \leq q_2 \leq \frac{\pi}{12}$ for two periods over 6 seconds was also tested. Figures 5.9 and 5.10 contain the specifics of the trajectory. In [16], the trajectory that excites inertia is defined as one that has periodic motion between two points, and that moves a single link. However, since it only utilizes the second link, it exhibits the same theoretical rank deficiency as seen in the second link trajectory above. These results are found in Tables 5.13, 5.14, 5.15, and 5.16. The oscillating trajectory where $-\frac{\pi}{12} \leq q_2 \leq \frac{\pi}{12}$ produced the best overall estimation of the inertia.

The condition number of the reduced regressor $W$ decreased as the period increased, indicating that the slower arm oscillates, the better the results. The inertia calculations can be seen in Table 5.17 and 5.18.
Figure 5.9: Joint positions, velocities, and accelerations for oscillating trajectory $-\frac{\pi}{12} \leq q_2 \leq \frac{\pi}{12}$.

Table 5.13: Estimation Results for Oscillating Trajectory $-\frac{\pi}{6} \leq q_2 \leq \frac{\pi}{6}$ with a 5 lb Load.

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>Estimate</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_2$ (kg)</td>
<td>3.401</td>
<td>-259.333</td>
<td>7725.200</td>
</tr>
<tr>
<td>$l_{c_2}$ (m)</td>
<td>0.2744</td>
<td>-0.0015</td>
<td>100.5485</td>
</tr>
<tr>
<td>$I_2$ (kg m$^2$)</td>
<td>0.0296</td>
<td>0.1112</td>
<td>275.7088</td>
</tr>
</tbody>
</table>

Table 5.14: Estimation Results for Trajectory $-\frac{\pi}{6} \leq q_2 \leq \frac{\pi}{6}$ with a 10 lb Load.

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>Estimate</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_2$ (kg)</td>
<td>5.6209</td>
<td>-240.742</td>
<td>4383.000</td>
</tr>
<tr>
<td>$l_{c_2}$ (m)</td>
<td>0.2849</td>
<td>-0.0020</td>
<td>100.7161</td>
</tr>
<tr>
<td>$I_2$ (kg m$^2$)</td>
<td>0.0435</td>
<td>0.1050</td>
<td>141.3387</td>
</tr>
</tbody>
</table>
Figure 5.10: Joint positions, velocities, and accelerations for oscillating trajectory $-\frac{\pi}{6} \leq q_2 \leq \frac{\pi}{6}$.

Table 5.15: Estimation Results for Oscillating Trajectory $-\frac{\pi}{12} \leq q_2 \leq \frac{\pi}{12}$ with a 5 lb Load.

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>Estimate</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_2$ (kg)</td>
<td>3.401</td>
<td>-244.7933</td>
<td>7297.7000</td>
</tr>
<tr>
<td>$l_c$ (m)</td>
<td>0.2744</td>
<td>-0.0019</td>
<td>100.6856</td>
</tr>
<tr>
<td>$I_2$ (kg m$^2$)</td>
<td>0.0296</td>
<td>0.0723</td>
<td>144.0913</td>
</tr>
</tbody>
</table>

Table 5.16: Estimation Results for Trajectory $-\frac{\pi}{12} \leq q_2 \leq \frac{\pi}{12}$ with a 10 lb Load.

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>Estimate</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_2$ (kg)</td>
<td>5.6209</td>
<td>-200.7191</td>
<td>3670.9500</td>
</tr>
<tr>
<td>$l_c$ (m)</td>
<td>0.2849</td>
<td>-0.0032</td>
<td>101.1221</td>
</tr>
<tr>
<td>$I_2$ (kg m$^2$)</td>
<td>0.0435</td>
<td>0.0680</td>
<td>56.1823</td>
</tr>
</tbody>
</table>
Table 5.17: Inertia Estimation Results for Trajectory $-\frac{\pi}{12} \leq q_2 \leq \frac{\pi}{12}$ using reduced regressor.

<table>
<thead>
<tr>
<th>Load (lb)</th>
<th>Estimated I (kg m$^2$)</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-0.1588</td>
<td>636.4982</td>
</tr>
<tr>
<td>10</td>
<td>-0.3370</td>
<td>874.6887</td>
</tr>
</tbody>
</table>

Table 5.18: Inertia Estimation Results for Trajectory $-\frac{\pi}{6} \leq q_2 \leq \frac{\pi}{6}$ using reduced regressor.

<table>
<thead>
<tr>
<th>Load (lb)</th>
<th>Estimated I (kg m$^2$)</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-0.1140</td>
<td>484.9580</td>
</tr>
<tr>
<td>10</td>
<td>-0.2865</td>
<td>758.5419</td>
</tr>
</tbody>
</table>

5.2.5 Exciting Trajectory Summary

Of all trajectories, the vertical trajectories produce the best estimates for the mass and center of mass. These are the most important values to identify since they have the largest effect on the dynamic model. The horizontal trajectory did not seem to excite the inertia effect as much as desired. Although it performed the best in simulations, in application it performed much worse than all other trajectories. This may be due to a low excitation of the gravity dependent terms in the inverse dynamic model.

Using the same scaling factor for the trajectories that only utilized the second link produced higher error in the estimates. Since the gearing system at this link is different from the principal link, the efficiency is different. Having $\alpha = 1.36$ may be the best tuning factor when both joints are considered, but when only using the second link, a different factor may be required. The best inertia estimates came from the oscillating trajectory between $-\frac{\pi}{12} \leq q_2 \leq \frac{\pi}{12}$. When loaded with 5 lb, the error was 79%, but when loaded with 10 lb, the estimation error was only 14%. This indicates that a small oscillating trajectory will produce better estimates when the payload is increased.

The inertia tends to be an order of magnitude less than either the mass or the center of mass, causing only a small effect on the torque. The torque value is mostly influenced by the mass term. These factors make the identification of the inertia very difficult. In addition, the simulations indicate that the inertia is the most sensitive parameter to noise in the data. This is evident in that most of the trajectories estimated the inertia with over 100 percent error.
error, or even estimated a negative inertia. Using the reduced matrix $W$ technique and moving only the second link did not produce good estimations, since it continually underestimated the inertial parameter vector. This leads to the negative inertia estimates seen in tables 5.12 - 5.18. The difficulty in estimating the inertia of the system is a product of the noise present when sampling. If the relative standard deviation of the noise is greater than the ratio of the inertia to $\chi_3$, then the value of the inertia is lost in the noise.
CHAPTER 6

Conclusion

A method for identification of the inertial parameters of the payload was presented in this research. The inverse dynamic model was employed with a least squares approximation to produce estimates of the inertial parameters based on measured torque values. Using the difference between the inertial parameters of the unloaded system and the loaded system, the inertial parameters of the payload were estimated. It was determined that the set of adequate exciting trajectories are a vertical trajectory to determine the mass and center of mass, and an oscillating trajectory to determine the inertia. These trajectories produce the best estimates for the inertial parameters overall. However, determining the inertia is difficult due to it being an order of magnitude smaller than the other inertial parameters. Thus, the inertia has a small effect on the torque compared to the mass and center of mass. As a result, identifying the mass and center of mass can be prioritized over identifying the inertia. Even if there is error in the estimated inertia, SBMPO can still produce a feasible trajectory for the manipulator if good estimates of the mass and center of mass are provided. The result of this research will be used in conjunction with SBMPO to perform estimation of the payload’s inertial parameters for small and lightweight manipulators.

6.1 Future Work

It would be beneficial to determine a trajectory that will produce the best estimations of inertia. This may require a more complex trajectory, and it may be of use to examine exciting trajectory generators as seen in [14]. However, these generators would have to be modified to ensure production of a feasible trajectory. One route to examine is the use of a force sensor connected to the end effector. By using this equipment, better estimates of the inertia may be obtained.
Appendices
APPENDIX A

Matlab Simulation Code

```matlab
%%Author: Ryan – David Reyes
%%Simulation of the 2 DOF manipulator
%%
%%
%% Physical parameters of the CISCOR robot arm. (m, kg, s)

%% Length of links
l1 = 0.375;
l2 = 0.3;

%% Distance from origin of each link to center of mass
lc1 = 0.195;
lc2_base = 0.22;

%% Mass of links
m1 = 2.883;
m2_base = 1.083;

%% Inertia relative to center of mass
I1 = 0.0345;
I2_base = 0.013;

%% Gravity term
g = 9.81;

%% Load parameters
l2 = 0.0435;
lc2 = 0.2849;
m2 = 4.732;
l2 = 0.0296;
lc2 = 0.2744;
m2 = 3.401;

%%
%% Trajectory specifications
x_initial = -.1;
x_final = .1;
x_velocity_initial = 0;
```
x.velocity_final = 0;
x.acceleration_initial = 0;
x.acceleration_final = 0;

y_initial = -.575;
y_final = -.575;
y.velocity_initial = 0;
y.velocity_final = 0;
y.acceleration_initial = 0;
y.acceleration_final = 0;

t_0 = 0;
t_final = 2;
delta_t = 0.001;
time = t_0 : delta_t : 2;
num_samples = 2000; % this is the number of samples across the whole trajectory

%%%
% Generate the polynomial coefficients of the trajectory
x.coefficients = GenerateTaskTrajectoryCoeff(t_0, ...
                t_final, ...
                x_initial, ...
                x.velocity_initial, ...
                x.acceleration_initial, ...
                x_final, ...
                x.velocity_final, ...
                x.acceleration_final);

y.coefficients = GenerateTaskTrajectoryCoeff(t_0, ...
                t_final, ...
                y_initial, ...
                y.velocity_initial, ...
                y.acceleration_initial, ...
                y_final, ...
                y.velocity_final, ...
                y.acceleration_final);

%%%
% Generate Task Space Trajectories
x_trajectory = GenerateTaskTrajectory(time, delta_t, x.coefficients);
y_trajectory = GenerateTaskTrajectory(time, delta_t, y.coefficients);

%%%
% Generate Joint Space Trajectory
[ q1_trajectory, q1_velocity, q1_acceleration, ...
  q2_trajectory, q2_velocity, q2_acceleration] = ...
  GenerateJointTrajectory(x_trajectory, y_trajectory, delta_t, l1, l2);

%calculate joint torques over time
torque.1 = (m1 * lc1^2 + m2 * l1^2 + m2 * lc2^2 ... + 2 * m2 * lc2 * cos(q2_trajectory) + l1 + l2) .* q1_acceleration ...
  + (m2 * lc2^2 + m2 * l1 * lc2 * cos(q2_trajectory) + l2) .* q2_acceleration ...
  - 2 * m2 * l1 * lc2 * sin(q2_trajectory) .* q1_velocity .* q2_velocity ...
  - m2 * l1 * lc2 * sin(q2_trajectory) .* q1_velocity .*^2 ...
  + (m1 * lc1 + m2 * l1) * g * sin(q1_trajectory) ...
  + m2 * g * lc2 * sin(q1_trajectory + q2_trajectory);

torque.2 = (m2 * lc2^2 + m2 * l1 * lc2 * cos(q2_trajectory) + l2) .* q1_acceleration ...
  + (m2 * lc2^2 + l2) * q2_acceleration ...
  + m2 * l1 * lc2 * sin(q2_trajectory) .* q1_velocity .* 2 ...
  + m2 * g * lc2 * sin(q1_trajectory + q2_trajectory);

% calculate actual parameters
chi1 = m1 * lc1^2 + m2 * (l1^2 + lc2^2) + l1 + l2;
chi2 = m2 * l1 * lc2;
chi3 = m2 * lc2^2 + l2;
chi4 = m1 * lc1 + m2*l1;
chi5 = m2 * lc2;
chi = [chi1 ; chi2 ; chi3 ; chi4 ; chi5];
Y = double.empty;
torque = double.empty;

% sample and fill the regressor
for n = 1:length(time)
    q1_accel_n = q1_acceleration(n);
    q1_veloc_n = q1_velocity(n);
    q1_traj_n = q1_trajectory(n);
    q2_accel_n = q2_acceleration(n);
    q2_veloc_n = q2_velocity(n);
    q2_traj_n = q2_trajectory(n);
    Y.12n = cos(q2_traj_n)*(2*q1_accel_n + q2_accel_n) - sin(q2_traj_n)*(
        q1_veloc_n^2 + 2*q1_veloc_n*q2_veloc_n);
    Y.22n = cos(q2_traj_n)*q1_accel_n + sin(q2_traj_n)*(q1_veloc_n^2);
    Y_n = [q1_accel_n, Y.12n, q2_accel_n, g*sin(q1_traj_n), g*
        sin(q1_traj_n + q2_traj_n); ...
    end

% calculate the estimated vector parameters
chihat = (transpose(Y) * Y) \ transpose(Y) * torque;
chi_percent_error = abs(chihat - chi) ./ chi * 100;
%calculate the estimated inertial parameters
m2_hat = (chihat(4)−m1*l1)/l1;
lc2_hat = chihat(5)/m2_hat;
l2_hat = chihat(3)−m2_hat*l2_hat^2;

m2_percent_error = abs(m2_hat − m2)./m2 * 100;
lc2_percent_error = abs(lc2_hat − lc2)./lc2 * 100;
l2_percent_error = abs(l2_hat − l2)./l2 * 100;

chi_results = [chi chihat chi_percent_error];
param_results = ...
    [m2 m2_hat m2_percent_error; ...]
    [lc2 lc2_hat lc2_percent_error; ...]
    [l2 l2_hat l2_percent_error];

printf_v2(chi_results, 'Chi Results', 'chi1 chi2 chi3 chi4 chi5', 'chi&
    chihat& chi.%error', '&')
printf_v2(param_results, 'Parameter Results', 'mass ctr_of_mass inertia', 'orig&
    calc& %error', '&')

C++ Identification Code

/*
   Author: Ryan - David Reyes
   This program implements the inertial parameter identification
   method in C++. Uses the Eigen Linear Algebra Library.
   */
#include <cstdio>
#include <vector>
#include <iostream>
#include <Eigen/Dense>

using namespace Eigen;

const float l1 = 0.375;
const float m1 = 2.883;
const float lc1 = 0.195;

struct regressor_data {
    float entry0;
    float entry1;
    float entry2;
    float entry3;
    float entry4;
} entries;

int main(int argc, char ** argv) {
    char * regressor_file_name = NULL;
    FILE * regressor_file = NULL;
}
char * torque_file_name = NULL;
FILE * torque_file = NULL;

unsigned long regressor_row_count = 0;
unsigned long torque_row_count = 0;

if (argc > 2) {
    regressor_file_name = argv[1];
    torque_file_name = argv[2];
}
else {
    printf("Usage: %s [ regressor_file ] [ torque_file]\n", argv[0]);
    return 0;
}

// open the regressor and torque data files.
regressor_file = fopen(regressor_file_name, "r");
torque_file = fopen(torque_file_name, "r");
if (regressor_file == NULL) {
    printf("%s not found!", regressor_file_name);
    return 0;
} else if (torque_file == NULL) {
    printf("%s not found!", torque_file_name);
    return 0;
}

// determine the number of lines in each file
while ( fscanf(regressor_file, "%f,%f,%f,%f,%f", &entries.entry0,
    &entries.entry1, &entries.entry2, &entries.entry3,
    &entries.entry4) != EOF)
    regressor_row_count++;
while ( fscanf(torque_file, "%f", &entries.entry0) != EOF)
    torque_row_count++;

if (torque_row_count != regressor_row_count) {
    printf("Number of rows do not match!\n");
    printf("Regressor rows: %ld\n", regressor_row_count);
    printf("Torque rows: %ld", torque_row_count);
    return 0;
}

rewind(regressor_file);
rewind(torque_file);

// declare the matrices
MatrixXf Y(regressor_row_count, 5);
VectorXf tau(torque_row_count);
// scan in the regressor into the matrix
for (unsigned long i = 0; i < regressor_row_count; ++i) {
    fscanf(regressor_file, "%f,%f,%f,%f,%f", &entries.entry0,
            &entries.entry1, &entries.entry2, &entries.entry3,
            &entries.entry4);
    Y(i, 0) = entries.entry0;
    Y(i, 1) = entries.entry1;
    Y(i, 2) = entries.entry2;
    Y(i, 3) = entries.entry3;
    Y(i, 4) = entries.entry4;
}

// scan in the torque samples into the matrix
for (unsigned long i = 0; i < torque_row_count; ++i) {
    fscanf(torque_file, "%f", &entries.entry0);
    tau(i) = entries.entry0;
}

// pseudo inverse
VectorXf chihat = Y.jacobiSvd(ComputeThinU | ComputeThinV).solve(tau);
float m2_hat = (chihat(3) - m1*l1c1)/l1;
float lc2_hat = chihat(4)/m2_hat;
float l2_hat = chihat(2) - m2_hat*pow(lc2_hat, 2);
std::cout << m2_hat << "\n";
std::cout << lc2_hat << "\n";
std::cout << l2_hat << "\n";
fclose(regressor_file);
fclose(torque_file);
return 0;
}
REFERENCES


