

## **Copyright Warning & Restrictions**

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

**Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation**

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

## **ABSTRACT**

### **REALIZATION OF DYNAMIXEL SERVO PLANT PARAMETERS TO IMPROVE ADMITTANCE CONTROL FOR A COMPLIANT HUMAN-ROBOT INTERACTION**

**by  
Ahmad Zahid Rao**

In theory, admittance control offers a very effective method of implementing smooth human-robot interaction. It allows the user's applied force to control the movement of a powerful robot as if the robot were a small, passive mass. However, the real-world application of admittance control faces limitation posed by the dynamics of servo motors, the accuracy of the force sensors, and the computation speed of processors.

This research investigates the limitations on achieving compliant passive behavior when using state-of-the-art actuators, sensors and processors. The work involves characterizing the dynamic behavior of the servo motors, development of improved differential equations representing admittance control, and testing to determine the ability of a robotic system to represent the behavior of passivity. A method has been developed for experimentally determining the inertial, and dissipative (damping and friction) characteristics of three different models of Dynamixel motors. These parameters are optimized using data from a pendulum drop test with mass at various distances from the center of rotation. With these parameters, we assess the ability of our motor model to generate an ideal motion based upon a torque input from the user. The aim is to understand the limitations of our control paradigm to allow users to be unable to feel any difference between the performances of the passive and motor joints.

**REALIZATION OF DYNAMIXEL SERVO PLANT PARAMETERS TO  
IMPROVE ADMITTANCE CONTROL FOR A COMPLIANT  
HUMAN-ROBOT INTERACTION**

**by  
Ahmad Zahid Rao**

**A Thesis  
Submitted to the Faculty of  
New Jersey Institute of Technology  
in Partial Fulfillment of the Requirements for the Degree of  
Master of Science in Biomedical Engineering**

**Department of Biomedical Engineering**

**January 2016**

Blank Page

**APPROVAL PAGE**

**REALIZATION OF DYNAMIXEL SERVO PLANT PARAMETERS TO  
IMPROVE ADMITTANCE CONTROL FOR A COMPLIANT  
HUMAN-ROBOT INTERACTION**

**Ahmad Zahid Rao**

---

Dr. Richard Foulds, Thesis Advisor Date  
Associate Professor of Biomedical Engineering, NJIT

---

Dr. Sergei Adamovich, Committee Member Date  
Associate Professor of Biomedical Engineering, NJIT

---

Dr. Antje Ihlefeld, Committee Member Date  
Assistant Professor of Biomedical Engineering, NJIT

## **BIOGRAPHICAL SKETCH**

**Author:** Ahmad Zahid Rao

**Degree:** Master of Science

**Date:** January 2016

### **Undergraduate and Graduate Education:**

- Master of Science in Biomedical Engineering,  
New Jersey Institute of Technology, Newark, NJ, 2016
- Bachelor of Engineering in Biomedical Engineering,  
NED University of Engineering and Technology, Karachi, Pakistan, 2014

**Major:** Biomedical Engineering





## ACKNOWLEDGMENT

I would like to give my deepest regards to Dr. Richard Foulds who has been very generous with guidance, motivation and insight throughout the thesis work. I would not have been able to finish this work on time, if it was not for his keen interest in the work and his willingness to dedicate time for discussions. I would also like to express gratitude to Dr. Sergei Adamovich and Dr. Antje Ihlefeld for their encouragement and time to time advice on the topics regarding this work.

Many obstacles that I faced during this thesis would not have been easy and quick to overcome had it not been the kind help of friends, fellow lab members and teacher. To name a few, I remember when Kiran Karunakaran helped me understand the ‘abnormal’ behavior of the motor and the sensors and when Kevin Abbruzzese helped me learn how to use the force sensor. I remember when Madeline Corrigan taught me how to level the bed of the 3D printer and when Peter Michael, Dr. Ghaith Androwis and Dr. George Mytalas helped me figure out the statistics involved in my work. I remember how John Hoinowske would always be there with all the resources I would need for the experimental setup. And I remember when Ahmad Alokaily and Ahmet Asan with all their sincerity would prompt me to write my thesis. I am obligated for all their help.

Last, but not least, I am extremely grateful to my parents who not only instilled in me the importance of higher education but also supported me in all the decisions and with all their resources. It was the continual care, concern and prayers of my parents and siblings’ which boosted my optimism and self-confidence and without whom I would not have been able to achieve this dream.

## TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION .....	1
1.1 Review of Literature .....	2
2 METHODOLOGY .....	8
2.1 Experimental Setup for Pendulum Test .....	8
2.2 Analysis of Pendulum Test Data.....	10
2.3 Experimental Setup for Friction Estimation .....	10
2.4 Analysis of Friction Estimation Data.....	12
2.5 Optimization of Inertial and Dissipative Parameters .....	13
3 RESULTS .....	17
3.1 Frictional Torque Estimation for Motors.....	17
3.2 Optimized Moment of Inertia, Damping and Friction of Motors .....	19
3.3 Testing the Results.....	24
3.4 Improved ODE for Implementing Admittance Control.....	25
4 DISCUSSION.....	27
APPENDIX A PLOTS FOR OPTIMIZATION .....	30
APPENDIX B PLOTS FOR TEST DATA .....	46
B.1 Testing Trials Plots for MX-28 .....	46
B.2 Testing Trials Plots for MX-64 .....	61

**TABLE OF CONTENTS**  
**(continued)**

<b>Chapter</b>	<b>Page</b>
B.3 Testing Trials Plots for MX-106 .....	76
APPENDIX C CODING SCRIPTS .....	92
C.1 Script for Friction Estimation.....	92
C.2 Script for Data Recording Synchronization .....	93
C.3 Script used for Optimization .....	95
C.4 Script used for Testing .....	96
C.5 Script used for ODE .....	98
REFERENCES .....	99

## LIST OF TABLES

<b>Table</b>		<b>Page</b>
3.1	Summarized results from frictional torque (Nm) estimation in counter clockwise (ccw) and clockwise (cw) directions.....	18
3.2	Summarized results from optimization for MX-28 with a movable mass at various locations .....	19
3.3	Summarized results from optimization for MX-64 with a movable mass at various locations .....	21
3.4	Summarized results from optimization for MX-106 with a movable mass at various locations .....	22

## LIST OF FIGURES

<b>Figure</b>		<b>Page</b>
2.1	3D printer attachments for the experimental setup .....	8
2.2	Pendulum drop test in progress .....	9
2.3	Setup for recording data for friction estimation .....	11
2.4	Synchronization of OptiTrack system using Arduino .....	12
2.5	Free body diagram of the pendulum drop test .....	14
2.6	Simulink model for parameter optimization .....	15
3.1	Rotational friction estimation from a trial .....	17
3.2	Simulink model for testing .....	24
3.3	Step, hyperbolic tangent and a coefficient times hyperbolic function .....	26
A.1	MX-28 trials with mass at 40 cm .....	31
A.2	MX-28 trials with mass at 35 cm .....	32
A.3	MX-28 trials with mass at 30 cm .....	33
A.4	MX-28 trials with mass at 25 cm .....	34
A.5	MX-28 trials with mass at 20 cm .....	35
A.6	MX-64 trials with mass at 40 cm .....	36
A.7	MX-64 trials with mass at 35 cm .....	37
A.8	MX-64 trials with mass at 30 cm .....	38
A.9	MX-64 trials with mass at 25 cm .....	39
A.10	MX-64 trials with mass at 20 cm .....	40
A.11	MX-106 trials with mass at 40 cm .....	41
A.12	MX-106 trials with mass at 35 cm .....	42

**LIST OF FIGURES**  
(continued)

<b>Figure</b>	<b>Page</b>
A.13 MX-106 trials with mass at 30 cm.....	43
A.14 MX-106 trials with mass at 25 cm.....	44
A.15 MX-106 trials with mass at 20 cm.....	45
B.1 MX-28 slow to fast trial 1, measured position and torque.....	46
B.2 MX-28 slow to fast trial 1, measured and calculated torque .....	47
B.3 MX-28 slow to fast trial 1, recorded and predicted position .....	47
B.4 MX-28 slow to fast trial 2, measured position and torque.....	48
B.5 MX-28 slow to fast trial 2, measured and calculated torque .....	48
B.6 MX-28 slow to fast trial 2, recorded and predicted position .....	49
B.7 MX-28 fast to slow trial 1, measured position and torque.....	49
B.8 MX-28 fast to slow trial 1, measured and calculated torque .....	50
B.9 MX-28 fast to slow trial 1, recorded and predicted position .....	50
B.10 MX-28 fast to slow trial 2, measured position and torque.....	51
B.11 MX-28 fast to slow trial 2, measured and calculated torque .....	51
B.12 MX-28 fast to slow trial 2, recorded and predicted position .....	52
B.13 MX-28 fast trial 1, measured position and torque .....	52
B.14 MX-28 fast trial 1, measured and calculated torque .....	53
B.15 MX-28 fast trial 1, recorded and predicted position .....	53
B.16 MX-28 fast trial 2, measured position and torque .....	54
B.17 MX-28 fast trial 2, measured and calculated torque .....	54

**LIST OF FIGURES**  
**(continued)**

<b>Figure</b>	<b>Page</b>
B.18 MX-28 fast trial 2, recorded and predicted position .....	55
B.19 MX-28 normal trial 1, measured position and torque.....	55
B.20 MX-28 normal trial 1, measured and calculated torque .....	56
B.21 MX-28 normal trial 1, recorded and predicted position .....	56
B.22 MX-28 normal trial 2, measured position and torque.....	57
B.23 MX-28 normal trial 2, measured and calculated torque .....	57
B.24 MX-28 normal trial 2, recorded and predicted position .....	58
B.25 MX-28 slow trial 1, measured position and torque.....	58
B.26 MX-28 slow trial 1, measured and calculated torque .....	59
B.27 MX-28 slow trial 1, recorded and predicted position .....	59
B.28 MX-28 slow trial 2, measured position and torque.....	60
B.29 MX-28 slow trial 2, measured and calculated torque .....	60
B.30 MX-28 slow trial 2, recorded and predicted position .....	61
B.31 MX-64 slow to fast trial 1, measured position and torque.....	61
B.32 MX-64 slow to fast trial 1, measured and calculated torque .....	62
B.33 MX-64 slow to fast trial 1, recorded and predicted position .....	62
B.34 MX-64 slow to fast trial 2, measured position and torque.....	63
B.35 MX-64 slow to fast trial 2, measured and calculated torque .....	63
B.36 MX-64 slow to fast trial 2, recorded and predicted position .....	64
B.37 MX-64 fast to slow trial 1, measured position and torque.....	64

**LIST OF FIGURES**  
(continued)

<b>Figure</b>	<b>Page</b>
B.38 MX-64 fast to slow trial 1, measured and calculated torque .....	65
B.39 MX-64 fast to slow trial 1, recorded and predicted position .....	65
B.40 MX-64 fast to slow trial 2, measured position and torque.....	66
B.41 MX-64 fast to slow trial 2, measured and calculated torque .....	66
B.42 MX-64 fast to slow trial 2, recorded and predicted position .....	67
B.43 MX-64 fast trial 1, measured position and torque .....	67
B.44 MX-64 fast trial 1, measured and calculated torque .....	68
B.45 MX-64 fast trial 1, recorded and predicted position .....	68
B.46 MX-64 fast trial 2, measured position and torque .....	69
B.47 MX-64 fast trial 2, measured and calculated torque .....	69
B.48 MX-64 fast trial 2, recorded and predicted position .....	70
B.49 MX-64 normal trial 1, measured position and torque.....	70
B.50 MX-64 normal trial 1, measured and calculated torque .....	71
B.51 MX-64 normal trial 1, recorded and predicted position .....	71
B.52 MX-64 normal trial 2, measured position and torque.....	72
B.53 MX-64 normal trial 2, measured and calculated torque .....	72
B.54 MX-64 normal trial 2, recorded and predicted position .....	73
B.55 MX-64 slow trial 1, measured position and torque.....	73
B.56 MX-64 slow trial 1, measured and calculated torque .....	74
B.57 MX-64 slow trial 1, recorded and predicted position .....	74



**LIST OF FIGURES**  
**(continued)**

<b>Figure</b>	<b>Page</b>
B.58 MX-64 slow trial 2, measured position and torque.....	75
B.59 MX-64 slow trial 2, measured and calculated torque .....	75
B.60 MX-64 slow trial 2, recorded and predicted position .....	76
B.61 MX-106 slow to fast trial 1, measured position and torque.....	76
B.62 MX-106 slow to fast trial 1, measured and calculated torque .....	77
B.63 MX-106 slow to fast trial 1, recorded and predicted position .....	77
B.64 MX-106 slow to fast trial 2, measured position and torque.....	78
B.65 MX-106 slow to fast trial 2, measured and calculated torque .....	78
B.66 MX-106 slow to fast trial 2, recorded and predicted position .....	79
B.67 MX-106 fast to slow trial 1, measured position and torque.....	79
B.68 MX-106 fast to slow trial 1, measured and calculated torque .....	80
B.69 MX-106 fast to slow trial 1, recorded and predicted position .....	80
B.70 MX-106 fast to slow trial 2, measured position and torque.....	81
B.71 MX-106 fast to slow trial 2, measured and calculated torque .....	81
B.72 MX-106 fast to slow trial 2, recorded and predicted position .....	82
B.73 MX-106 fast trial 1, measured position and torque .....	82
B.74 MX-106 fast trial 1, measured and calculated torque .....	83
B.75 MX-106 fast trial 1, recorded and predicted position .....	83
B.76 MX-106 fast trial 2, measured position and torque .....	84
B.77 MX-106 fast trial 2, measured and calculated torque .....	84

**LIST OF FIGURES**  
**(continued)**

<b>Figure</b>	<b>Page</b>
B.78 MX-106 fast trial 2, recorded and predicted position .....	85
B.79 MX-106 normal trial 1, measured position and torque.....	85
B.80 MX-106 normal trial 1, measured and calculated torque .....	86
B.81 MX-106 normal trial 1, recorded and predicted position .....	86
B.82 MX-106 normal trial 2, measured position and torque.....	87
B.83 MX-106 normal trial 2, measured and calculated torque .....	87
B.84 MX-106 normal trial 2, recorded and predicted position .....	88
B.85 MX-106 slow trial 1, measured position and torque.....	88
B.86 MX-106 slow trial 1, measured and calculated torque .....	89
B.87 MX-106 slow trial 1, recorded and predicted position .....	89
B.88 MX-106 slow trial 2, measured position and torque.....	90
B.89 MX-106 slow trial 2, measured and calculated torque .....	90
B.90 MX-106 slow trial 2, recorded and predicted position .....	91

# CHAPTER 1

## INTRODUCTION

Among the two most common Human-Robot Interaction control paradigms are admittance and impedance control which develop a relationship between the user's input and the robot's output. Admittance control uses the user's applied force to drive the robot [1]. When the user applies a force to the robot, the robot decides how much it should move, in what direction and with what velocity? Newton's second law of motion applies here.

$$\text{Force} = \text{mass} \times \text{acceleration} \quad (1.1)$$

Double integration of the acceleration gets us the position which should be reached with the applied force. The robots are usually big and bulky, requiring a considerable amount of force to make them move. However, instead of putting the actual mass of the robot, a small virtual frictionless mass is put in the equation and the controller calculates the position for that frictionless mass in response to the applied force. The robot is then commanded to reach there. Thus, this allows the users to operate bulky robots with very little force, which is a core advantage of admittance control.

The compliance of an admittance controlled robot, however, is limited by factors such as the characteristics of the actuators, the behavior of the sensors and the computational processing speed of the controllers. A lack of full understanding of the dynamics involved leads to inclusion of unwanted stiffness and oscillation in the system [2].

In this thesis, these limitations have been explored. The characteristic parameters of the servo motor system are optimized to make the admittance control using these

servos as smooth as possible. To do this, an experimental setup is designed to apply the pendulum drop test to the Dynamixel motors, MX-28, MX-64 and MX-106. The analysis is then carried out to find the inertial and dissipative characteristics on these motors.

## **1.1 Review of Literature**

### **A. Parameter Estimation**

Characterization and modeling of some Dynamixel motors have been attempted in previous work. An attempt was made to characterize the Dynamixel AX-12+ servo [3]. The servo was physically opened to examine the parts inside to understand how it works. In addition, mechanical and electrical measurements were done on the servo and the responses to various inputs were recorded to obtain parameters such as torque and friction. These obtained estimated parameters and measured movement profiles were used to develop a simulation model of the motor. However, this work was incomplete to fully understand the behavior of the servo.

In another attempt, the researchers made a reference trajectory using Dynamixel MX-106 servo and used simple Iterative Learning Control method to find the servo commands that would produce a trajectory similar to the reference trajectory. They used a single model to estimate all the parameters together. To validate the developed model, an MX-106 servo based robot was tested for position control. The accuracy and energy efficiency were found to be satisfactory [2].

A number of researchers have worked to estimate the moment of inertia of various servo systems for various applications. Most researchers have used observer based algorithms to estimate these parameters.

Some use estimated parameters such as motor speed and torque disturbance as observers to estimate the moment of inertia, others use direct output from filters such as Kalman filter as the estimate of moment of inertia, and yet some others use mathematical models to estimate motor speed based on a moment of inertia and compare it with actual motor speed until both speeds are close. In one study, the researcher has proposed an algorithm which multiplies the motor position to the reference torque input to get an estimate of the moment of inertia [4]. Change of inertia observer and torque observer have also been used by some researchers to estimate the inertia of a DC motor. However simpler, the method was found to be more accurate for larger DC motors [5].

Induced torque harmonics have also been implemented in an attempt to determine the mechanical motor parameters such as mass moment of inertia and viscous damping in permanent magnet synchronous motor systems. These harmonics are chosen to be the ones that do not excite the motor in normal operation and thus could be used when the motor is actually operating. Motor's velocity profile is measured as a result of these torque harmonics which is then used to estimate the parameters. These parameters are then used to calculate the torque and compared to the actual torque read by the piezoelectric sensors to prove the accuracy of the proposed method [6].

In another study, a velocity controlled servo has been analyzed to estimate its inertia, viscous friction, Coulomb friction and a constant disturbance parameters using a two-step steady-state response method that employs a simple numerical model to estimate these parameters [7]. A brushless servo motor is then used to positively validate the model using step and sine-wave signals as the input. The results produced actually are comparable to that obtained by the standard recursive least squares method.

Some researchers have also used genetic algorithms, neural networks and other such computational intelligence procedures to identify the mechanical motor parameters. In one study [7], the authors used Levenberg-Marquardt algorithm for optimization of motor parameters. Predefined standard signals have been used for parameter estimation too. Friction identification in open loop have also been attempted by [7] by applying a ramp to the servo. In closed loop system, researchers estimated damping in addition to inertia in a reaction wheel using a curve-fitting technique on responses to step signals. In closed loop, Dahl model of a linear motion was employed to estimate friction too [7].

## **B. Dynamixel Servos**

The Dynamixel servos are being used widely in various research and development projects. They are robust, fast, and are easy to program and thus become a smart actuator for precise control. Their application ranges from educational uses to medical uses to industrial uses. Some of the recent research utilizing these actuators are discussed below.

Considering the importance of robotics education for undergraduate and graduate students, a Coin Handling Arm for Robotic Mastery (CHARM) robot is developed which is aimed at teaching robotics in a conceptual way [8]. A camera is used to identify coins of different types on a rotating table that also have other objects on it. Once the coins are identified, each type of coins are put into their specific bins by the arm which is driven by the Dynamixel servomotors. This set up provides an alternate approach to the traditional LEGO Mindstorms kits for teaching purposes overcoming its limitations, not only in robotics but also in other fields such as physics and mathematics.

Another work is also directed towards the development and control of a robotic arm for educational purpose. The open-source Python software is used to control the Dynamixel motors in the 6-degrees of freedom BRACON robot turning this robot into a much cheaper and reliable solution for its purpose as compared to other available alternates [9].

Many biomedical related researches have also been utilizing the Dynamixel servos. In one work, the researchers developed a motorized remote probe control method for transesophageal echocardiography [10]. The conventional manual probe control faces problems such as poor ergonomics, leading to musculoskeletal disorders in the operators and hazardous X-ray exposures to its operators. To overcome these issues a remote control system is developed that uses two Dynamixel MX-28 motors as actuators for the probe movement and utilizes the Fuzzy Logic Toolbox in MATLAB as the controller.

In another study, the researchers used the Dynamixel MX-106 smart servos to build a one DOF ankle prosthesis for the amputees that could provide flexion and dorsiflexion movements [11]. A fuzzy controller is used to control the position and torque of the motor depending upon the force inputs from the force sensors in the sole of the prosthesis. From this setup, they were able to obtain angular motion of the foot that was in the normal walking range.

The Shoulder-Elbow-Forearm Robotics Economic (SEFRE) rehabilitation system is aimed for use by patients that have no upper limb muscle force or whose muscle force is limited due to any reason [12]. The SEFRE system has a KUKA robotic manipulator to which is attached a Forearm Supportive Mechanism (FSM). This FSM utilizes two Dynamixel motors that are used for elbow and forearm movement. A 1:3 gearbox is

added to increase the torque bearing capability for the elbow motion. This, however, reduces the speed and range of motion of the Dynamixel motor. To overcome the range of motion issue, 'wheel mode' has to be used instead of 'joint mode'. A PID controller and an encoder algorithm is implemented to make the 'wheel mode' perform as a 'joint mode'. The FSM controller gets user's movement in the form of a Kalman filtered force data from a six-axis force sensor and drives the Dynamixel motors accordingly. Using the KUKA robot and the FSM, the intelligent controller is able to make efficient movements as concurrent and natural as a natural human motion.

One of the research attempts to enable a humanoid robot to imitate the human motion. Microsoft Kinect is used for human motion tracking [13]. Since no body markers are used, the OpenNI skeleton tracking software is used to identify the human joints. This whole body data is then realized to an 18 DOF robot that has Dynamixel AX-12 servos as the actuators. The researchers are able to produce good real-time imitation of a dance using this setup.

Another research involves the Dynamixel servos from the humanoid Bioloid kit from Robotis [14]. A webcam is used to capture the image of a free-hand drawing made by a human being. This image is then processed by curve fitting using the polygonal approximation which utilizes only 10% of the total points to reconstruct the image. Two AX-12 motors are used to build a robotic arm that would draw this reconstructed image. Their goal is to be able to use the whole humanoid robot to do this task in the future research work.

In another work the researchers set to classify the surface on which a snake-like robot is moving on [15]. They built a snake-like robot with 16 Dynamixel AX-12



actuators, each attached at 90 degrees from the previous actuator. For three different gait patterns, using the information from the torque and current sensors that are built in these actuators, the analysis was carried out by applying the support vector machines and the K-means in order to predict the surface on which locomotion is being carried upon.

In yet another work, the researchers developed an online VC++ based software to communicate with AVR microcontroller to control multiple Dynamixel MX-28 for a cockroach robot [16]. Since the cockroach robot has many degrees of freedom and it is required to have small size and weight while having high torque and strength, the MX-28 thus becomes ideal for use. Their experimental paradigm produced an accurate and efficient robot.

These selected works show the diverse applications for which the Dynamixel actuators are being used in the world. Their characteristics, ease of use and the capability to be controlled and feedback so many parameters makes these actuators a smart choice for researchers.

Keeping in view the vast use of Dynamixel motors for robotics applications, where good estimates of parameters is very crucial for the optimum operation and to avoid erroneous performance, the characterization and modeling of these servos would play a fundamental role for improving the performance of admittance control with these servos and achieving compliant human robot interaction.

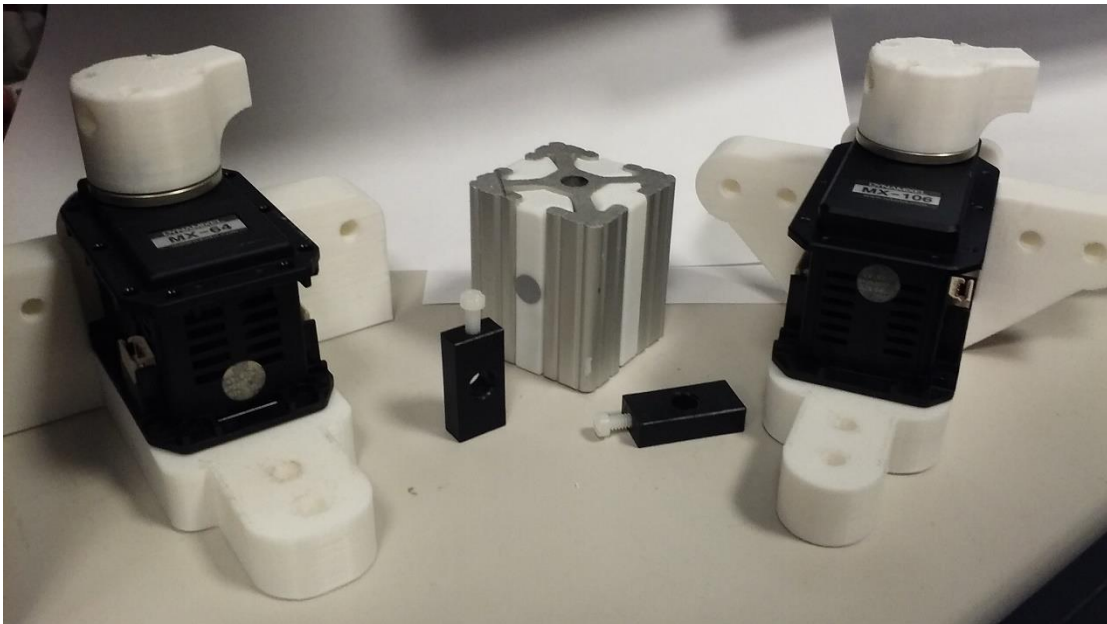
## CHAPTER 2

### METHODOLOGY

#### 2.1 Experimental Setup for Pendulum Test

The pendulum drop test is carried out with all the three motors, Dynamixel MX-28, MX-64 and MX-106. In our case the pendulum drop test involves attaching a mass to the motor at some distance such that the mass can fall freely under the gravity's influence in the vertical plane.

The base for each motor is designed such that it could mount the motor vertically. The head for each motor is designed such that it can hold a thin solid Aluminum rod for the movement with no backlash. The Aluminum rod has a length of 45.6 cm and is graduated at each 5 cm intervals starting at 20 cm from the proximal end which is at the center of rotation of the motor. It has a mass of 40.0 grams.



**Figure 2.1** 3D printer attachments for the experimental setup.

Attached to this rod is a moveable solid mass of 97 grams having a cubic geometry with a through-hole in the exact center, having the same diameter as of the rod, so that its center of mass is along the same axis as that of the center of mass of the rod. This mass can be moved up or down the rod very easily and can be held in place with small clips on its either side along the Aluminum rod.

The purpose of this moveable mass is to allow us to vary the gravitational torque during the pendulum test while having the mass of the exact same geometry for each trial. This negates the variations caused by the geometry of the mass from trial to trial. The mass is fixed at each distance, 20 cm, 25 cm, 30 cm, 35 cm and 40 cm, from the center of the rotation during the pendulum test and at each particular position, 6 pendulum drop trials are carried out, with 3 drops in clockwise direction and 3 drops in counter clockwise direction. This is repeated for each of the three motors.



**Figure 2.2** Pendulum drop test in progress.

To carry out the pendulum drop test, the free end of the aluminum rod is raised manually to about 90 degrees from the vertical and then released. The position data for this movement is recorded until the rod comes to a rest, using the OptiTrack Motion Capture System. One reflective marker is put at the center of rotation on the motor's head and the other is put at the center of mass of the fixed moveable mass. The cameras are put at a distance where the whole pendulum motion can be viewed. The camera system is then also calibrated for the axes before any recording is done so that it knows its orientation with respect to the 'pendulum'.

## **2.2 Analysis of Pendulum Test Data**

The pendulum drop test data recorded for each of the 90 trials is analyzed in MATLAB 2015b. The OptiTrack software, Motive, allows to save the data in .csv (comma separated variable) file and this file can easily be opened in MATLAB.

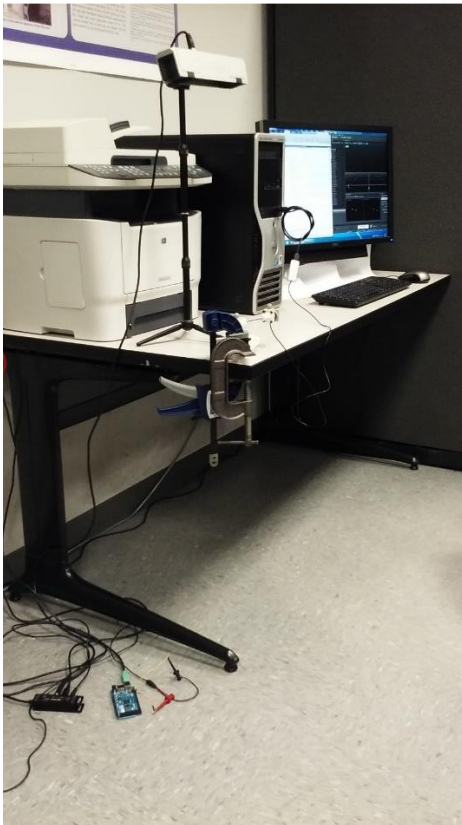
An imaginary line connecting the two reflective markers, representing the aluminum rod, is made and its angle from the vertical is computed. Since the pendulum is released from around an angle of around 90 degrees from the vertical, the data are checked for its rationality by making sure it starts from around that angle. As the pendulum motion continues, the pendulum should oscillates about the vertical, which is zero degrees angle. This change of angle data is later used for parameter estimation.

## **2.3 Experimental Setup for Friction Estimation**

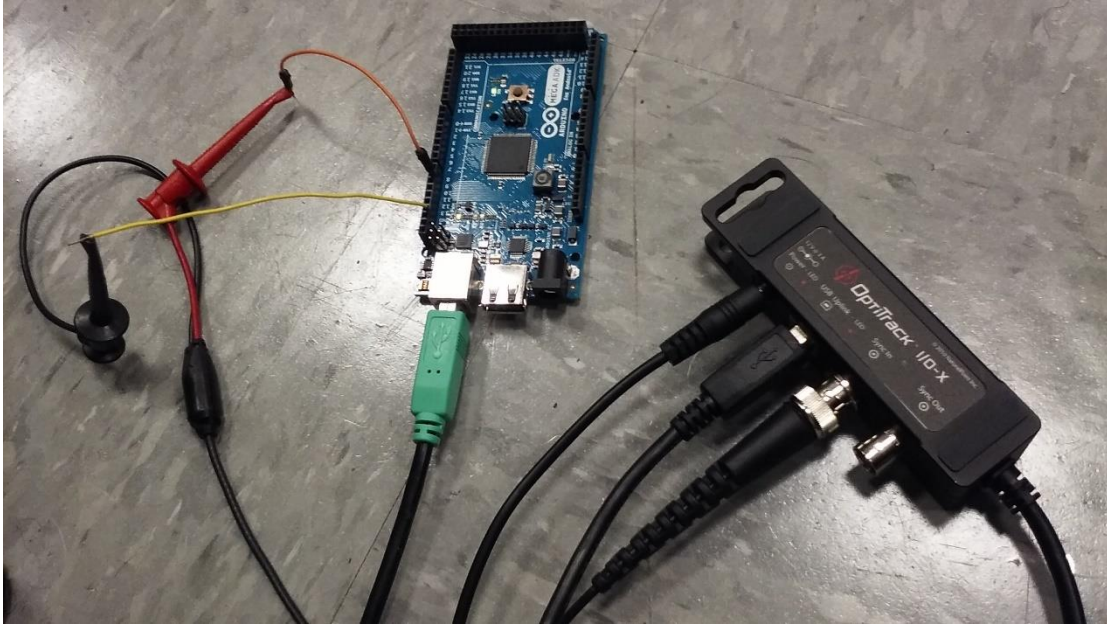
The friction in the motors needs to be known for the analysis. To get an estimate of the friction, a gradual force is applied to the free end of a small 10.5 cm long Aluminum rod,

whose one end is attached to the motor's head at the center of rotation. At the precise instance when friction is overcome by the applied force, the rod starts moving. This force, which just overcame friction is considered as an estimation for the friction.

To set up for recording data for this, the OptiTrack Motion Capture system is used again and is synchronized with a high speed OptoForce force sensor which is attached to the free end of the rod and it is this sensor where the gradual force is applied. To define the location of center of rotation and of the force sensor, one reflective marker is put at the center of rotation of the motor while another is put in the same plane at the base of the force sensor, respectively. Position and force data collection are both triggered and stopped at the same instance.



**Figure 2.3** Setup for recording data for friction estimation.



**Figure 2.4** Synchronization of OptiTrack system using Arduino.

Arduino Mega ADK board is used to send the sync commands from the PC to the OptiTrack. The ‘Legacy MATLAB and Simulink Support for Arduino’ is used for communication between MATLAB and Arduino.

Six trials with each motor are carried out, 3 applying the force to move the rod in clockwise direction and other 3 for counter clockwise direction.

## **2.4 Analysis of Friction Estimation Data**

The position data is saved in .csv file format and imported in MATLAB for analysis. The imaginary line connecting both the reflective markers, representing the small Aluminum rod, is computed and its angle is found with time.

The force data given by the force sensor is imported and calibrated using its sensitivity report provided by the manufacturer, to convert the counts to Newtons.

The force and position data are then plotted together, and the force, at the point in time when the position starts to change, is noted as the estimated friction.

This is done with each trial of each motor and a single value for an estimated friction for each motor is obtained by averaging the trials for that motor. This friction is then multiplied with the distance, from the center of the rotation to the force sensor, to get the frictional torque having units of Newton-meter.

## 2.5 Optimization of Inertial and Dissipative Parameters

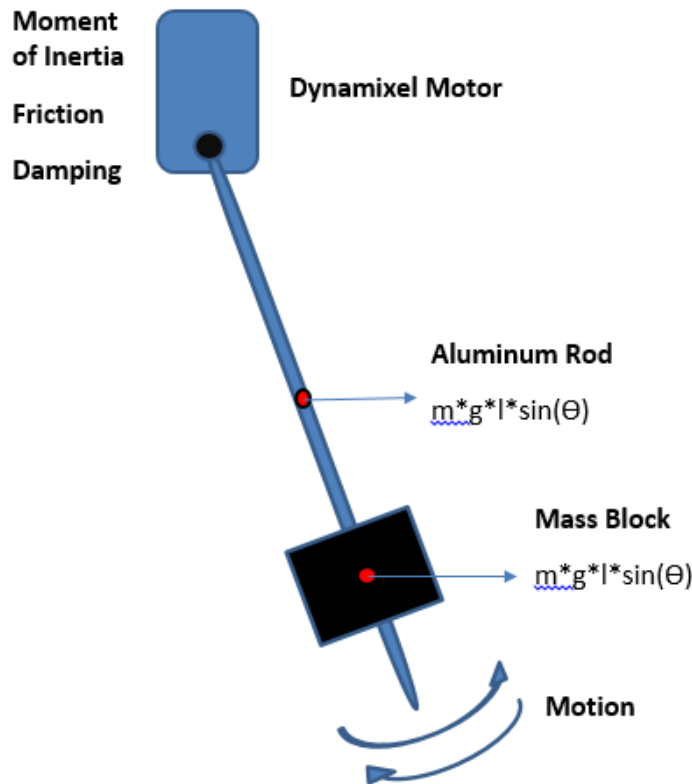
The pendulum motion can be represented using the equations of motion by summing the moments in the system. The Dynamixel servos have their inherent inertial and dissipative parameters. The Aluminum rod and the moveable mass also apply their gravitational torque on the system during the free fall. The equation of motion for the pendulum drop hence becomes:

$$I\ddot{\theta} + B\dot{\theta} + F = -m_{rod}gl_{rod_{com}}\sin(\theta) - m_{mass}gl_{mass_{com}}\sin(\theta) \quad (2.1)$$

Where ‘I’ is the moment of inertia of the system, ‘B’ is its damping, ‘F’ is the rotational friction of the Dynamixel. The change in angle of the pendulum is denoted by ‘ $\theta$ ’, ‘g’ is the gravitational acceleration, ‘ $m_{rod}$ ’ and ‘ $l_{rod_{com}}$ ’ are the mass and the distance of center of mass of the Aluminum rod from the center of rotation, respectively and ‘ $m_{mass}$ ’ and ‘ $l_{mass_{com}}$ ’ are the mass and the distance of center of mass of the moveable mass from the center of rotation, respectively. The negative sign with

gravitational torques indicate the force of gravity in the negative direction i.e. downwards. This equation of motion can be re-arranged to be written as:

$$I\ddot{\theta} = -m_{rod}gl_{rod\_com}\sin(\theta) - m_{mass}gl_{mass\_com}\sin(\theta) - B\dot{\theta} - F \quad (2.2)$$

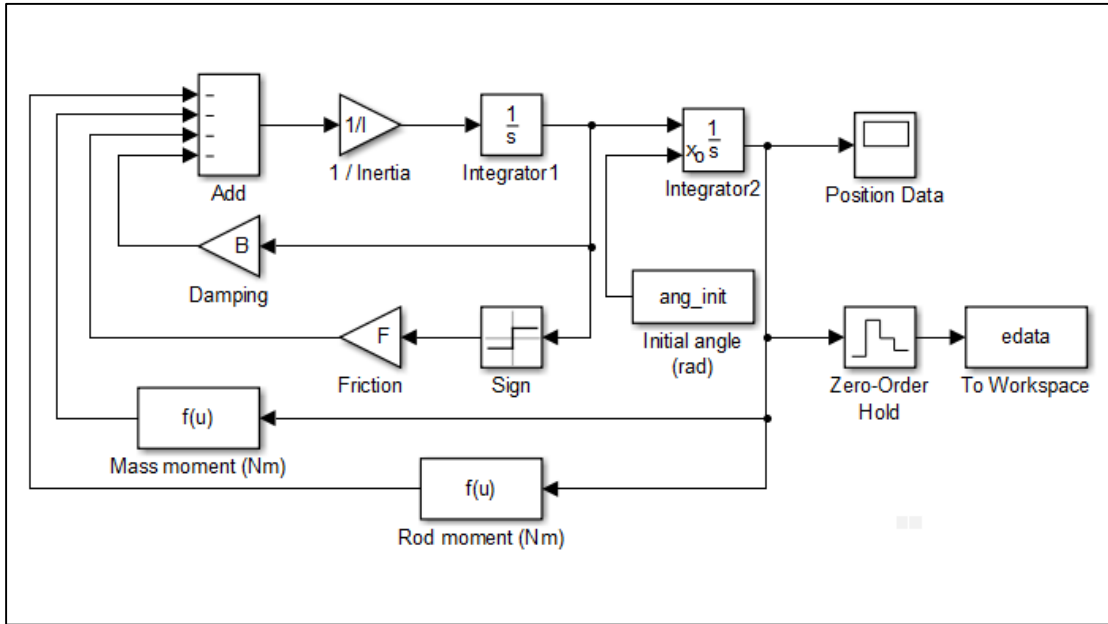


**Figure 2.5** Free body diagram of the pendulum drop test.

Previously, Simon et. al. [17] have developed a similar model to optimize the moment of inertia and damping of a leg's pendulum motion under free fall. They used these estimations to develop a quantifying method for patients with spasticity, to help the care providers in diagnosis and rehabilitation for these individuals. The model uses the



Levenberg-Marquardt optimization which is a least-squares based algorithm to optimize our parameters of interest. Our model for the optimization is shown in figure below:



**Figure 2.6** Simulink model for parameter optimization.

This Simulink model is fed with the position data from the pendulum test and a logically guessed value of ‘I’ and ‘B’ and the estimated value of ‘F’ found earlier. The model first generates a motion data based on the provided ‘I’, ‘B’ and ‘F’ values. It then runs its optimization algorithm and adjusts the parameters while calculating the least-squares and attempting to generate a motion data with the least error and the least deviation from the actual data. Since ‘B’ and ‘F’ are both dissipative parameters, they cannot both be optimized in the same model at the same time. Therefore, two instances of this model are used. With one model, ‘I’ and ‘B’ are optimized. These optimized ‘I’ and ‘B’ are fed to the other instance of the model to optimize for ‘F’. Optimized ‘F’ from the second model is then fed back to the first model to again optimize for ‘I’ and ‘B’. Once

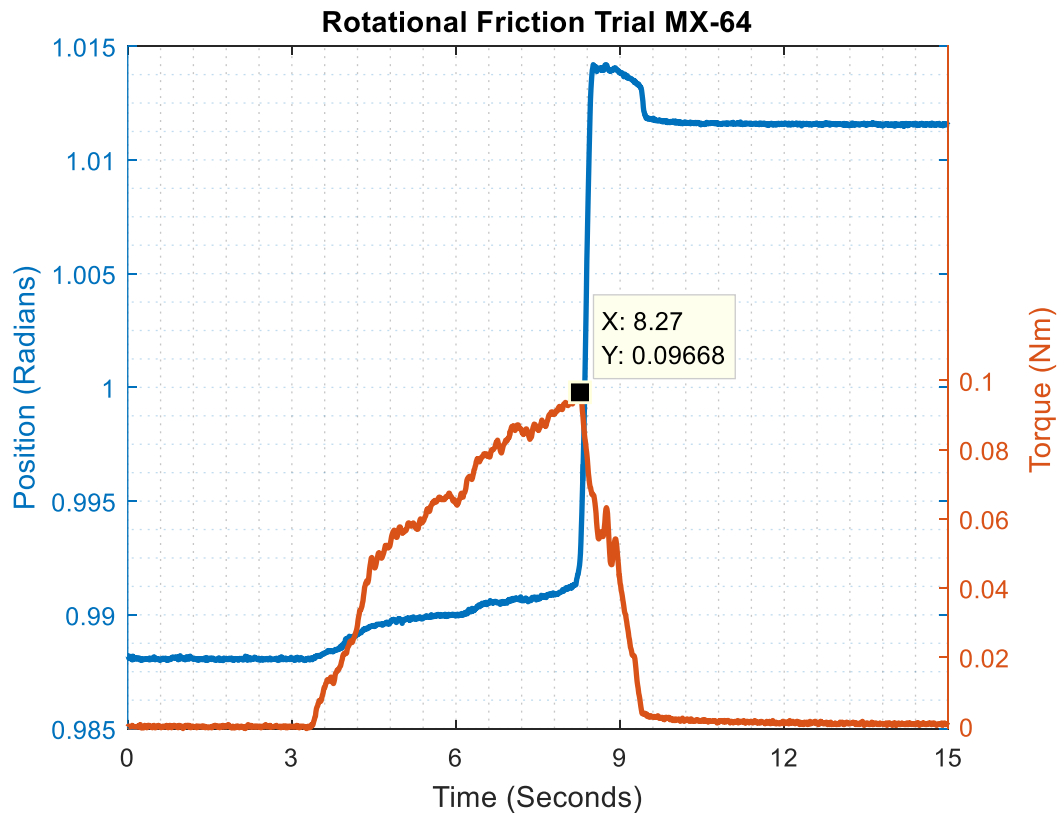
the second model starts giving similar optimized values for 'F', we run the first model again for once to get the final optimized 'I' and 'B' values. This process is repeated for each of the 90 trials. Single averages of the 'B', 'I' and 'F' are recorded for each motor.

## CHAPTER 3

### RESULTS

#### 3.1 Frictional Torque Estimation for Motors

A small gradual torque is applied to the motor's head while recording the position and torque. When the motor's friction is overcome, a sudden change in the position takes place. And the torque at that very instance is recorded as the estimate of the frictional torque for the trial. In the Figure 3.1, the blue plot show the position with time while the red plot shows the torque applied with time. The marker shows the value of the torque peak at 8.27 seconds.



**Figure 3.1** Rotational friction estimation from a trial.

In similar fashion the friction from the trials for all the motors is noted. The results from the frictional torque estimation trials for all the motors are summarized in the Table 3.1 below.

**Table 3.1** Summarized results from frictional torque (Nm) estimation in counter clockwise (ccw) and clockwise (cw) directions

All values are in Newton-meters	<b>MX-28</b>	<b>MX-64</b>	<b>MX-106</b>
<b>Trial 1 ccw</b>	0.01657	0.10100	0.11380
<b>Trial 2 ccw</b>	0.01415	0.09668	0.12610
<b>Trial 3 ccw</b>	0.01915	0.09522	0.11100
<b>Average for ccw</b>	<b>0.01660</b>	<b>0.09760</b>	<b>0.11700</b>
<b>Trial 1 cw</b>	0.01499	0.10710	0.14140
<b>Trial 2 cw</b>	0.01931	0.11240	0.12870
<b>Trial 3 cw</b>	0.02140	0.12390	0.11920
<b>Average for cw</b>	<b>0.01860</b>	<b>0.11450</b>	<b>0.12980</b>
<b>Overall average</b>	<b>0.01760</b>	<b>0.10610</b>	<b>0.12340</b>

The table lists averages of rotational friction torque in counter clockwise and clockwise direction separately. It also gives a single value averaged from all the trials. These values are used as the initial estimates of friction in the optimization model.

### 3.2 Optimized Moment of Inertia, Damping and Friction of Motors

Table 3.2, 3.3 and 3.4 summarizes the optimization results from the trials of MX-28, MX-64 and MX-106 respectively. They record the values of ‘I’, ‘B’ and ‘F’ that the optimization model estimated. The optimization model takes the actual position data and tries to match a new position data on it, and tells us what the parameters’ values are to produce this new data. The closeness of the match of the two position data, actually, represents the closeness of the values of the parameters given by the model to the actual values of those parameters for the servos.

**Table 3.2** Summarized results from optimization for MX-28 with a movable mass at various locations

<b>MX-28 Trials</b>	<b>Friction</b>	<b>Moment of Inertia (motor)</b>	<b>Moment of Inertia (total)</b>	<b>Damping</b>
MX28_40cm_ccw_1	0.0085	0.0060	0.0233	0.0319
MX28_40cm_ccw_2	0.0114	0.0056	0.0229	0.0306
MX28_40cm_ccw_3	0.0088	0.0065	0.0237	0.0322
MX28_40cm_cw_1	0.0046	0.0063	0.0236	0.0353
MX28_40cm_cw_2	0.0039	0.0068	0.0240	0.0362
MX28_40cm_cw_3	0.0044	0.0063	0.0235	0.0348
<b>Average at 40cm</b>	<b>0.0069</b>	<b>0.0063</b>	<b>0.0235</b>	<b>0.0335</b>
MX28_35cm_ccw_1	0.0097	0.0064	0.0198	0.0334
MX28_35cm_ccw_2	0.0090	0.0066	0.0199	0.0336
MX28_35cm_ccw_3	0.0079	0.0066	0.0199	0.0357
MX28_35cm_cw_1	0.0064	0.0067	0.0201	0.0364
MX28_35cm_cw_2	0.0070	0.0065	0.0198	0.0349
MX28_35cm_cw_3	0.0071	0.0064	0.0198	0.0347
<b>Average at 35cm</b>	<b>0.0079</b>	<b>0.0065</b>	<b>0.0199</b>	<b>0.0348</b>

MX28_30cm_ccw_1	0.0087	0.0067	0.0167	0.0344
MX28_30cm_ccw_2	0.0086	0.0067	0.0167	0.0346
MX28_30cm_ccw_3	0.0090	0.0062	0.0162	0.0332
MX28_30cm_cw_1	0.0077	0.0069	0.0169	0.0343
MX28_30cm_cw_2	0.0076	0.0068	0.0168	0.0341
MX28_30cm_cw_3	0.0078	0.0066	0.0166	0.0337
<b>Average at 30cm</b>	<b>0.0082</b>	<b>0.0067</b>	<b>0.0167</b>	<b>0.0341</b>
MX28_25cm_ccw_1	0.0053	0.0065	0.0137	0.0360
MX28_25cm_ccw_2	0.0051	0.0068	0.0140	0.0367
MX28_25cm_ccw_3	0.0054	0.0068	0.0139	0.0367
MX28_25cm_cw_1	0.0053	0.0068	0.0139	0.0388
MX28_25cm_cw_2	0.0047	0.0071	0.0142	0.0392
MX28_25cm_cw_3	0.0041	0.0072	0.0143	0.0400
<b>Average at 25cm</b>	<b>0.0050</b>	<b>0.0069</b>	<b>0.0140</b>	<b>0.0379</b>
MX28_20cm_ccw_1	0.0036	0.0076	0.0124	0.0416
MX28_20cm_ccw_2	0.0028	0.0076	0.0124	0.0423
MX28_20cm_ccw_3	0.0440	0.0075	0.0123	0.0405
MX28_20cm_cw_1	0.0052	0.0074	0.0123	0.0386
MX28_20cm_cw_2	0.0041	0.0079	0.0127	0.0421
MX28_20cm_cw_3	0.0051	0.0074	0.0122	0.0390
<b>Average at 20cm</b>	<b>0.0108</b>	<b>0.0076</b>	<b>0.0124</b>	<b>0.0407</b>
<b>MX-28 averages</b>	<b>0.0078</b>	<b>0.0068</b>		<b>0.0362</b>

**Table 3.3** Summarized results from optimization for MX-64 with a movable mass at various locations

<b>MX-64 Trials</b>	<b>Friction</b>	<b>Moment of Inertia (motor)</b>	<b>Moment of Inertia (total)</b>	<b>Damping</b>
MX64_40cm_ccw_1	0.0866	0.0145	0.0317	0.0358
MX64_40cm_ccw_2	0.0866	0.0145	0.0317	0.0361
MX64_40cm_ccw_3	0.0866	0.0145	0.0317	0.0361
MX64_40cm_cw_1	0.0871	0.0133	0.0305	0.0379
MX64_40cm_cw_2	0.0834	0.0148	0.0320	0.0397
MX64_40cm_cw_3	0.0873	0.0134	0.0307	0.0367
<b>Average at 40cm</b>	<b>0.0863</b>	<b>0.0142</b>	<b>0.0314</b>	<b>0.0371</b>
MX64_35cm_ccw_1	0.0929	0.0148	0.0282	0.0348
MX64_35cm_ccw_2	0.0902	0.0145	0.0279	0.0349
MX64_35cm_ccw_3	0.0902	0.0145	0.0297	0.0353
MX64_35cm_cw_1	0.0835	0.0146	0.0280	0.0364
MX64_35cm_cw_2	0.0842	0.0142	0.0276	0.0373
MX64_35cm_cw_3	0.0855	0.0142	0.0276	0.0381
<b>Average at 35cm</b>	<b>0.0878</b>	<b>0.0145</b>	<b>0.0282</b>	<b>0.0361</b>
MX64_30cm_ccw_1	0.0915	0.0141	0.0241	0.0305
MX64_30cm_ccw_2	0.0852	0.0152	0.0252	0.0359
MX64_30cm_ccw_3	0.0905	0.0150	0.0250	0.0318
MX64_30cm_cw_1	0.0872	0.0150	0.0250	0.0330
MX64_30cm_cw_2	0.0896	0.0159	0.0259	0.0338
MX64_30cm_cw_3	0.0836	0.0140	0.0240	0.0341
<b>Average at 30cm</b>	<b>0.0879</b>	<b>0.0149</b>	<b>0.0249</b>	<b>0.0332</b>
MX64_25cm_ccw_1	0.0874	0.0144	0.0216	0.0320
MX64_25cm_ccw_2	0.0819	0.0146	0.0218	0.0357
MX64_25cm_ccw_3	0.0855	0.0147	0.0218	0.0339
MX64_25cm_cw_1	0.0750	0.0148	0.0219	0.0383

MX64_25cm_cw_2	0.0747	0.0148	0.0220	0.0385
MX64_25cm_cw_3	0.0747	0.0134	0.0206	0.0367
<b>Average at 25cm</b>	<b>0.0799</b>	<b>0.0145</b>	<b>0.0216</b>	<b>0.0359</b>
MX64_20cm_ccw_1	0.0817	0.0159	0.0208	0.0292
MX64_20cm_ccw_2	0.0815	0.0154	0.0202	0.0295
MX64_20cm_ccw_3	0.0787	0.0146	0.0194	0.0317
MX64_20cm_cw_1	0.0787	0.0161	0.0209	0.0320
MX64_20cm_cw_2	0.0781	0.0143	0.0192	0.0323
MX64_20cm_cw_3	0.0783	0.0160	0.0208	0.0332
<b>Average at 20cm</b>	<b>0.0795</b>	<b>0.0154</b>	<b>0.0202</b>	<b>0.0313</b>
<b>MX-64 averages</b>	<b>0.0843</b>	<b>0.0147</b>		<b>0.0347</b>

**Table 3.4** Summarized results from optimization for MX-106 with a movable mass at various locations

<b>MX-106 Trials</b>	<b>Friction</b>	<b>Moment of Inertia (motor)</b>	<b>Moment of Inertia (total)</b>	<b>Damping</b>
MX106_40cm_ccw_1	0.1174	0.0250	0.0422	0.0491
MX106_40cm_ccw_2	0.1157	0.0270	0.0442	0.0492
MX106_40cm_ccw_3	0.1192	0.0308	0.0480	0.0502
MX106_40cm_cw_1	0.1198	0.0256	0.0429	0.0444
MX106_40cm_cw_2	0.1201	0.0274	0.0447	0.0447
MX106_40cm_cw_3	0.1155	0.0292	0.0465	0.0514
<b>Average at 40cm</b>	<b>0.1180</b>	<b>0.0275</b>	<b>0.0448</b>	<b>0.0482</b>
MX106_35cm_ccw_1	0.1142	0.0298	0.0430	0.0510
MX106_35cm_ccw_2	0.1123	0.0280	0.0414	0.0526
MX106_35cm_ccw_3	0.1113	0.0271	0.0405	0.0537
MX106_35cm_cw_1	0.1127	0.0257	0.0390	0.0591
MX106_35cm_cw_2	0.1083	0.0365	0.0499	0.0600



MX106_35cm_cw_3	0.1066	0.0297	0.0431	0.0559
<b>Average at 35cm</b>	<b>0.1109</b>	<b>0.0295</b>	<b>0.0428</b>	<b>0.0554</b>
MX106_30cm_ccw_1	0.1045	0.0284	0.0384	0.0532
MX106_30cm_ccw_2	0.1036	0.0294	0.0394	0.0547
MX106_30cm_ccw_3	0.1112	0.0291	0.0391	0.0475
MX106_30cm_cw_1	0.1234	0.0288	0.0388	0.0402
MX106_30cm_cw_2	0.1234	0.0251	0.0351	0.0410
MX106_30cm_cw_3	0.1178	0.0250	0.0350	0.0404
<b>Average at 30cm</b>	<b>0.1140</b>	<b>0.0276</b>	<b>0.0376</b>	<b>0.0462</b>
MX106_25cm_ccw_1	0.1234	0.0284	0.0356	0.0333
MX106_25cm_ccw_2	0.1234	0.0272	0.0344	0.0330
MX106_25cm_ccw_3	0.1234	0.0278	0.0350	0.0347
MX106_25cm_cw_1	0.1234	0.0259	0.0331	0.0285
MX106_25cm_cw_2	0.1234	0.0267	0.0339	0.0292
MX106_25cm_cw_3	0.1234	0.0264	0.0336	0.0291
<b>Average at 25cm</b>	<b>0.1234</b>	<b>0.0271</b>	<b>0.0343</b>	<b>0.0313</b>
MX106_20cm_ccw_1	0.1234	0.0272	0.0320	0.0287
MX106_20cm_ccw_2	0.1233	0.0262	0.0310	0.0277
MX106_20cm_ccw_3	0.1224	0.0271	0.0319	0.0279
MX106_20cm_cw_1	0.1234	0.0283	0.0332	0.0238
MX106_20cm_cw_2	0.1234	0.0270	0.0318	0.0234
MX106_20cm_cw_3	0.1234	0.0282	0.0330	0.0233
<b>Average at 20cm</b>	<b>0.1232</b>	<b>0.0273</b>	<b>0.0322</b>	<b>0.0258</b>
<b>MX-106 averages</b>	<b>0.1179</b>	<b>0.0278</b>		<b>0.0414</b>

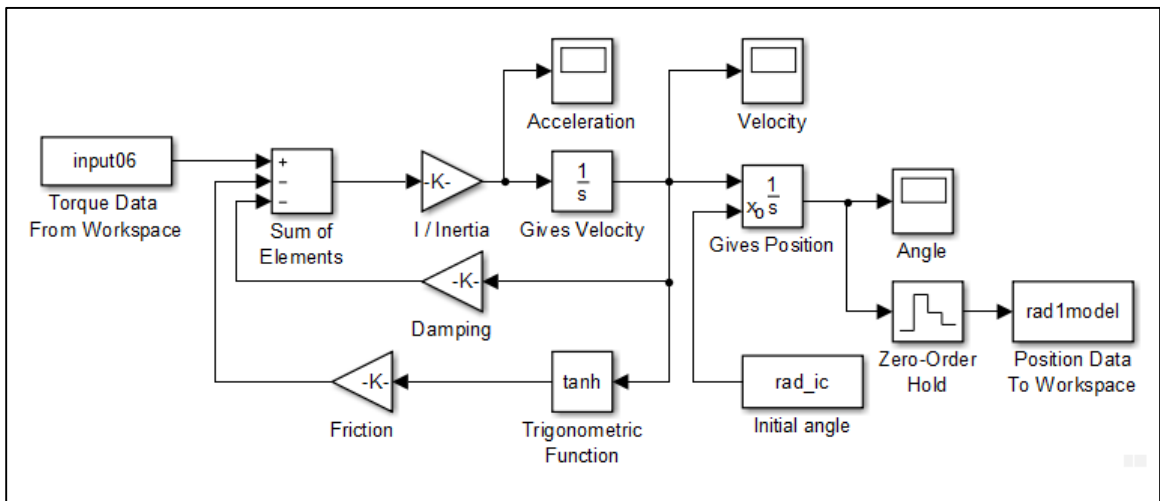
The ‘Friction’ has the units of Newton-meters, ‘Inertia (motor)’ and ‘Inertia (total)’ have the units of kilogram-meter<sup>2</sup>, and the ‘Damping’ has the units of Newton-meter-second<sup>-1</sup>. The figures in Appendix A display the plots for the optimization of all

the trials with all the motors corresponding to the trials listed in the three tables above. The optimization here tells that the values of the parameters gained are from a system that would produce a position data like the one shown if it undergoes the pendulum drop test.

### 3.3 Testing the Results

The results are tested to see if a torque, that was used to make simple movements with the motor passively, is input to the admittance control model, with the obtained values of the moment of inertia, damping and friction, would produce a position output that is same as the movements made in reality with the motor passively.

The admittance control model, shown in Figure 3.2, is a Simulink model that gets the torque input from the MATLAB workspace and uses the parameters obtained earlier to predict the position that would be reached with our Dynamixel motors.



**Figure 3.2** Simulink model for testing.

The testing trials are divided into five categories depending upon the speed with which the motor was acted upon. These are: ‘Slow’, ‘Normal’, ‘Fast’, ‘Slow to Fast’, and ‘Fast to Slow’. In the ‘Slow’, ‘Normal’ and ‘Fast’ trials, a constant speed was attempted to be maintained. In the ‘Slow to Fast’ and ‘Fast to Slow’ trials, the speed was varied within each trial.

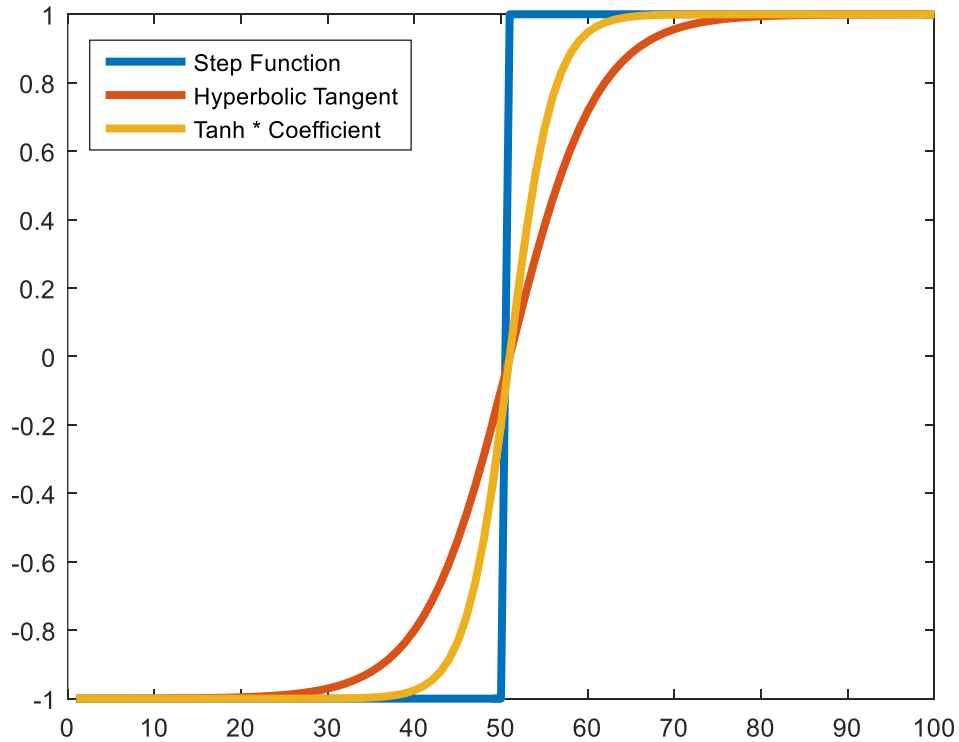
For each trial, there are three plots, and these plots can be found in Appendix A.2. The first plot shows both the recorded Position and Torque with time. The second plot shows the measured torque and compares it with the reconstructed calculated torque, both of these are plotted on top of each other with respect to time. The third plot compares the actual change in position, recorded by the OptiTrack Duo motion capture system, with the change in position predicted by the admittance control model shown in Figure 3.2.

### **3.4 Improved ODE for Implementing Admittance Control**

The improved Ordinary Differential Equation code for implementing admittance control incorporates the moment of inertia and damping as well as the friction parameters to predict the target position based on a torque input. This is a C code that is turned into a mex (MATLAB executable) file so that it can be run using the MATLAB platform while not compromising on the computational speed.

The code for the ODE can be found in the Appendix B.1. The code is basically the code version of the Simulink model shown in Figure 3.2. Since the Friction cannot change instantaneously from a positive value to a negative value when the motion reverses direction from positive to negative in real world, the Friction is modeled as a hyperbolic tangent so that there is no tremendous instantaneous change in the value of

friction. In the code, the Friction is multiplied by a coefficient too. This coefficient is found by trial and error method to produce a hyperbolic tangent that is very close to the step function. Figure 3.3 shows the difference between a step function, a hyperbolic function and hyperbolic function that is multiplied with a coefficient.



**Figure 3.3** Step function verses hyperbolic tangent and a coefficient times hyperbolic function.

## CHAPTER 4

### DISCUSSION

A simple practical method has been shown in this research that could be used to optimize the parameters of a plant; in our case the plant are the Dynamixel servos. No specialized electrical measurement tools are needed and the optimization results obtained are quite reasonable. A few points can be deduced from the results.

With a faster speed there seems to be a delay in the measured torques from the expected or calculated torques. This might have been due to the fact in order to move faster, the force sensor was held tightly between the fingers exerting a higher force in both the positive and the negative directions. It can be seen from the slow to fast and fast to slow trials, that within a single trial, if the speed is lower, this unexpected delay goes away.

Measurements from the force sensor are however consistent. There seems to be a drift in the readings since the measured torque produces a position data that although has the expected shape but starts drifting in a direction. The direction of this drift is not always the same. Sometimes within a single trial, the drift changes direction. A possible explanation of it could be that the application of force on the force sensor is not correct. This can be built upon the fact that the measured torque is always less than the calculated torque. And the smaller measured torque, in reality is producing the position data that a higher torque must have produced. This drift also seems to be proportional to the speed of the motion, which is the higher the speed, the more drift there is.

However, with the bigger motor, the MX-106, the measured torque is much closer to the calculated torque. So the issue might be with the capability of the force sensor, since this particular force sensor is designed for higher loads, up to 200 N.

The reason behind the recorded and the modeled position not entirely overlapping could also be the model's inability to perfectly model the servos. There are various kinds of friction in a real servo system; static, viscous, coulomb etc., however, in our simple model we have only considered one friction. With the incorporation of the estimated friction of all the different kinds, the model would produce even better results. Also, as can be seen from the initial estimated friction and the optimized friction, during the optimization the friction value always decreases. This is because the initial estimates are the estimates of the static friction which is the friction needed to make the rotors move from a resting position. However, during a motion, this friction is lesser and since the optimization is done on a motion, the optimized value of friction is lower. Since the motion used in optimization started from a rest point, went through motion in either direction and comes to a stop at the end, the optimized value of friction can be thought of as a combination of all the different kinds of friction present in the servo system.

The admittance control model that is used to test the optimization not only allows to simulate the control algorithm but also to compare the accuracy of the improved ODE code. The ODE code, with the inclusion of friction parameter and with the modeling of this friction parameter as a hyperbolic tangent provides a more realistic approach.

Although, this research is focused on understanding the plant parameters of the Dynamixel servos, the results from test trials suggest a need to understand the force sensor in order to completely master the implementation of admittance control.

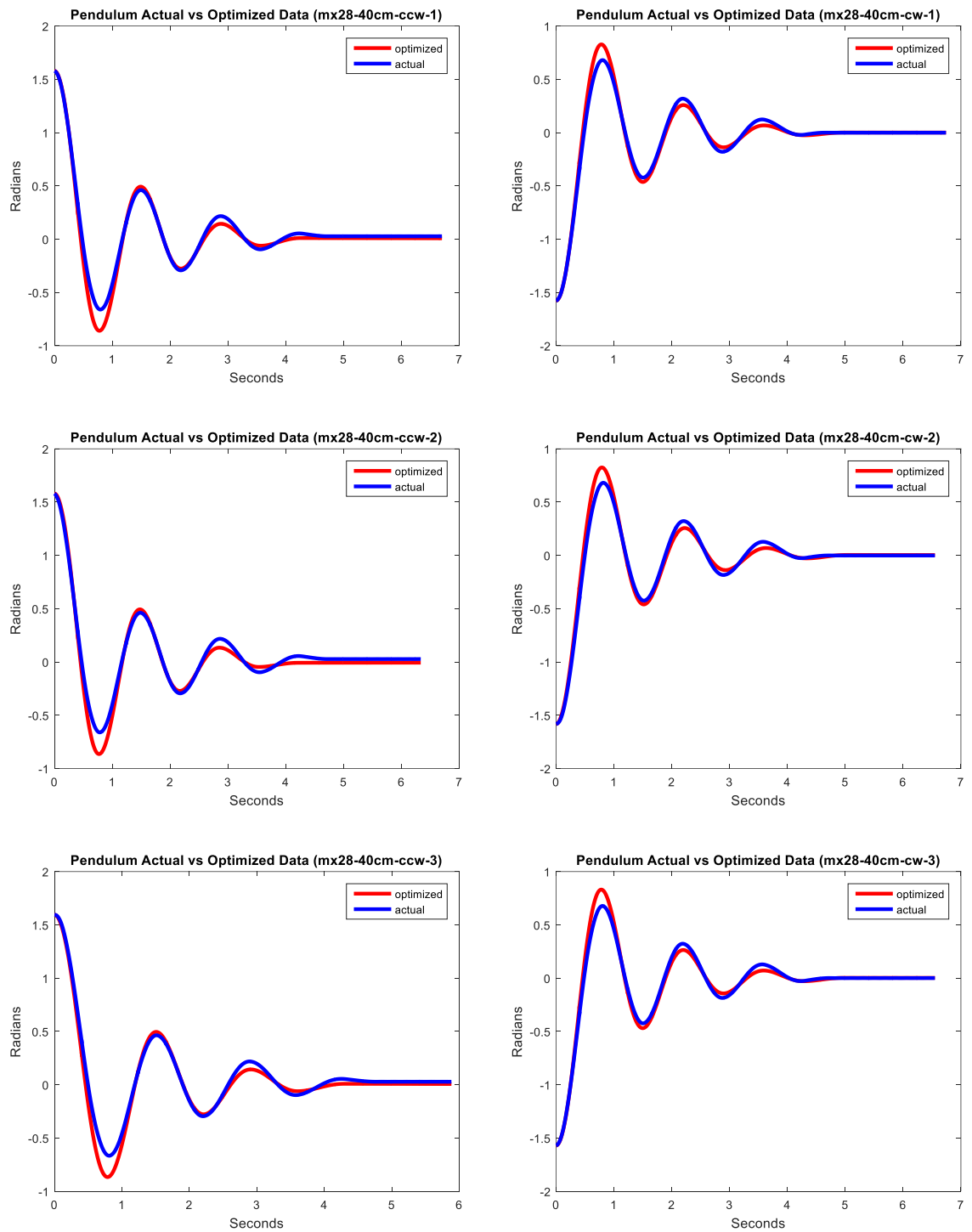
An understanding of the force sensor's sensitivity, accuracy, drift with time as well as drift with prolong force application and the effects of various orientation of force sensor on the force reception will be a valuable contribution to the improvement of performance of the control algorithms utilizing the force sensors. Improved force reading algorithm can also be developed and implemented that removes tremors from the applied force in real-time.

The application of the improved ODE to the Dynamixel servos would also allow to understand the variation between simulated and practical implementation of the admittance control with optimized values for the plant parameters.

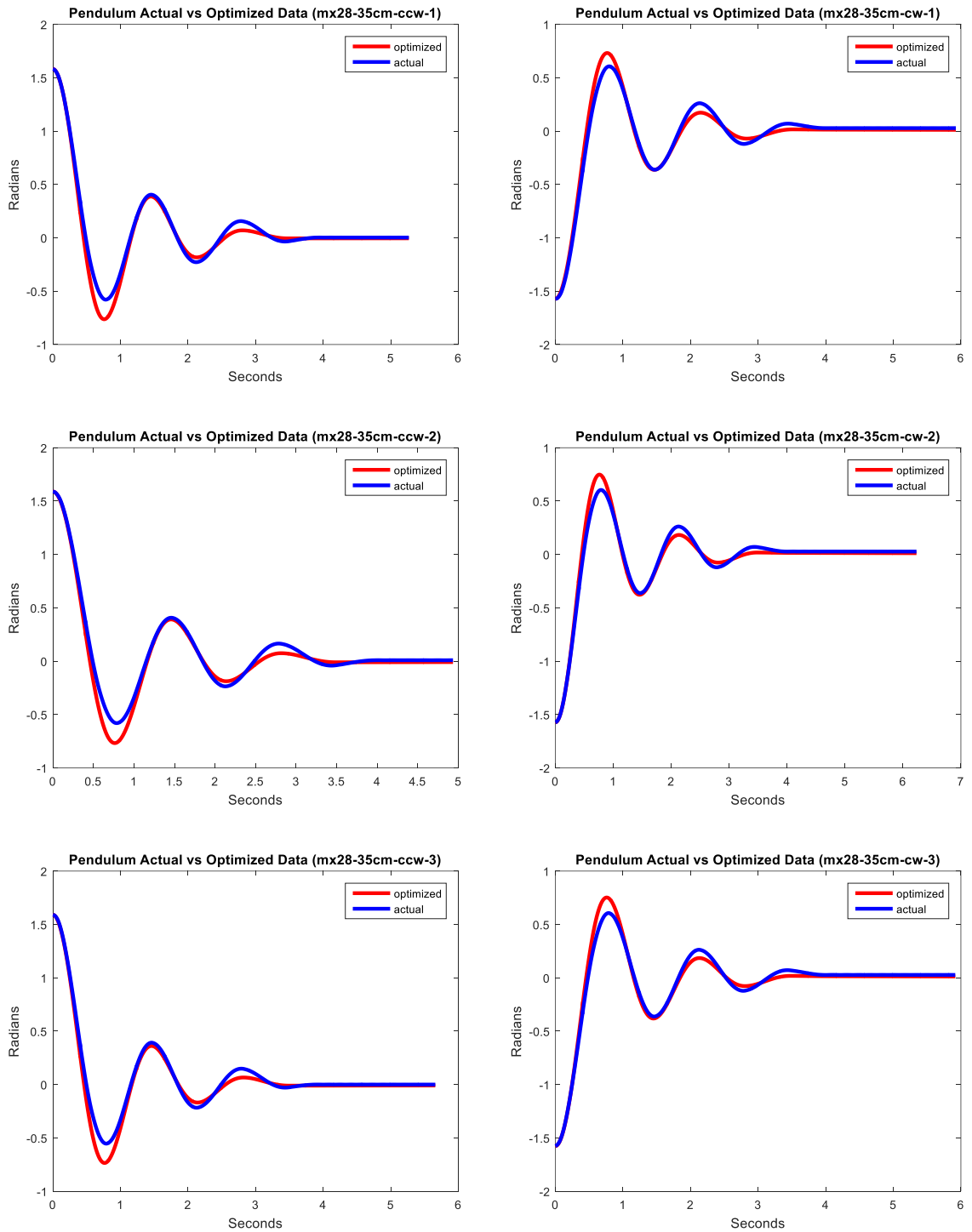
**APPENDIX A**  
**PLOTS FOR OPTIMIZATION**

This appendix would contain all the optimization plots for all the motors. Each motor has 6 trials with the mass at each of the 5 locations from motor's center of rotation.

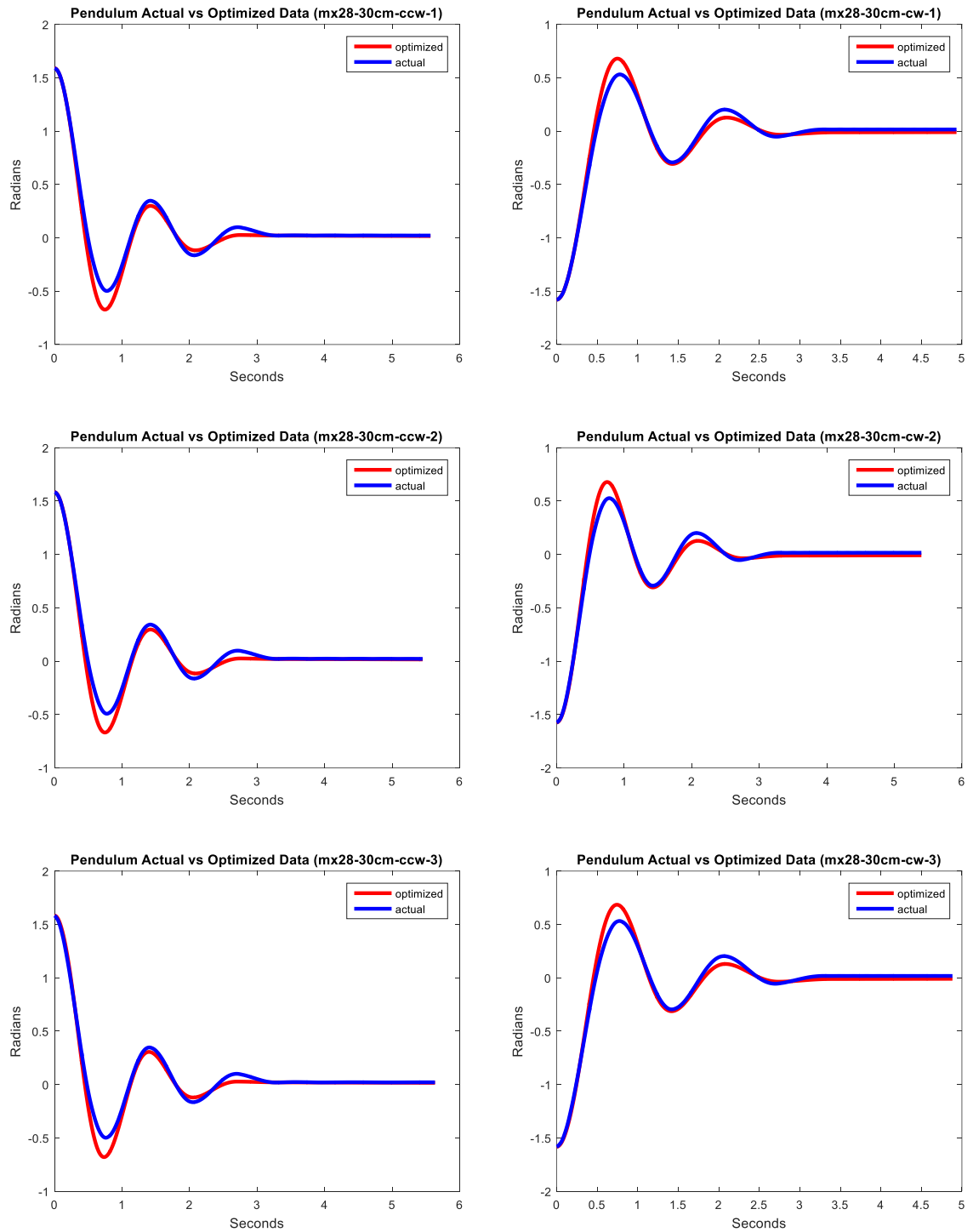




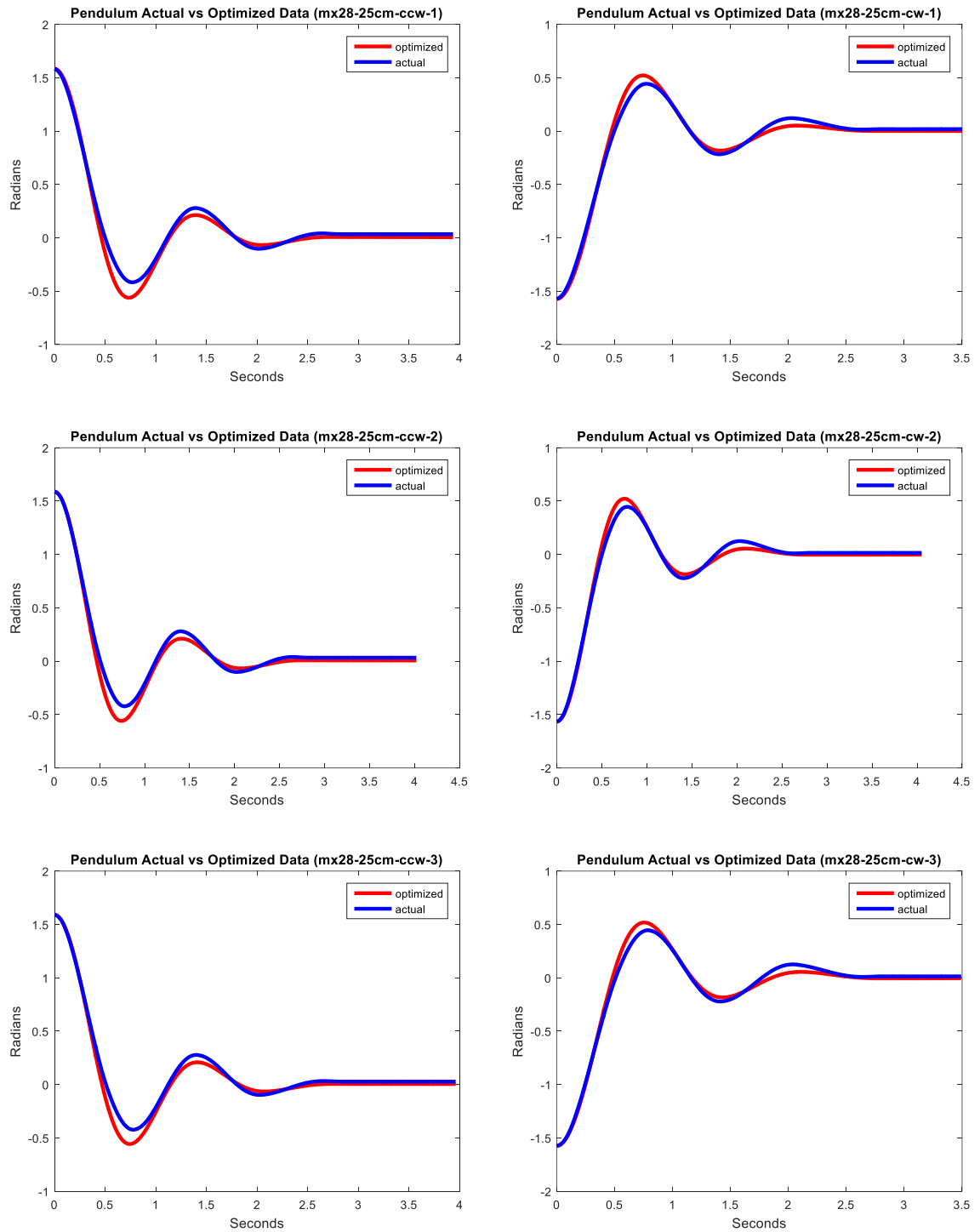
**Figure A.1** MX-28 trials with mass at 40 cm.



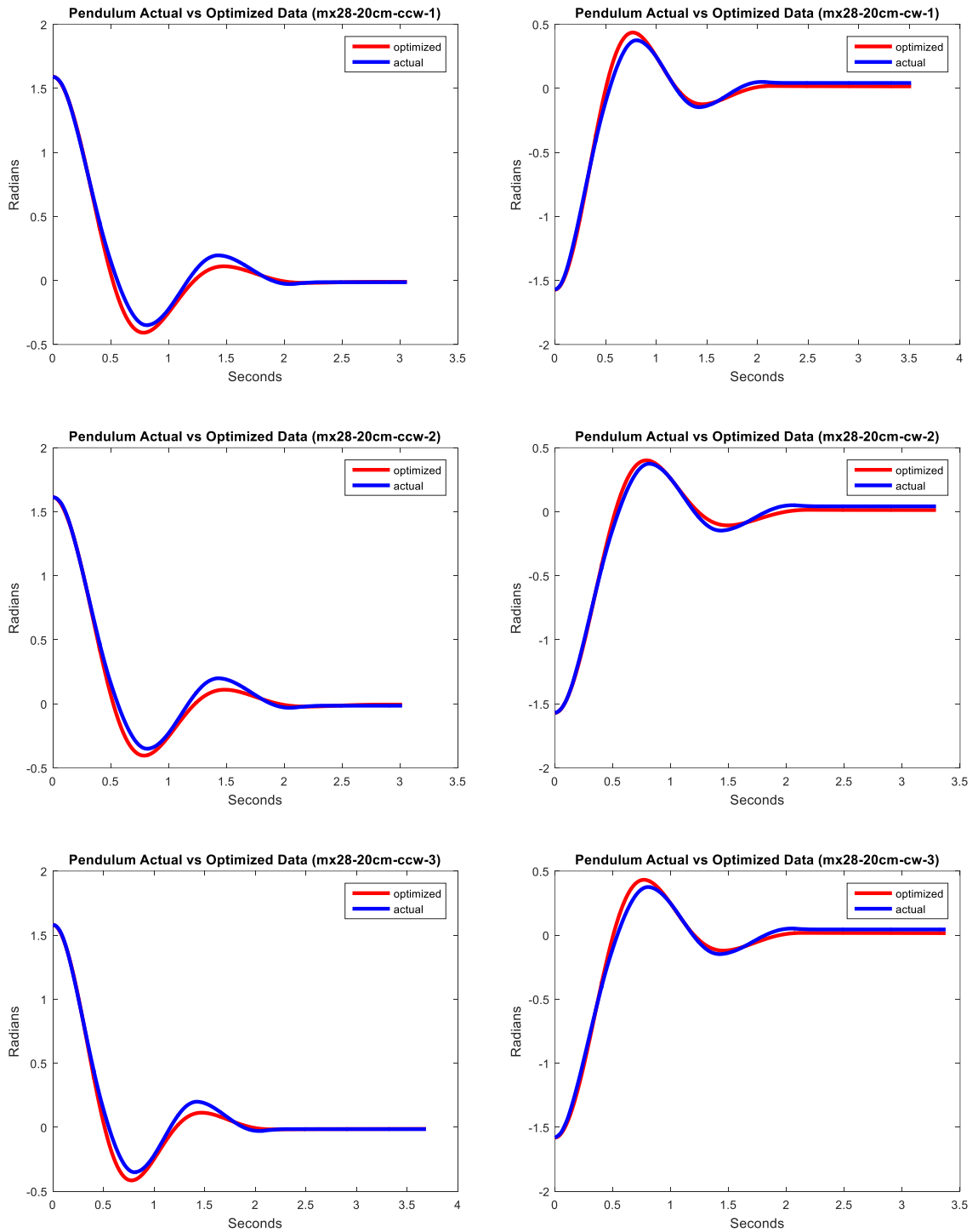
**Figure A.2** MX-28 trials with mass at 35 cm.



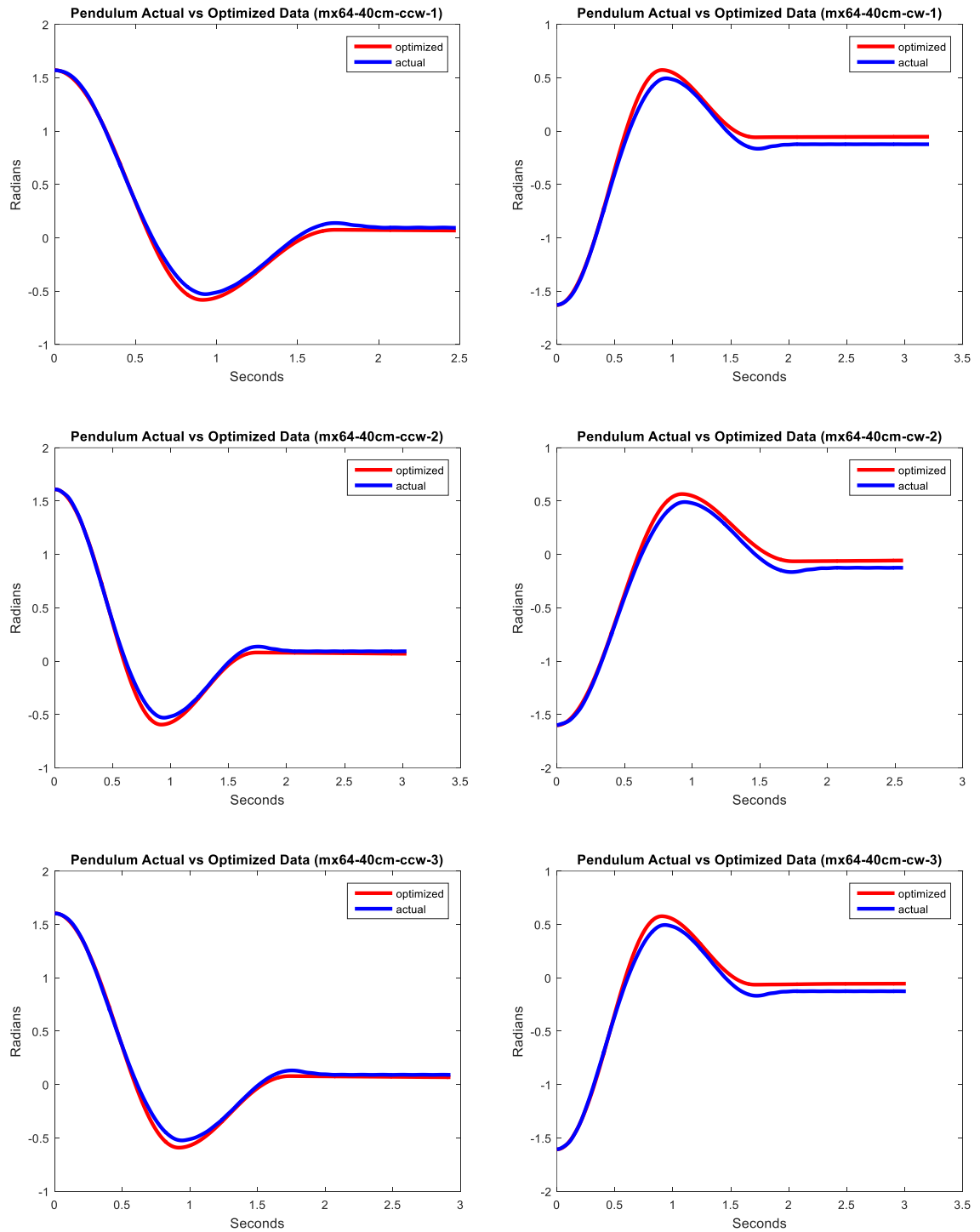
**Figure A.3** MX-28 trials with mass at 30 cm.



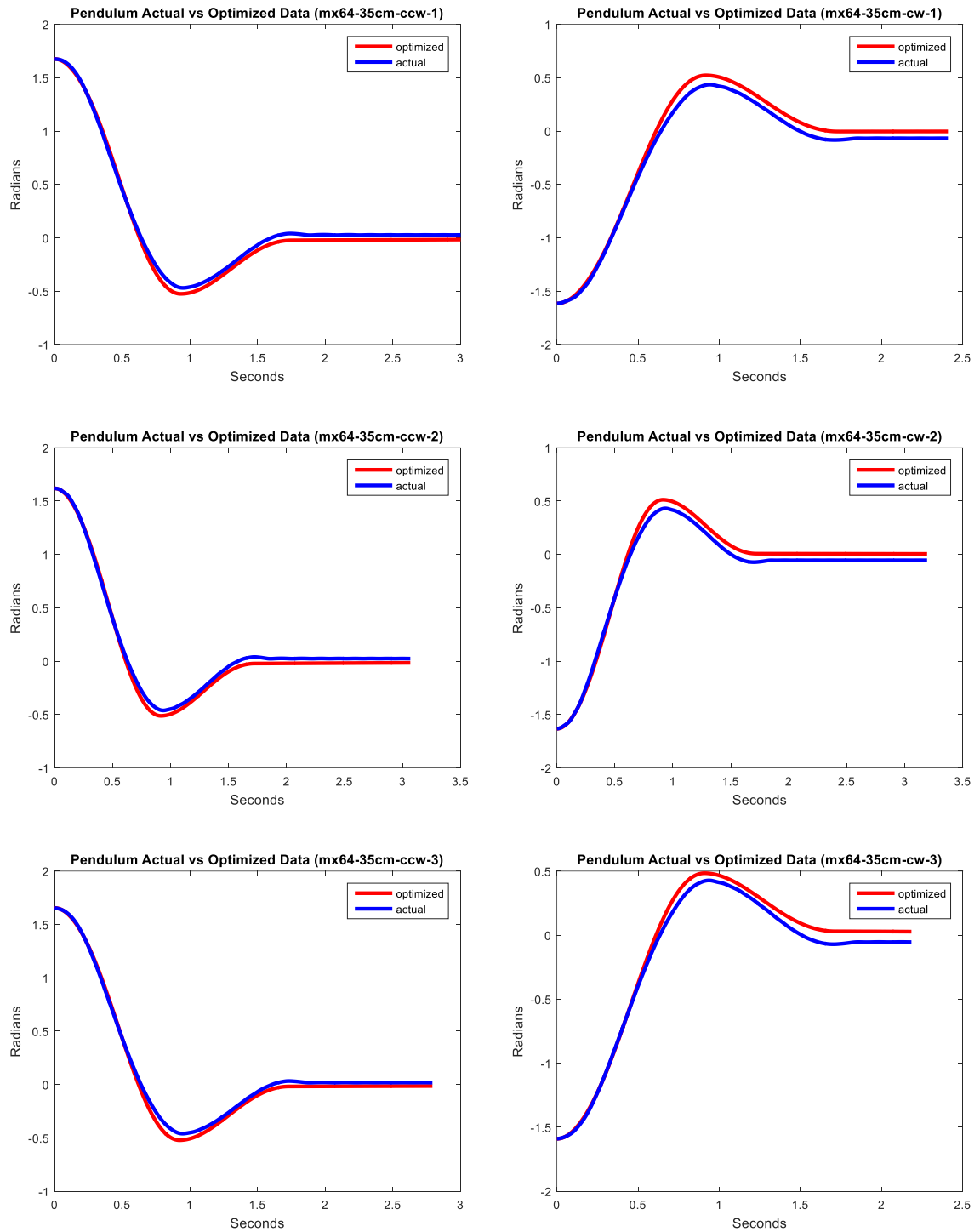
**Figure A.4** MX-28 trials with mass at 25 cm.



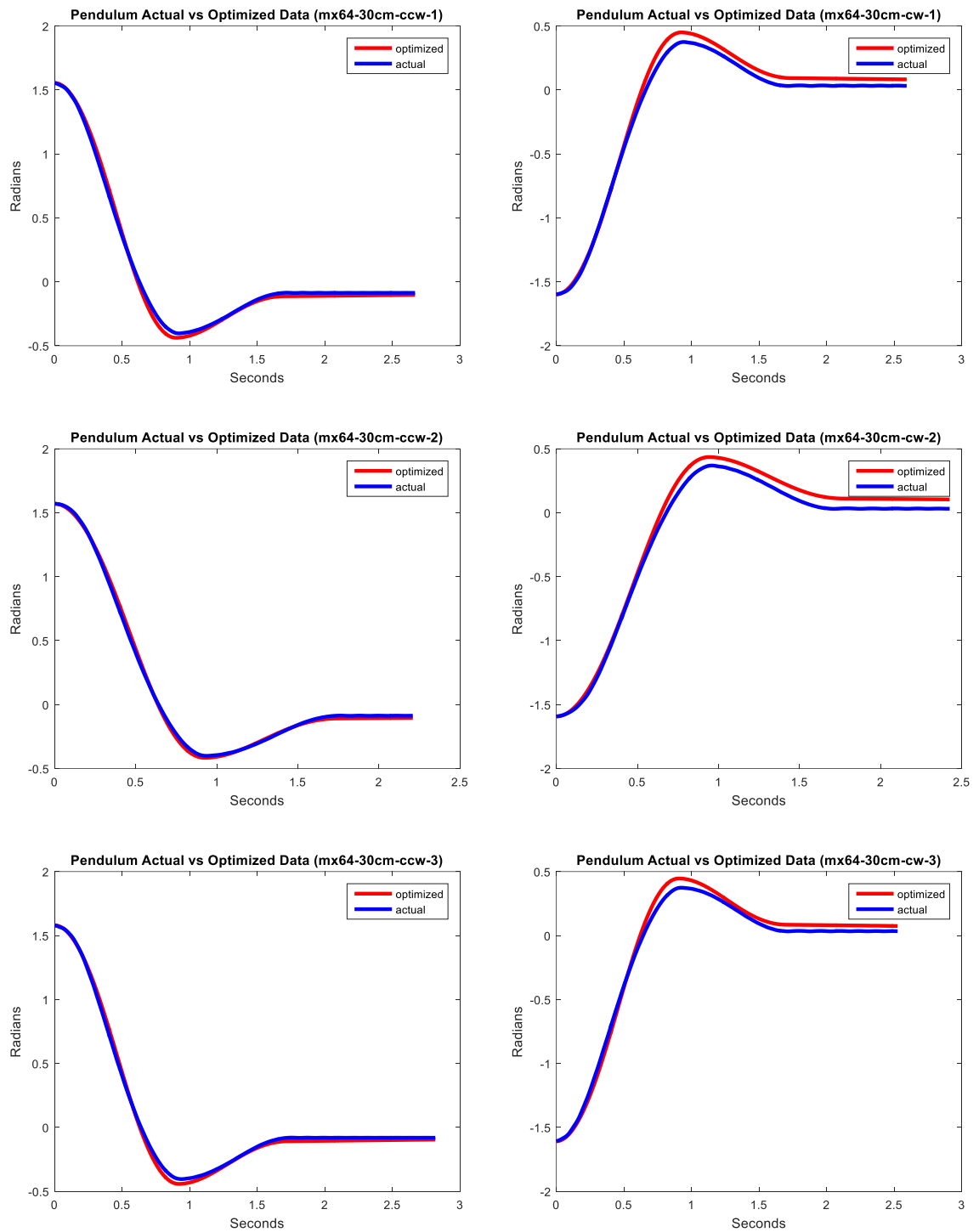
**Figure A.5** MX-28 trials with mass at 20 cm.



**Figure A.6** MX-64 trials with mass at 40 cm.

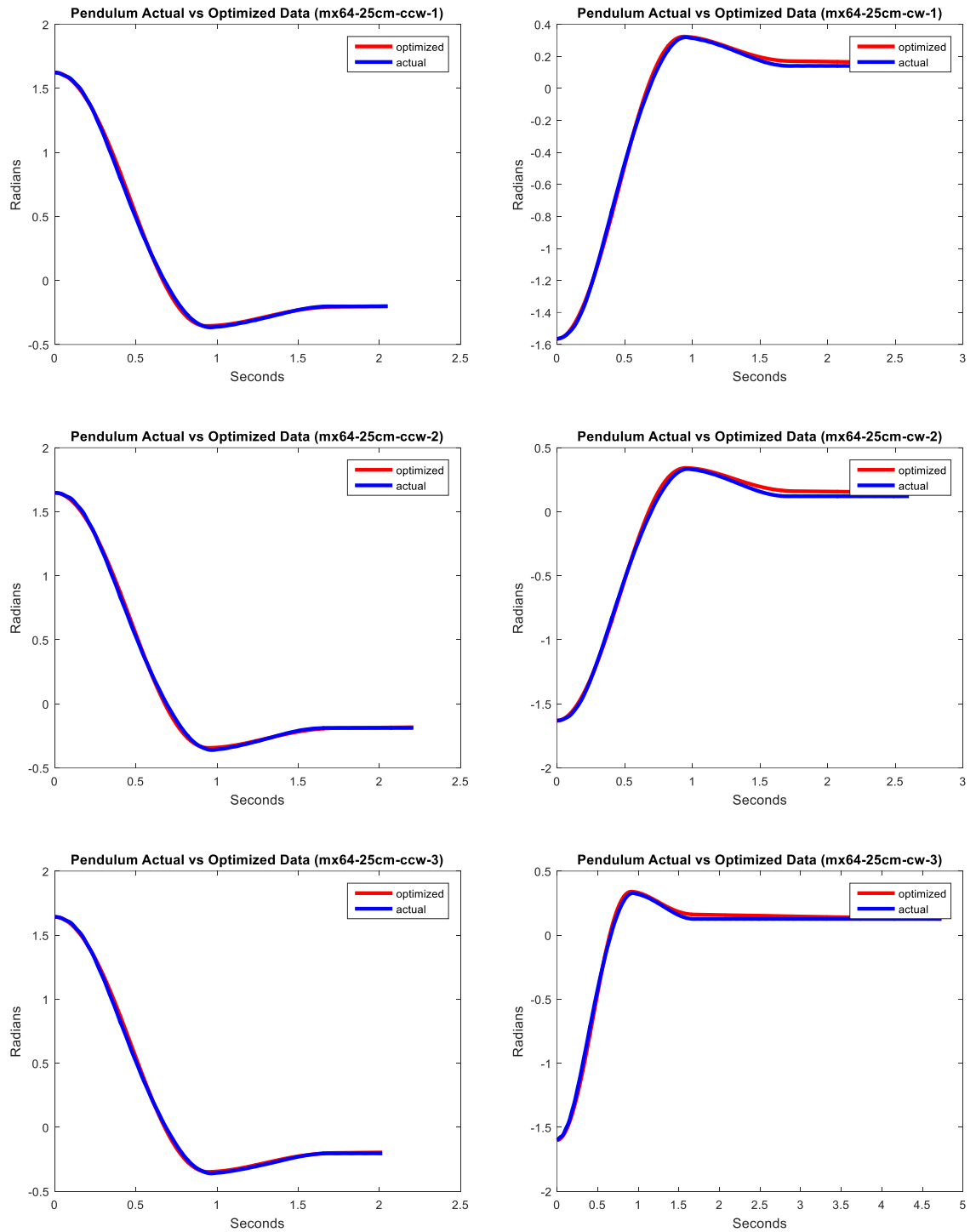


**Figure A.7** MX-64 trials with mass at 35 cm.

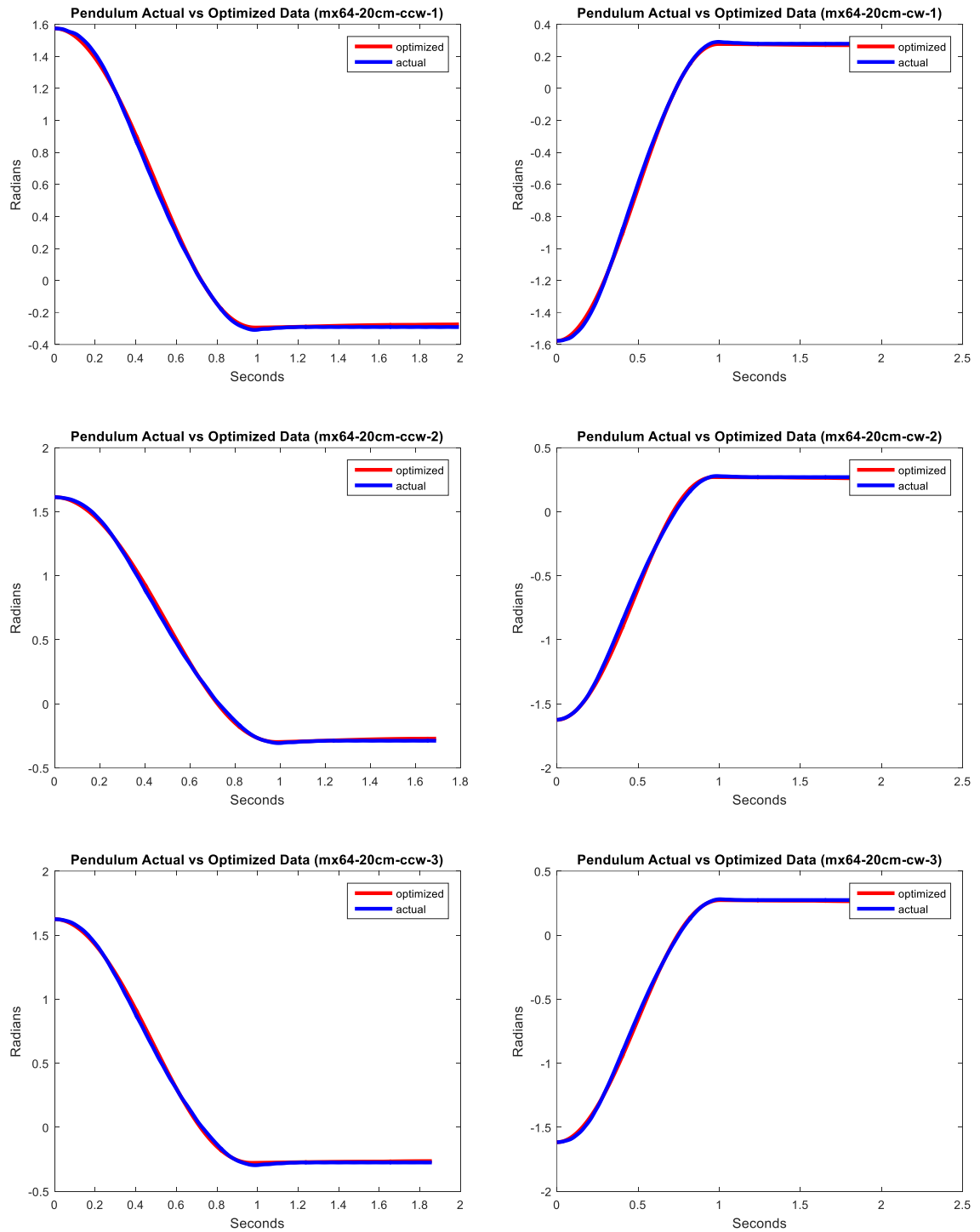


**Figure A.8** MX-64 trials with mass at 30 cm.

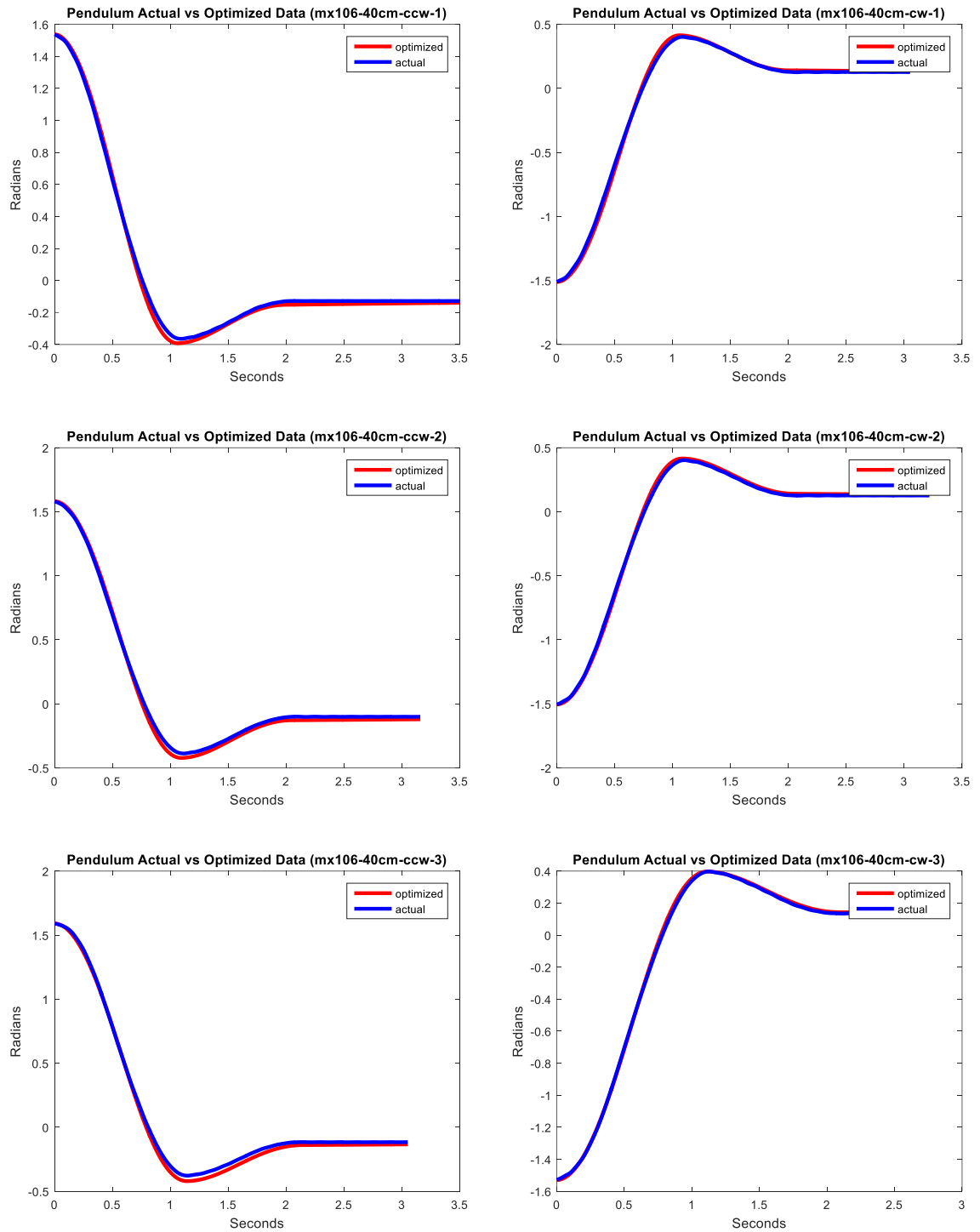




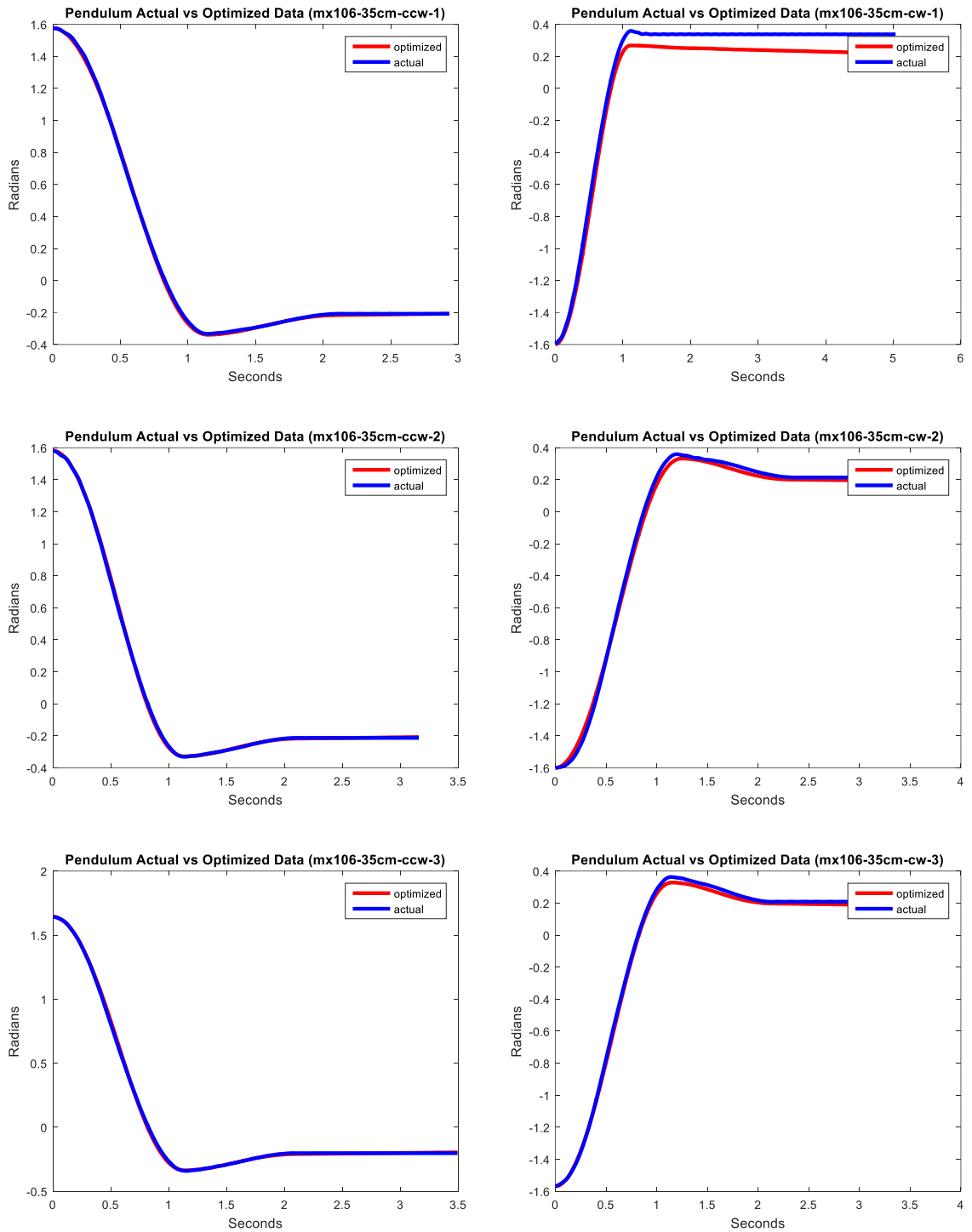
**Figure A.9** MX-64 trials with mass at 25 cm.



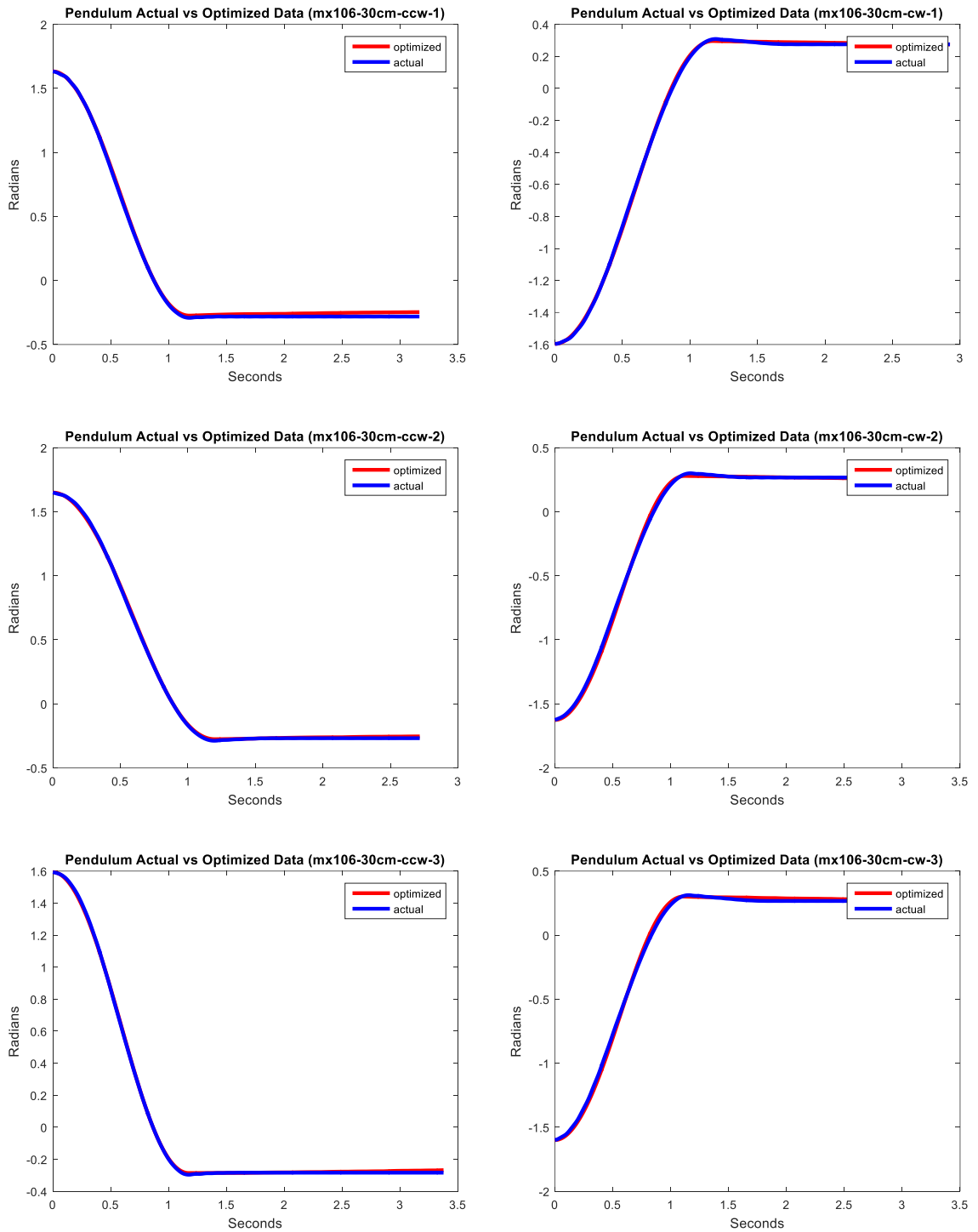
**Figure A.10** MX-64 trials with mass at 20 cm.



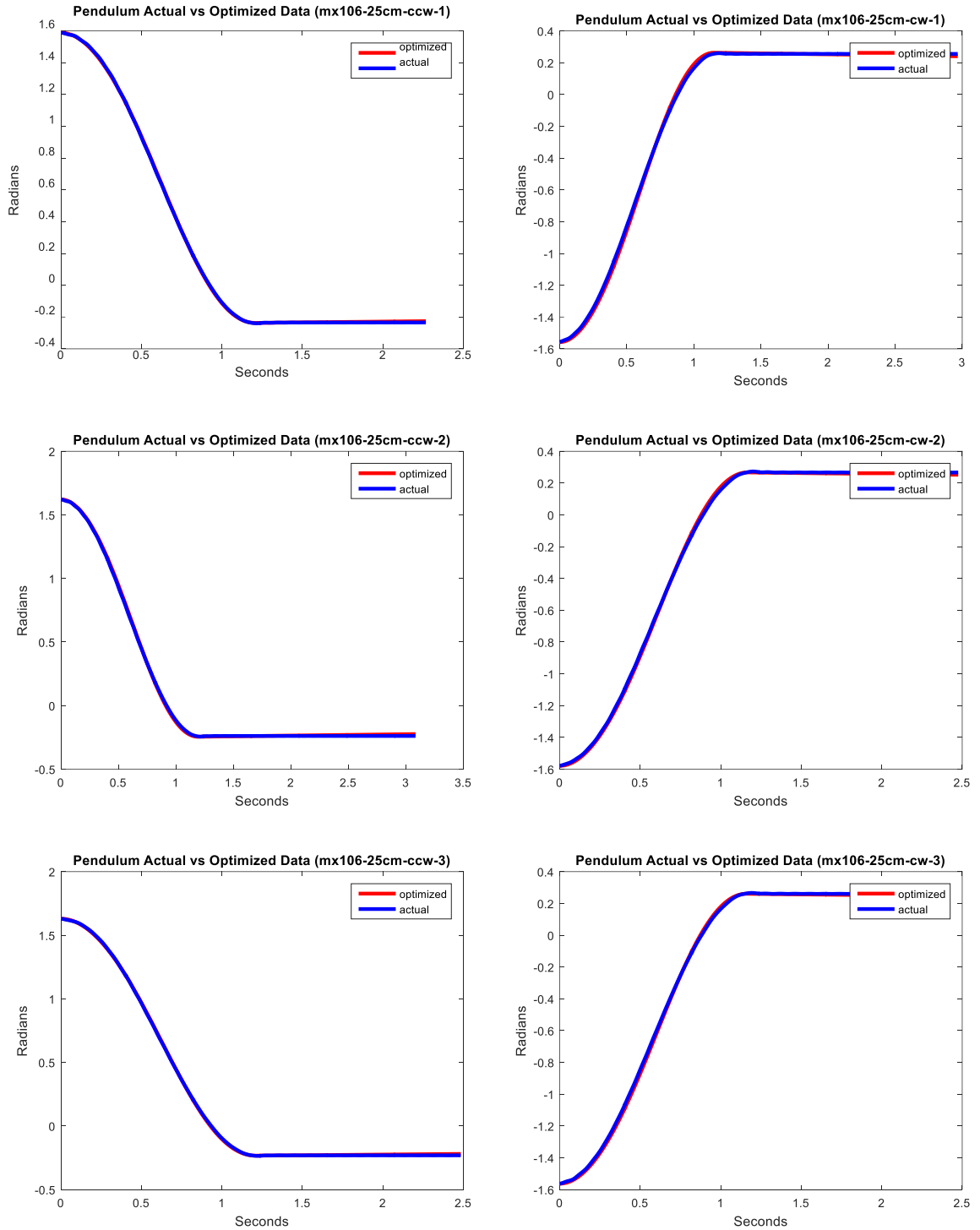
**Figure A.11** MX-106 trials with mass at 40 cm.



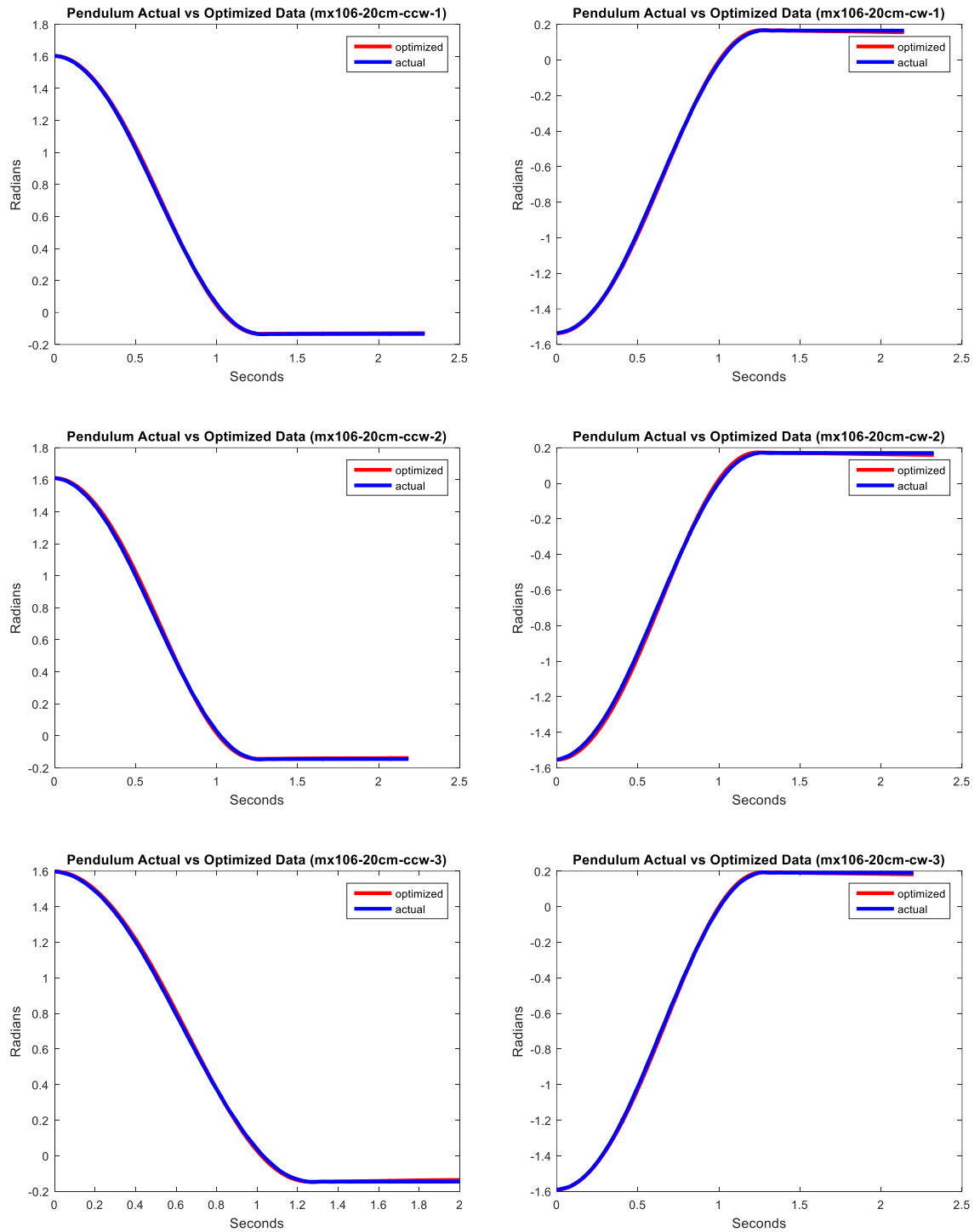
**Figure A.12** MX-106 trials with mass at 35 cm.



**Figure A.13** MX-106 trials with mass at 30 cm.



**Figure A.14** MX-106 trials with mass at 25 cm.



**Figure A.15** MX-106 trials with mass at 20 cm.

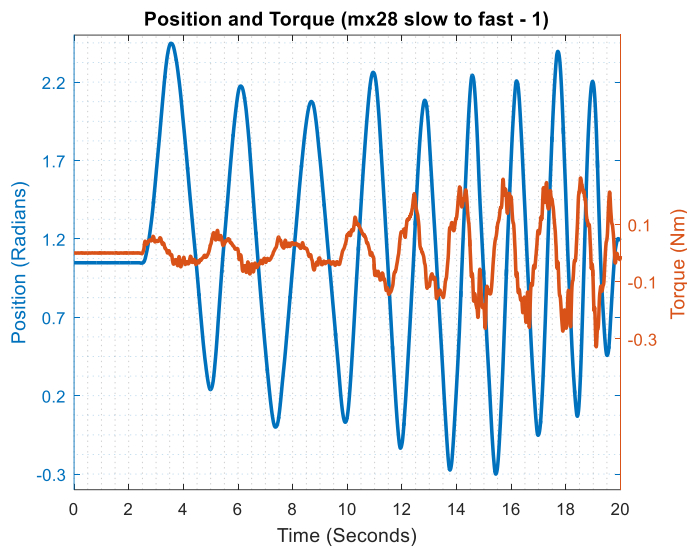
## APPENDIX B

### PLOTS FOR TEST DATA

This appendix covers the plots for all the testing trials with all the motors, MX-28, MX-64 and MX-106. Each motor has 10 trials and each trial has 3 plots.

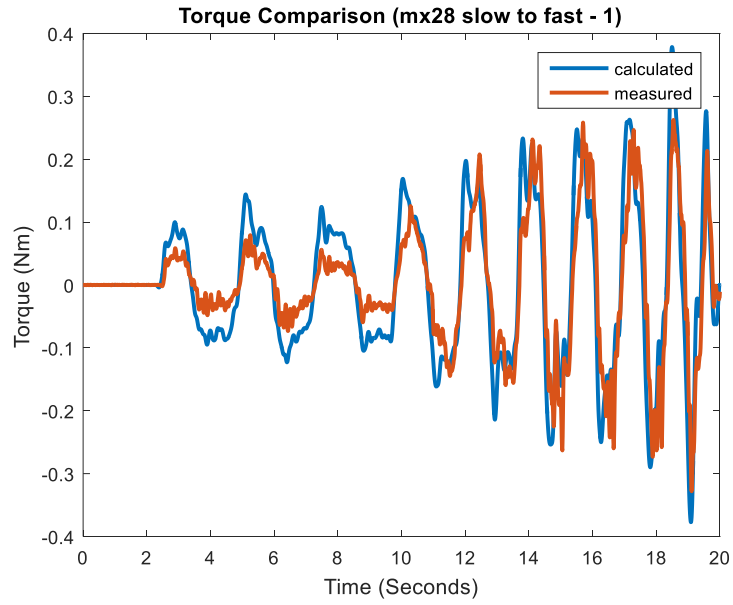
#### B.1 Testing Trials Plots for MX-28

##### B.1.1 Slow to Fast Trial 1 with MX-28

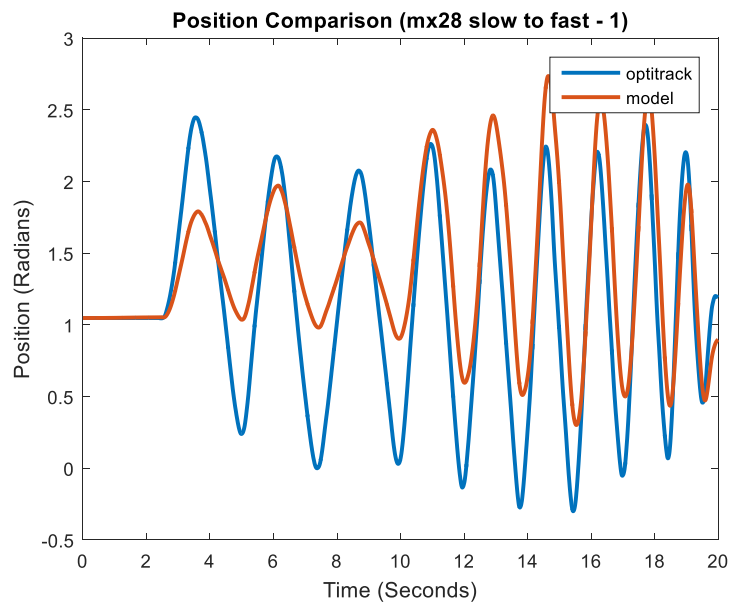


**Figure B.1** MX-28 slow to fast trial 1, measured position and torque.



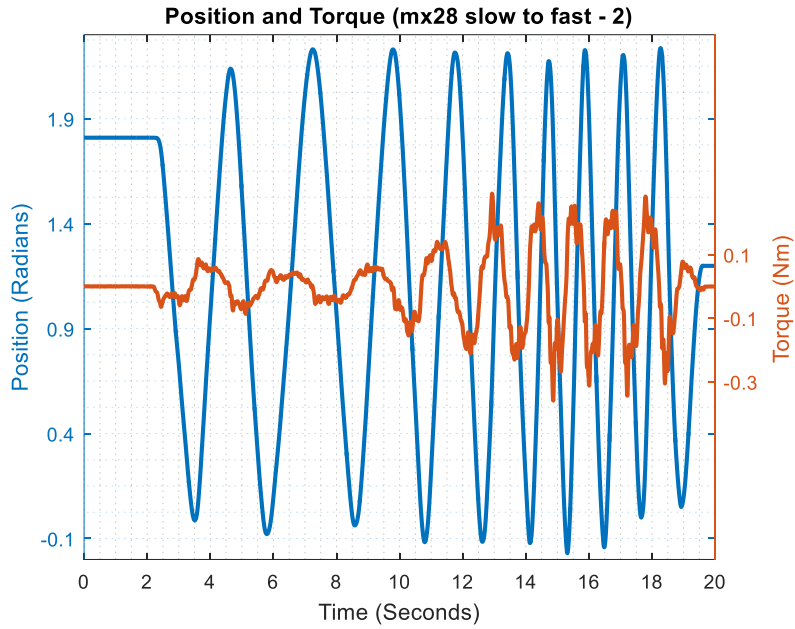


**Figure B.2** MX-28 slow to fast trial 1, measured and calculated torque.

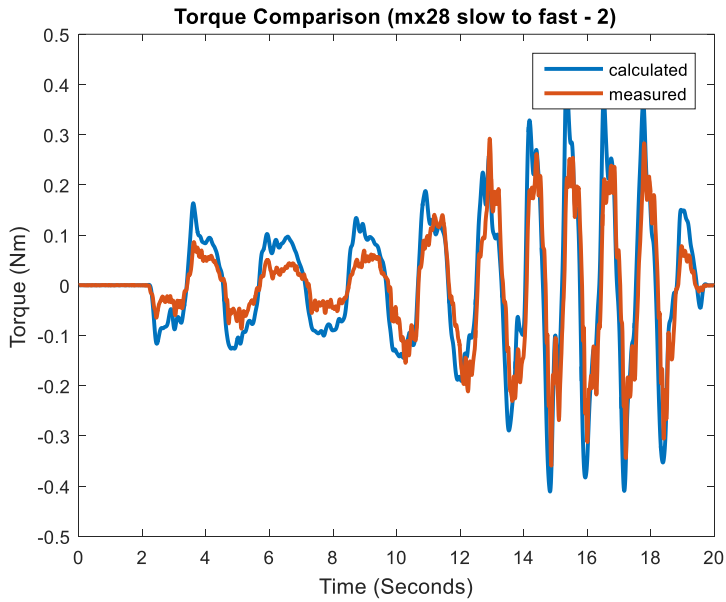


**Figure B.3** MX-28 slow to fast trial 1, recorded and predicted position.

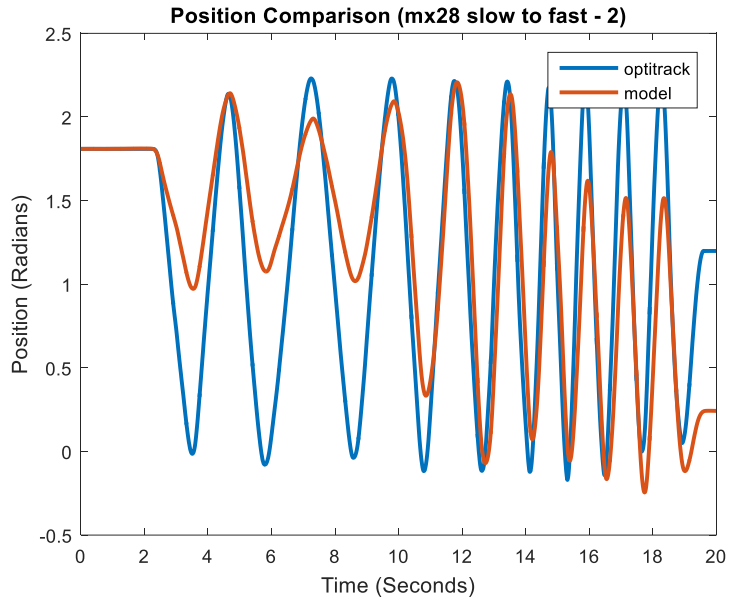
### B.1.2 Slow to Fast Trial 2 with MX-28



**Figure B.4** MX-28 slow to fast trial 2, measured position and torque.

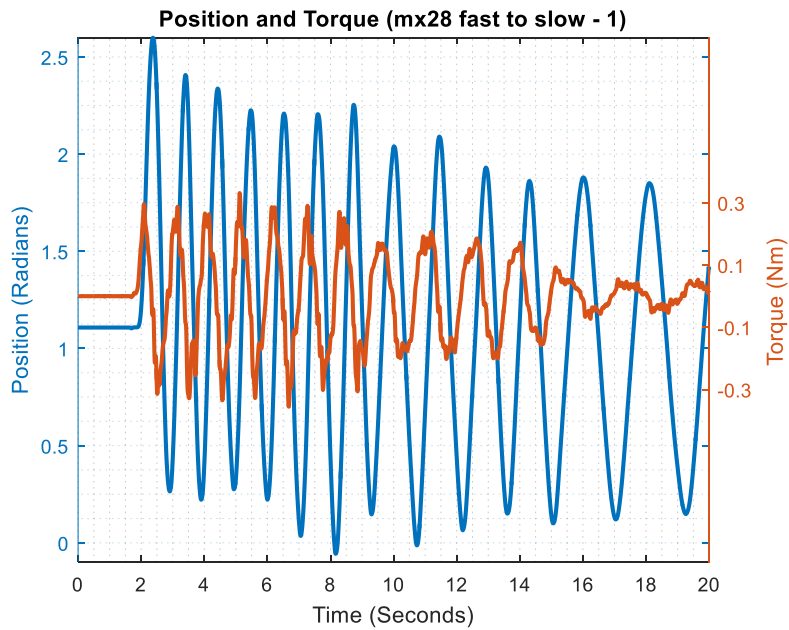


**Figure B.5** MX-28 slow to fast trial 2, measured and calculated torque.

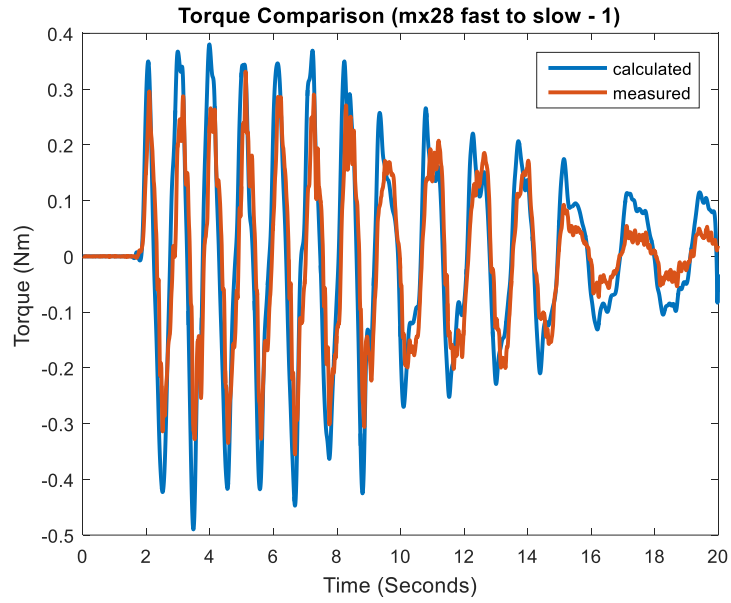


**Figure B.6** MX-28 slow to fast trial 2, recorded and predicted position.

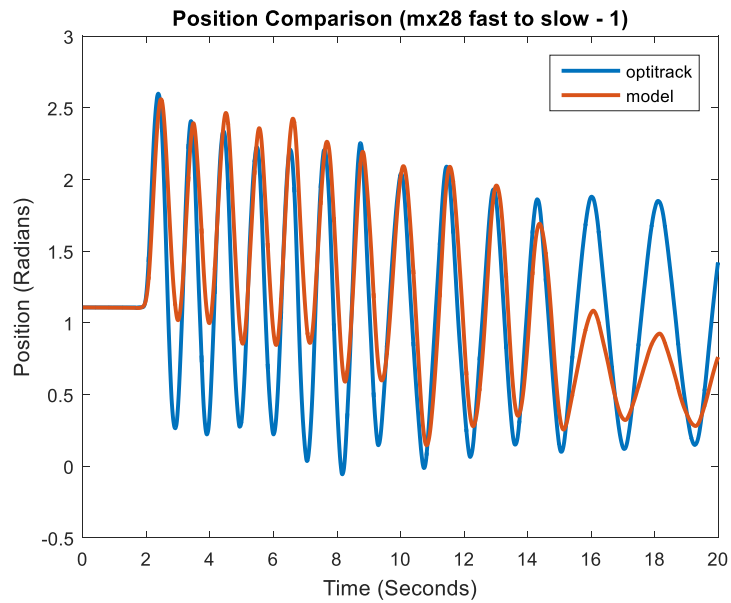
### B.1.3 Fast to Slow Trial 1 with MX-28



**Figure B.7** MX-28 fast to slow trial 1, measured position and torque.

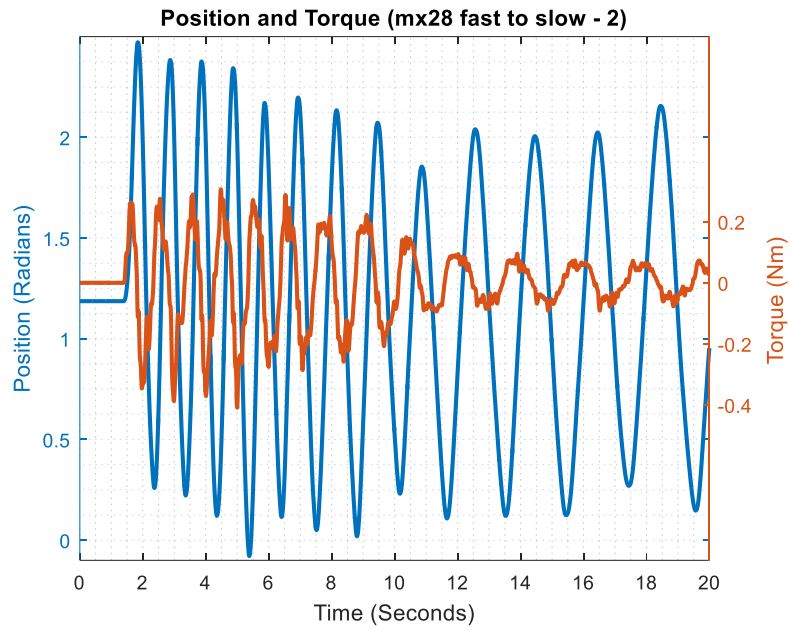


**Figure B.8** MX-28 fast to slow trial 1, measured and calculated torque.

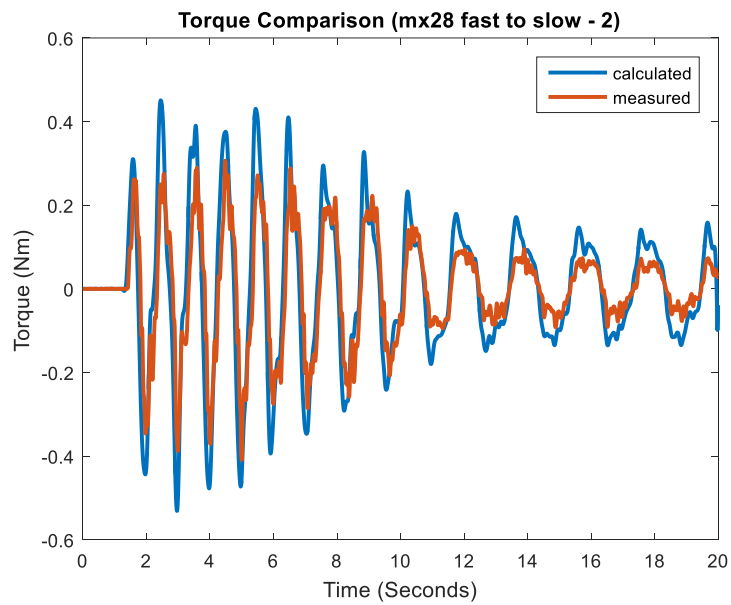


**Figure B.9** MX-28 fast to slow trial 1, recorded and predicted position.

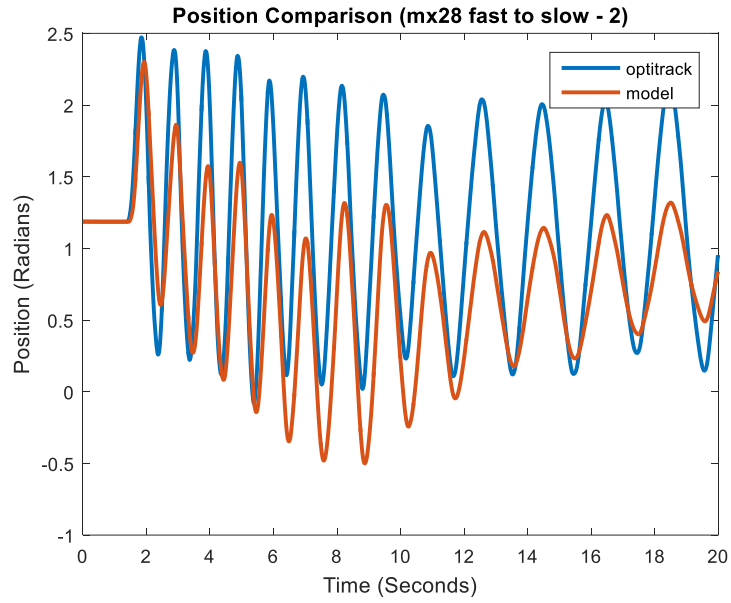
### B.1.4 Fast to Slow Trial 2 with MX-28



**Figure B.10** MX-28 fast to slow trial 2, measured position and torque.

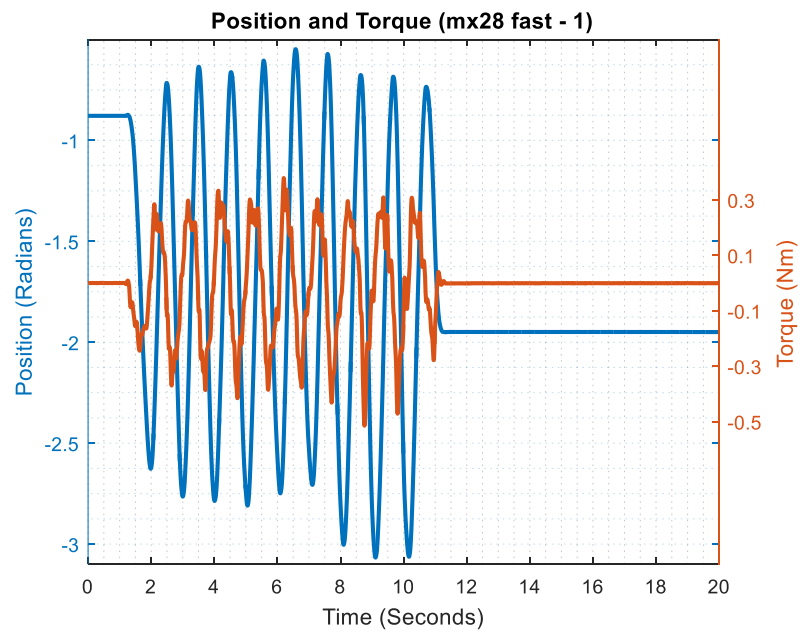


**Figure B.11** MX-28 fast to slow trial 2, measured and calculated torque.

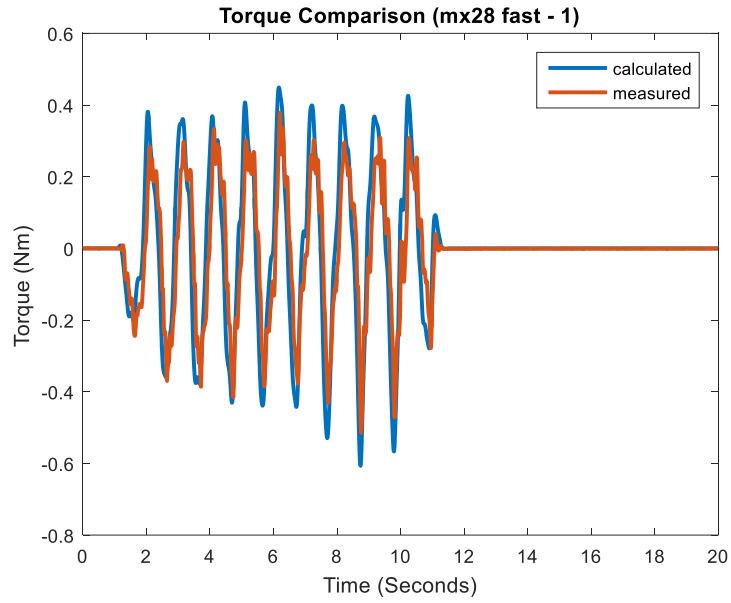


**Figure B.12** MX-28 fast to slow trial 2, recorded and predicted position.

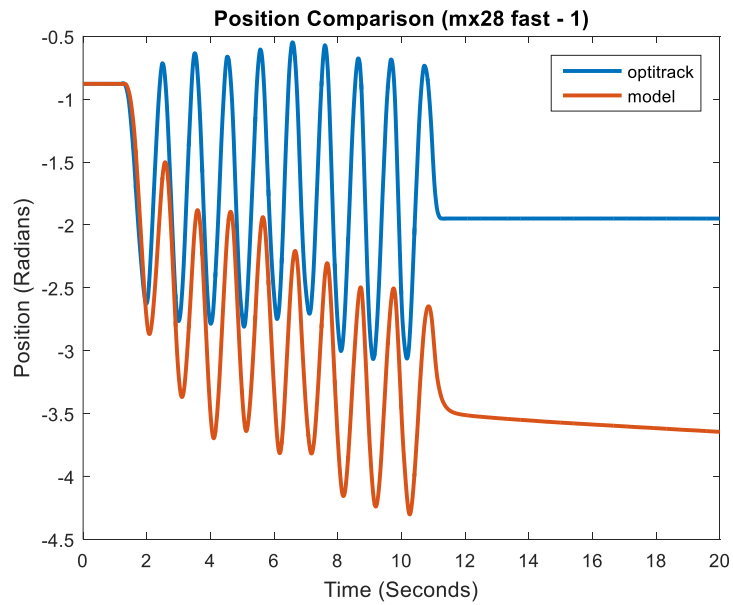
### B.1.5 Fast Trial 1 with MX-28



**Figure B.13** MX-28 fast trial 1, measured position and torque.

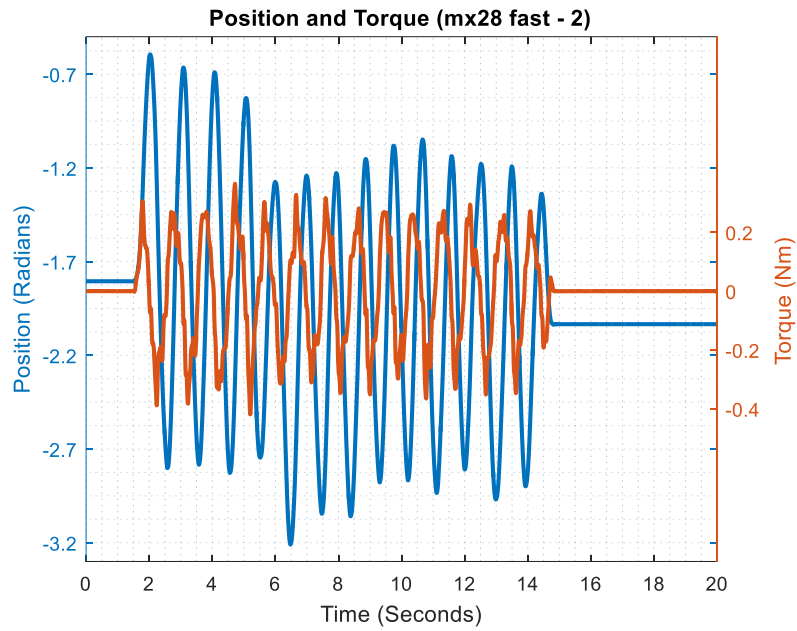


**Figure B.14** MX-28 fast trial 1, measured and calculated torque.

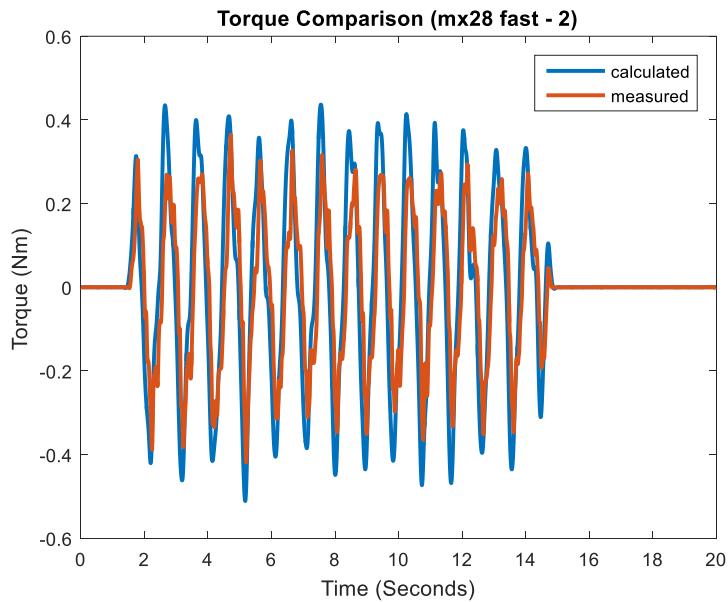


**Figure B.15** MX-28 fast trial 1, recorded and predicted position.

### B.1.6 Fast Trial 2 with MX-28

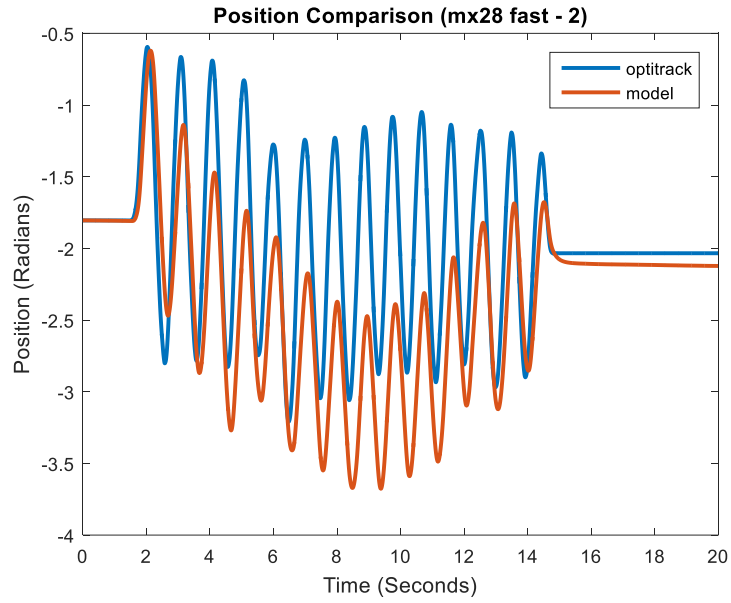


**Figure B.16** MX-28 fast trial 2, measured position and torque.



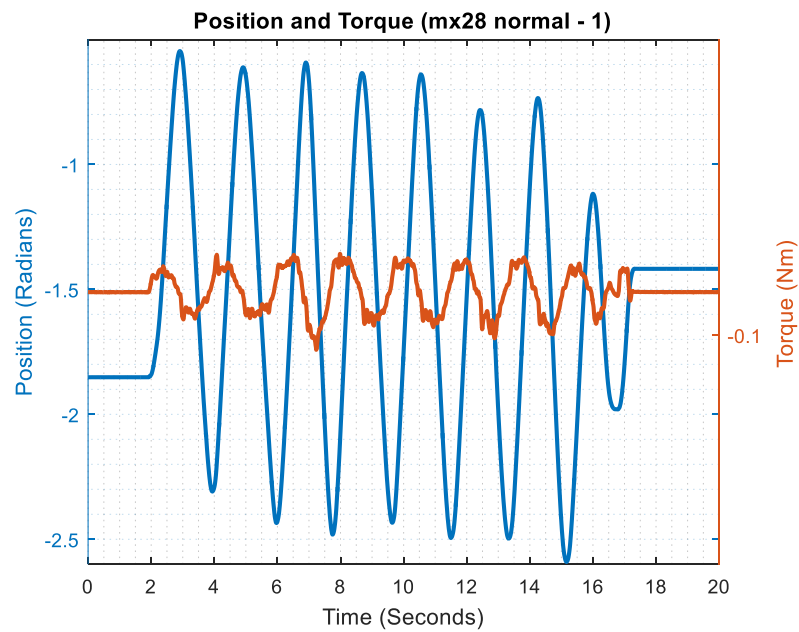
**Figure B.17** MX-28 fast trial 2, measured and calculated torque.



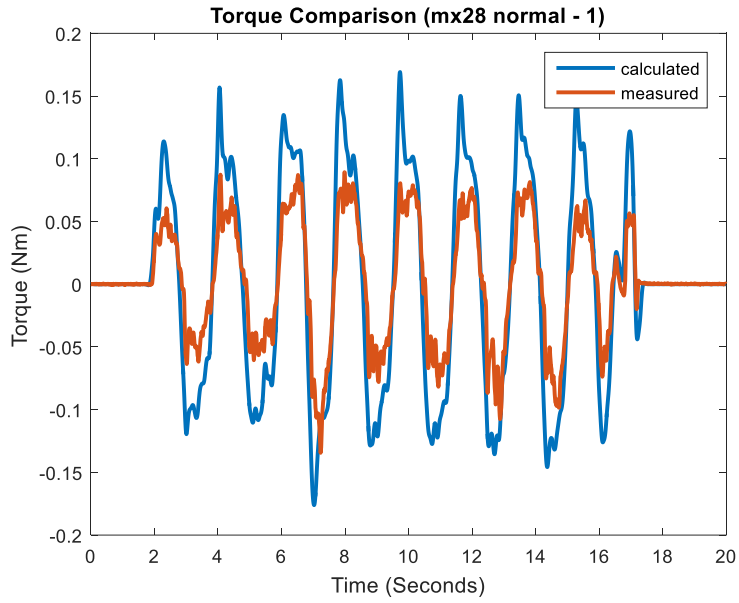


**Figure B.18** MX-28 fast trial 2, recorded and predicted position.

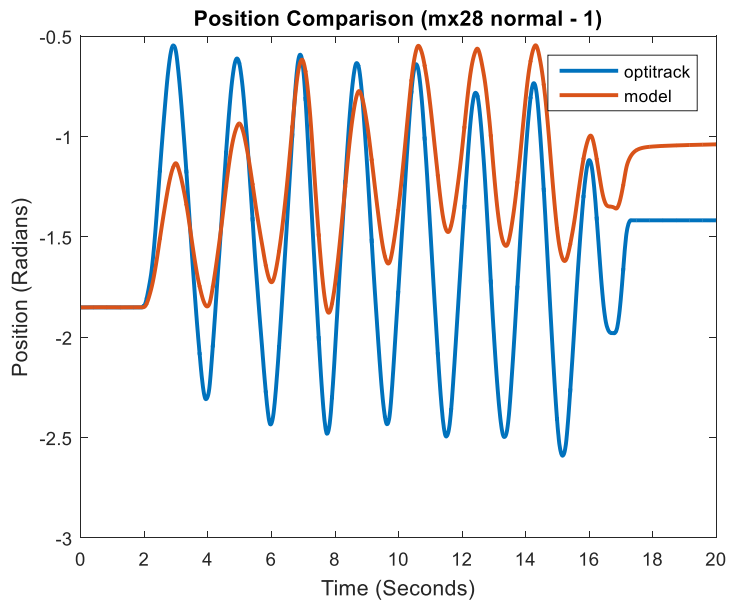
**B.1.7 Normal Trial 1 with MX-28**



**Figure B.19** MX-28 normal trial 1, measured position and torque.

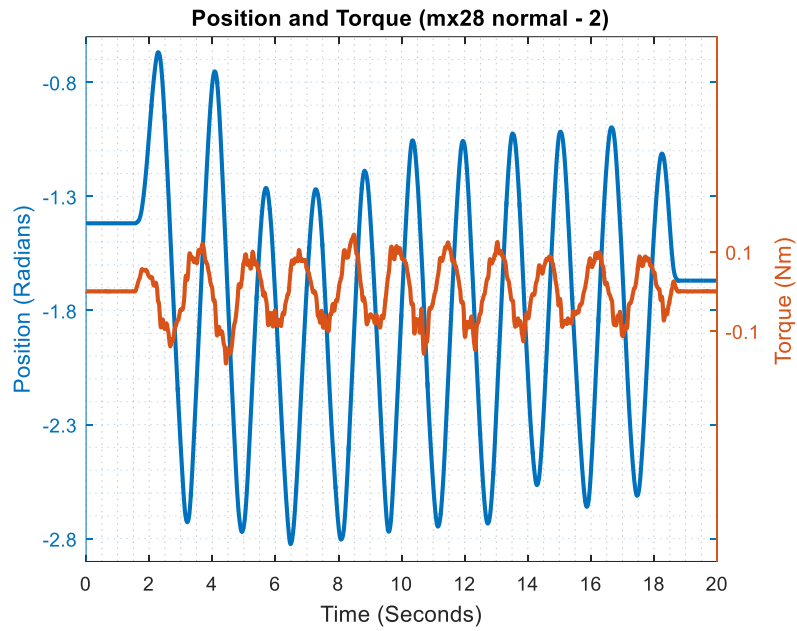


**Figure B.20** MX-28 normal trial 1, measured and calculated torque.

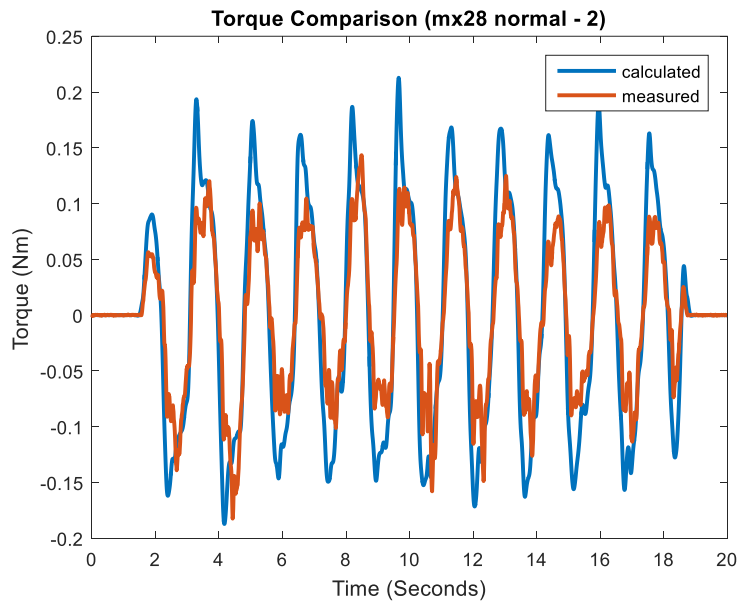


**Figure B.21** MX-28 normal trial 1, recorded and predicted position.

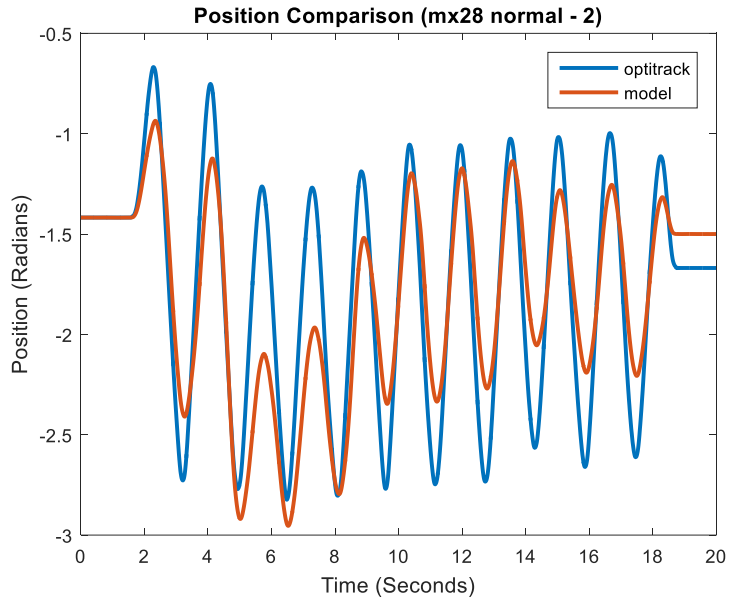
### B.1.8 Normal Trial 2 with MX-28



**Figure B.22** MX-28 normal trial 2, measured position and torque.

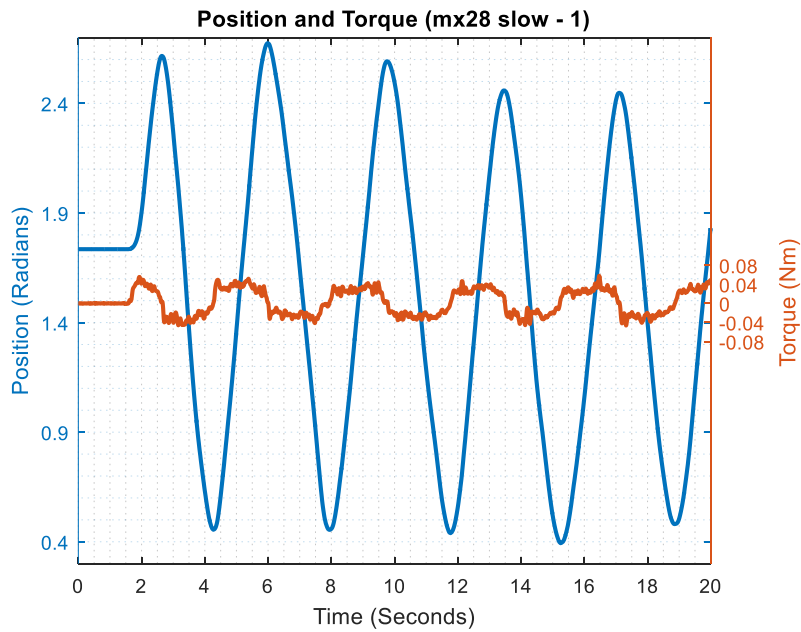


**Figure B.23** MX-28 normal trial 2, measured and calculated torque.

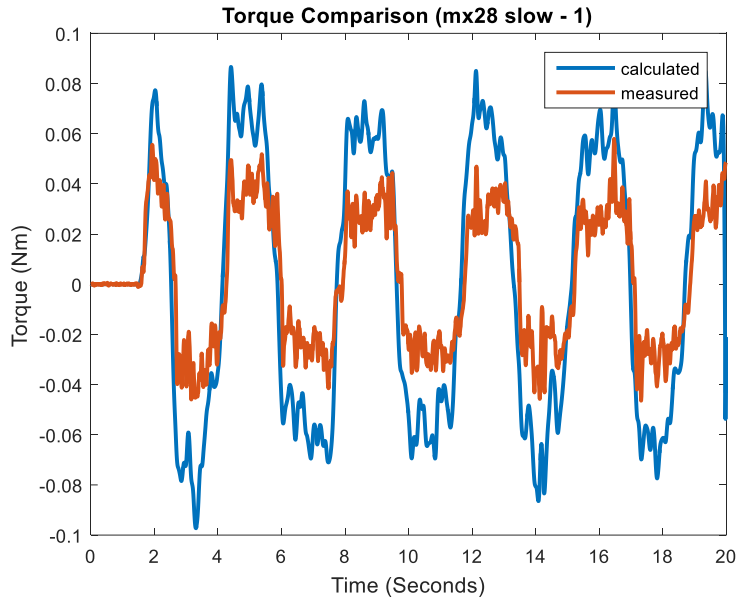


**Figure B.24** MX-28 normal trial 2, recorded and predicted position.

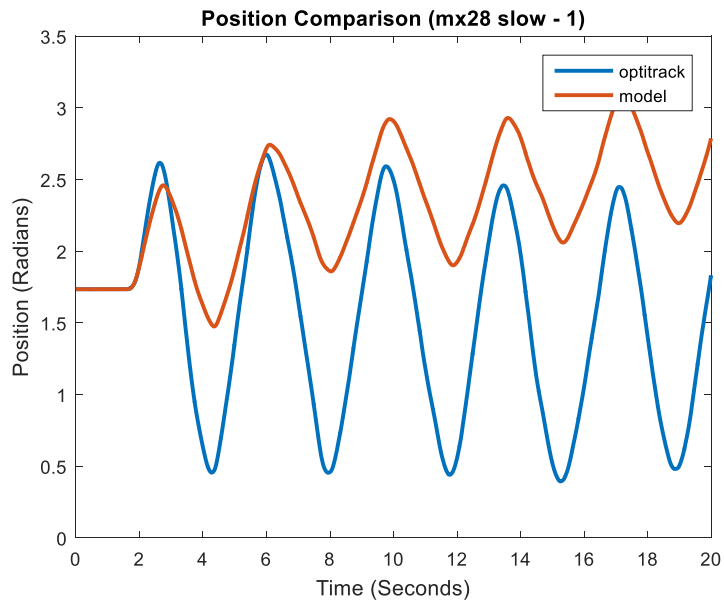
### B.1.9 Slow Trial 1 with MX-28



**Figure B.25** MX-28 slow trial 1, measured position and torque.

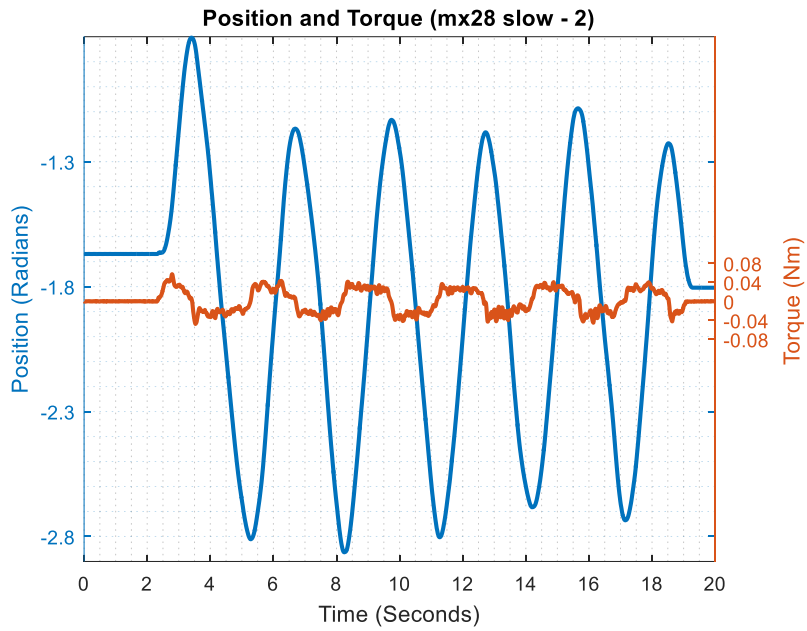


**Figure B.26** MX-28 slow trial 1, measured and calculated torque.

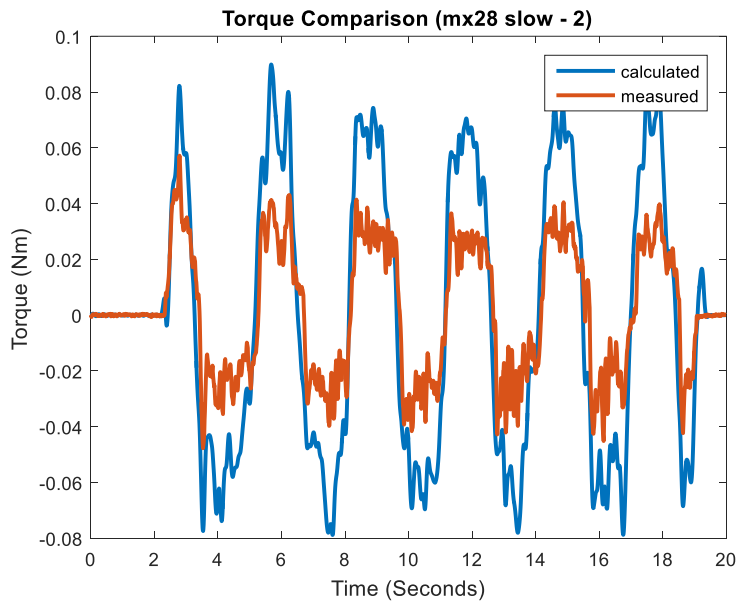


**Figure B.27** MX-28 slow trial 1, recorded and predicted position.

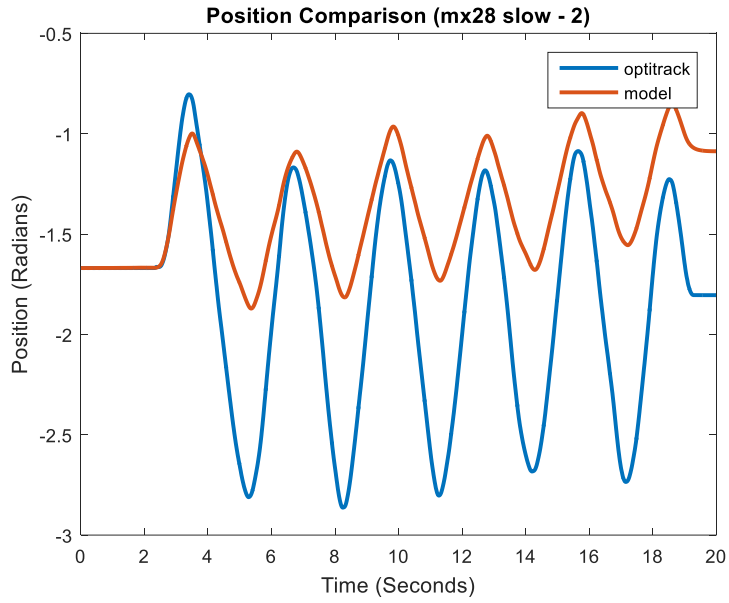
### B.1.10 Slow Trial 2 with MX-28



**Figure B.28** MX-28 slow trial 2, measured position and torque.



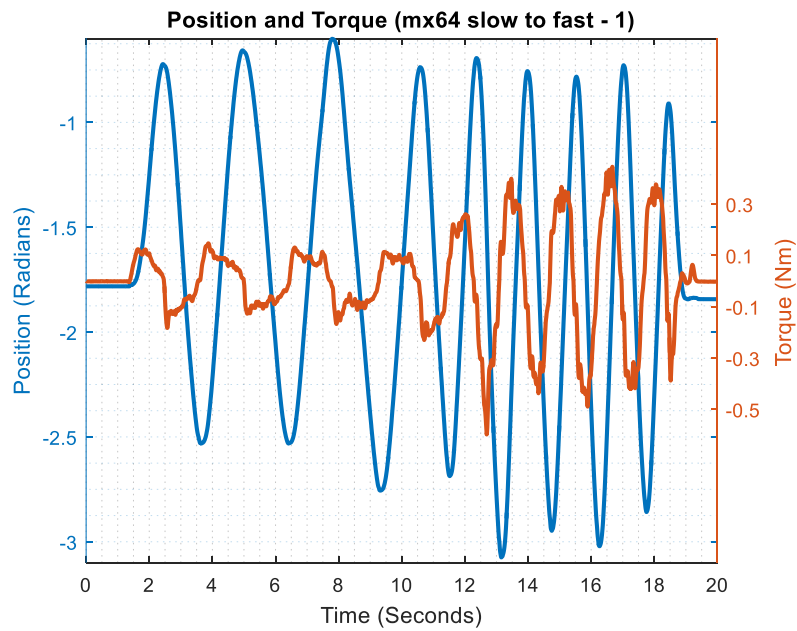
**Figure B.29** MX-28 slow trial 2, measured and calculated torque.



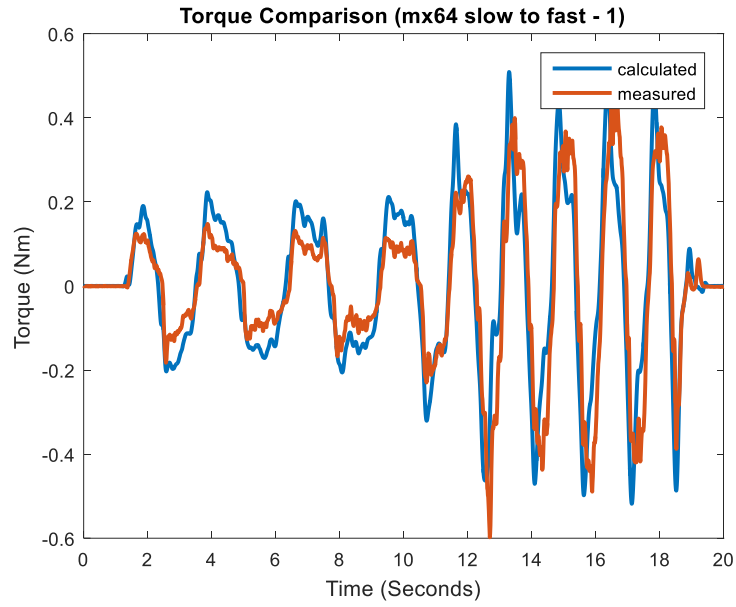
**Figure B.30** MX-28 slow trial 2, recorded and predicted position.

## B.2 Testing Trials Plots for MX-64

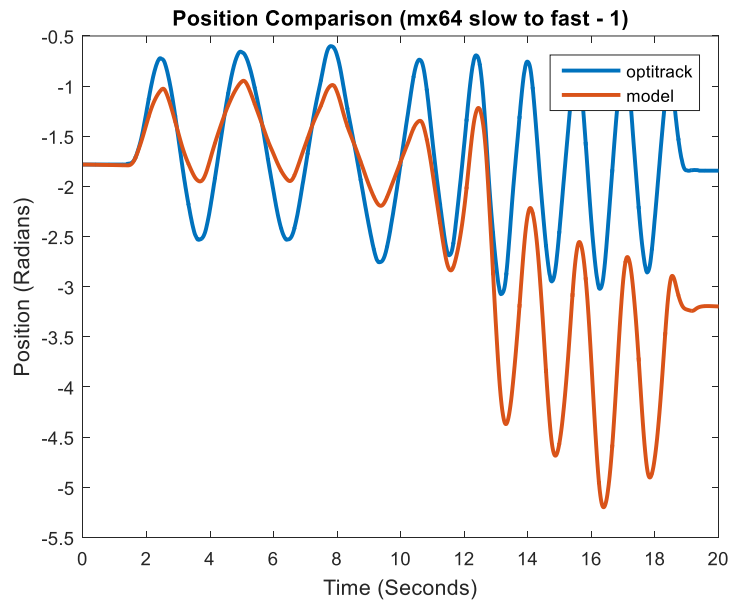
### B.2.1 Slow to Fast Trial 1 with MX-64



**Figure B.31** MX-64 slow to fast trial 1, measured position and torque.



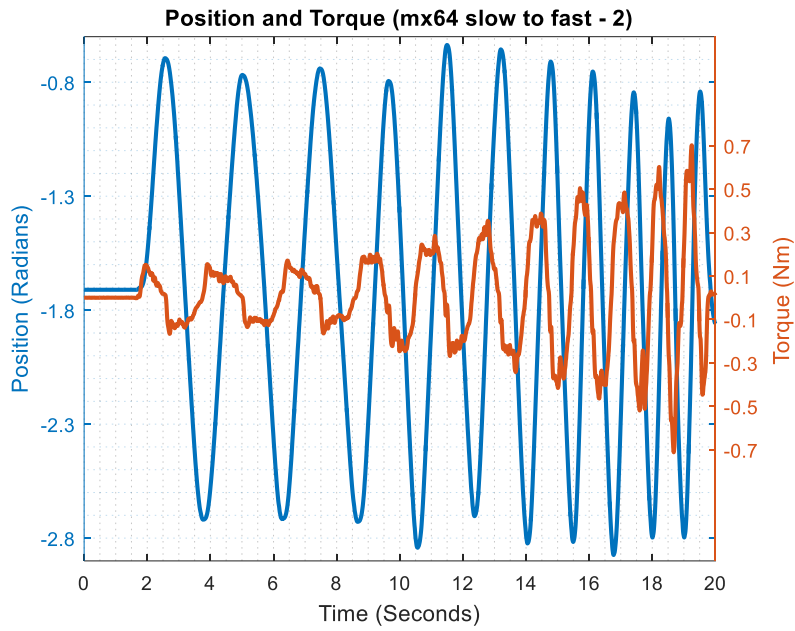
**Figure B.32** MX-64 slow to fast trial 1, measured and calculated torque.



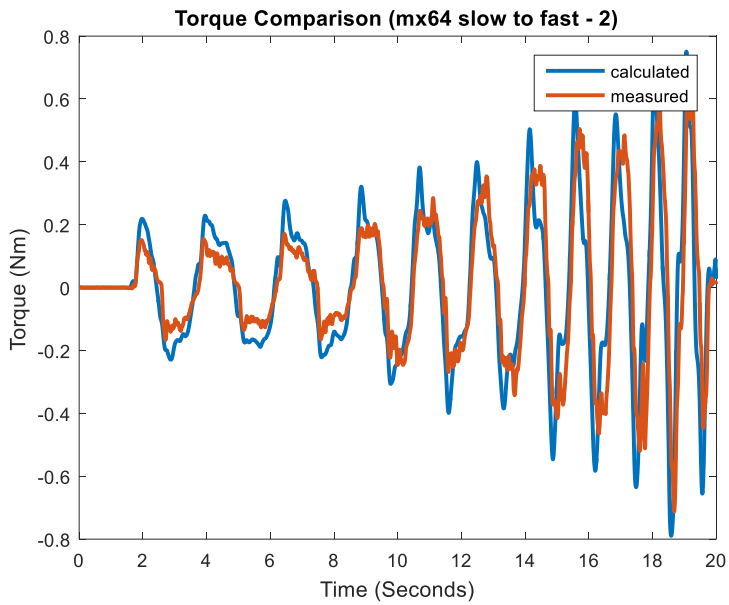
**Figure B.33** MX-64 slow to fast trial 1, recorded and predicted position.



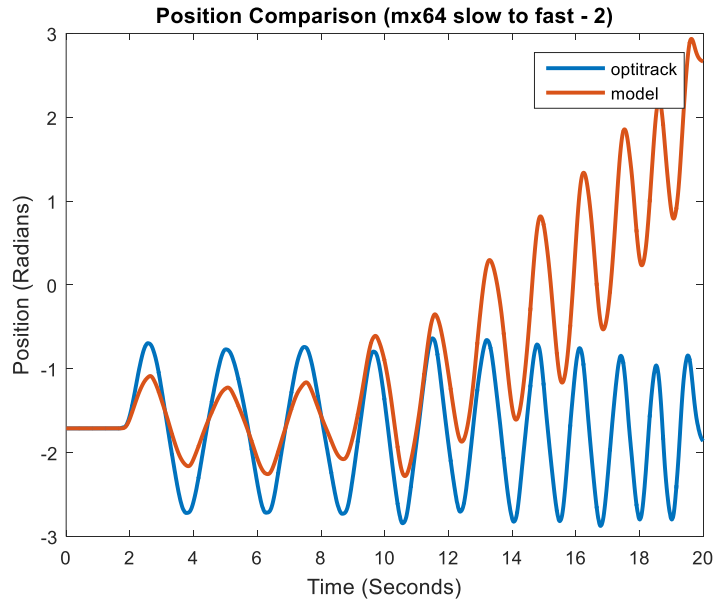
## B.2.2 Slow to Fast Trial 2 with MX-64



**Figure B.34** MX-64 slow to fast trial 2, measured position and torque.

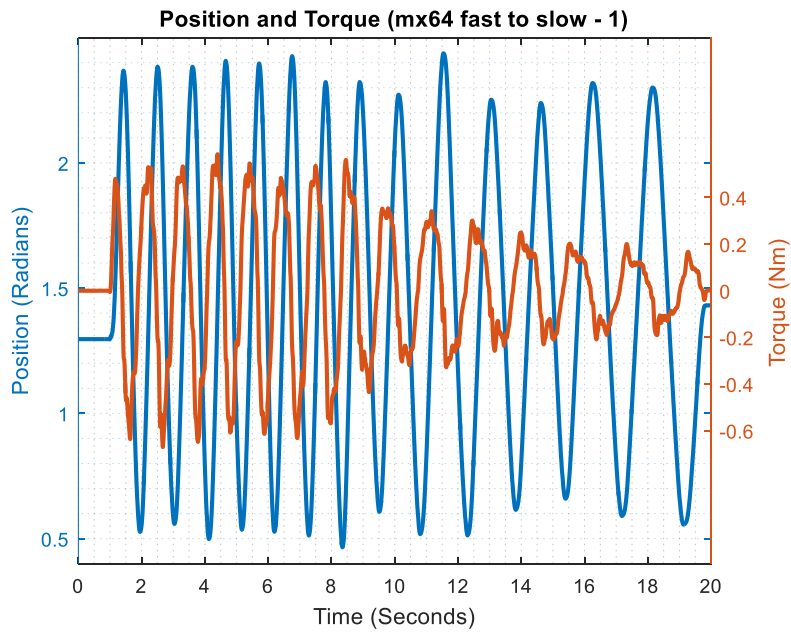


**Figure B.35** MX-64 slow to fast trial 2, measured and calculated torque.

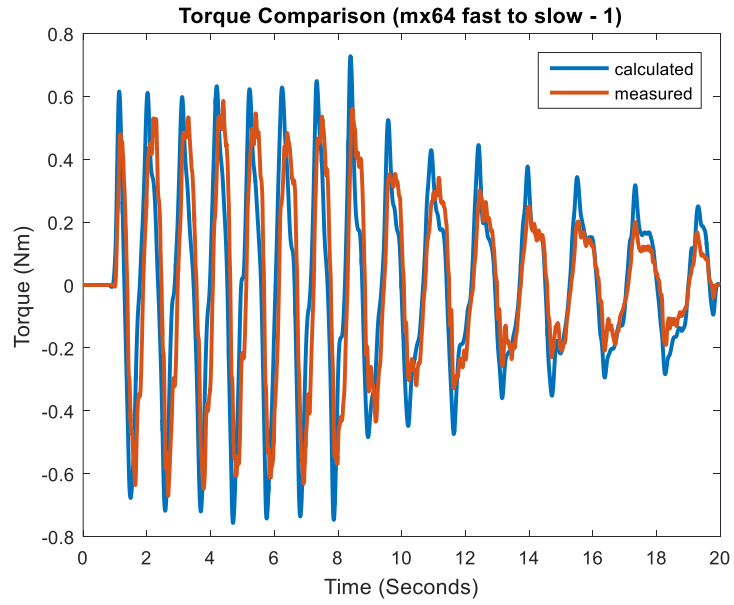


**Figure B.36** MX-64 slow to fast trial 2, recorded and predicted position.

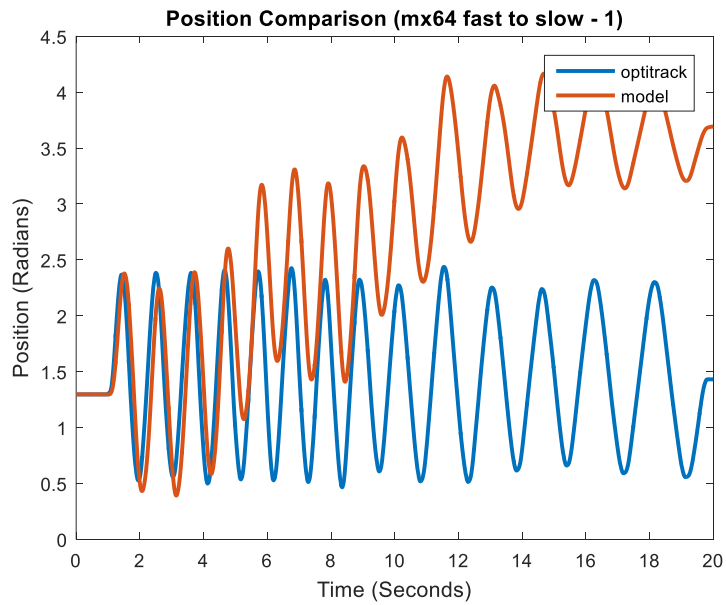
### B.2.3 Fast to Slow Trial 1 with MX-64



**Figure B.37** MX-64 fast to slow trial 1, measured position and torque.

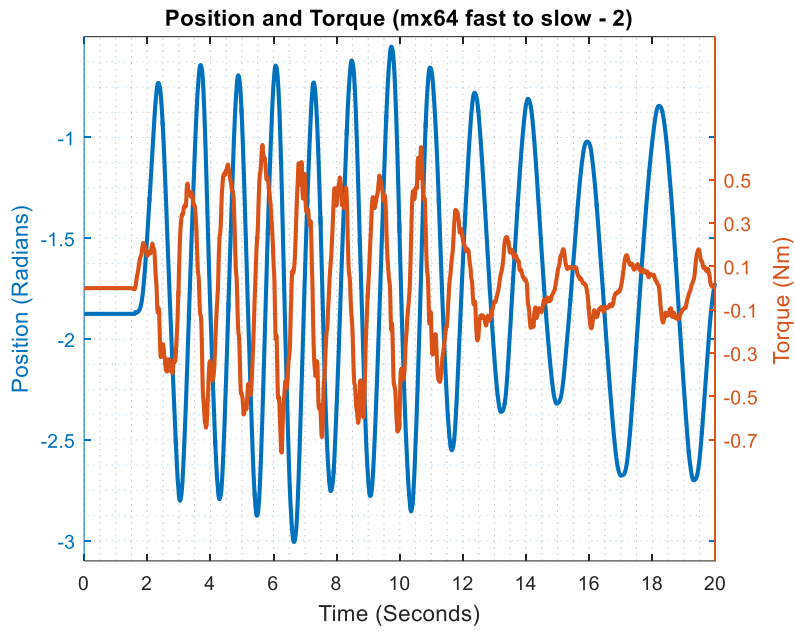


**Figure B.38** MX-64 fast to slow trial 1, measured and calculated torque.

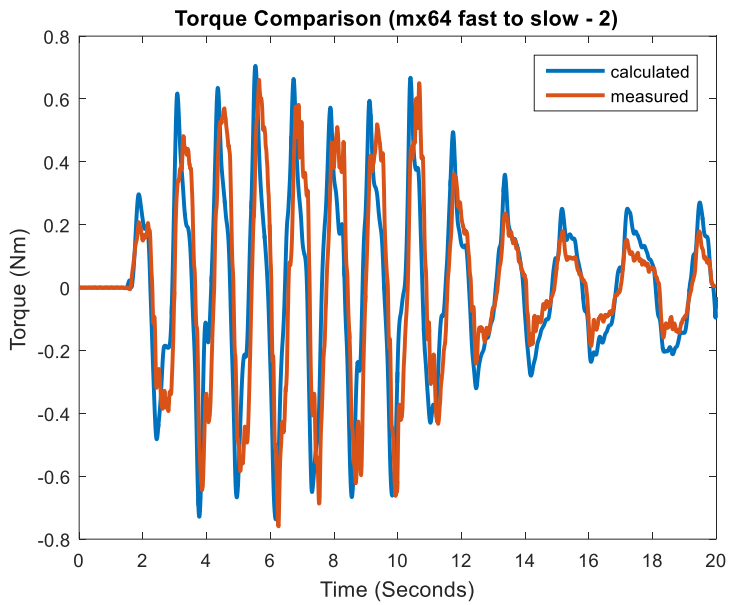


**Figure B.39** MX-64 fast to slow trial 1, recorded and predicted position.

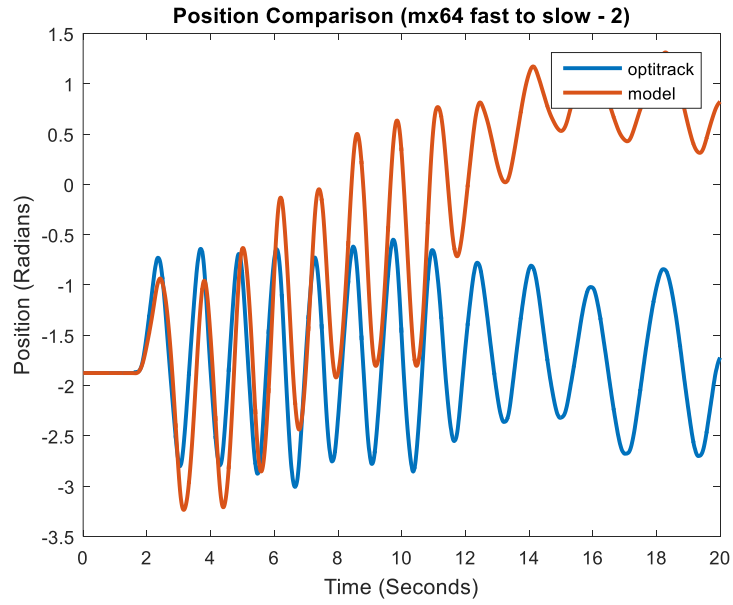
## B.2.4 Fast to Slow Trial 2 with MX-64



**Figure B.40** MX-64 fast to slow trial 2, measured position and torque.

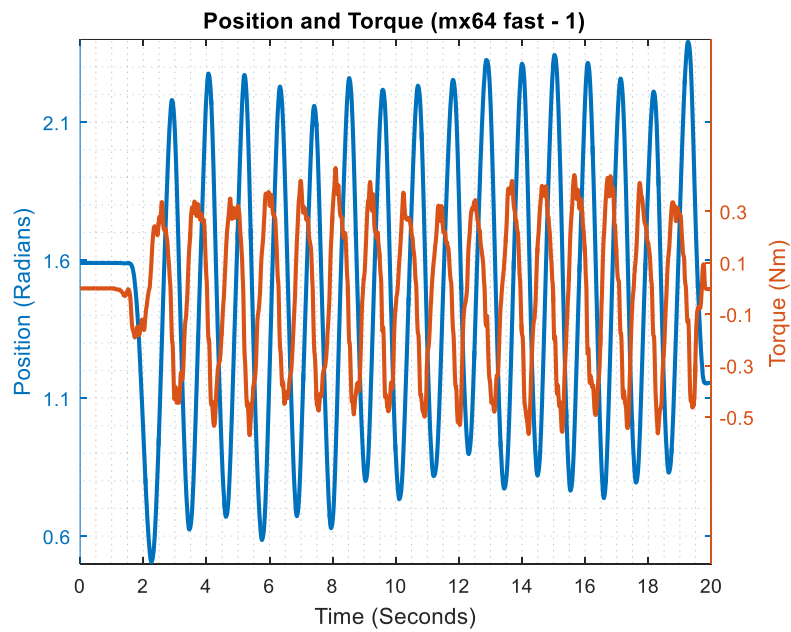


**Figure B.41** MX-64 fast to slow trial 2, measured and calculated torque.

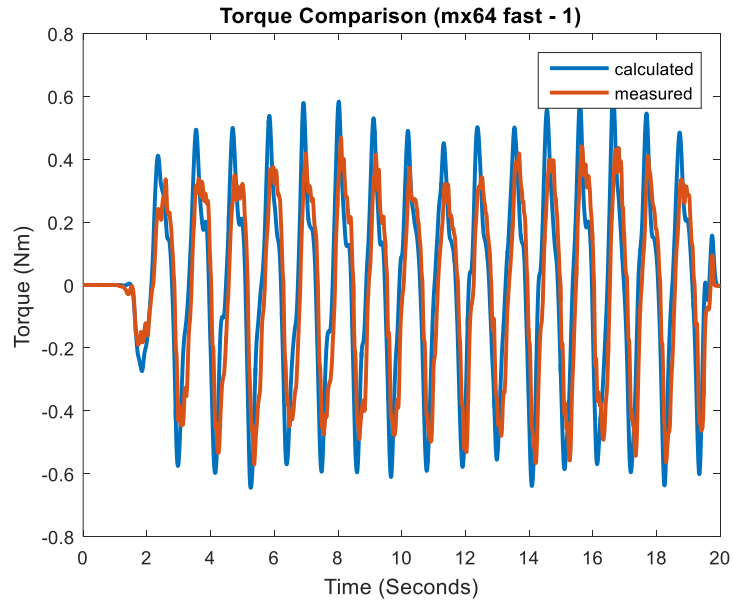


**Figure B.42** MX-64 fast to slow trial 2, recorded and predicted position.

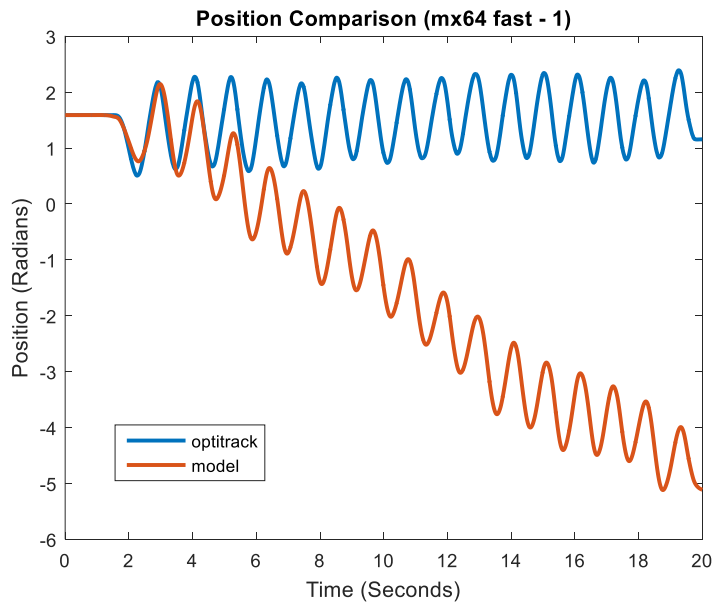
### B.2.5 Fast Trial 1 with MX-64



**Figure B.43** MX-64 fast trial 1, measured position and torque.

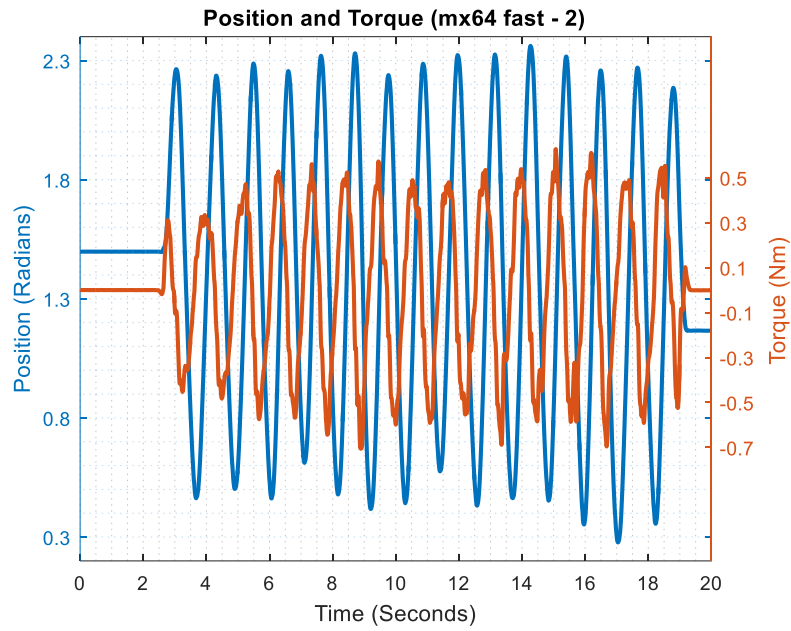


**Figure B.44** MX-64 fast trial 1, measured and calculated torque.

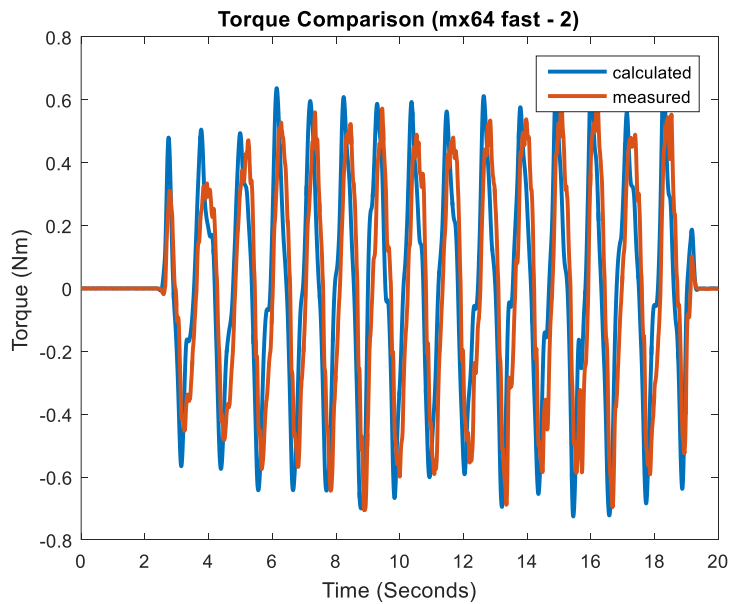


**Figure B.45** MX-64 fast trial 1, recorded and predicted position.

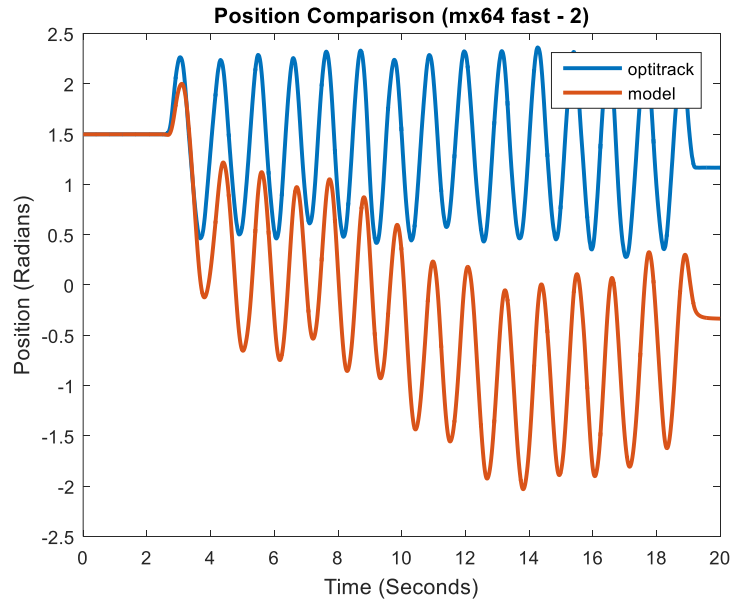
## B.2.6 Fast Trial 2 with MX-64



**Figure B.46** MX-64 fast trial 2, measured position and torque.

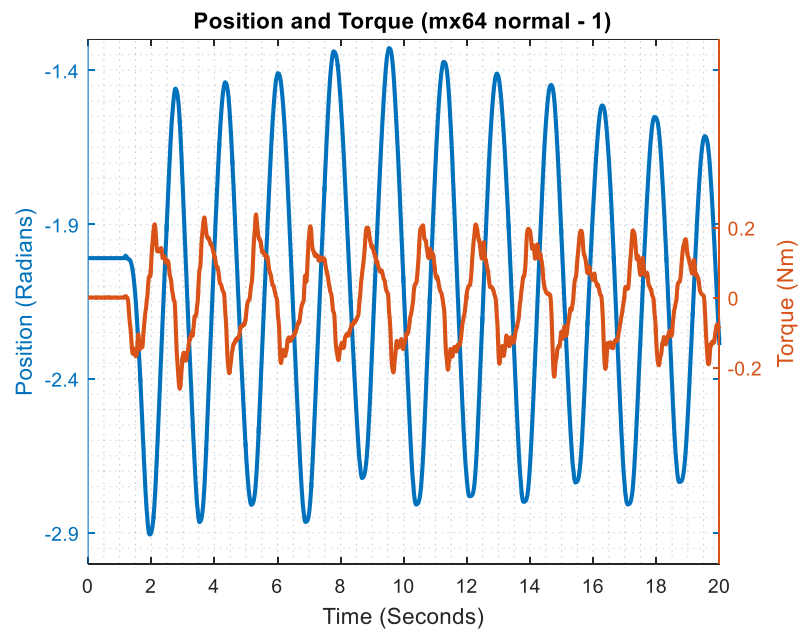


**Figure B.47** MX-64 fast trial 2, measured and calculated torque.



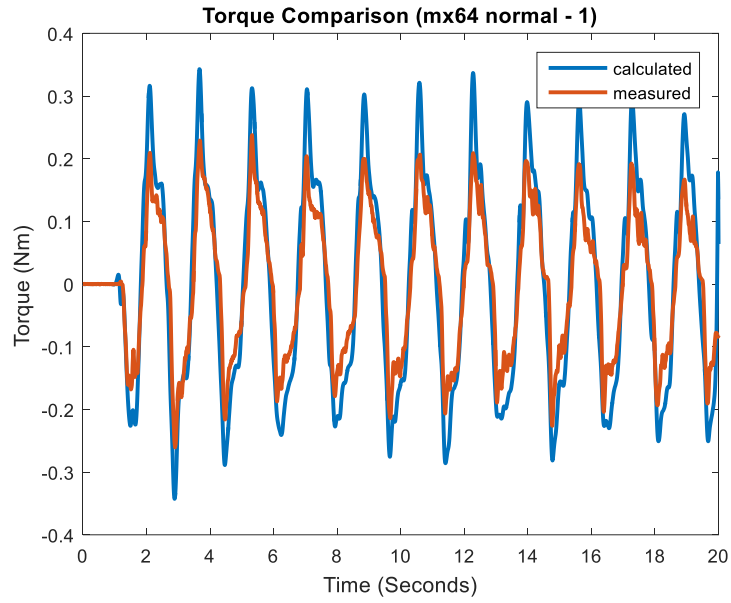
**Figure B.48** MX-64 fast trial 2, recorded and predicted position.

### B.2.7 Normal Trial 1 with MX-64

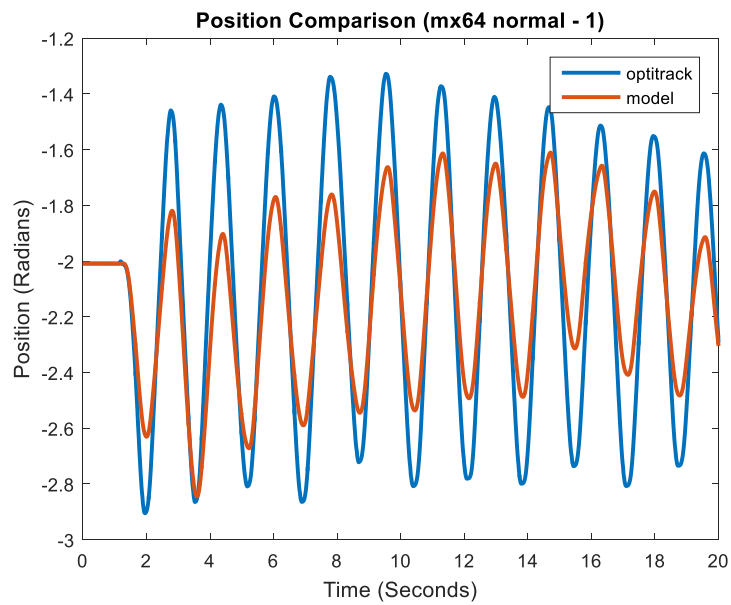


**Figure B.49** MX-64 normal trial 1, measured position and torque.





**Figure B.50** MX-64 normal trial 1, measured and calculated torque.



**Figure B.51** MX-64 normal trial 1, recorded and predicted position.

### B.2.8 Normal Trial 2 with MX-64

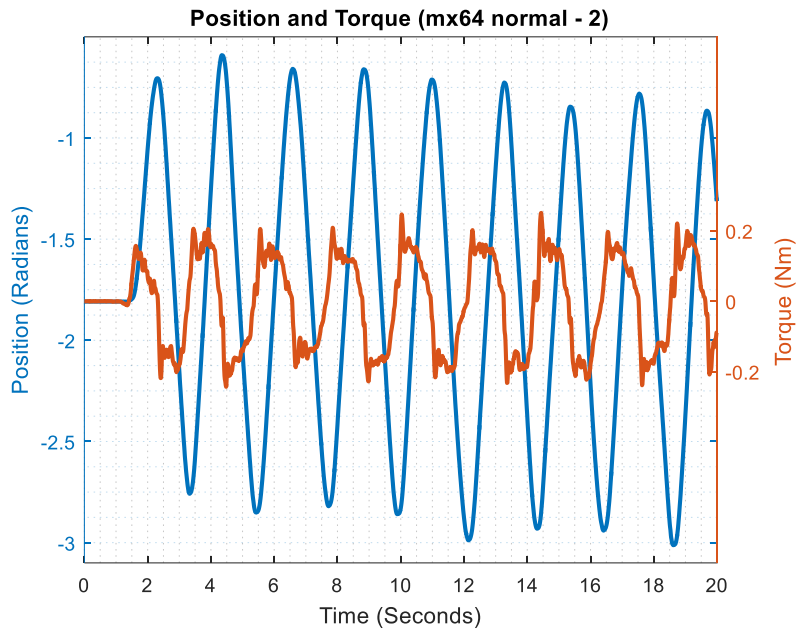


Figure B.52 MX-64 normal trial 2, measured position and torque.

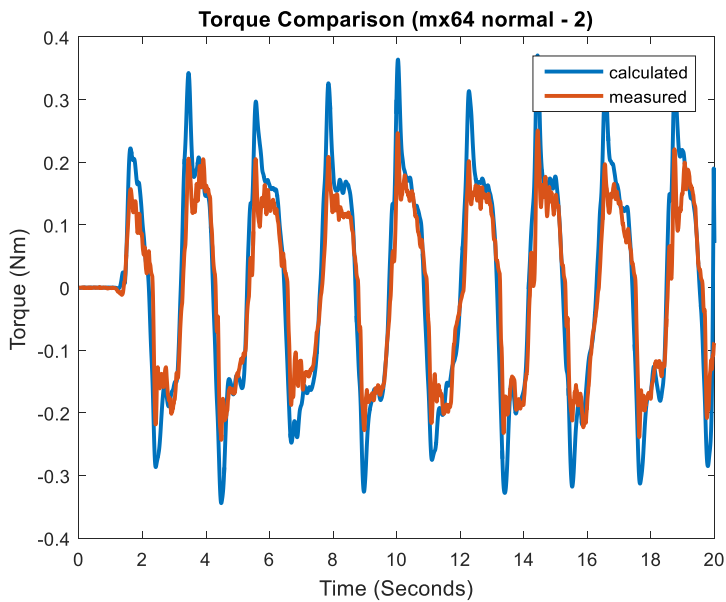
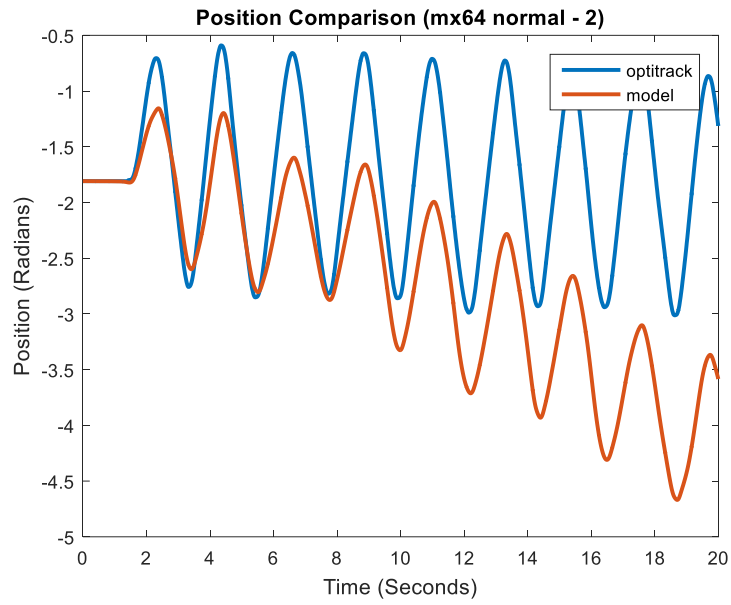
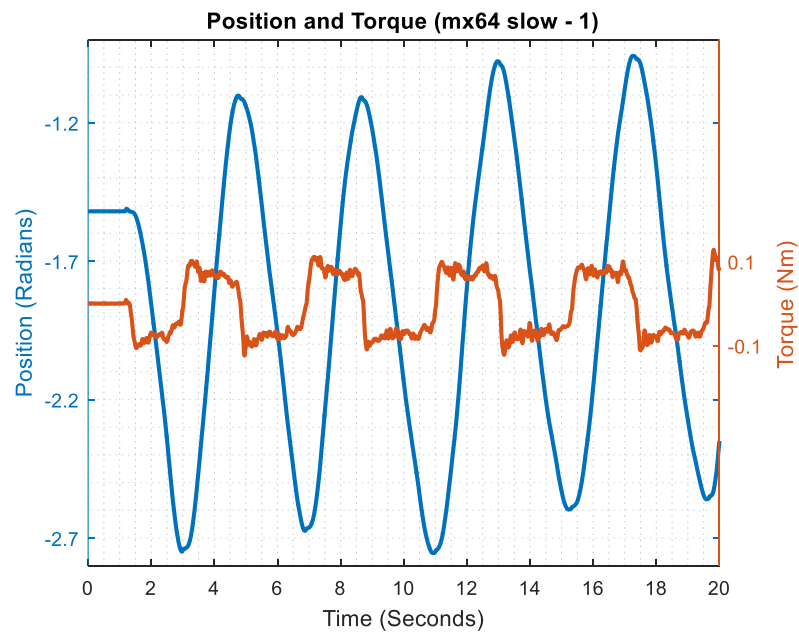


Figure B.53 MX-64 normal trial 2, measured and calculated torque.

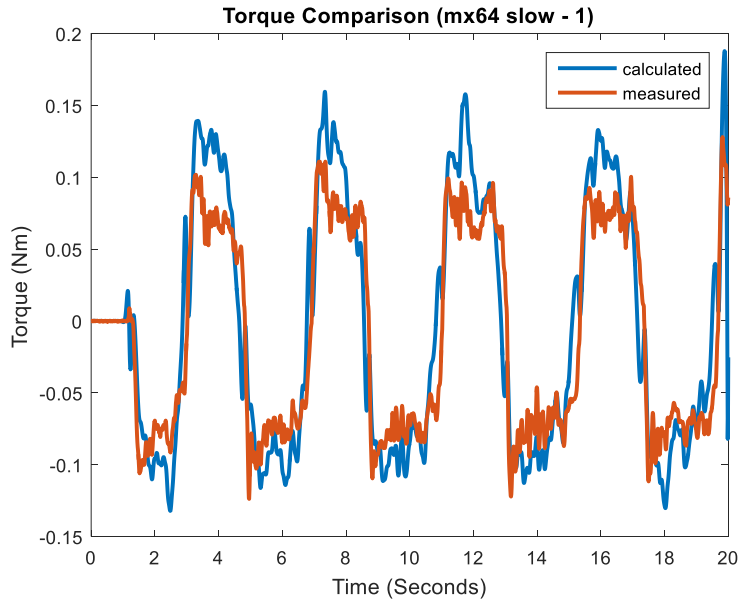


**Figure B.54** MX-64 normal trial 2, recorded and predicted position.

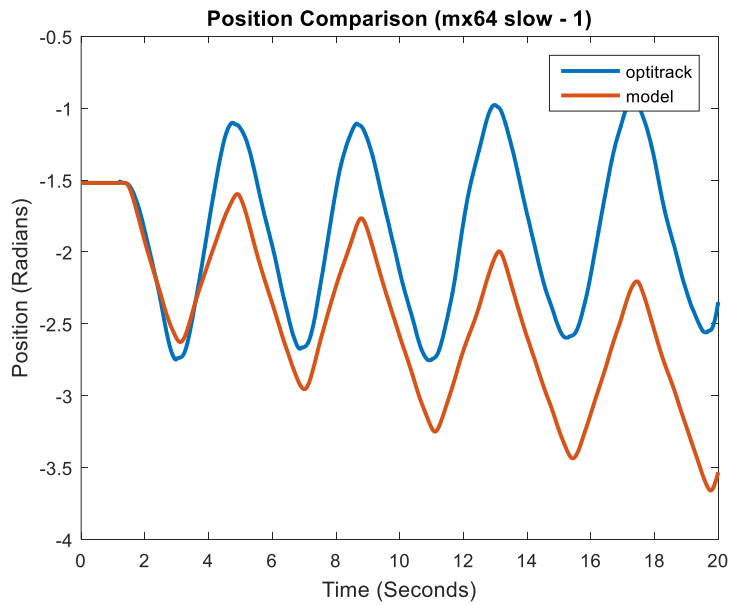
### B.2.9 Slow Trial 1 with MX-64



**Figure B.55** MX-64 slow trial 1, measured position and torque.

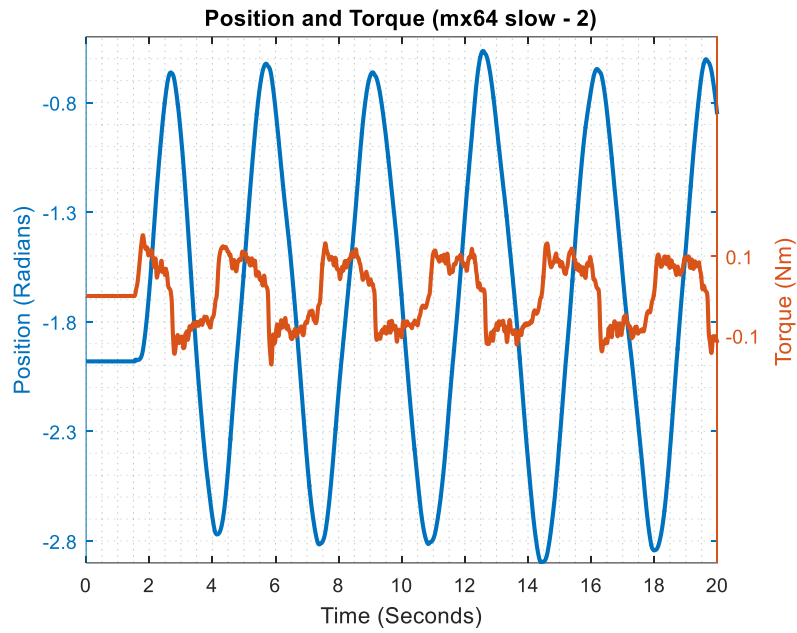


**Figure B.56** MX-64 slow trial 1, measured and calculated torque.

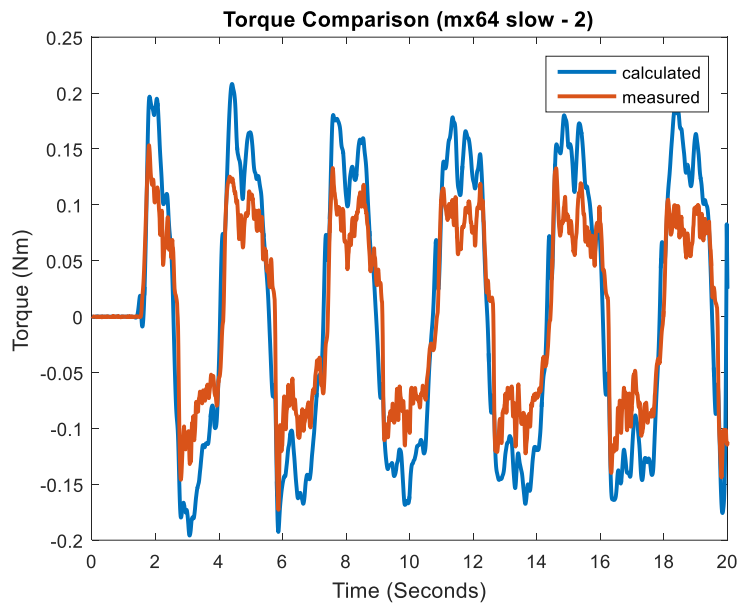


**Figure B.57** MX-64 slow trial 1, recorded and predicted position.

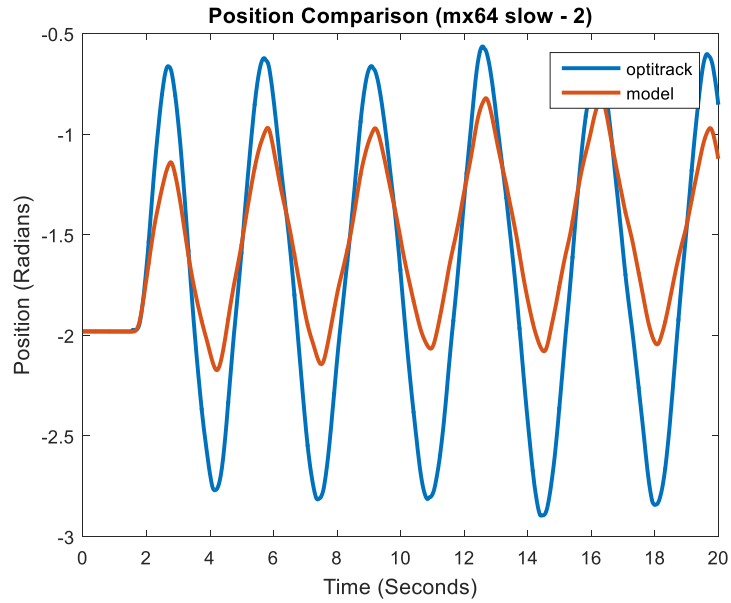
### B.2.10 Slow Trial 2 with MX-64



**Figure B.58** MX-64 slow trial 2, measured position and torque.



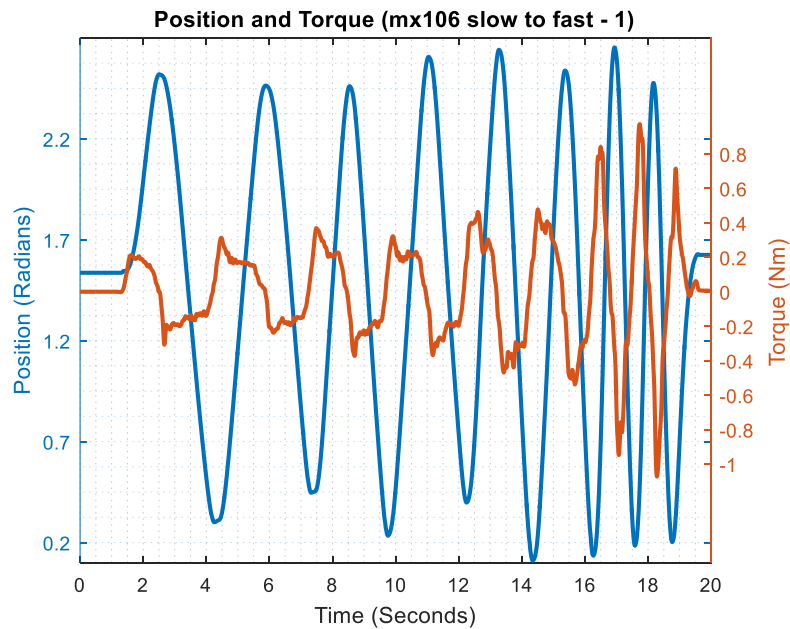
**Figure B.59** MX-64 slow trial 2, measured and calculated torque.



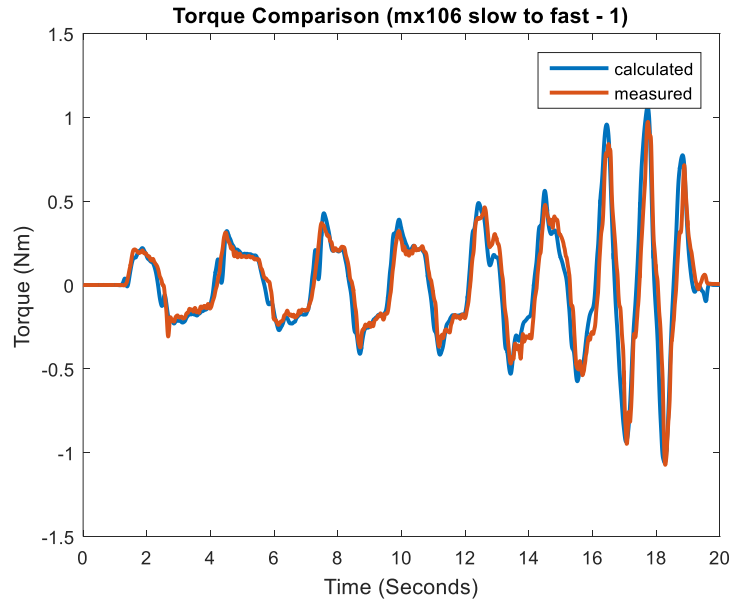
**Figure B.60** MX-64 slow trial 2, recorded and predicted position.

### B.3 Testing Trials Plots for MX-106

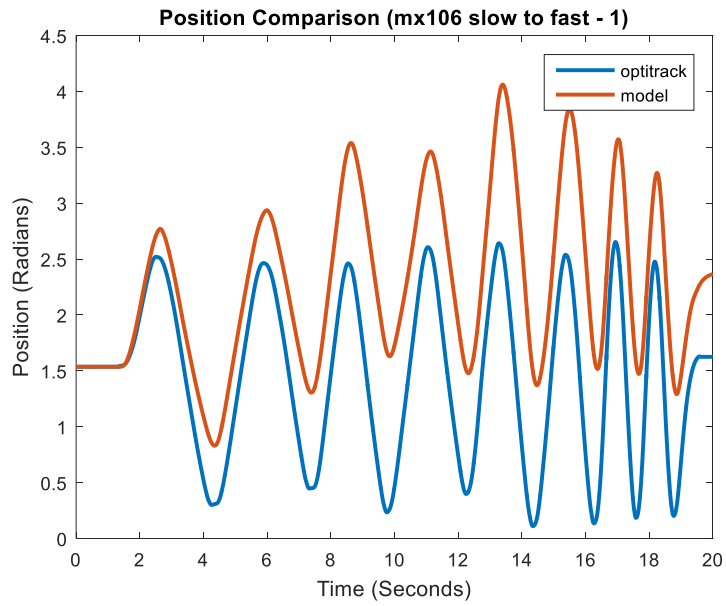
#### B.3.1 Slow to Fast Trial 1 with MX-106



**Figure B.61** MX-106 slow to fast trial 1, measured position and torque.

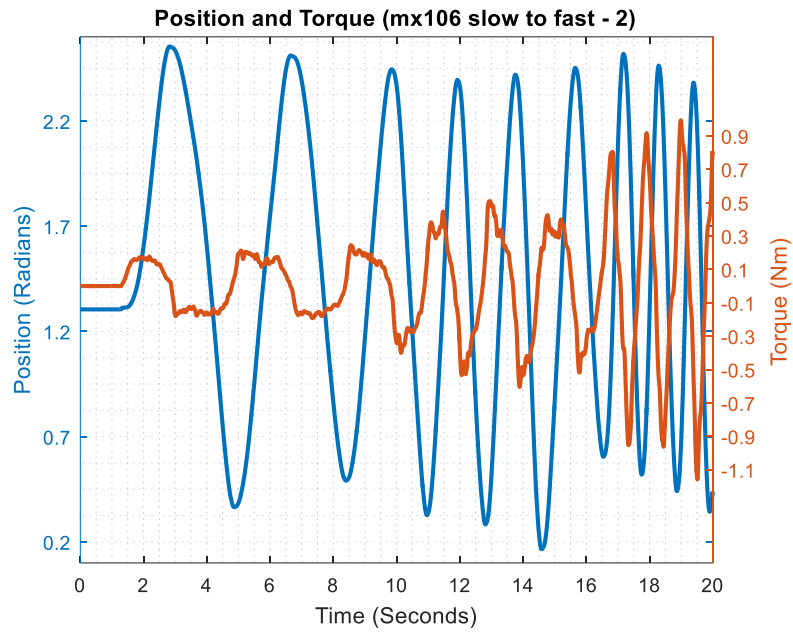


**Figure B.62** MX-106 slow to fast trial 1, measured and calculated torque.

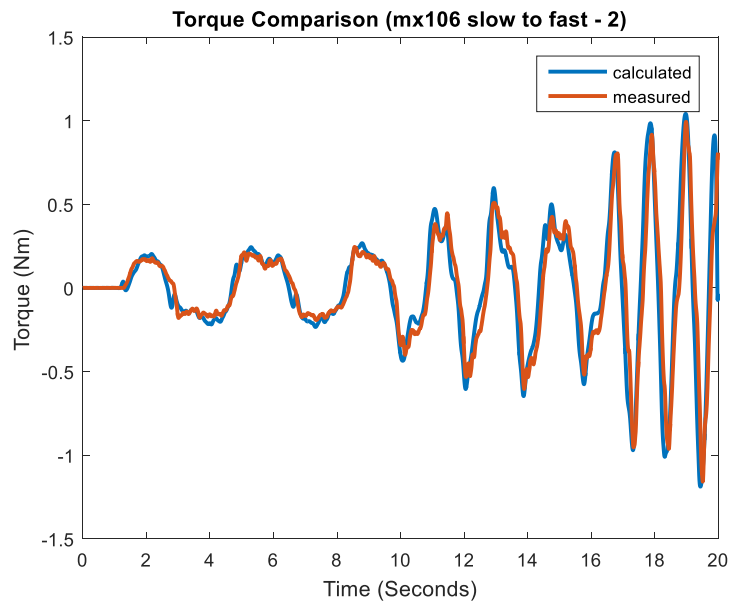


**Figure B.63** MX-106 slow to fast trial 1, recorded and predicted position.

### B.3.2 Slow to Fast Trial 2 with MX-106

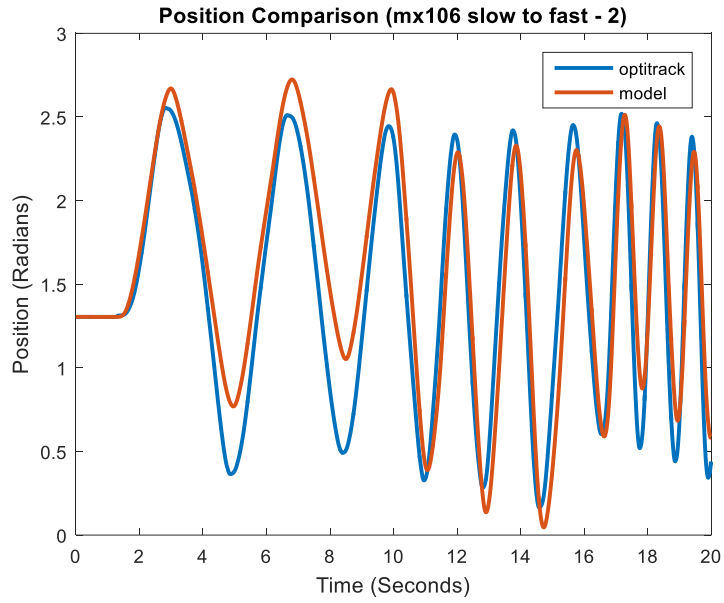


**Figure B.64** MX-106 slow to fast trial 2, measured position and torque.



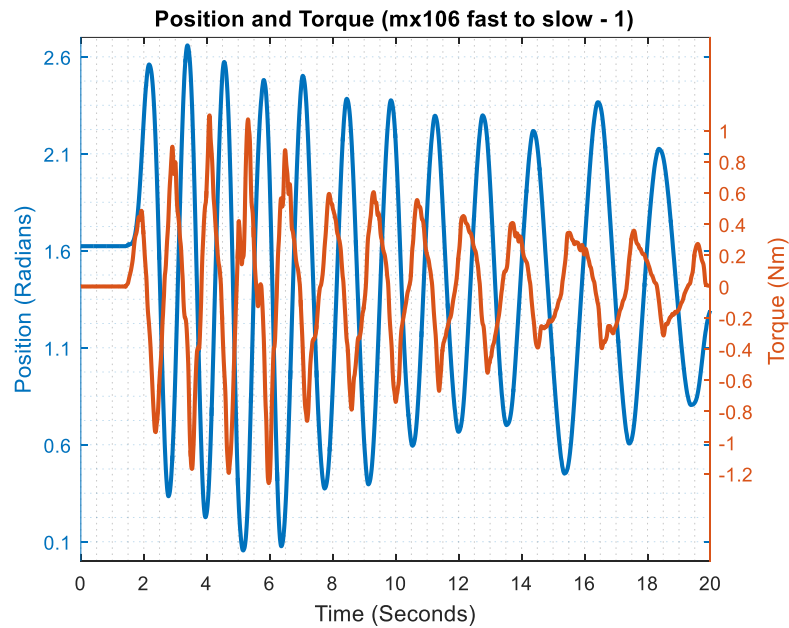
**Figure B.65** MX-106 slow to fast trial 2, measured and calculated torque.



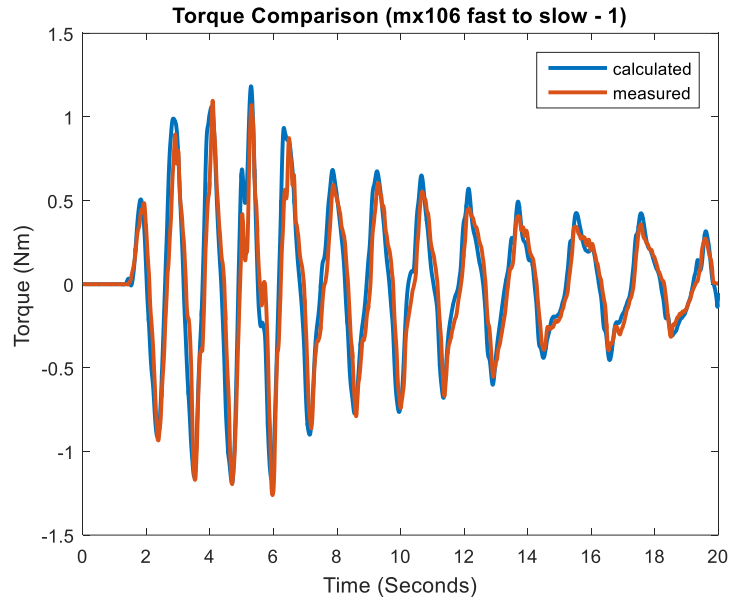


**Figure B.66** MX-106 slow to fast trial 2, recorded and predicted position.

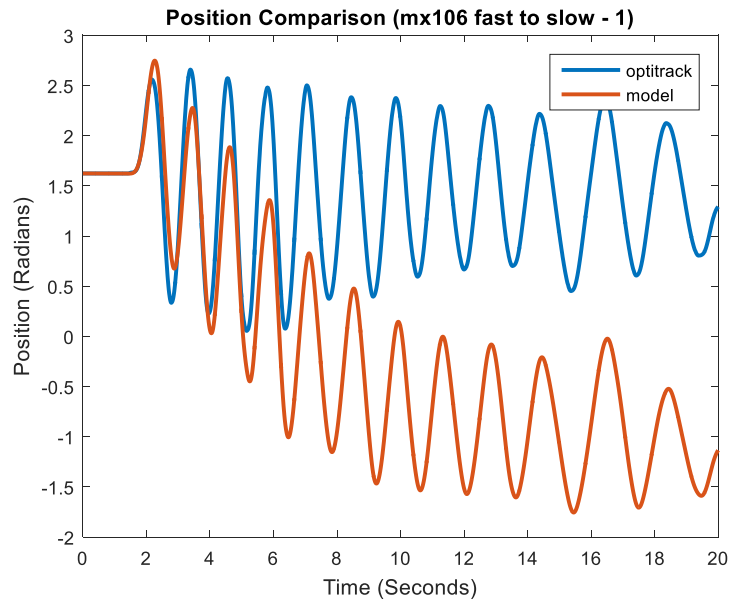
### B.3.3 Fast to Slow Trial 1 with MX-106



**Figure B.67** MX-106 fast to slow trial 1, measured position and torque.

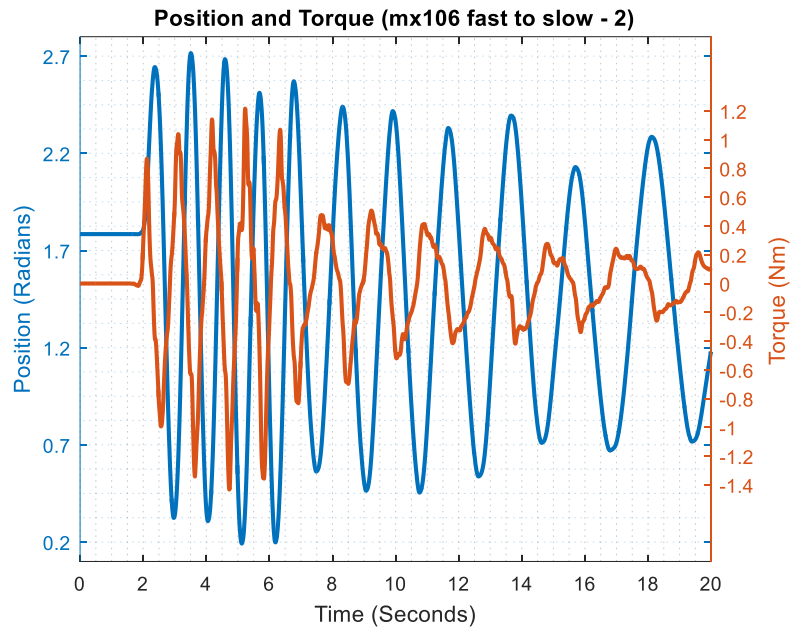


**Figure B.68** MX-106 fast to slow trial 1, measured and calculated torque.

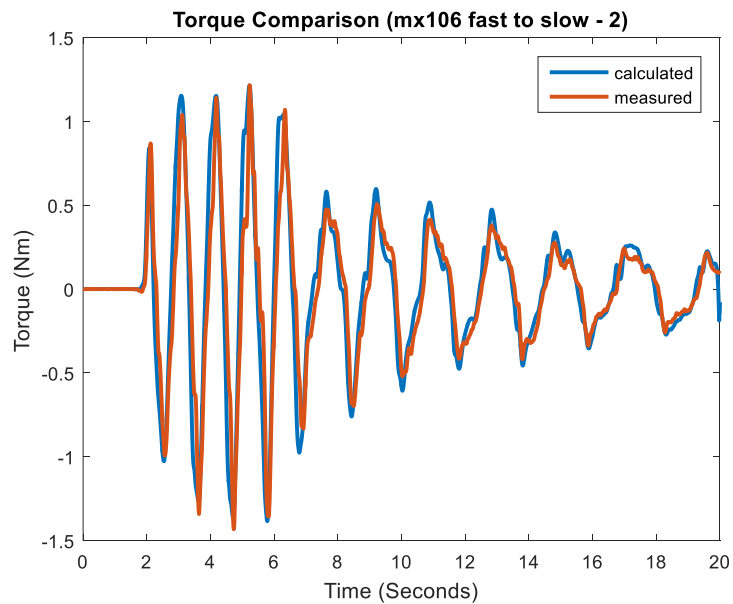


**Figure B.69** MX-106 fast to slow trial 1, recorded and predicted position.

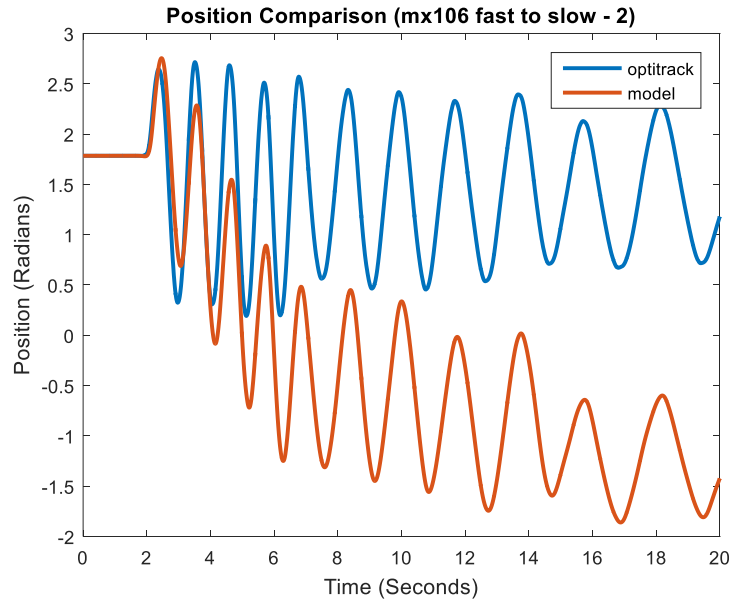
### B.3.4 Fast to Slow Trial 2 with MX-106



**Figure B.70** MX-106 fast to slow trial 2, measured position and torque.

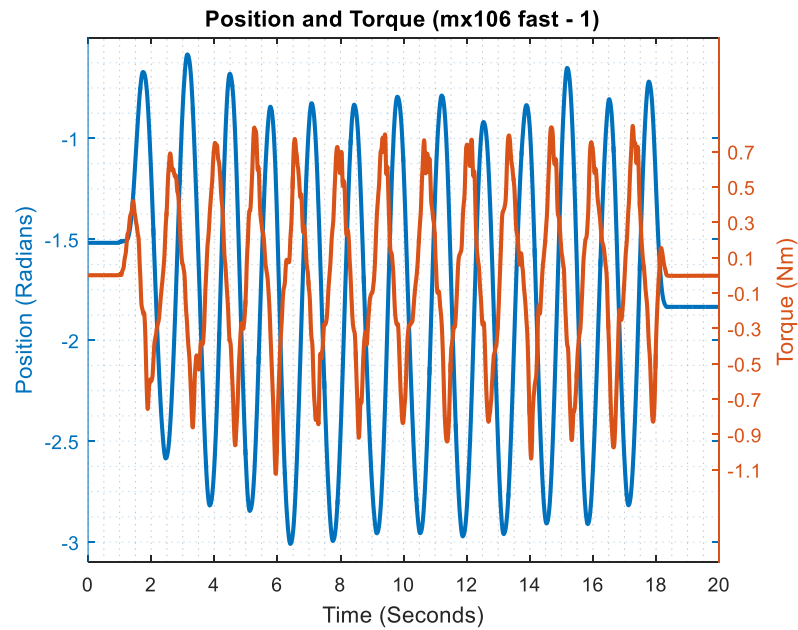


**Figure B.71** MX-106 fast to slow trial 2, measured and calculated torque.

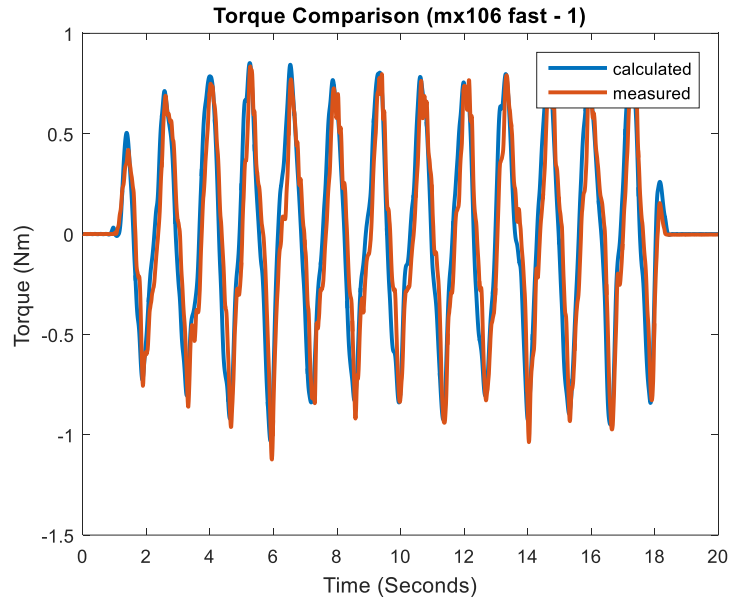


**Figure B.72** MX-106 fast to slow trial 2, recorded and predicted position.

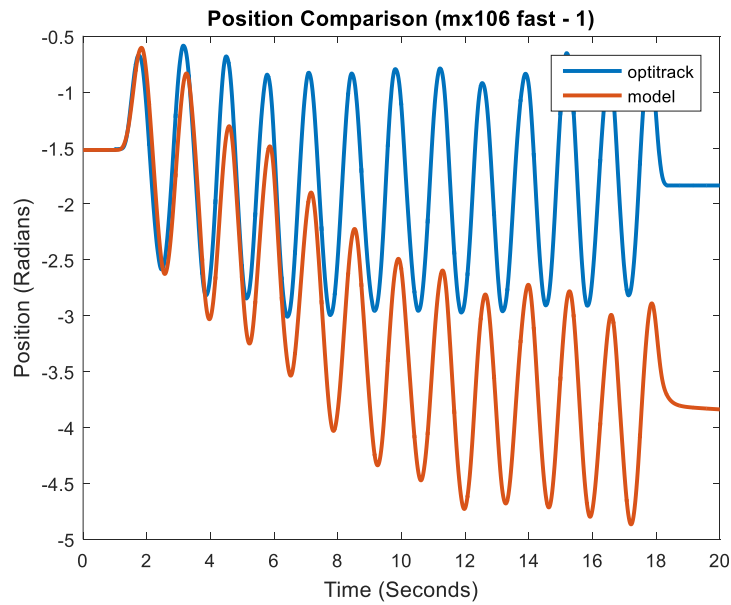
### B.3.5 Fast Trial 1 with MX-106



**Figure B.73** MX-106 fast trial 1, measured position and torque.



**Figure B.74** MX-106 fast trial 1, measured and calculated torque.



**Figure B.75** MX-106 fast trial 1, recorded and predicted position.

### B.3.6 Fast Trial 2 with MX-106

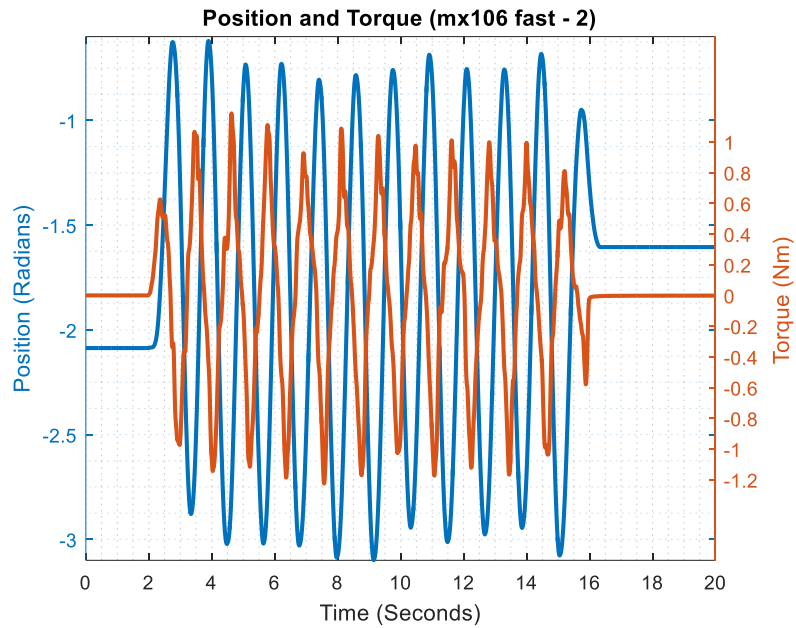


Figure B.76 MX-106 fast trial 2, measured position and torque.

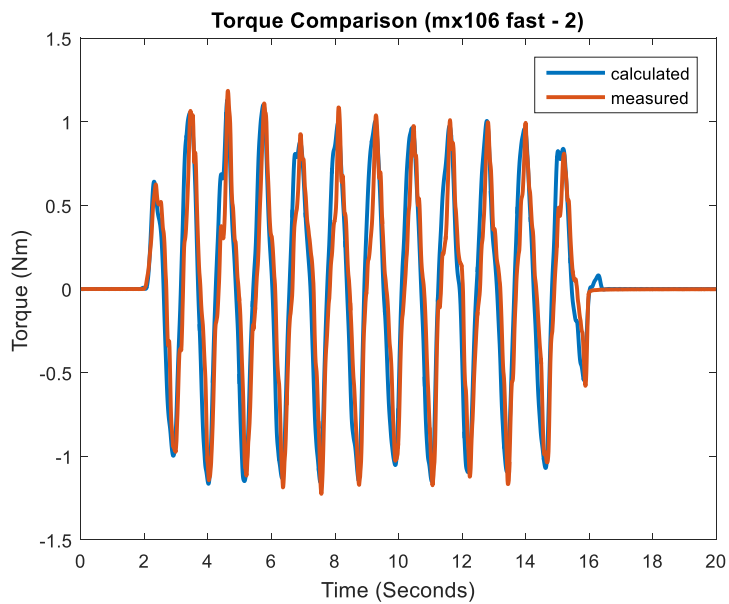
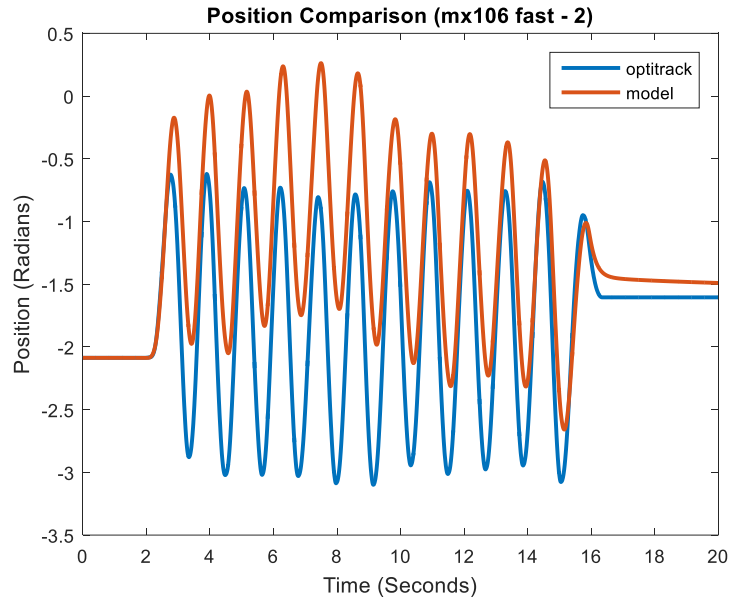
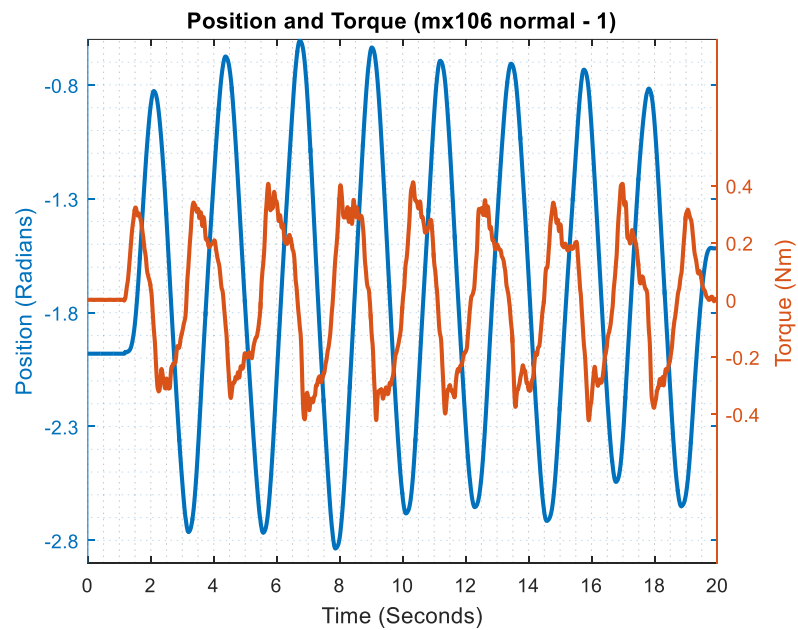


Figure B.77 MX-106 fast trial 2, measured and calculated torque.

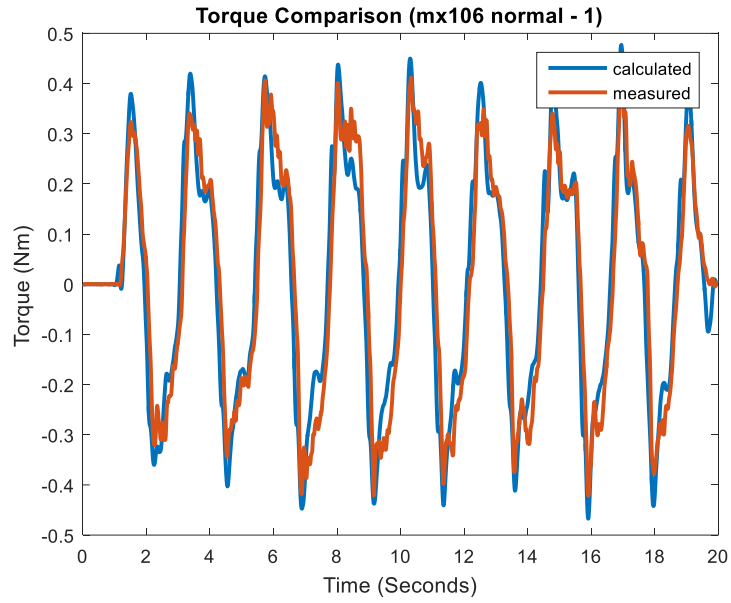


**Figure B.78** MX-106 fast trial 2, recorded and predicted position.

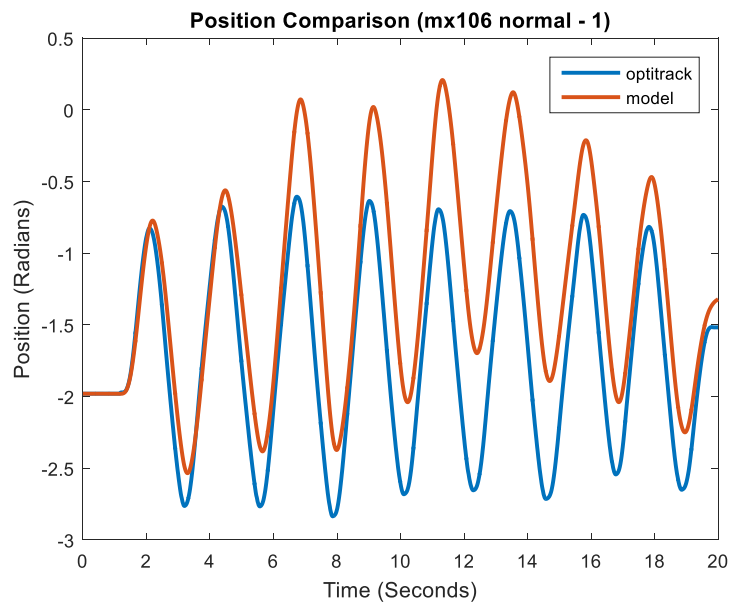
### B.3.7 Normal Trial 1 with MX-106



**Figure B.79** MX-106 normal trial 1, measured position and torque.



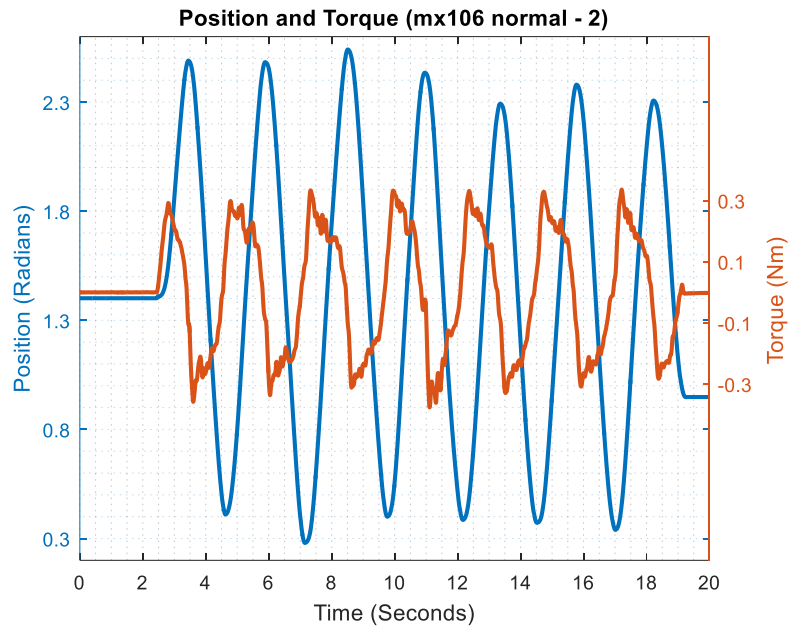
**Figure B.80** MX-106 normal trial 1, measured and calculated torque.



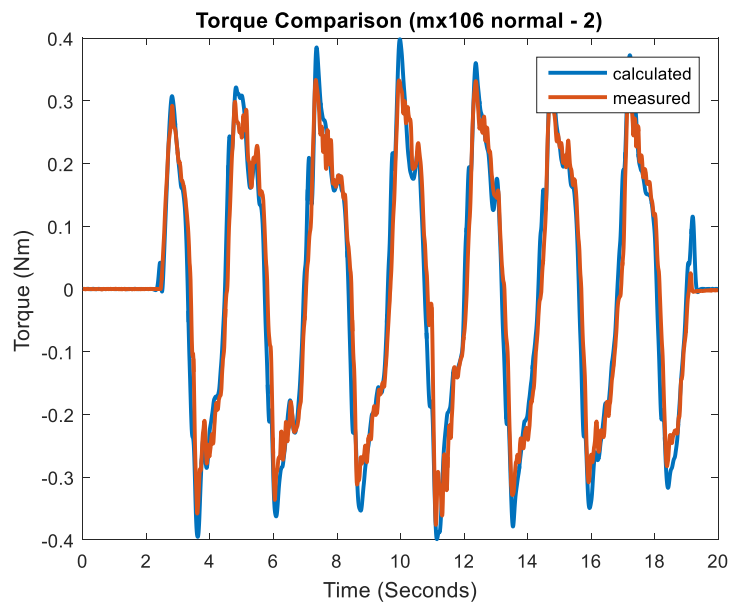
**Figure B.81** MX-106 normal trial 1, recorded and predicted position.



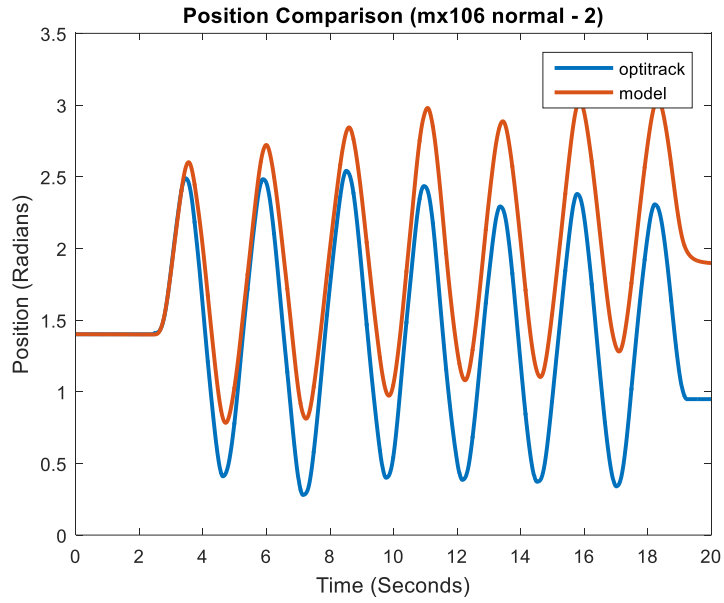
### B.3.8 Normal Trial 2 with MX-106



**Figure B.82** MX-106 normal trial 2, measured position and torque.

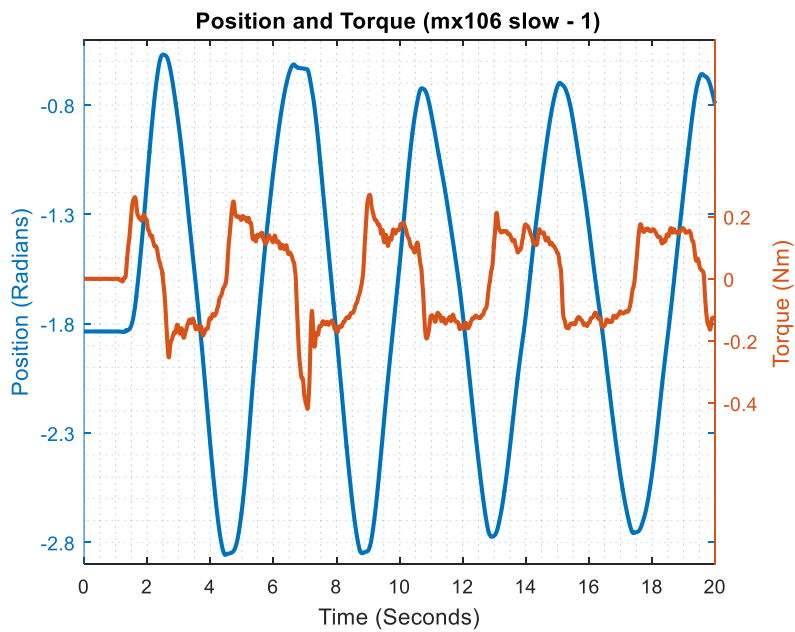


**Figure B.83** MX-106 normal trial 2, measured and calculated torque.

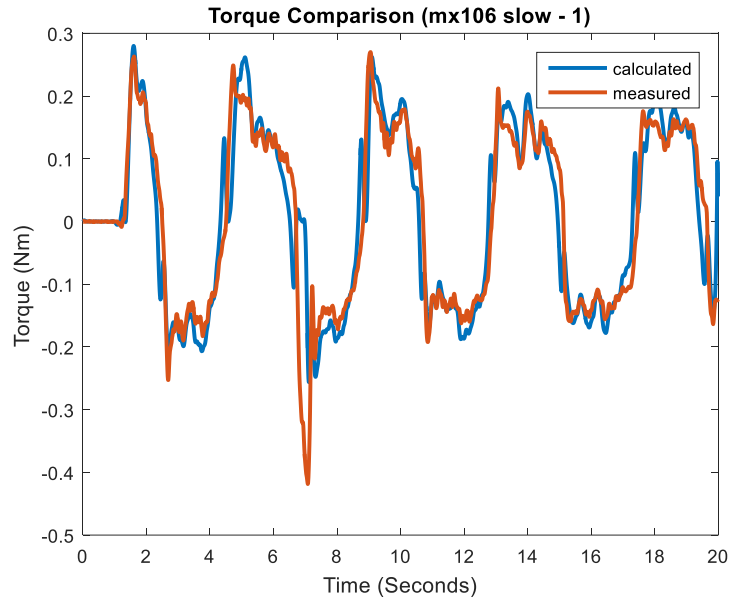


**Figure B.84** MX-106 normal trial 2, recorded and predicted position.

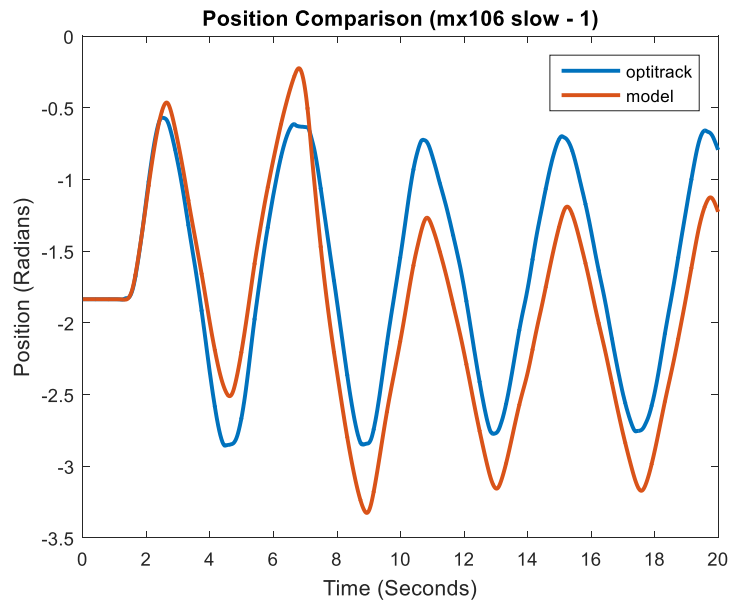
### B.3.9 Slow Trial 1 with MX-106



**Figure B.85** MX-106 slow trial 1, measured position and torque.

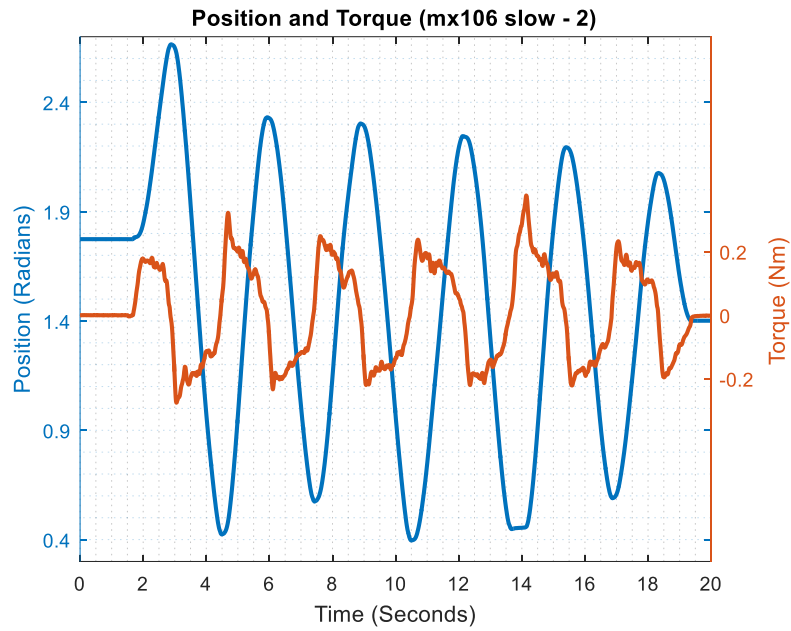


**Figure B.86** MX-106 slow trial 1, measured and calculated torque.

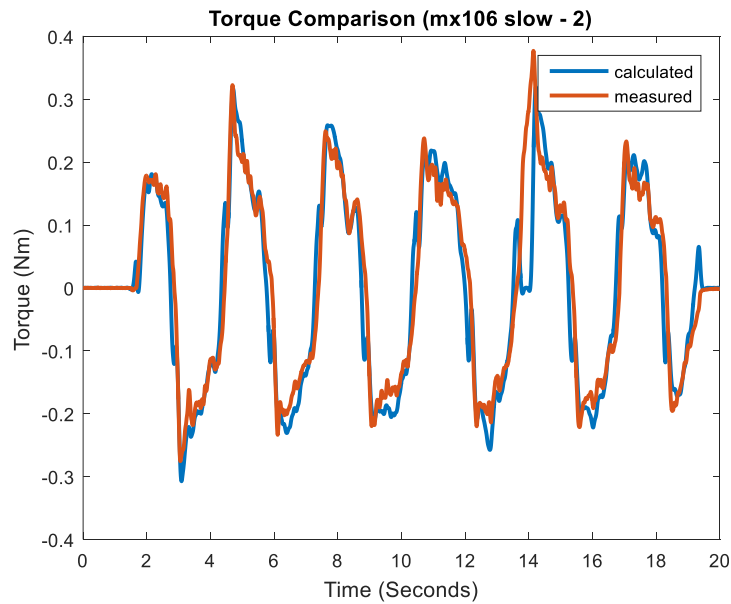


**Figure B.87** MX-106 slow trial 1, recorded and predicted position.

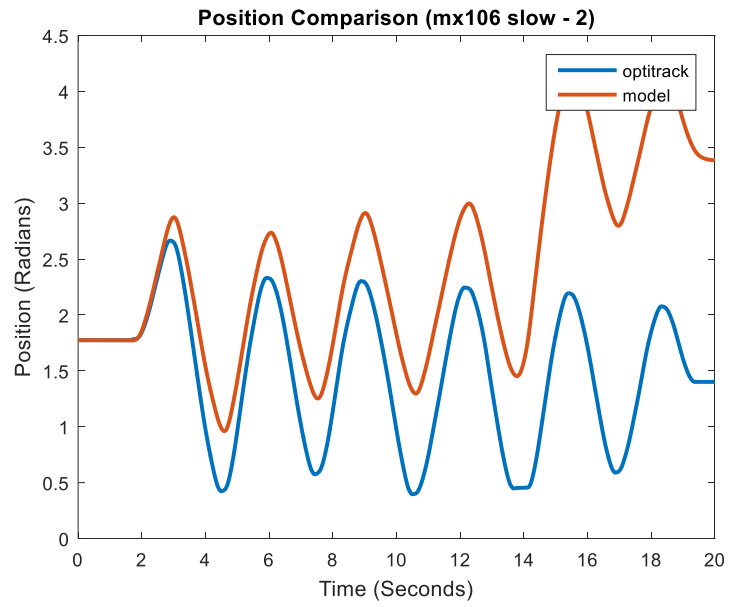
### B.3.10 Slow Trial 2 with MX-106



**Figure B.88** MX-106 slow trial 2, measured position and torque.



**Figure B.89** MX-106 slow trial 2, measured and calculated torque.



**Figure B.90** MX-106 slow trial 2, recorded and predicted position.

## APPENDIX C

### CODING SCRIPTS

This appendix contains the code scripts used in the experiments.

#### C.1 Script for Friction Estimation

This code is used to find the estimate of friction.

```
%% OptiTrack Data at 120 Hz
M1 = xlsread('f-mx64-ccw-2.csv'); % position data
f1 = M1(4:end,1); % frame
x1_1 = M1(4:end,5)*1000; % mm
z1_1 = M1(4:end,7)*1000; % mm
x1_2 = M1(4:end,10)*1000; % mm
z1_2 = M1(4:end,12)*1000; % mm
x1 = x1_2 - x1_1; % change signs here if abnormal plot
z1 = z1_2 - z1_1; % change signs here if abnormal plot
rad = atan2(z1,x1); % atan2(x_free,z_free)
rad(1) = rad(2);
rad_ic = rad(1);

%% OptoForce Data at 100 Hz
fnm1 = load('ft-mx64-ccw-2.mat'); % force data
measured_torque = fnm1.Fn;

%% Plot
figure(1)
t120 = 0:1/120:(length(rad)/120)-1/120;
t100 = 0:1/100:(length(measured_torque)/100)-1/100;
[hAx,hLine1,hLine2] = plotyy(t120,rad,t100,measured_torque);

title('Rotational Friction Trial MX-64');
xlabel('Time (Seconds)');
ylabel(hAx(1),'Position (Radians)'); % left y-axis
ylabel(hAx(2),'Torque (Nm)'); % right y-axis
set(hLine1,'linewidth',2);
set(hLine2,'linewidth',2);
% axis(hAx(1),[0 15 (min(rad)-.001) (max(rad)+0.001)]);
axis(hAx(2),[0 15 min(measured_torque) max(measured_torque)+0.1]);
set(hAx, 'Position', [.13 .11 .735 .815]);
hAx(1).XTick = [0:3:15];
grid minor;
```

## C.2 Script for Data Recording Synchronization

This code is used to synchronize the recording from OptiTrack and OptoForce with the help of Arduino.

```
% OptoForce initialization and checks
if strcmp(computer('arch'),'win32'),
    addpath './mex_files\32bit'; end % If the MATLAB is 32bit
if strcmp(computer('arch'),'win64'),
    addpath './mex_files\64bit'; end % If the MATLAB is 64bit
ports = OptoPorts(3); % For 3 axis sensors - Get an instance of
the OptoPorts class (3 - only 3D sensors; 6 - only 6D sensors )
available_ports = ports.listPorts;% Get the list of the available ports
if (isempty(available_ports)),
    disp('No DAQ is connected...'); else disp(available_ports); end;
if (ports.getLastSize()>0), % Is there at least 1 available port?
    port = available_ports(1,:);% If at least 1 port is available then
select the first one
    daq = OptoDAQ();% Get an instance of the OptoDAQ class (this class
handles the actual sensor reading)
    isOpen = daq.open(port,0);% Open the previously selected port (the
second argument: 0 - high-speed mode; 1 - slower debug mode)
    if (isOpen==1),

        speed = 100; % Set the required DAQ's internal sampling speed
(valid options: 1000Hz,333Hz, 100Hz, 30Hz)
        filter = 0;% Set the required DAQ's internal filtering-cutoff
frequency (valid options: 0(No filtering),150Hz,50Hz, 15Hz)
        daq.sendConfig(speed,filter);% Sends the required configuration
        channel = 1;% Some DAQ support multi-channel, otherwise it
must be 1

        % Initialize Arduino for OptiTrack Sync In
        a = arduino('COM14');% Check port number from device manager
        a.pinMode(9,'output');% Specify the output pin
        a.digitalWrite(9,0);% Stop OptiTrack data collection

        % Calibrate force sensor
        countspernewton = 451.89;
        disp('Calibration in progress !');
        for u = 1:10
            output = daq.read3D(channel);% For 3 axis sensors - Reads
all the available samples (output.size) to empty the buffer
            pause(0.1);
            ex(u) = output.Fx(end);
            ey(u) = output.Fy(end);
        end
        errx = sum(ex)/10;
        erry = sum(ey)/10;
        Fx = ex(end) - errx;
        Fy = ey(end) - erry;

        radius = 0.105; n = 0;
```

```

disp('Start recording from Motive and press any key');
pause

% Data recording loop
a.digitalWrite(9,1);% Start OptiTrack data collection
disp('Start')
tic;
while (n<2000 && output.size>=0 ),% Loop for 15sec (quit if any
error)
    n = n + 1;          % records the number of iterations
    output = daq.read3D(channel); % For 3 axis sensors -
Reads all the available samples (output.size)
    Fx = [Fx output.Fx(end)-errx]; % Fz stores all the
received samples of output.Fz
    Fy = [Fy output.Fy(end)-erry];
    time(n) = toc;
    while toc<0.01;
    end
    tic
end
a.digitalWrite(9,0);% Stop OptiTrack data collection
disp('Stop recording from Motive and press any key');
pause

% Clean-up
daq.close(); % Close the already opened DAQ
delete(a); % Delete the object
clear daq; % Destroy the OptoDAQ class
clear ports; % Destroy the OptoPorts class

Fr = sqrt((Fx.^2)+(Fy.^2)); % Resultant force
Tr = (Fr(2:end)/countspersnewton)*(radius); % Rotational
friction in Nm
Tx = (Fx(2:end)/countspersnewton)*(radius);
Ty = (Fy(2:end)/countspersnewton)*(radius);
figure(1)
plot(Tr);
hold on;
plot(Tx, 'r');
plot(Ty, 'k');
else
disp('The DAQ could not be opened');
end
end
disp('Trial stopped');

```



### C.3 Script used for Optimization

This code is used to run the optimizations. The first script is where the initial estimates are entered and from where the optimization functions are called.

```
global B;
global I;
global F;
global l;
cycles = 5;

for i=1:cycles
    if(i == 1)
        B = 0.0286;
        I = 0.0385;
        F = 0.1234;
        l = 0.20;
    end
    [B,I] = motorparams(B,I,F,l);
    if(i == cycles)
        Fp = F;
    end
    F = F_optimization(B,I,F,l);
end

Irod = (1/3)*0.0402*0.228^2;    % kgm^2
Imass = 0.1034*l^2;           % kgm^2
Imotor = I - Irod - Imass     % kgm^2
```

Function that gets called for optimization of B and I. Similar function is used for F too.

```
function [B,I,Kp,Kd] = motorparams(B,I,F,l)
% MOTORPAREAMS demonstrates using LSQNONLIN with Simulink.
systemmodell    % Load the model
global err
pd0 = [I,B];    % Set initial values of I and B

options = optimset('Algorithm','levenberg-marquardt',...
    'Display','iter','TolX',0.001,'TolFun',0.001);
[params] = lsqnonlin(@tracklsq, pd0, [], [], options);
I = params(1); % Get optimized value of I
B = params(2); % Get optimized value of B

function err = tracklsq(pd)
    expdata = evalin('base','radnew'); % change variable name here
    global edata
```

```

I = pd(1); % set value of I into function workspace
B = pd(2); % set value of B into function workspace to be used
in Simulink model

% Initialize sim options
[simopt] = simset('solver','ode3','SrcWorkspace','Current');
ang_init = expdata(1,1); % get the initial angle from data
% Compute function value
[t,simOut] = sim('systemmodell1',[0 ((length(expdata)-
1)/120)],simopt);

figure(1)
plot(t,simOut(:,1),'r-','LineWidth',3);
drawnow;
hold on;
plot(t,expdata,'b-','LineWidth',3);
legend('optimized','actual');
drawnow;
hold off;
pause(.01);
err = simOut(:,1) - expdata;
save e106_20cm_cw_3.mat err simOut expdata
end
xlabel('Seconds');
ylabel('Radians');
title('Pendulum Actual vs Optimized Data (mx106-20cm-cw-3)');
end

```

## C.4 Script used for Testing

This script is used to test the data and do the comparisons of torques and positions.

```

%% OptiTrack Data at 120 Hz
M1 = xlsread('mx28_slow_2.csv'); % position data
f1 = M1(4:end,1); % frame
x1_1 = M1(4:end,5)*1000; % mm
z1_1 = M1(4:end,7)*1000; % mm
x1_2 = M1(4:end,10)*1000; % mm
z1_2 = M1(4:end,12)*1000; % mm
x1 = -x1_2 + x1_1; % change signs here if abnormal plot
z1 = -z1_2 + z1_1; % change signs here if abnormal plot
rvector = sqrt(x1.^2 + z1.^2);
radius = mean(rvector)/1000; % meters
rad = atan2(z1,x1); % atan2(x_free,z_free)
rad = rad(1:2400);
rad_ic = rad(1);

%% OptoForce Data at 100 Hz
forcesensor = load('mx28_slow_2.mat'); % force data
forceX = forcesensor.Fx(1:2000);
torqueX = forceX * radius / 451.89;

```

```

%% Plot
figure(1)
t120 = 0:1/120:(length(rad)/120)-1/120;
t100 = 0:1/100:(length(torqueX)/100)-1/100;
[hAx,hLine1,hLine2] = plotyy(t120,rad,t100,torqueX);
title('Position and Torque (mx28 slow - 2)');
xlabel('Time (Seconds)');
ylabel(hAx(1),'Position (Radians)'); % left y-axis
ylabel(hAx(2),'Torque (Nm)'); % right y-axis
set(hLine1,'linewidth',2);
set(hLine2,'linewidth',2);
axis(hAx(1),[0 20 ((ceil(min(rad)*10))/10)-0.1
((floor(max(rad)*10))/10)+0.1]);
axis(hAx(2),[0 20 (min(torqueX)-0.5) (max(torqueX)+0.5)]);
% axis(hAx(2),[0 15 0 .05]);
set(hAx, 'Position', [.13 .11 .735 .815]);
hAx(1).YTick =
[ ((ceil(min(rad)*10))/10):0.5:((floor(max(rad)*10))/10)];
hAx(2).YTick = [ ((ceil(min(torqueX)*100))/100)-
.00:0.04:((floor(max(torqueX)*100))/100)+.00];
hAx(1).XTick = [0:2:20];
grid minor;

%% Calculation for expected torque with filtration
I = 0.0068 + 0.0003; % 0.0003 for rod
D = 0.0362;
F = 0.0078;
[B,A]=butter(2,2*5/120); % (order,2*cut-off/sampling-rate).
frad=filtfilt(B,A,rad);
vel = deriv671(frad,t120);
acc = deriv671(vel,t120);
calc_torque = (I*acc)+(D*vel)+(tanh(vel)*F);

figure(2)
plot(t120,calc_torque,'-','LineWidth',2);hold on;plot(t100,torqueX,'-
','LineWidth',2);%plot(t100,drift_torque,'g--')
legend('calculated','measured');
title('Torque Comparison (mx28 slow - 2)');
xlabel('Time (Seconds)');
ylabel('Torque (Nm)');

%% for validation model
input05=[t120' calc_torque'];
input06=[t100' torqueX'];

%% Comparing position data
figure(3);
plot(t120,rad,'-','LineWidth',2);hold on;plot(t120,radlmodel(1:end-
1),'-','LineWidth',2);
legend('optitrack','model');title('Position Comparison (mx28 slow -
2)');
xlabel('Time (Seconds)');ylabel('Position (Radians)');

```

## C.5 Script used for ODE

This is the actual function that gets called from the mex file.

```
function f = frODEtanh(t,x, p, u,myStruct)

Fcoef=tanh(100*x(2));%tanh-shaped coefficient for friction force

F=p(3);%Friction force
B=p(2);%Damping
I=p(1);%Moment of Inertia
FF=F*Fcoef;

f(1) = x(2);
f(2)=(-B*x(2) + u(1) -FF)/I;
f = f(:);
```

Following code is a demo of how to use the ODE code

```
% odeRAF201550.mexw32 requires a time period (tspan) and initial
% conditions (2 element vector x0)
% tt and the upper limit of the for loop are determined by the length
of the applied torque vector.
% If the torque vector has length= 10000, the values below do not need
to be changed.

tspan = [0 .001]; % time span for solver
x0     = [pi/2 0]; % initial conditions x0(1) is angle, x0(2) is angular
velocity
xtraj= x0(1);      % initialize trajectory to first initial condition
tt=0:.001:10-.001;% Change to match length of applied torque vector
for i=2:10000 % execute for length(torque)..change for vector of
different length

% call odeRAF201555 to solve equation of motion Itheta''= applied
torque - Btheta'-Ffriction
% INPUT ARGUMENTS B third argument has 3 values... I, B and Ffriction,
respectively the ith value of applied torque (starting at index 2) is
passed in the 4th argument.
[t,x] = odeRAF201555(tspan, x0, [.0325,.0975,.0552],[torque1(i)],[1e-6,
1e-8, 10]);

x0=[x(1,end) x(2,end)]; % make currently computed angle and angvel the
ICs for
% the next iteration of the loop
xtraj(i)=x0(1); % collect all values of x that are computed.
end
```

## REFERENCES

- [1] C. Ott, R. Mukherjee, and Y. Nakamura, "Unified Impedance and Admittance Control," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 554-561.
- [2] M. Schwarz and S. Behnke, "Compliant Robot Behavior Using Servo Actuator Models Identified by Iterative Learning Control," in *RoboCup 2013: Robot World Cup XVII*. vol. 8371, S. Behnke, M. Veloso, A. Visser, and R. Xiong, Eds., ed: Springer Berlin Heidelberg, 2014, pp. 207-218.
- [3] A. Mensink, "Characterization and modeling of a dynamixel servo," *Trabajo Individual de Investigación en el Electrical Engineering Control Engineering de la University of Twente*, 2008.
- [4] F. Andoh, "Inertia identification method based on the product of the integral of torque reference input and motor speed," in *Control Applications, 2008. CCA 2008. IEEE International Conference on*, 2008, pp. 1151-1158.
- [5] G. V. A. G. Asanka Perera, M. B. Pillai, A. M. Harsha, and S. Abeykoon, "DC motor inertia estimation for robust bilateral control," in *Information and Automation for Sustainability (ICIAfS), 2014 7th International Conference on*, 2014, pp. 1-7.
- [6] G. Heins, F. G. De Boer, and S. Vafi, "Characterisation of the mechanical motor parameters for a permanent magnet synchronous motor using induced torque harmonics," in *Power Engineering Conference, 2007. AUPEC 2007. Australasian Universities*, 2007, pp. 1-5.
- [7] R. Garrido and A. Concha, "Inertia and Friction Estimation of a Velocity-Controlled Servo Using Position Measurements," *Industrial Electronics, IEEE Transactions on*, vol. 61, pp. 4759-4770, 2014.
- [8] S. P. N. Singh, H. Kurniawati, K. S. Naveh, J. Song, and T. Zastrow, "CHARM: A platform for algorithmic robotics education & research," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, 2014, pp. 2602-2607.
- [9] D. Rivas, M. Alvarez, P. Velasco, J. Mamarandi, J. L. Carrillo-Medina, V. Bautista, *et al.*, "BRACON: Control system for a robotic arm with 6 degrees of freedom for education systems," in *Automation, Robotics and Applications (ICARA), 2015 6th International Conference on*, 2015, pp. 358-363.

- [10] C. Pahl and E. Supriyanto, "Guide to fuzzy logic based transesophageal echocardiography motorized probe control," in *Methods and Models in Automation and Robotics (MMAR), 2015 20th International Conference on*, 2015, pp. 892-896.
- [11] G. C. Andres, C. A. Luis, and Z. A. Julio, "Walking cycle control for an active ankle prosthesis with one degree of freedom monitored from a personal computer," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, 2015, pp. 3651-3654.
- [12] N. Benalie, W. Chonnaparamutt, and W. Supsi, "Intelligent controller of SEFRE rehab system," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014 11th International Conference on*, 2014, pp. 1-6.
- [13] W. Indrajit and A. Muis, "Development of whole body motion imitation in humanoid robot," in *QiR (Quality in Research), 2013 International Conference on*, 2013, pp. 138-141.
- [14] L. A. Perez-Gaspar, F. Trujillo-Romero, S. O. Caballero-Morales, and F. H. Ramirez-Leyva, "Curve fitting using polygonal approximation for a robotic writing task," in *Electronics, Communications and Computers (CONIELECOMP), 2015 International Conference on*, 2015, pp. 184-189.
- [15] J. Florez, F. Calderon, and C. Parra, "Servo load analysis for the classification of surface of locomotion in a modular snake-like robot," in *Image, Signal Processing, and Artificial Vision (STSIVA), 2012 XVII Symposium of*, 2012, pp. 13-18.
- [16] K. Xin, S. Wenda, C. Weihai, and W. Jianhua, "The control of Dynamixel RX-28 based on VC++ for the locomotion of cockroach robot," in *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on*, 2009, pp. 495-499.
- [17] D. Simon, G. J. Androwis, and R. A. Foulds, "Equilibrium point model of knee joint spasticity," in *Bioengineering Conference (NEBEC), 2011 IEEE 37th Annual Northeast*, 2011, pp. 1-2.