

Dissertations and Theses

4-2020

7-DOF Robotic Arm XY-Plane Translational Stability

Christopher Dean Alkire

Follow this and additional works at: <https://commons.erau.edu/edt>



Part of the [Aerospace Engineering Commons](#), and the [Computer-Aided Engineering and Design Commons](#)

Scholarly Commons Citation

Alkire, Christopher Dean, "7-DOF Robotic Arm XY-Plane Translational Stability" (2020). *Dissertations and Theses*. 506.

<https://commons.erau.edu/edt/506>

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

7-DOF Robotic Arm XY-Plane Translational Stability

By

Christopher Dean Alkire

A Thesis Submitted to the Faculty of Embry-Riddle Aeronautical University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Aerospace Engineering

April 2020

Embry-Riddle Aeronautical University

Daytona Beach, Florida

7-DOF Robotic Arm XY-Plane Translational Stability

By

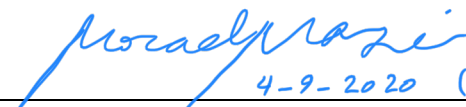
Christopher Dean Alkire

This Thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Richard Prazenica, Department of Aerospace Engineering, and has been approved by the members of the Thesis Committee. It was submitted to the Office of the Senior Vice President for Academic Affairs and Provost, and was accepted in the partial fulfillment of the requirements for the Degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE

 Chairman, Dr. Richard Prazenica

 Member, Dr. Troy Henderson



4-9-2020 (6:41 PM)

 Member, Dr. Morad Nazari

 Graduate Program Coordinator,
 Dr. Magdy Attia

 Date

 Dean of the College of Engineering,
 Dr. Maj Mirmirani

 Date

 Associate Provost of Academic Support,
 Dr. Christopher Grant

 Date

ACKNOWLEDGEMENTS

To my mother and father...

who have always shown me support in the most difficult times.

To Kendra...

who sees the best in me when I struggle to see it myself.

I would like to thank my advisor, Dr. Richard Prazenica, for keeping one foot in the frying pan of lecture and the other firmly into the fire that was advising this research. For always keeping yourself available to take the next step and never becoming as discouraged as I found myself.

-“Keep chopping wood”

The late Dr. Shahrdad Sajjadi, for being one of the most knowledgeable and capable Professors in my graduate mathematics coursework. Your time was too short, but the impact you left on your students will last forever.

Finally, I would like to thank each member of my committee. Dr. Morad Nazari, Dr. Troy Henderson, and Dr. Bogdan Udrea, for your feedback, leadership, guidance and patience. Your support and understanding of your students' need while challenging them each day makes each of you the backbone of the future for the aerospace dynamics and control field.

ABSTRACT

Robotic arms are traditionally mounted to rigid structures as they perform tasks. As a result, the arm's movement would not affect the position of its mount and its operational space that it is performing a task within would remain constant. If a robotic arm were to function in orbit, the motion of the arm would cause it to rotate and translate about its center of mass, which changes as the joints of the arm rotate.

The purpose of this thesis is to focus on the two-dimensional translational effects of an arm operating on a simulated zero-friction surface and provide a method to anticipate and stabilize these induced forces. By calculating the forces generated by the movement of a 7-DOF Sawyer robotic arm using the arm's Universal Robotics Description Format (URDF) parameters provided by Spear (2017) and a Denavit-Hartenberg method for the geometric solution of the arm's kinematics, the induced motion caused by the arm's movement can be arrested by an efficient control system. Using an XY-table to compensate for the induced motion of the arm, a comparison is made for an open-loop and closed-loop control of a cable driven XY-table. From this analysis, a better understanding of an active mount solution for robotic arms can be identified.

The key findings of this research are the validation of open-loop control response based on the calculated reaction force provided by kinematic analysis of the robotic arm's center of mass. Additionally, a closed-loop control response is assessed based on an applied external force to the system during operation. Both results lead to a controlled system displacement error that does not exceed $1e10^{-14}$ meters for the four test cases presented.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iii
ABSTRACT.....	iv
LIST OF FIGURES.....	vii
LIST OF TABLES.....	xi
NOMENCLATURE	xii
1. Introduction.....	1
1.1. Problem Statement.....	1
1.2. System Overview	2
1.2.1. Sawyer Robotic Arm.....	3
1.2.2. Cable Driven XY Table.....	3
1.3. Objectives	4
1.3.1. Uncontrolled.....	5
1.3.2. Open-Loop Control.....	5
1.3.3. Closed-Loop Control.....	5
1.4. Outline	5
2. Literature Review.....	7
2.1. Kinematic Modeling – Denavit Hartenberg Method.....	7
2.2. Kinematic Modeling – Screw Theory.....	9
2.3. Differential Kinematic – Denavit Hartenberg Method.....	11
2.4. Differential Kinematic – Screw Theory Method.....	11
3. Background.....	13
3.1. Kinematics Modeling.....	13
3.2. Simulation and Modeling.....	14
3.2.1. MATLAB.....	14
3.2.2. Simscape Multibody.....	14
3.2.3. Solidworks.....	15
4. Kinematics.....	16
4.1. Forward Kinematics and System Dynamics.....	16
4.1.1. Reliability Testing.....	17
4.1.2. Reliability Testing.....	18
4.1.3. Reliability Testing.....	19
4.1.4. Velocity Jacobian Matrices.....	21
4.1.5. Center of Mass	24
4.1.6. Gravitational Force	25
4.1.7. Inertial Force.....	26

4.1.8. Coriolis Force	27
4.1.9. Centripetal Force.....	27
4.1.10. Equations of Motion	27
5. Control Systems.....	31
5.1. Inverse Kinematic Control.....	31
5.1.1. Damped Least Squares Inverse Jacobian Control.....	32
5.1.2. Singularity Mitigation.....	32
5.2. Cable Driven XY Table Control.....	34
5.2.1. Open-Loop Control.....	34
5.2.2. Closed-Loop Control	35
6. Simulation Results	37
6.1. Uncontrolled Cases	37
6.1.1. Case I – Simple Movement I	37
6.1.2. Case II – Simple Movement II.....	40
6.1.3. Case III – 7-DOF Movement I.....	42
6.1.4. Case IV – 7-DOF Movement II	45
6.2. Open-Loop Control cases	47
6.2.1. Case I – Simple Movement I	47
6.2.2. Case II – Simple Movement II.....	49
6.2.3. Case III – 7-DOF Movement I.....	51
6.2.4. Case IV – 7-DOF Movement II	53
6.3. Closed-Loop Control cases.....	55
6.3.1. Case I – Simple Movement I with External Force.....	56
6.3.2. Case II – Simple Movement II with External Force.....	58
6.3.3. Case III – 7-DOF Movement I with External Force.....	60
6.3.4. Case IV – 7-DOF Movement II with External Force	62
7. Conclusions and Recommendations	64
REFERENCES.....	66

LIST OF FIGURES

Figure	Page
1.1 Sawyer robotic arm mounted atop the XY-table.....	2
1.2 XY-table (top view)	4
2.1 Rigid body within the inertial frame	8
2.2 General screw motion displaced from an inertial frame	9
3.1 Cycle of geometric parameters being manipulated to determine either the end effector position or joint angles.....	13
4.1 Solution process for calculating the forces of the Sawyer robot arm.....	16
4.2 Geometric properties of the Sawyer robotic arm	18
4.3 Velocity Jacobian matrix vectors for revolute joints.....	23
4.4 Gravitational force vector direction for Sawyer robot arm in start position	26
4.5 XY-table Free-Body Diagram(s) for force along the x-axis	28
4.6 XY-table Free-Body Diagram(s) for force along the y-axis	29
4.6 Coordinate frames of the XY-table including positive angle orientation.	30
5.1 Inverse kinematic control system utilizing the Jacobian matrix inverse.....	31
5.2 Time varying gain matrix for all test cases	33
5.3 Open-loop control system applied to XY-table.....	35
5.4 Closed-loop control system applied to XY-table.....	35
6.1a End-effector position throughout simulation of case I.....	38
6.1b Initial and final positions of case I	38
6.1c Joint angles from start position to fixed final positions when end-effector reaches target position for case I	39

Figure	Page
6.2 Planar displacement and acceleration of the XY-table pedestal center in case III	40
6.3a End-effector position throughout simulation of case II.....	40
6.3b Initial and final positions of case II	41
6.3c Joint angles from start position to fixed final positions when end-effector reaches target position for case II.....	41
6.4 Planar displacement and acceleration of the XY-table pedestal center in case II	42
6.5a End-effector position throughout simulation of case III	43
6.5b Initial and final positions of case I	43
6.5c Joint angles from start position to fixed final positions when end-effector reaches target position for case I	44
6.6 Planar displacement and acceleration of the XY-table pedestal center in case I	44
6.7a End-effector position throughout simulation of case IV	45
6.7b Initial and final positions of case IV	45
6.7c Joint angles from start position to fixed final positions when end-effector reaches target position for case IV	46
6.8 Planar displacement and acceleration of the XY-table pedestal center in case IV	46
6.9 Calculated displacement and measured displacement of the XY-table pedestal center in case I.....	47
6.10 Open-loop control output converted to the XY-table pedestal position, case III	48
6.11 Comparison of the residual error between the open-loop control with a measured system disturbance and a calculated disturbance for case III	49
6.12 Calculated displacement and measured displacement of the XY-table pedestal center in case II	49

Figure	Page
6.13 Open-loop control output converted to the XY-table pedestal position, case II	50
6.14 Comparison of the residual error between the open-loop control with a measured system disturbance and a calculated disturbance for case II.....	51
6.15 Calculated displacement and measured displacement of the XY-table pedestal center in case III	51
6.16 Open-loop control output converted to the XY-table pedestal position, case III	52
6.17 Comparison of the residual error between the open-loop control with a measured system disturbance and a calculated disturbance for case III	53
6.18 Calculated displacement and measured displacement of the XY-table pedestal center in case IV	53
6.19 Open-loop control output converted to the XY-table pedestal position, case IV	54
6.20 Comparison of the residual error between the open-loop control with a measured system disturbance and a calculated disturbance for case IV	54
6.21 Externally applied forces measured at the pedestal.....	55
6.22 Residual position error measured by the feedback of the closed-loop control system for case I.....	56
6.23 Closed-loop control output converted to the XY-table pedestal position, case I	57
6.24 Steady-state error of the pedestal position for closed-loop control case I	58
6.25 Residual position error measured by the feedback of the closed-loop control system for case II.....	58
6.26 Closed-loop control output converted to the XY-table pedestal position, case II	59
6.27 Steady-state error of the pedestal position for closed-loop control case II	60
6.28 Residual position error measured by the feedback of the closed-loop control system for case III	60
6.29 Closed-loop control output converted to the XY-table pedestal position, case III.....	61

Figure	Page
6.30 Steady-state error of the pedestal position for closed-loop control case III.....	62
6.31 Residual position error measured by the feedback of the closed-loop control system for case IV	62
6.32 Closed-loop control output converted to the XY-table pedestal position, case IV.....	63
6.33 Steady-state error of the pedestal position for closed-loop control case IV.....	63

LIST OF TABLES

Table	Page
4.1 DH Parameters (Sawyer robotic arm)	19

NOMENCLATURE

<i>CM</i>	Center of mass
<i>DH</i>	Denavit-Hartenberg parameters
<i>DOF</i>	Degree of freedom
<i>URDF</i>	Universal robotic description format

1. Introduction

The introduction consists of the problem statement to provide context for the research being conducted. The system overview section defines the various elements being modeling for simulation and their function. Objectives gives the testing criteria that will validate the problem-solving approach. Finally, research outline summarizes the entire process of problem solving leading up to the results section.

1.1. Problem Statement

Robotic arms are traditionally mounted to rigid structures as they perform tasks. As a result, the arm's movement would not affect the position of its mount, and the operational space of the robotic system would remain constant. Newton's third law of motion states for every force generated by the arm there will be an equal and opposite force applied to the mount. Typically, mounting the robotic arm to a rigid base would arrest the reactive force. This allows for a geometric solution to be derived for the arm and the operational space it occupies. This thesis considers the case in which the base mount can displace as a result of the operation of the robotic arm. If a robotic arm were to function in orbit, the motion of the arm would cause it to rotate and translate about its changing center of mass. For example, the inertial forces would be generated by the movement of its joints as it tries to carry out its task, similar to the challenges that astronauts face during their extravehicular activities. This thesis focuses on the two-dimensional translational effects of an arm operating on a simulated zero-friction surface and provides a method to stabilize these induced forces. By calculating the force generated by the movement of a 7-DOF Sawyer robotic arm and stabilizing the forces exerted at its base using an open-

loop and closed-loop control of an XY-table, a better understanding of an active mount solution for robotic arms will be established.

1.2. System Overview

The system considered in this study is comprised of two primary subsystems. The first subsystem is the 7-DOF Sawyer robotic arm, which is mounted on top of the second subsystem, the XY-table. Figure 1.1 depicts the Sawyer arm rigidly mounted to a pedestal that is a part of the XY-table. The table has free movement in only the planar directions. The robotic arm can move all of the seven rigid bodies attached to its base about revolute joints that fix each rigid body to the other in order. This order of rigid bodies is referred to sequentially with the base being frame 0 and the end-effector referenced as frame 7. The end-effector is a term used to describe the tool tip or point at the furthest extent of any robotic arm that the user intends to have reach a desired position. For this simulation, no tool is attached to frame 7, so the robotic arm's wrist joint is identified as the end-effector.

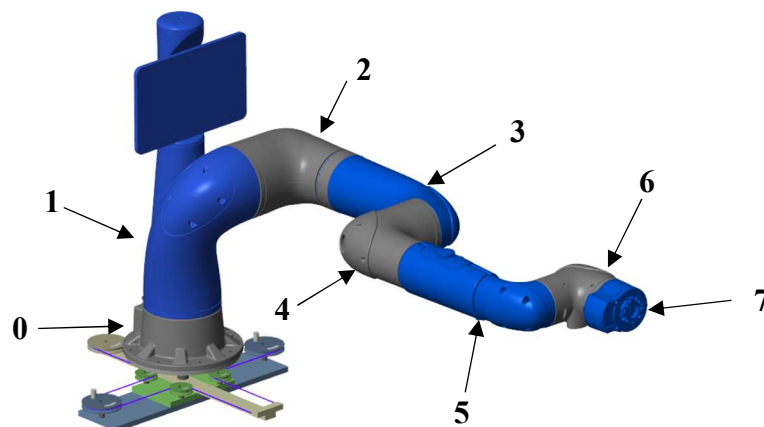


Figure 1.1 Sawyer robotic arm mounted atop the XY-table. Joints are referred to by their sequence number in the image.

1.2.1. Sawyer Arm

The Sawyer robotic arm is composed of a series of rigid bodies. All except two of the rigid bodies are joined by a set of revolute joints. The base is fixed to a desired mounting surface. For the purpose of this study, that surface is the XY-table pedestal. The wrist joint, frame 7, is the second joint with only a single revolute joint. Joints one through six are all shaped similar to 90 deg elbows, which allows for their revolute joint axes to be perpendicular throughout the robotic arm's operation. There is also a monitor that can rotate independently of joint number one for a user interface.

The internal servomotors, wiring and power structure, along with the shape of the rigid bodies makes a precise measurement of inertial properties difficult without the URDF. This information is provided by the manufacturer and contains the inertial characteristics of each joint to a high degree of precision, which aids in the calculation of kinematic and kinetic attributes of the dynamics system and modeling the robotic arm as realistically as possible in the simulation environment.

1.2.2. XY-Table

The XY-table is a cable-driven planar movement simulator with two low friction sleds mounted onto a base that is fixed in the system's inertial space. The two sleds can slide independently or together depending on the rotation of two torque inputs located at each end of the base. Figure 1.3 shows the XY-table without the Sawyer robotic arm mounted to it for clarity.

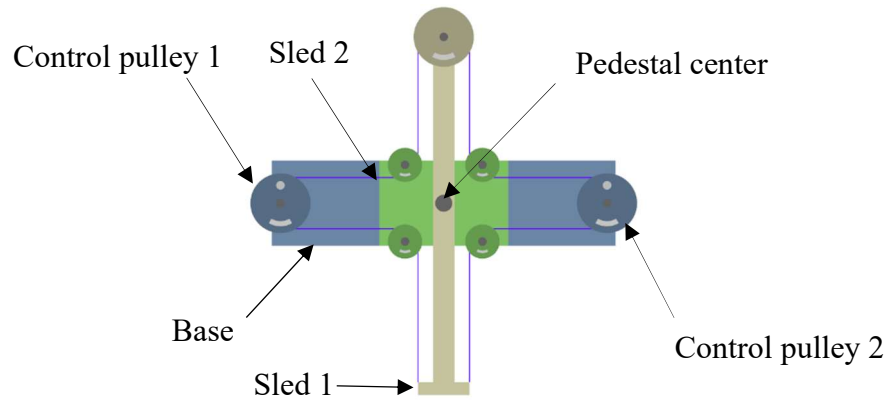


Figure 1.2 XY-table (top view) shown with robotic arm and pedestal removed for clarity (MathWorks, 2019).

1.3. Objectives

From Newton's third law it is expected that any object that is acted on by an external force will continue to move in the direction of that force vector until acted on by another force. For this system, the expectation is that the robotic arm's movement has a series of reaction forces that will transfer through the joints to the base of the robotic arm. Typically, these forces are arrested by a rigid mount that is anchored to a solid surface, but instead if the mount were in a free-fall, or a micro-gravity environment, the reactive forces would cause the base of the robotic arm to rotate and translate from the torque inputs. For the mount to remain stable, another input would have to be applied to the mount to counter the torque. In this simulation, the applied forces exerted through the robotic arm base cause the pedestal to translate along the XY-plane. The XY-table applies a counter torque through its two torque inputs to arrest this translation and stabilize the pedestal at its origin point. To test this stability, three control scenarios are designed and observed.

1.3.1. Uncontrolled

The first testing segment is an uncontrolled system. This would allow the pedestal free movement during the robotic arm's operation and any external forces transferred to the pedestal would be expected to cause unimpeded translation in accordance with Newton's third law.

1.3.2. Open-Loop Control

The second form of control is the open-loop control system. This controller will use the kinematic and kinetic properties calculated from the robotic arm's center of mass to determine the input torques the xy-table will use for maintaining the robotic arm at the systems equilibrium position.

1.3.3. Closed-Loop Control

The third form of control is the use of feedback from a position sensor where the displacement of the pedestal from its start position is known and the torque inputs will continue to reduce error based on the feedback from the position sensor. This would be challenging in a physical environment because the system would require a known fixed point to measure displacement error.

1.4. Outline

This is organized as follows. Chapter 2 provides a review of key concepts and relevant literature. Chapter 3 discusses the simulated Sawyer robot arm modeling. Chapter 4 provides details regarding the method of calculating the reaction forces and inverse kinematic controller for the Sawyer robot arm and the controller development for the XY-table's response to these forces. The results section is contained within Chapter 5

and the final chapter is Chapter 6, which provides a summary of the research method and conclusions determined from the research.

2. Literature Review

The kinematic modeling of a robotic arm describes the relationship between joint positions and the end-effector location in three-dimensional space. Forward kinematics require calculating the end-effector position from the joint positions given, while inverse kinematics is the process of obtaining the joint positions required to reach a desired end-effector position (Stramigioli & Bruyninckx, 2001 p. 39). The first is used in the operation and simulation of linked rigid bodies or “robotic arms”, while the second is essential for motion planning. Two common issues occur when performing inverse kinematics. One is the possibility of multiple solutions, and the second is the potential for singularities within the results (Siciliano, 2008).

The term “differential kinematics” refers to the relationship between the derivatives of the position being calculated. The computation of differential kinematics consists of determining linear and angular velocities of the end-effector from the knowledge of joint velocities. Along with its relationship to motion analysis, differential kinematics is often used to solve inverse kinematics problems by integrating the calculated velocities to determine a desired joint position, especially when the manipulator has a high degree of freedom (DOF) kinematic structure. There are two methods commonly used for kinematic modeling, one based on the Denavit–Hartenberg (DH) parameters method and another based on screw theory. While the DH parameters method is largely used in the robotics industry, the screw theory approach is less common.

2.1. Kinematic Modeling – Denavit–Hartenberg Method

The DH parameter method describes the kinematic relations between the linked rigid bodies connected by 1 degree-of-freedom rotational, prismatic and spherical joints. All

versions of the convention require four parameters to completely describe the relative pose of a link relative to the previous link in the series. Determining the DH parameters requires defined reference frames (i) for every linked rigid body. The DH parameter method described by Hartenberg is summarized by the following steps:

1. Determine z-axis rotation from frame i-1 to frame i.
2. Determine z-axis displacement from frame i-1 to frame i.
3. Determine x-axis rotation from frame i-1 to frame i.
4. Determine x-axis displacement from frame i-1 to frame I (Hartenberg, 1955, pp. 215-221).

With the successful completion of these steps it is possible to determine the homogeneous transformation matrices for each joint and finally the overall homogeneous transformation matrix utilizing all of the joint transformation matrices can be calculated.

The resulting transformation matrix (T) or matrices can then be utilized by associating a vector within the body frame of a single rigid body to the inertial frame of the series of rigid bodies. This is demonstrated by Figure 2.1 and Equation 2.1.

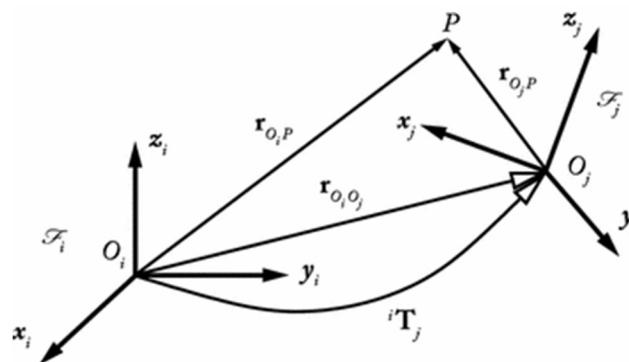


Figure 2.1 Rigid body (j) within the inertial frame (i) with two vectors defining the location of point P in both frames (Briot & Khalil, 2015, p. 20).

$$\mathbf{r}_{o_i P} = T_{ij} \mathbf{r}_{o_j P} \quad (2.1)$$

2.2. Kinematic Modeling – Screw Theory Method

Screw theory is a tool used for kinematic analyses of linked rigid body kinematic systems like robotic arms. Its origins in Mozzi and Chasles Theorem date back to 1763; however, its application to kinematic rigid body systems, like robotic arms was first published in 1999 when Roth and Tsai developed a method to represent the pose of a kinematic chain based on Chasles Theorem (Tsai, 1999, p. 42).

A screw has geometric properties that represent both rotational and translational components. It has a singular axis, on which both quantities are defined, and a scalar pitch, which relates translation and rotation (Davidson, 2004, p. 763). Screw theory associates physical meaning to a purely geometric entity, by its use to express velocities (angular and linear ones) as twists, and forces as wrenches. A twist (ϵ) represents the instantaneous motion of a rigid body relative to a referenced axis of rotation. The typical expression denotes ω as the angular velocity of rotation about a certain axis and v is the instantaneously linear velocity of translation along the same axis, as shown in Figure 2.2.

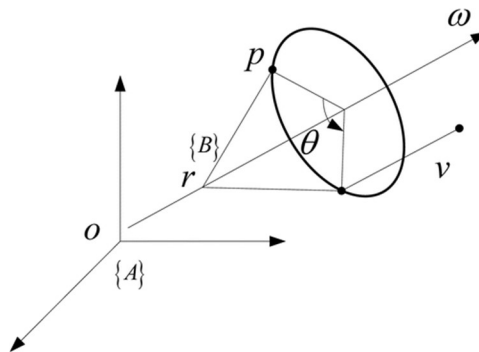


Figure 2.2 General screw motion displaced from an inertial frame (Chen, 2015, p. 2).

For a general case, the transformation matrix from the origin to a point on the rigid body is referred to as the initial configuration of the body coordinate system B relative to the inertial coordinate A at the base. After the twist motion of the rigid body, the final configuration of B to A in terms of a twist exponential is Equation 2.2 (Chen, 2015, p. 3).

$$T_{ab}(\theta) = e^{\hat{\epsilon}\theta}T_{ab}(0) \quad (2.2)$$

For a set of rigid bodies connected in series, the expression for twist is represented by Equation 10. The subscript (i) of Equation 2.3 refers to the sequence number for the rigid body in the system (Chen, 2015, p. 3).

$$\hat{\epsilon}_i = \begin{bmatrix} \omega_i^\times & r_i \\ 0 & 0 \end{bmatrix} \quad (2.3)$$

For an open-chain serial robot with total number of n revolute joints, the forward kinematics can be achieved by the product of the exponential formula in Equation 2.1 multiplied for each joint and finally multiplied against the initial static configuration of the robotic arm to achieve Equation 2.4 (Chen, 2015, p. 3).

$$T_{0n}(\theta) = e^{\hat{\epsilon}_i\theta_i}e^{\hat{\epsilon}_{i+1}\theta_{i+1}} \dots e^{\hat{\epsilon}_{n-1}\theta_{n-1}}T_{0n}(0) \quad (2.4)$$

The DH parameter and Screw Theory methods result in the same transformation matrices. For a dynamic system, the expression is maintained in its analytical form to be utilized for any kinematic process that requires the system to be defined geometrically. The transformation matrix is able to identify the forward kinematic transformation matrix for any set of joint angles of a linked series of rigid bodies. The process of obtaining the kinematic model of a manipulator is summarized from Chen (2015) by following steps:

1. Choose a fixed coordinate system from where the screws will be expressed.
2. Define a reference configuration for the manipulator, from which screw parameters \mathbf{s} and \mathbf{s}_0 for each joint of the manipulator are determined.

3. For each joint, identify the Rodrigues parameters as well as the joint variable (i).
4. Determine the homogeneous transformation matrices for each joint.
5. Determine the overall homogeneous transformation matrix.

Screw theory is always represented relative to a reference coordinate system.

Although in manipulators the base reference system is often of interest, any reference system can be used to determine the kinematic equations of the robot. That allows the choice of one that simplifies the computation of the screws which will in turn simplify the transformation matrices.

2.3. Differential Kinematics – Denavit-Hartenberg Method

There are two ways to derive the map between individual body frame and inertial frame velocities. The simple geometric analytical approach observes that each joint contributes to the end-effector movement. Therefore, the total velocity of it is the sum of these contributions, shown in Equation 2.5, where \mathbf{P}_{i-1} is the position vector of the end-effector from the origin of the $i-1$ frame, and $\dot{\mathbf{q}}$ is the column vector of joint angular velocity (Rocha & Tonetto, 2011, p. 726).

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{v}_n \\ \boldsymbol{\omega}_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n [\dot{\theta}_i (\mathbf{z}_{i-1} \times \mathbf{P}_{i-1}) + \mathbf{z}_{i-1} \dot{d}_i] \\ \sum_{i=1}^n \dot{\theta}_i \mathbf{z}_{i-1} \end{bmatrix} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (2.5)$$

2.4. Differential Kinematics – Screw Theory Method

In this method, the end-effector velocity frame is relative to a point instantaneously coincident with the origin of the inertial frame (to which all joint screws are expressed). The end-effector velocity is expressed by the twist from Equation 2.2, which is equal to the sum of the twist of all joints shown in Equation 2.6 (Rocha & Tonetto, 2011, p. 726).

$$\sum_{i=1}^n \hat{e}_i \dot{q}_i = [\hat{e}_1 \quad \cdots \quad \hat{e}_n] \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} = J(\mathbf{q}) \dot{\mathbf{q}} \quad (2.6)$$

The columns of the Jacobian are the normalized screws of each joint, while the magnitudes of the twists compose the joint velocities column vector $\dot{\mathbf{q}}$. Screw-based kinematic modeling has some advantages compared to modeling using DH parameters, although Screw Theory appears more complex to execute compared to a direct geometric approach like the DH parameter method.

The flexibility of reference frames using the Screw Theory transformation matrix setup method is a beneficial feature, since it eases the parameter identification process and can be used to obtain simplified analytical expressions for use with forward and inverse kinematics in robotic arm operations.

For this research the DH parameter method was utilized primarily because of the large quantity of supporting documentation available during the derivation process. Both methods yield the necessary homogenous transformation matrices with the only challenge being the limit of available resources to process these analytical expressions. More detailed discussion of the execution of the DH parameter method and the resulting Jacobian velocities as it applies to this research can be found in section 4.1.2.

3. Background

The background section provides the details regarding the modeling and tools used to produce the simulated testing environment. It also includes the general robotic operation processes, typical for all robotic systems.

3.1. Kinematics Modeling

In this simulation, system analysis needs to be done in order to study the dynamics of each part of the system. The kinematics analysis is divided into forward and inverse kinematics. Forward kinematics consists of finding the position of the end-effector in the inertial space knowing joint angles as $f(\theta_1, \theta_2, \dots, \theta_n) = [x, y, z]$ and the inverse kinematics consist of determining the required joint angles to achieve a known end-effector position, $f(x, y, z) = [\theta_1, \theta_2, \dots, \theta_n]$. The relationship between the joint angles and end-effector position is more clearly displayed in block diagram form in Figure 3.1.

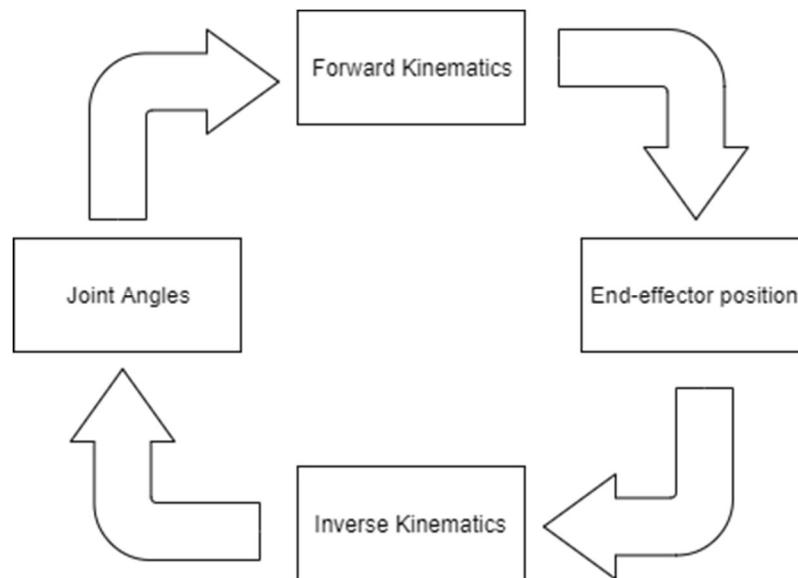


Figure 3.1 Cycle of geometric parameters being manipulated to determine either the end effector position or joint angles.

Forward and inverse kinematics utilize transformation matrices which consist of displacement and rotation matrices to determine the desired geometric parameters. A more detailed discussion of this calculation is provided in section 4.1.

3.2. Simulation and Modeling

Effective modeling of robotic arms can be done with a variety of existing software resources. For the purpose of this simulation, the robotic arm requires a mathematical programming language to calculate the kinematic and kinetics. The controllers operating on the robotic arm and the XY-table require a graphical simulation environment. Finally, for 3D modeling of the system, a computer animated design (CAD) software is required.

3.2.1. MATLAB

The programming language MATLAB provides the appropriate tools and resources for performing the kinematics and kinetic calculations. This software includes symbolic variable solutions and matrix algebra functions for rapid processing. Performing these calculations by hand is very tedious and allows significant opportunity for error. Without the support of computing resources, complex dynamic modeling would not be possible. The programming language has a robotics package and SolidWorks plug-in to support building the simulation into a 3D environment.

3.2.2. Simscape Multibody

Although MATLAB does have the resources to complete the simulation of joint position angles and end-effector position error, Simulink offers a graphical structure for the controller and methods to display the physical model through its Simscape Multibody add-on. This add-on allows 3D CAD models to be imported into a simulation environment

where the user can apply dynamic motion to the system using a controller. This forms the simulation environment for analyzing the robotic arm's movement.

3.2.3. SolidWorks

A 3D CAD software like SolidWorks provides the ability to verify the mass properties of each rigid body within the Sawyer robotic arm for use in verifying the inertia matrices for the individual rigid bodies and the system inertia matrix in its inertial frame. Additionally, it provides a resource for applying accurate origins and coordinate frame orientations matching those used for the coordinate frames from the DH parameters. Resolving this issue in a CAD program prevents complications when the 3D models are imported into the simulation environment. Conflicts between the inherited joint origin and the origin needed for dynamic modeling prevent proper visualization of the robotic arm's movement and the XY tables response.

4. Kinematics

The solution method proposed for determining the reaction forces at the base of the robotic arm require knowledge of the joint angles, angular velocities and angular accelerations for the Sawyer robot arm. These variables can be provided by an inverse kinematic controller, then applied to a set of analytical forward kinematic equations to determine the center of mass (CM) of the robot arm at a given time (t). The differential kinematics of the CM position provide the analytical expressions for inertial, centripetal and Coriolis forces that comprise the force vector at the center of mass and in turn the reaction force at the base of the robotic arm. This solution process for calculating the forces generated by the Sawyer arm is summarized by the block diagram in Figure 4.1.

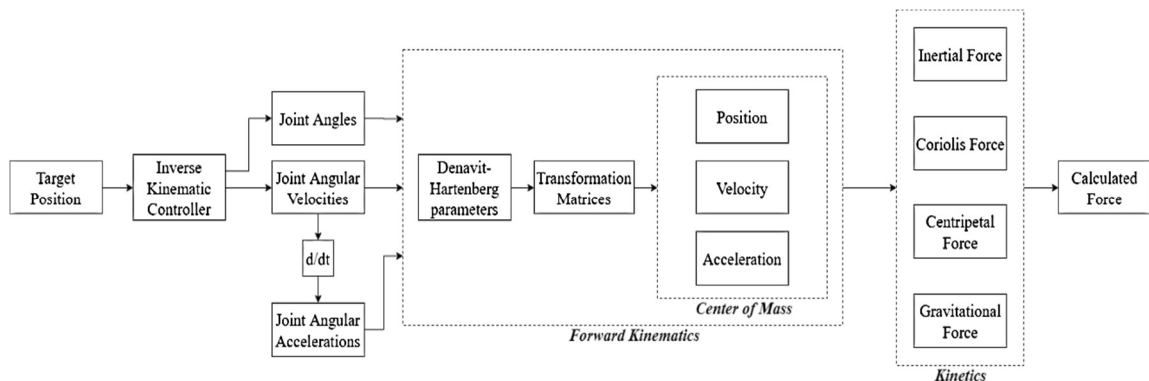


Figure 4.1 Solution process for calculating the forces of the Sawyer robot arm.

4.1. Forward Kinematics and System Dynamics

For a robotic arm, each of the rigid bodies within the system must be characterized by determining a geometric relationship between all of the joints that can be utilized for determining the end-effector location. The goal is to determine the displacement of the end-effector as a vector of cartesian coordinates in the robotic arm's inertial frame as a

function of the unique joint angles. The general function for the position of the end-effector appears in the following form $f(q_1, q_2, \dots, q_n) = [x, y, z] \in \mathcal{R}^3$. There are two accepted methods for determining the end-effector location. one is a more direct geometric method utilizing rotation matrices with series of trigonometric functions; the second method is referred to as Screw Theory, which is a method of performing algebraic operations on vector pairs in three-dimensional space, which leads to exponential representations for the system transformation matrices. Both of these methods utilize DH parameters and provide an accurate end effector position, but vary in complexity to set up. This analysis will continue utilizing the DH parameters.

4.1.1. Coordinate Frame Assignment

One of the critical elements for the application of DH parameters is the location and orientation of the coordinate frames throughout the robotic arm. In some instances, it is possible to reduce the complexity of the final function with respect to joint angles by limiting rotations about unnecessary axes and having a displacement vector that has only one or two non-zero cartesian elements. The benefits of this effort will be clarified in the next sections. Figure 4.2 depicts the Sawyer robotic arm and the arrangement of each rigid body coordinate frame in the system.

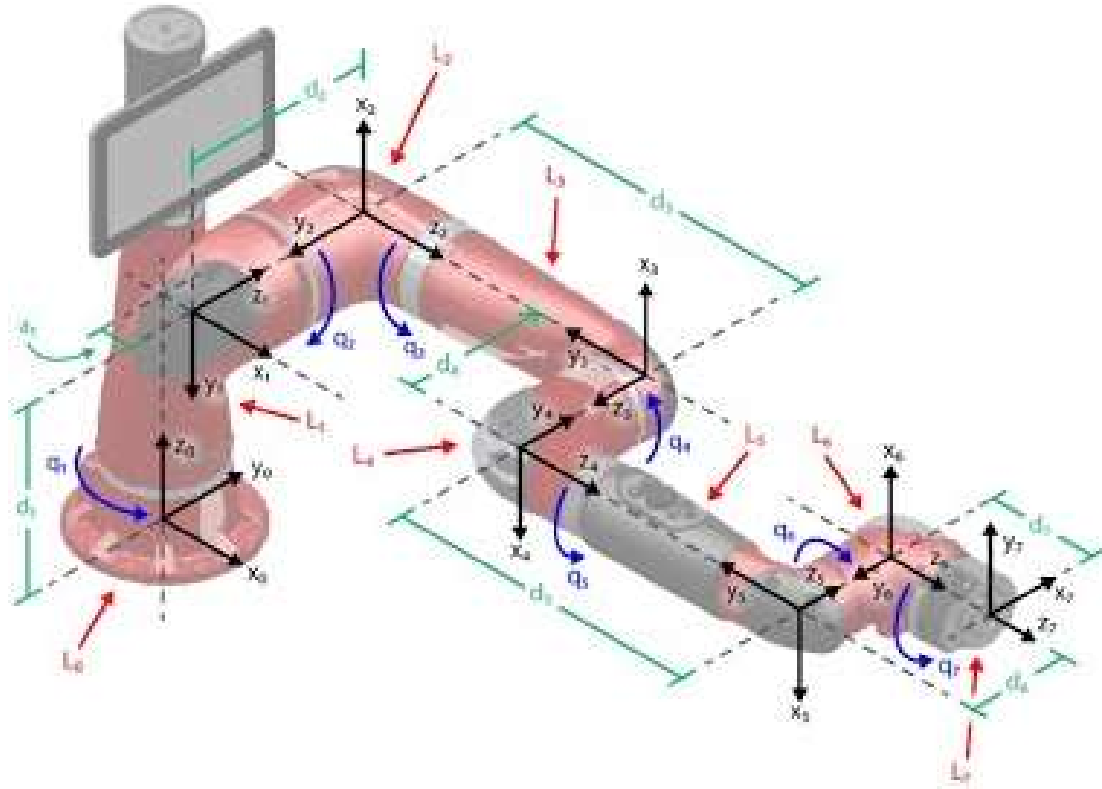


Figure 4.2 Geometric properties of the Sawyer robotic arm. Coordinate frames (black). Displacements (green). Rigid bodies (red). Revolute joints (blue) (Layeghi, 2017).

Figure 4.2 shows that there is a rotation centered about the z-axis of every coordinate system and all displacements occur along the z-axis of the rigid bodies, with the exception of frame 0 to frame 1, where the displacement vector has a single component of x-axis displacement. The overall robotic arm coordinate system layout minimizes the necessary calculations for finding the end-effector position as shown in the next sections.

4.1.2. Denavit-Hartenberg Parameters

With an efficient coordinate system placement, the DH parameters can be determined. These parameters are determined by following a series of steps to compose a table of values specific to the rigid body system. Summarizing the method used by Hayat in 2013 the steps are as follows for any two joined frames:

- 1) Determine z-axis rotation from frame $i-1$ to frame i .
- 2) Determine z-axis displacement from frame $i-1$ to frame i .
- 3) Determine x-axis rotation from frame $i-1$ to frame i .
- 4) Determine x-axis displacement from frame $i-1$ to frame i .

Continuing to perform this procedure from frame i to frame n results in a table of DH parameters. The DH parameters for the Sawyer robotic arm shown in Figure 4.1 are given in Table 4.1.

Table 4.1

DH Parameters (Sawyer robotic arm)

Link	q (deg)	d (mm)	a (mm)	α (deg)
0-1	0	317	81	-90
1-2	270	194.5	0	-90
2-3	0	400	0	-90
3-4	180	168.5	0	-90
4-5	0	400	0	-90
5-6	180	136.3	0	-90
6-7	270	134.75	0	0

(Layeghi, 2017)

4.1.3. Homogeneous Transformation Matrix

A homogeneous transformation matrix can be defined by combining the rotation matrix between any two connected rigid body frames with the corresponding displacement vector. The general form of the homogeneous transformation matrix for the system is shown in Equation 4.1. The transformation matrix (T) is a $[4 \times 4]$ matrix comprised of a $[3 \times 3]$ rotation matrix (R), the $[3 \times 1]$ displacement vector (D), and augmented by a series of zeroes and 1 in the 4th row to create a square matrix.

$$T = \begin{bmatrix} R & D \\ 0 & 1 \end{bmatrix} \quad (4.1)$$

A rotation matrix has the following form when defining a standard 3-2-1 Euler rotation angle for rotation of a coordinate frame about x, y and z axes.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \quad (4.2)$$

$$R_y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \quad (4.3)$$

$$R_z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

For the Sawyer robot arm the rotation matrix about the y-axis is not used because of the efficient placement of the coordinate systems for the DH parameters. As shown in Table 4.1, there was no definition for rotation about the y-axis so that element can be removed from the transformation matrix for the Sawyer arm.

The displacement vectors along the z-axis and the x-axis are D and A, respectively. The number of displacement vectors needed to calculate the transformation matrix is reduced because there is no displacement in the y-axis direction between any two frames.

$$D = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}; d_x = 0, d_y = 0, d_z = d \quad (4.5)$$

$$A = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}; a_x = \alpha, a_y = 0, a_z = 0 \quad (4.6)$$

By multiplying the rotation matrices and displacement vectors together the following general homogeneous transformation matrix between two frames can be found as

$$T = R_z D A R_x = \begin{bmatrix} \cos q & -\sin q \cos \alpha & \sin q \sin \alpha & a \cos q \\ \sin q & \cos q \cos \alpha & -\cos q \sin \alpha & a \sin q \\ 0 & \sin \alpha & \cos \alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

Each row of Table 4.1 provides a set of values that can be applied to Equation 4.7 which yields a transformation matrix for every set of linked rigid bodies (i.e. 0→1, ..., 6→7).

When a series of homogeneous transformation matrices is combined, the result is a matrix that defines the rotation and displacement between the initial frame and final frame of the sequence as follows.

$$T_{07} = T_{01} T_{12} T_{23} T_{34} T_{45} T_{56} T_{67} = \begin{bmatrix} 0 & 0 & 1 & 1015.75 \\ 1 & 0 & 0 & 160.3 \\ 0 & 1 & 0 & 317 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

The transformation matrix shown for T_{07} in Equation 4.8 is the result of inputting all of the DH parameters in Table 4.1 that were determined from Figure 4.1. Therefore, Equation 4.8 represents the transformation of the end effector into the base frame 0 for the initial configuration of the Sawyer robotic arm. In general, the joint angles (q_i) about the z-axis are not constants, but rather are variables because these joint angles change throughout the robotic arm's operation. For that reason, the transformation matrices must be expressed in their analytical form for all general calculations using the transformation matrices.

4.1.4. Velocity Jacobian Matrices

For use in the robotic arm controller described in detail in section 4.2, the velocity Jacobian matrices are comprised of two primary components: the linear and angular velocity Jacobian. These matrices relate the linear and angular velocity of the end-effector to the angular velocity of the individual joints. Each Jacobian matrix is a [3x7]

matrix and the concatenated final matrix is a [6x7] Jacobian matrix that is utilized for determining joint velocity or system kinetic energy, shown in Equation 4.9.

$$J = \begin{bmatrix} J_v(\mathbf{q}) \\ J_\omega(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} \frac{\partial v_x}{\partial q_1} & \frac{\partial v_x}{\partial q_2} & \frac{\partial v_x}{\partial q_3} & \frac{\partial v_x}{\partial q_4} & \frac{\partial v_x}{\partial q_5} & \frac{\partial v_x}{\partial q_6} & \frac{\partial v_x}{\partial q_7} \\ \frac{\partial v_y}{\partial q_1} & \frac{\partial v_y}{\partial q_2} & \frac{\partial v_y}{\partial q_3} & \frac{\partial v_y}{\partial q_4} & \frac{\partial v_y}{\partial q_5} & \frac{\partial v_y}{\partial q_6} & \frac{\partial v_y}{\partial q_7} \\ \frac{\partial v_z}{\partial q_1} & \frac{\partial v_z}{\partial q_2} & \frac{\partial v_z}{\partial q_3} & \frac{\partial v_z}{\partial q_4} & \frac{\partial v_z}{\partial q_5} & \frac{\partial v_z}{\partial q_6} & \frac{\partial v_z}{\partial q_7} \\ \frac{\partial \omega_x}{\partial q_1} & \frac{\partial \omega_x}{\partial q_2} & \frac{\partial \omega_x}{\partial q_3} & \frac{\partial \omega_x}{\partial q_4} & \frac{\partial \omega_x}{\partial q_5} & \frac{\partial \omega_x}{\partial q_6} & \frac{\partial \omega_x}{\partial q_7} \\ \frac{\partial \omega_y}{\partial q_1} & \frac{\partial \omega_y}{\partial q_2} & \frac{\partial \omega_y}{\partial q_3} & \frac{\partial \omega_y}{\partial q_4} & \frac{\partial \omega_y}{\partial q_5} & \frac{\partial \omega_y}{\partial q_6} & \frac{\partial \omega_y}{\partial q_7} \\ \frac{\partial \omega_z}{\partial q_1} & \frac{\partial \omega_z}{\partial q_2} & \frac{\partial \omega_z}{\partial q_3} & \frac{\partial \omega_z}{\partial q_4} & \frac{\partial \omega_z}{\partial q_5} & \frac{\partial \omega_z}{\partial q_6} & \frac{\partial \omega_z}{\partial q_7} \end{bmatrix} \quad (4.9)$$

The linear velocity Jacobian (J_v) is composed of a series of partial derivatives of the cartesian velocity components with respect to the joint angles, as shown in Equation 4.10.

$$J_v(q_i) = \mathbf{z}_{i-1} \times (\mathbf{o}_n - \mathbf{o}_{i-1}) = \begin{bmatrix} \frac{\partial v_x}{\partial q_i} \\ \frac{\partial v_y}{\partial q_i} \\ \frac{\partial v_z}{\partial q_i} \end{bmatrix}, \quad (4.10)$$

The vector \mathbf{z}_{i-1} refers to the z-axis element, or third column of the rotation matrix within the transformation matrix (Equation 4.1) for a set of rigid bodies. The vector \mathbf{o}_n refers to the vector from the base frame (frame 0) to the end-effector in frame 7. Vector \mathbf{o}_{n-1} references the vector from the base frame to the center of mass (CM) of the rigid body that contains the rotation vector \mathbf{z}_{i-1} . This vector is important because the desired velocity of a rigid body is always uniform through its CM in the robotic arm's inertial frame. For additional reference, these vectors are shown in a general layout for an arbitrary series of rigid bodies in Figure 4.3.

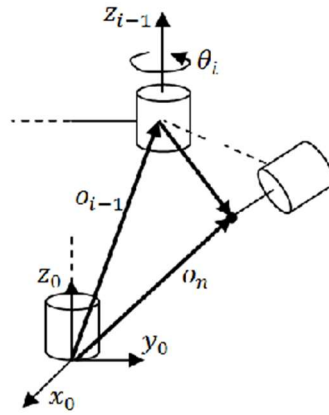


Figure 4.3 Velocity Jacobian matrix vectors for revolute joints (TTU Advanced Robotics, 2010).

The expression given in Equation 4.10 is specifically for determining the linear velocity Jacobian of a single revolute joint. Alternate forms of this expression are required for spherical joints and prismatic joints. For the Sawyer robotic arm, all joints are revolute. By applying Equation 4.10 for each joint, the linear velocity Jacobian matrix takes the following form:

$$J_v(\mathbf{q}) = \begin{bmatrix} \frac{\partial v_x}{\partial q_1} & \frac{\partial v_x}{\partial q_2} & \frac{\partial v_x}{\partial q_3} & \frac{\partial v_x}{\partial q_4} & \frac{\partial v_x}{\partial q_5} & \frac{\partial v_x}{\partial q_6} & \frac{\partial v_x}{\partial q_7} \\ \frac{\partial v_y}{\partial q_1} & \frac{\partial v_y}{\partial q_2} & \frac{\partial v_y}{\partial q_3} & \frac{\partial v_y}{\partial q_4} & \frac{\partial v_y}{\partial q_5} & \frac{\partial v_y}{\partial q_6} & \frac{\partial v_y}{\partial q_7} \\ \frac{\partial v_z}{\partial q_1} & \frac{\partial v_z}{\partial q_2} & \frac{\partial v_z}{\partial q_3} & \frac{\partial v_z}{\partial q_4} & \frac{\partial v_z}{\partial q_5} & \frac{\partial v_z}{\partial q_6} & \frac{\partial v_z}{\partial q_7} \end{bmatrix} \quad (4.11)$$

The elements of the linear velocity Jacobian matrix are composed of the same analytical expressions from the transformation matrices; therefore, the velocity Jacobian matrix will vary as the joint angles change.

The angular velocity Jacobian (J_ω) matrix takes a similar form as the linear velocity Jacobian matrix. As before the expression results in a $[3 \times 7]$ matrix comprised of partial

derivatives with respect to the joint angles, but the vector is simply the third column of the previous frame's rotation matrix, as shown in Equation 4.12.

$$J_v(q_i) = \mathbf{z}_{i-1} = \begin{bmatrix} \frac{\partial v_x}{\partial q_i} \\ \frac{\partial v_y}{\partial q_i} \\ \frac{\partial v_z}{\partial q_i} \end{bmatrix}, \quad (4.12)$$

Also, similar to the linear velocity Jacobian, Equation 4.12 only applies to revolute joints. The matrix formed when Equation 4.12 is applied to all joints in the Sawyer robotic arm becomes

$$[J_v(\mathbf{q})] = \begin{bmatrix} \frac{\partial \omega_x}{\partial q_1} & \frac{\partial \omega_x}{\partial q_2} & \frac{\partial \omega_x}{\partial q_3} & \frac{\partial \omega_x}{\partial q_4} & \frac{\partial \omega_x}{\partial q_5} & \frac{\partial \omega_x}{\partial q_6} & \frac{\partial \omega_x}{\partial q_7} \\ \frac{\partial \omega_y}{\partial q_1} & \frac{\partial \omega_y}{\partial q_2} & \frac{\partial \omega_y}{\partial q_3} & \frac{\partial \omega_y}{\partial q_4} & \frac{\partial \omega_y}{\partial q_5} & \frac{\partial \omega_y}{\partial q_6} & \frac{\partial \omega_y}{\partial q_7} \\ \frac{\partial \omega_z}{\partial q_1} & \frac{\partial \omega_z}{\partial q_2} & \frac{\partial \omega_z}{\partial q_3} & \frac{\partial \omega_z}{\partial q_4} & \frac{\partial \omega_z}{\partial q_5} & \frac{\partial \omega_z}{\partial q_6} & \frac{\partial \omega_z}{\partial q_7} \end{bmatrix} \quad (4.13)$$

The concatenated [6x7] velocity Jacobian matrix will have the form shown in Equation 4.9. This matrix will be a supporting element in the derivation of the equations of motion in the following sections.

4.1.5. Center of Mass

At any time (t), the robotic arm can be considered a single rigid body, fixed in its inertial frame. At that time, the center of mass (CM) has a specific location as a function of the joint angles $\{q_i\}_{i=1}^7$. Finding this position from the known CM in each link's body frame requires first transforming the body frame CM vectors to the inertial frame. This is done by transforming the body frame CM vector (\mathbf{p}_{CM}) to the inertial frame CM vector (\mathbf{P}_{CM}). The general expression for this process is shown in Equation 4.14.

$$\mathbf{P}_{CM_{0 \rightarrow i}} \Big|_{inertial} = \begin{pmatrix} x_{CM_i} \\ y_{CM_i} \\ z_{CM_i} \end{pmatrix} = T_{0 \rightarrow i} \mathbf{p}_{CM_{0 \rightarrow i}} \Big|_{body} \quad (4.14)$$

With seven rotating joints and eight total bodies, the base frame 0 is simply multiplied by identity to determine its inertial frame CM.

Next, the entire robotic arm's CM vector is determined by combining the mass of each joint (m_i) with its corresponding CM vector as shown in Equation 4.15.

$$\mathbf{CM}(\mathbf{q}) = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{CM} = \frac{\sum_{i=0}^n m_i * P_{CM_i}}{\sum_{i=0}^n m_i} \quad (4.15)$$

Equation 4.15 is kept in an analytical form that is dependent on the joint angles at time (t). The first and second derivatives with respect to time of the \mathbf{CM} shown in Equation 4.16 are utilized to determine the inertial, Coriolis and centripetal forces described in the following sections.

$$\mathbf{v}_{CM}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{d}{dt} \mathbf{CM}(\mathbf{q}) \quad (4.16)$$

$$\mathbf{a}_{CM}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \frac{d^2}{dt^2} \mathbf{CM}(\mathbf{q}) \quad (4.17)$$

The expressions of the time derivatives of the CM vector are functions of angular velocity and angular acceleration at each joint for the given time t.

4.1.6. Gravitational Force

The next portion of the system dynamics derivation is to determine the gravitational torque applied by the gravitational force of the system. This requires knowledge of the mass for each link (m_i), the gravitational acceleration vector acting on the link (\mathbf{g}) and the current position CM vector (\mathbf{CM}) as a function of the joint angles at time (t) The gravitational force of each link is given by Equation 4.18.

$$\mathbf{F}_g = \sum_{i=1}^n m_i * \mathbf{g} \cdot \mathbf{CM} \quad (4.18)$$

As the joints rotate, the gravitational force will change as a function of $\mathbf{q}(t)$ much like the velocity Jacobians. All elements supporting the dynamics derivation are dependent on

the joint angles at a given time. The force vector shown in Figure 4.4 is subtracted from the measured forces of the robotic arm about the inertial frames x and y axes to remove the contribution of gravitational effects acting on the system. This removes the effect of external forces except the ones that are provided by the XY-table's control inputs described in section 5.2.

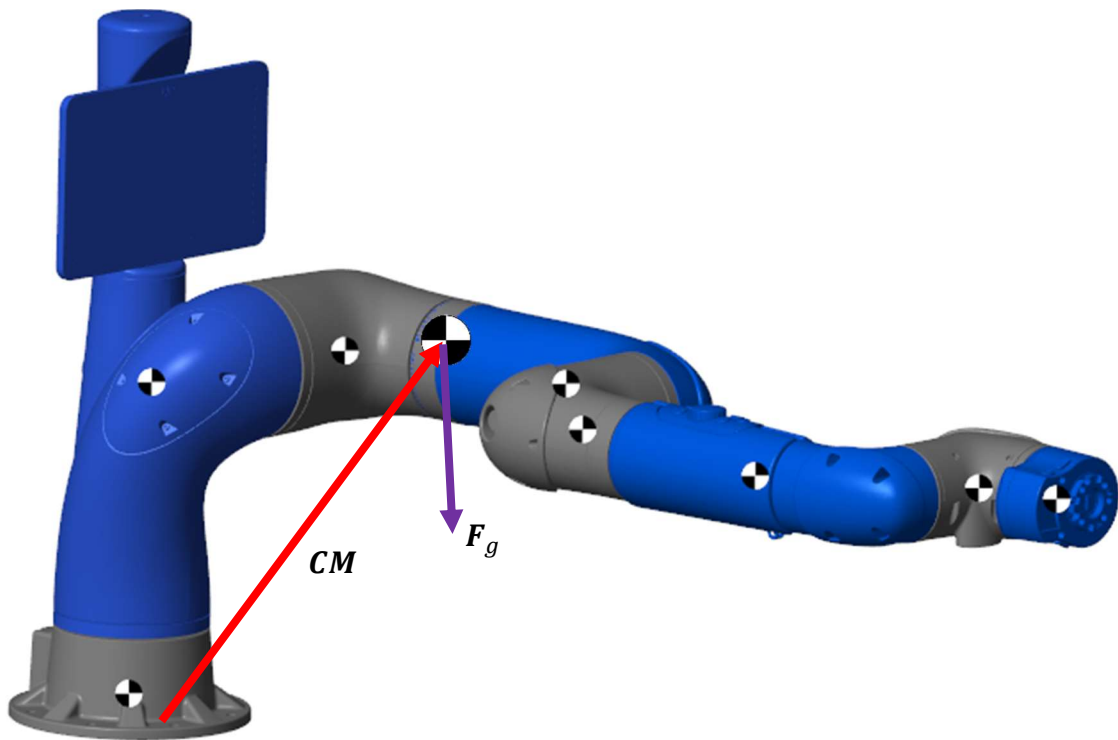


Figure 4.4 Gravitational force vector direction for Sawyer robot arm in start position.

4.1.7. Inertial Force

The inertial forces are referred to in Newton's second law of inertia, Equation 4.19.

These forces act on the robotic arm **CM**.

$$\mathbf{F}_{In} = \sum_{i=0}^n m_i * \mathbf{a}_{CM} \quad (4.19)$$

4.1.8. Coriolis Force

In order to account for the Coriolis forces acting on the system at the **CM**, the following expression for Coriolis forces acting on a single rigid body $\in \mathcal{H}^3$ is written as shown in Equation 4.20.

$$\mathbf{F}_{Co} = -2 * \boldsymbol{\omega}_{CM} \times \mathbf{v}_{CM} \quad (4.20)$$

Equation 4.18 can be re-written using the linear vectors of **CM** and \mathbf{v}_{CM} .

$$\mathbf{F}_{Co} = -2 * (\mathbf{CM} \times \mathbf{v}_{CM}) \times \mathbf{v}_{CM} \quad (4.21)$$

4.1.9. Centripetal Force

The centripetal forces of the system acting on a single rigid body that $\in \mathcal{H}^3$ is given in Equation 4.22.

$$\mathbf{F}_{Ce} = -\sum_{i=0}^n m_i * \boldsymbol{\omega}_{CM} \times (\boldsymbol{\omega}_{CM} \times \mathbf{CM}) \quad (4.22)$$

Equation 4.23 can be re-written similar to Equation 4.18 as follows.

$$\mathbf{F}_{Ce} = -\sum_{i=0}^n m_i * (\mathbf{CM} \times \mathbf{v}_{CM}) \times [(\mathbf{CM} \times \mathbf{v}_{CM}) \times \mathbf{CM}] \quad (4.23)$$

4.1.10. Equations of Motion

The resulting equation of motion for the Sawyer robot arm, Equation 4.24, is a summation of the forces acting at the robotic arm's center of mass at time (t) and the reaction forces at the pedestal (\mathbf{F}_R) acting opposite to the forces at the robotic arm center of mass (\mathbf{F}_{CM}), Equation 4.25 (Schaub & Junkins, 2009, p. 28).

$$\mathbf{F}_{CM} = \mathbf{F}_{In} + \mathbf{F}_{Ce} + \mathbf{F}_{Co} \quad (4.24)$$

$$\mathbf{F}_{CM} = -\mathbf{F}_R \quad (4.25)$$

Equation 4.25 relates all of the angular rates, angular accelerations, joint angles, inertias and masses for the robotic arm into a singular force vector in the inertial frame

that can be applied as a calculated disturbance to the XY-table in the open and closed-loop control systems for the robotic arm.

The base of the robotic arm is fixed to the pedestal upon which it is mounted. As a result, the reaction force in Equation 4.25 acts directly on the pedestal of the XY-table. For this reason, an expression relating the force applied at the pedestal by the robotic arm and the control wheels of the XY-table must be derived.

The four free-body diagrams representing the cable tensions about each pulley segment for applied force along the x-axis of the inertial frame (F_x) of the XY-table are shown in Figure 4.5.

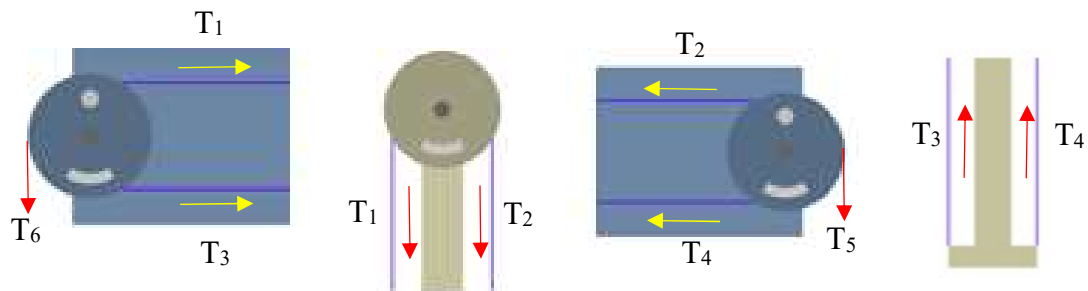


Figure 4.5 XY-table Free-Body Diagram(s) for force along the x-axis.

Force can only be exerted in the +x-direction if both control pulleys are applying additional tension, T_5 and T_6 , in the directions shown in Figure 4.3. The summation of forces for these segments leads to the relationship between the forces in 2D and the applied control pulley forces, Equation 4.26.

$$\sum F_x = -T_2 + T_4 - T_1 + T_3 - T_5 + T_6 - F_x = 0 \quad (4.26)$$

A single connected cable has equal tension throughout the XY-table so all segments can be defined as a single value of tension (T) in Equation 4.27.

$$T = T_1 = T_2 = T_3 = T_4 \quad (4.27)$$

The tensions at the control pulleys are applied when a torque (τ) is needed. For this reason, they add to the tension on either side of the pulley depending on the direction the torque is applied.

$$\tau_5 = rT_5 \quad (4.268)$$

$$\tau_6 = rT_6 \quad (4.29)$$

$$F_x = \frac{1}{r}(-\tau_5 + \tau_6) \quad (4.30)$$

For force along the +y-axis (F_y) of the XY-table inertial frame, the control pulleys must apply the additional tensions, T_5 and T_6 , in the directions shown in Figure 4.6.

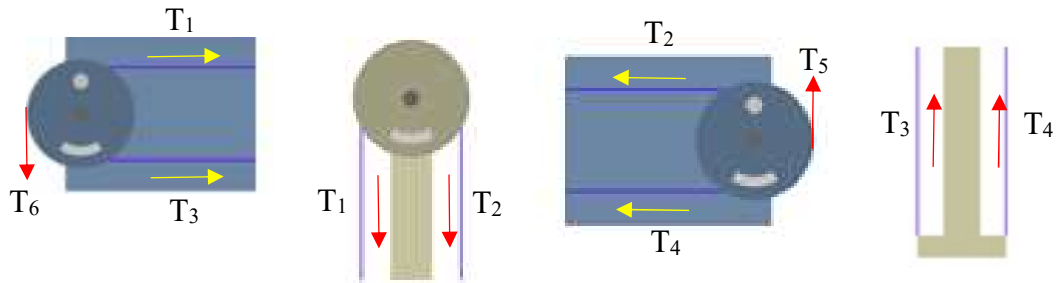


Figure 4.6 XY-table Free-Body Diagram(s) for force along the y-axis.

$$\sum F_y = T_1 - T_2 - T_4 + T_5 + T_3 + T_6 - F_y = 0 \quad (4.31)$$

$$F_y = \frac{1}{r}(\tau_5 + \tau_6) \quad (4.32)$$

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \tau_5 \\ \tau_6 \end{bmatrix} \quad (4.33)$$

$$\begin{bmatrix} \tau_5 \\ \tau_6 \end{bmatrix} = r \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} F_x \\ F_y \end{bmatrix} \quad (4.34)$$

Recalling the XY-table has been reduced to a system of approximately massless solid elements to imitate an air-bearing frictionless surface, Equation 4.34 must be represented without mass; otherwise it would result in trivial solutions.

$$\begin{bmatrix} mra_5 \\ mra_6 \end{bmatrix} = r \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} ma_x \\ ma_y \end{bmatrix} \quad (4.35)$$

$$\begin{bmatrix} \alpha_5 \\ \alpha_6 \end{bmatrix} = r \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} a_x \\ a_y \end{bmatrix} \quad (4.36)$$

The resulting equation of motion for the XY-table relates the linear acceleration of the pedestal in the inertial frame to the angular acceleration of the control pulleys. One final step is to relate Equation 4.36 to the position of the pedestal in the inertial frame using the angle of rotation of the torque pulleys. This is done by the double integration of Equation 4.36 with respect to time in Equation 4.37

$$\begin{bmatrix} \theta_5 \\ \theta_6 \end{bmatrix} = r \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} p_x \\ p_y \end{bmatrix} \quad (4.37)$$

Equation 4.37 is associated with the XY-table as shown in Figure 4.7. The system is now completely defined and a control system for the XY-table can be developed based on Equation 4.37 (AbdelHamid, Abdeldayem, & Mabrouk, 2018, p. 65).

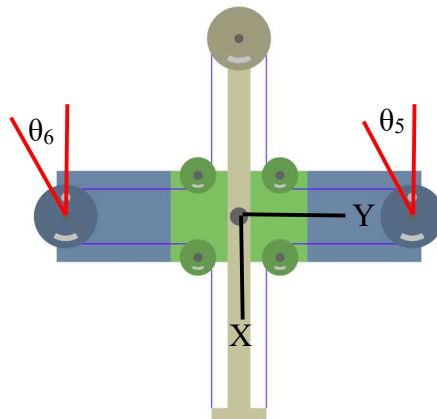


Figure 4.8 Coordinate frames of the XY-table including positive angle orientation.

5. Control Systems

The control systems section defines the various controllers used throughout the simulation. Primarily the robotic arm and XY-table controllers. This section also includes the details regarding their operation and controller development.

5.1. Inverse Kinematic Control

In order to control the system when a target location for the end-effector is selected, the inverse kinematics must be derived (Ali, Park, & Lee, 2010, p. 704). Several methods exist for inverse kinematics. The Sawyer arm is not avoiding obstacles within its operating area so there is no risk of collisions in the simulation. In order to develop a control system that assigns the joints angular rates $\{\dot{q}_i\}_{i=1}^7$ as the arm moves from its initial position (x_e) to the target position (x_d), a control system like the one shown in Figure 5.1 is required.

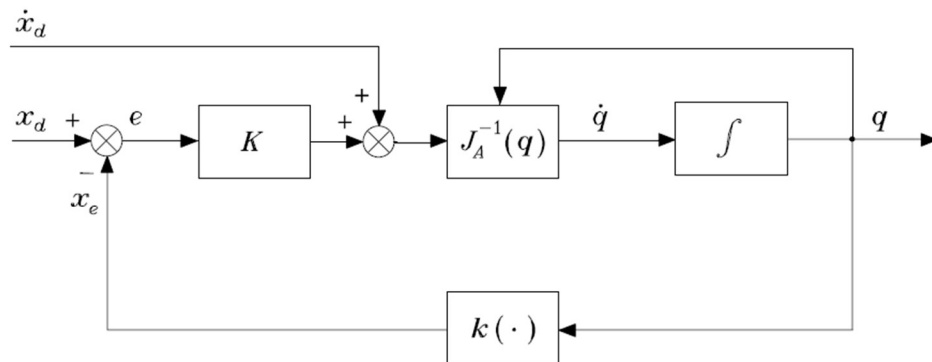


Figure 5.1 Inverse kinematic control system utilizing the Jacobian matrix inverse (Siciliano et al., 2010, p. 133).

For this control system the position error (e) and its relation to the Jacobian matrix (J) are shown as follows.

$$e = x_d - x_e \quad (5.1)$$

$$e = J\Delta q \quad (5.2)$$

The inverse Jacobian matrix will take the error of the position (e) and provide the necessary joint angle rates $\{\dot{q}_i\}_{i=1}^7$.

$$\Delta q = J^{-1}e \quad (5.3)$$

5.1.1. Damped Least Squares Inverse Jacobian Control

Due to issues with the inverse Jacobian at joint angles that create mathematical singularities within the Jacobian matrices a damped least squares method is implemented in the following form, where α is a weighing factor (Buss, 2004, p. 10).

$$J'_{LS} = J^T (JJ^T + \alpha^2 I)^{-1} \quad (5.4)$$

A large weighting factor increases the ability for the damped least squares to overcome singularities at the expense of final position accuracy.

The position error gain is given by the [6x6] matrix (K). The target velocity is added to the position error to better enable the arm to continue tracking a dynamic target position. The feedback value ($k(\cdot)$) applies the forward kinematics expressions described in section 4.1 for converting the resulting joint positions $\{q_i\}_{i=1}^7$ to cartesian coordinates in the base frame of the robotic arm.

5.1.2. Singularity Resolution

Due to the constant scalar weighting factor (α) applied to the damped least squares inverse Jacobian controller, the singularity mitigation causes the final position error not to converge to zero. The larger a weighting value is introduced to the controller, the less accurate the final position of the end-effector (Siciliano, 1990, p. 204). This result is caused by the weighting factor creating an additional error within the Jacobian inverse of

the controller to avoid elements along the diagonal of the Jacobian matrix from becoming zero. Any zeros within the Jacobian matrix would prevent the Jacobian matrix from being inverted and the controller would be unable to solve for the joint angular velocities (Siciliano & Bruno, 1990, p. 202).

To overcome this issue, the scalar gain matrix (K) is adjusted from a series of scalar values to a series of gains $K(t)$ that are functions of the simulation time t . For the controller being applied to the Sawyer robot arm, the gain matrix takes the form of Equation 5.5.

$$K(t) = \begin{bmatrix} 350 & 0 & 0 & 0 & 0 & 0 \\ 0 & 350 & 0 & 0 & 0 & 0 \\ 0 & 0 & 350 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.001 + (t * 300) \end{bmatrix} \quad (5.5)$$

The gain matrix increases the gain applied to joint 6 the longer the controller operates, which can be observed in Figure 5.2.

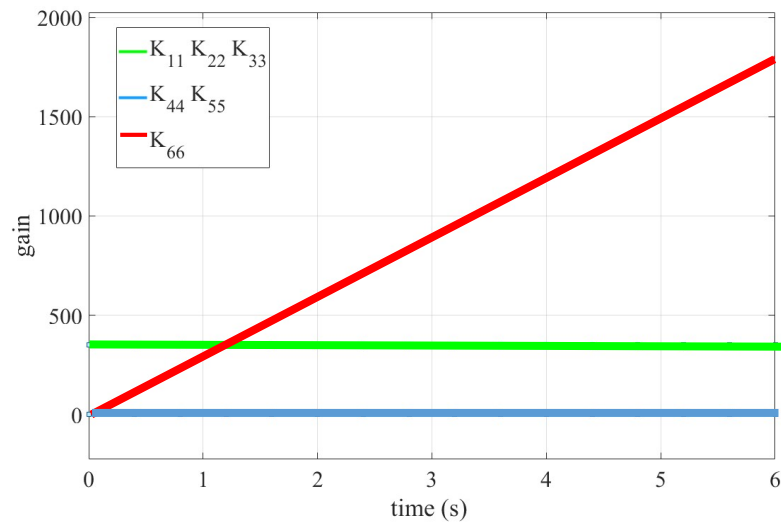


Figure 5.2 Time varying gain matrix for all test cases.

The result is the end-effector position error is continuously decreased by the controller to compensate for the additional error generated by adding weighting factor to the Jacobian control expression.

5.2. XY-table Control

The XY-table controls the force applied by the robotic arm at the pedestal by using a simple PID control system with a goal of reducing the steady state error for the XY-table once all of the joints of the robotic arm have arrived at fixed positions and no additional force is applied to the XY-table. The open-loop response attempts to do this strictly by utilizing the EOM for the Sawyer robotic arm to define the forces applied to the pedestal. From this result, the position of the table at time (t) is calculated and the XY-table reduces the steady-state error. The success of the open-loop system response is largely based on the successful calculation of the reaction forces at the pedestal.

The closed-loop response applies a position feedback loop that will combine the disturbance caused by the robotic arm with any remaining steady-state error. Once the robotic arm's joints reach fixed positions, the remaining steady-state error provided by the position sensor will be reduced by the controller to within machine precision tolerance.

5.2.1. Open-Loop Control

The open-loop controller consists of a target position in the system xy-plane and a position error at time t, caused by a disturbance which is generated by the movement of the Sawyer robotic arm. The block diagram of this controller is shown in Figure 5.3.

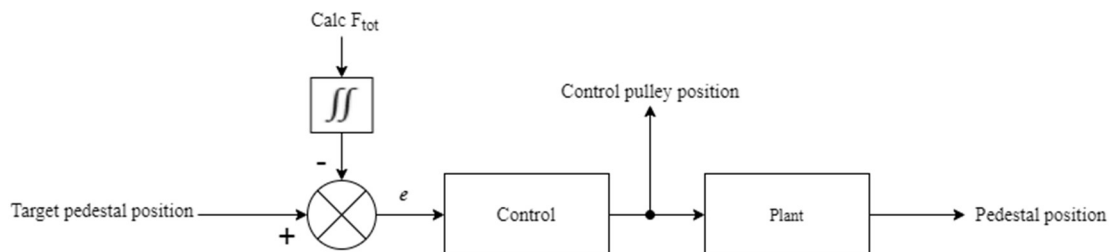


Figure 5.3 Open-loop control system applied to XY-table.

The control block for this controller is Equation 4.35, which generates the angular velocity of the control pulleys for the XY-table. The control blocks output is then applied to the control pulleys in order to maintain the pedestal center at the system origin.

5.2.2. Closed-loop

The closed-loop controller consists of the same design as the open-loop in section 5.2.1 with the addition of a position feedback loop. The block diagram of this controller is shown in Figure 5.4.

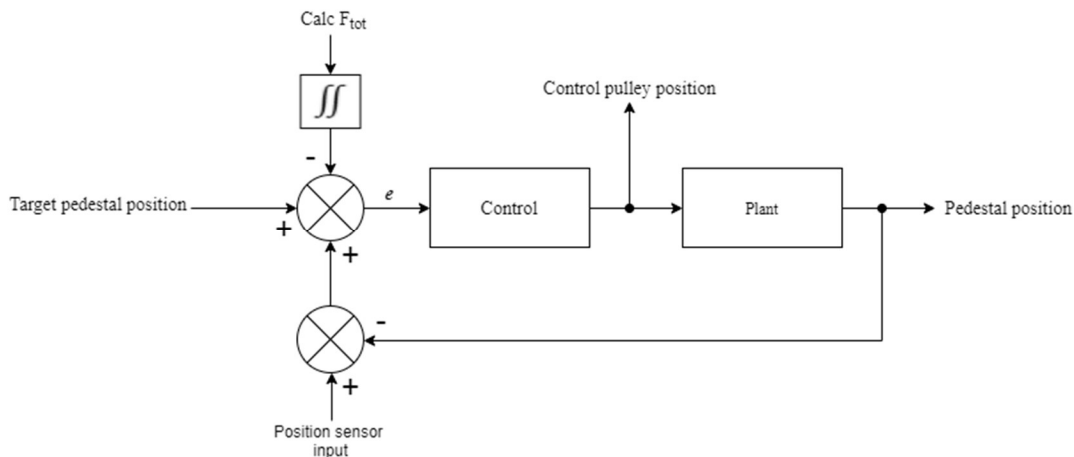


Figure 5.4 Closed-loop control system applied to XY-table.

Unlike the open-loop control system, the feedback loop in Figure 5.4 determines the difference between the current pedestal position and the measured pedestal position from the position sensor. The result is the residual error which is summed with the calculated position error to provide the controller with the actual position error of the pedestal. Once the arm's joint velocities have been reduced to zero, the remaining position error from the feedback will be corrected by the controller. This will allow the system to continue reducing the steady-state error to within machine precision. The feedback block in this controller will convert the control pulley output signal to the current corrected position before being compared to the measured position sensor input. The difference of these two signals will be passed on to the summing block to be combined with the disturbance to become the total system error at time (t).

6. Simulation Results

The following sections provide the detailed results of the various test cases performed within the Simscape Multibody simulation. As stated previously in this research the test cases performed are a series of robotic arm movements with 1-DOF and 7-DOF's with the XY-table providing open and closed loop stability to the system. The uncontrolled cases validate the problem statement in addition to the validity of the simulation environment.

6.1. Uncontrolled Cases

The implementation of the damped least squares inverse Jacobian controller yields an infinite number of test cases. For the purpose of this thesis, four cases have been chosen for analysis. Two cases represent simple movements of the robotic arm that provide a fundamental understanding of the applied and reaction forces for the complete system. The remaining two cases demonstrate the complex dynamic movements of a 7-DOF system and how these reaction forces provide unique and complex results for the open and closed-loop XY-table control systems.

6.1.1. Case I – Simple Movement I

For case I the movement only requires the end-effector to acquire the target position using a single axis. To ensure the controller does not use other joints to arrive at the target location and therefore keep the movement of the arm using as few degrees of freedom as possible, all undesired joint rotations have been restricted within the robotic arm controller to output an angular velocity of zero at all times. Reduced DOF movements add clarity to the model's overall performance. The 2-DOF movement rotates at joint 2

(q_2) and causes the robotic arm to raise directly upward along the zy-plane as shown in Figure 6.1a and b to arrive at the target end-effector position of $\vec{r} = \langle 0.081, 0.16, 1.25 \rangle$ m.

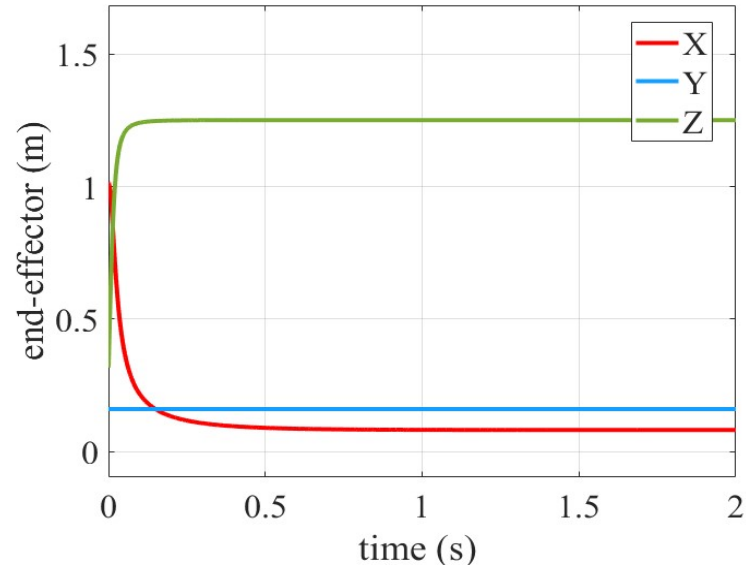


Figure 6.1a End-effector position throughout simulation of case I.

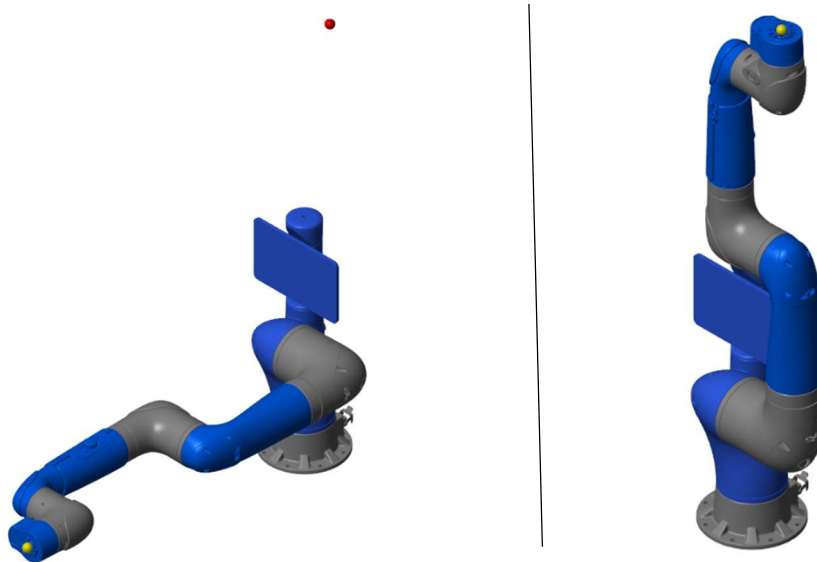


Figure 6.1b Initial and final positions of case I. XY-table removed for clarity. End-effector position (yellow). Target end-effector position (red).

The joint angles shown in Figure 5.1b are fixed throughout the simulation, with the exception of joint 2, which supports the observed movement in Figure 6.1b.

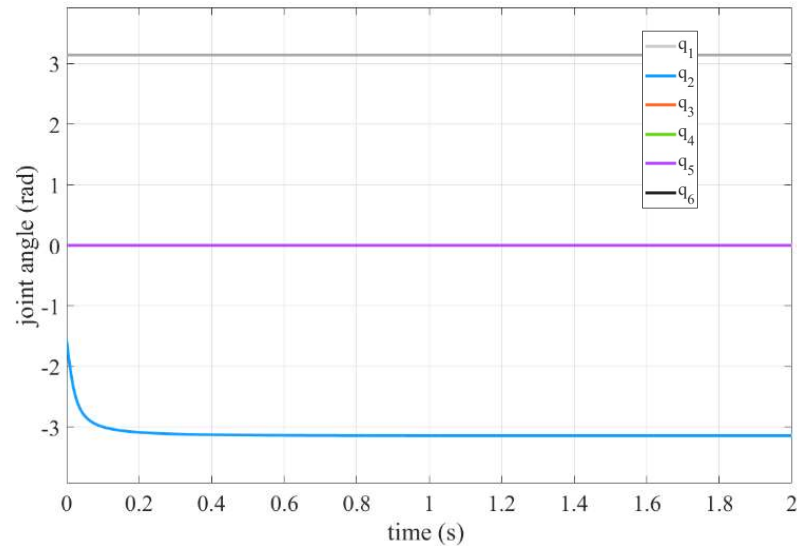


Figure 6.1c Joint angles from start position to fixed final positions when end-effector reaches target position for case I.

The resulting displacement and acceleration in Figure 6.2 demonstrate a result that is unique to the robotic arm's mass distribution and the aggressive response of the model due to high displacement error in the first 0.2 seconds. The magnitude of the force applied is larger in cases III and IV because the robotic arm is extended fully throughout the movement. As a result, the displacement of the pedestal occurs more rapidly and reaches 4 meters of displacement along a single axis. The expectation is that a uniform rotation along the zy-plane would cause the reaction force to accelerate the pedestal along the y-axis. This is observed in Figure 6.2.

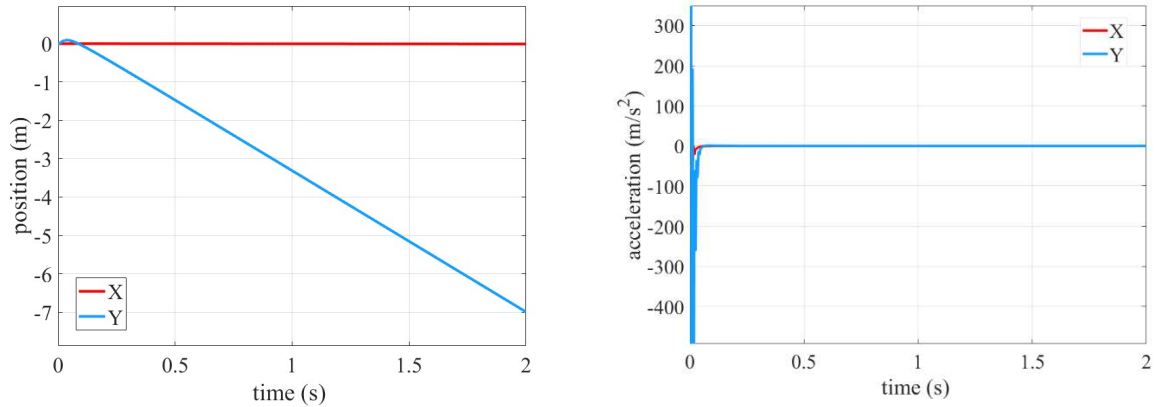


Figure 6.2 Planar displacement and acceleration of the XY-table pedestal center in case I.

6.1.2. Case II – Simple Movement II

Similar to case I, the movement only requires the end-effector to acquire the target position using a single axis of rotation. All undesired joint movements have been restricted within the robotic arm controller to output an angular velocity of zero at all times. The only joint rotating in case II is joint 1 (q_1), as shown in Figure 6.3a and b with a target end-effector position of $\vec{r} = \langle -0.1603, 1.016, 0.317 \rangle$ m.

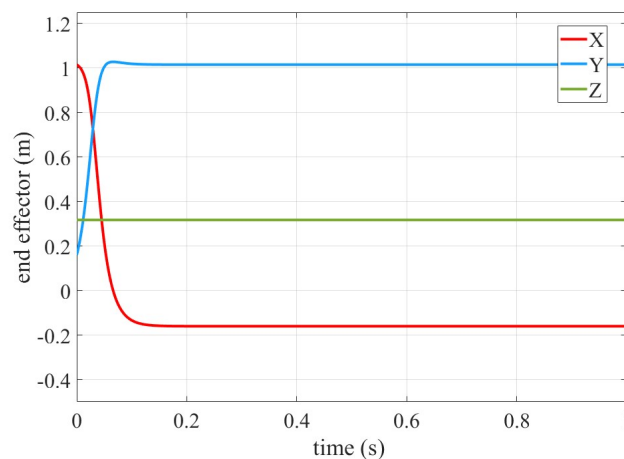


Figure 6.3a End-effector position throughout simulation of case II.

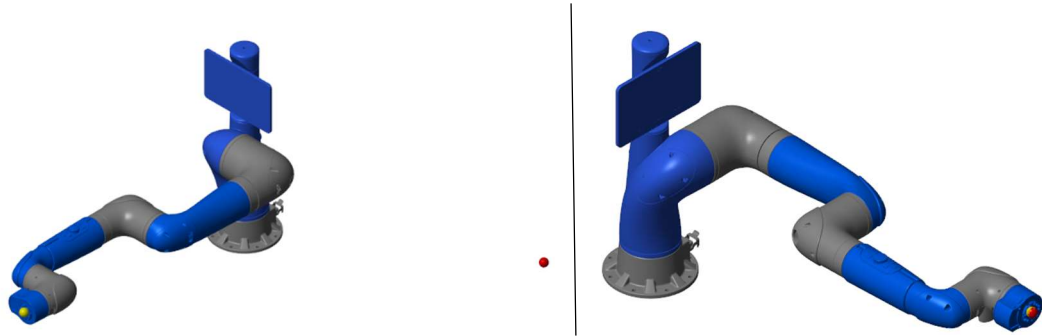


Figure 6.3b Initial and final positions of case II. XY-table removed for clarity. End-effector position (yellow). Target end-effector position (red).

Figure 6.3b displays joint 1 as the only joint increasing from $\frac{\pi}{2}$ to π in the xy-plane. All other joints remain fixed throughout the simulation.

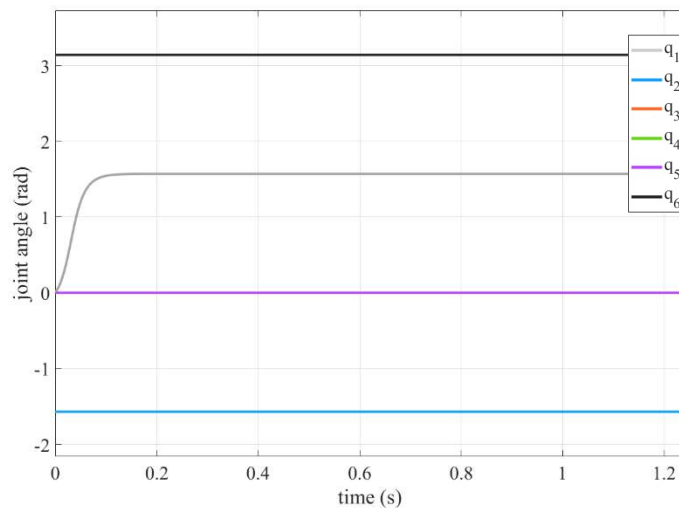


Figure 6.3c Joint angles from start position to fixed final positions when end-effector reaches target position for case II.

Figure 6.4 shows the pedestal acceleration starts with the rapid movement to reduce error with the reaction force vector along the y-axis and shortly after; as the angular velocity for joint 1 decrease, the applied force along the x-axis and the force along the y-

axis inverts, reducing acceleration along the y-axis. The results in a displacement that begins rapidly along the y-axis and sharply increases in magnitude along the x-axis shortly after in the simulation.

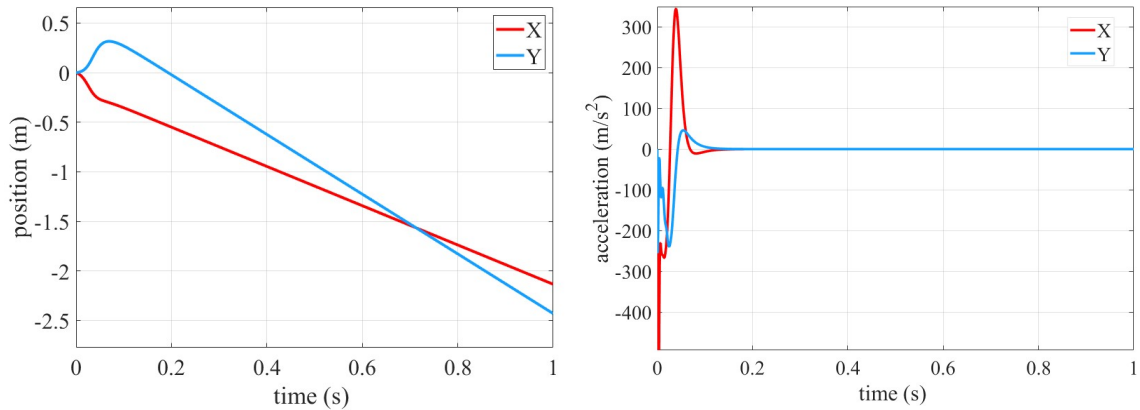


Figure 6.4 Planar displacement and acceleration of the XY-table pedestal center in case II.

6.1.3. Case III – 7-DOF Movement I

For the third case, a target end-effector position has been chosen that will demonstrate the high DOF dynamic capability of the robotic arm. The result is a variety of reaction forces acting on the pedestal of the XY-table that will require a response from the XY-table's control systems. In the inertial frame of the combined XY-table and robotic arm the target end-effector position is $\vec{r} = \langle 0.6, 0.6, 0.6 \rangle$ m. Figure 6.5a and b shows the initial configuration and the final configuration once the end-effector reaches its target position.

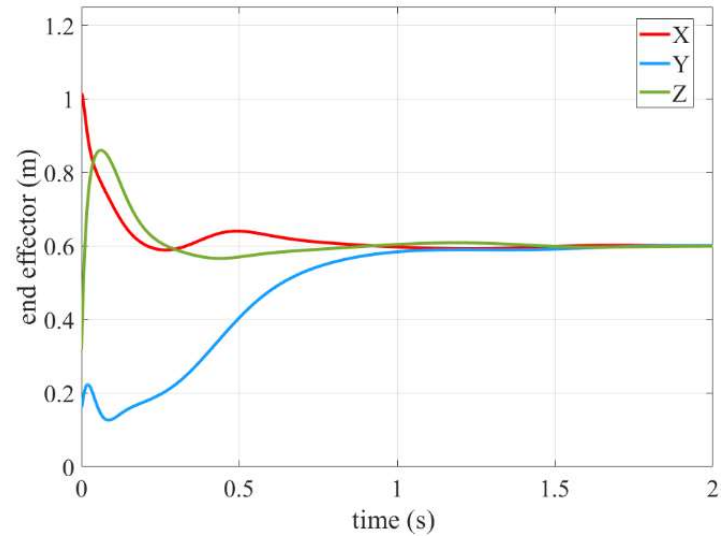


Figure 6.5a End-effector position throughout simulation of case III.

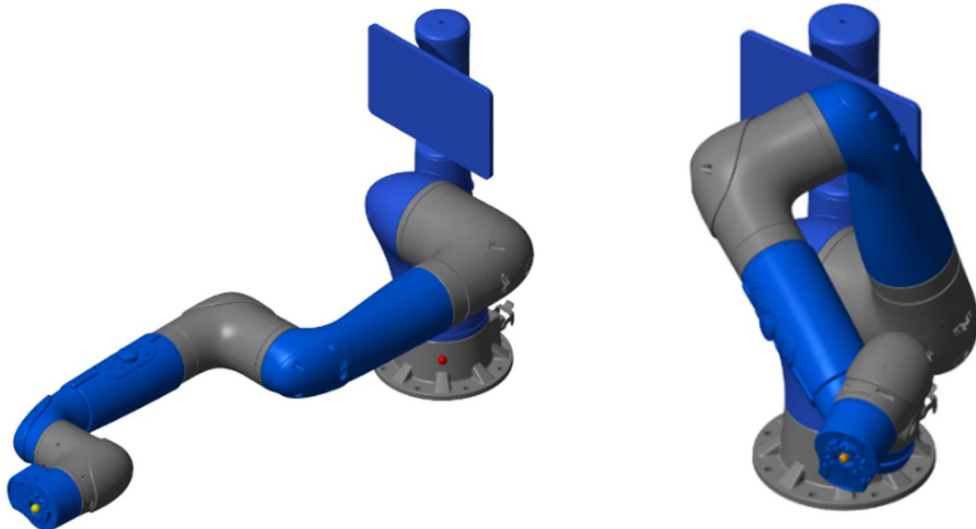


Figure 6.5b Initial and final positions of case I. XY-table removed for clarity. End-effector position (yellow). Target end-effector position (red).

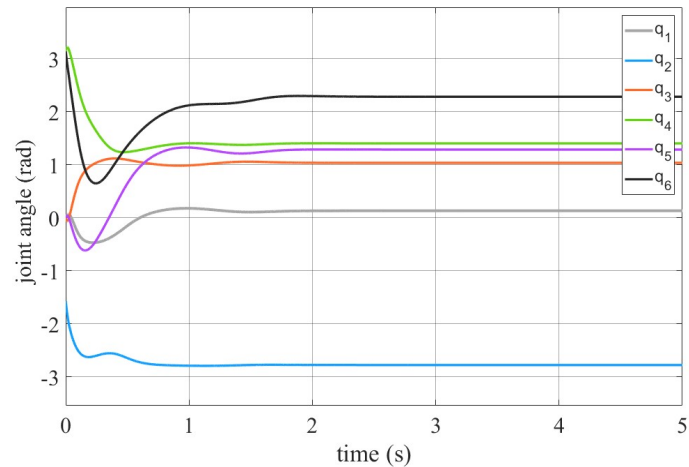


Figure 6.5c Joint angles from start position to fixed final positions when end-effector reaches target position for case I.

The reaction force applied to the pedestal of the XY-table causes a similar displacement error as observed in previous test cases, which is shown in Figure 6.6. At the time the joint angles have reached their final configuration, the acceleration of the pedestal becomes zero because there is no longer a reaction force from the robotic arm acting on the pedestal.

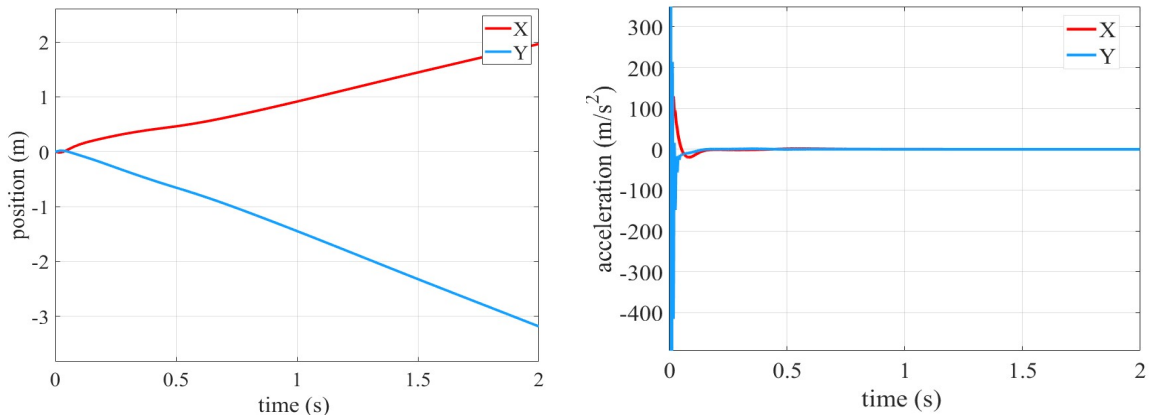


Figure 6.6 Planar displacement and acceleration of the XY-table pedestal center in case I.

6.1.4. Case IV – 7-DOF Movement II

For case IV, the movement displayed in Figure 6.7a and b will utilize all of its joints to arrive at a separate, unique end-effector position at $\vec{r} = \langle 0.8, 0.0, 0.6 \rangle$ m in the robotic arm's inertial frame.

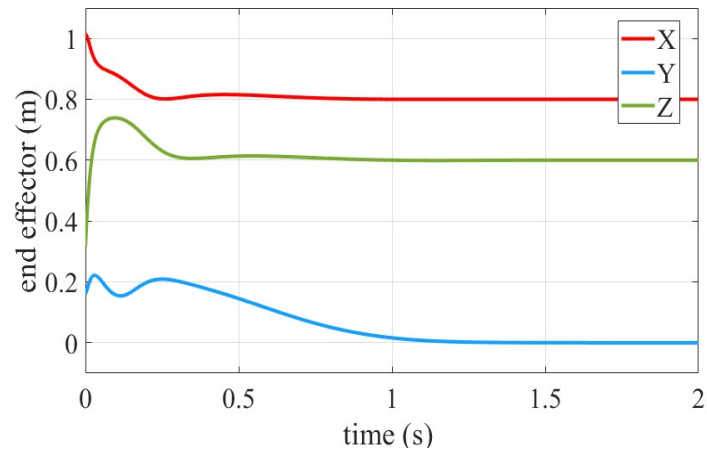


Figure 6.7a End-effector position throughout simulation of case IV.

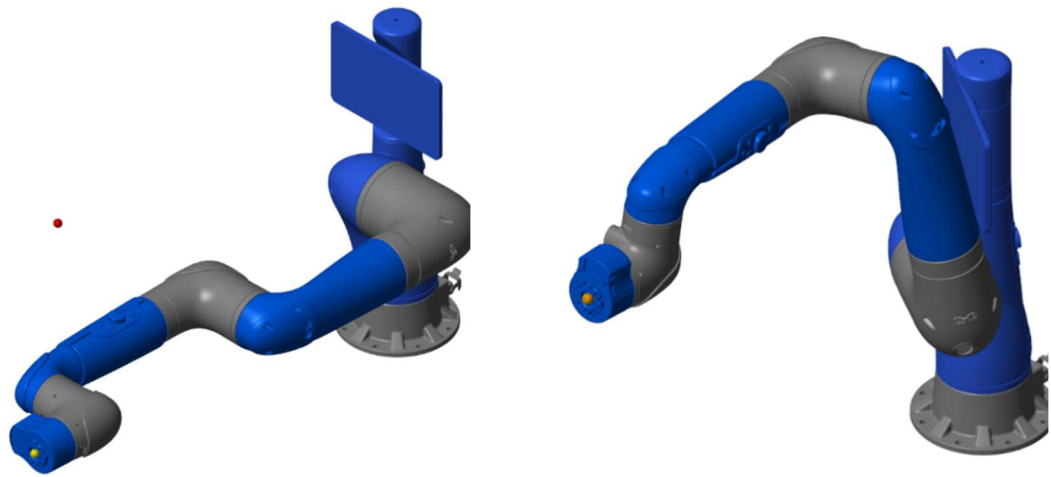


Figure 6.7b Initial and final positions of case IV. XY-table removed for clarity. End-effector position (yellow). Target end-effector position (red).

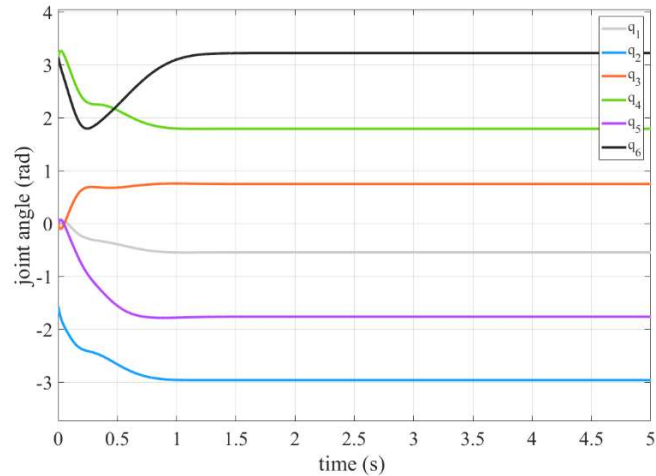


Figure 6.7c Joint angles from start position to fixed final positions when end-effector reaches target position for case IV.

The joint angles follow a similar pattern as in case I, achieving the completed movement in $t < 2$ seconds as shown in Figure 6.7c.

The robotic arm's dynamic movement provides a reduced reaction force to the pedestal, which causes the XY-table pedestal to accelerate at half the rate observed in case III. In Figure 6.8, the pedestal's displacement does not exceed 2 meters along the x or y axes when compared with the results of case III.

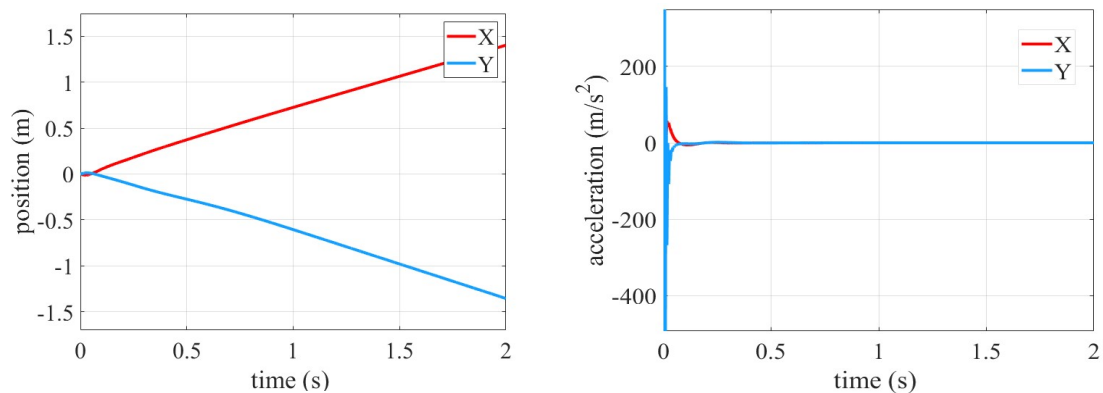


Figure 6.8 Planar displacement and acceleration of the XY-table pedestal center in case IV.

The largest movement of the robotic arm occurs within the first 0.2 seconds of the simulation. This is the period where most of the reaction force at the pedestal is applied and accounts for most of the displacement.

6.2. Open-Loop Control

With control applied to the system by the two control pulleys of the XY-table, the table will use the calculated forces exerted by the robotic arm as disturbances to the table's position in the system's inertial frame, with the difference being the control pulleys resisting the forces by applying a counter-force to the pedestal to maintain equilibrium. The more accurately the disturbances are calculated, the smaller the steady-state position error for the system will be.

6.2.1. Case I – Simple Movement I

Comparing the calculated position error to the simulated position error that is provided by the Simscape Multibody platform, the results shown in Figure 6.9 demonstrate similar plots for each result respectively.

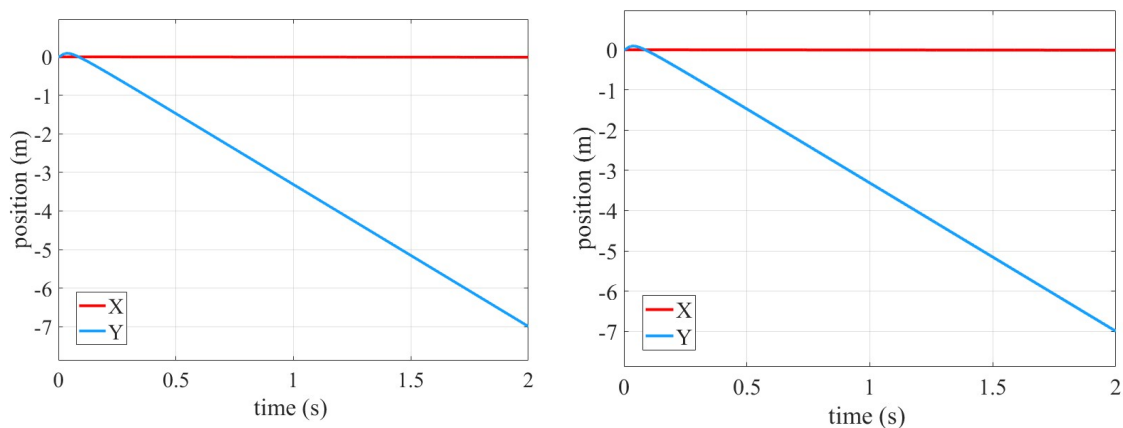


Figure 6.9 Calculated displacement (left) and measured displacement (right) of the XY-table pedestal center in case I.

When the open-loop control is applied based on the calculated disturbance the control pulleys follow a path opposite that of the applied disturbance which results in the pedestal attempting controlled motion in the opposite direction along each axis as shown in Figure 6.10.

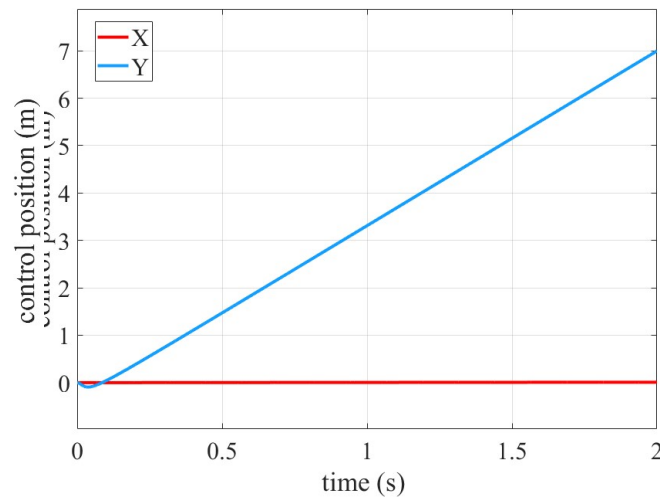


Figure 6.10 Open-loop control output converted to the XY-table pedestal position, case I.

When the control is applied, the reaction forces of the robotic arm are cancelled out and the final position maintains equilibrium in the system inertial frame throughout the simulation.

For comparison, the controller is then applied using the measured disturbance with the open-loop control. The two results are compared in Figure 6.11 to observe differences between the measured displacement of the modeled system and the error calculated within the controller.

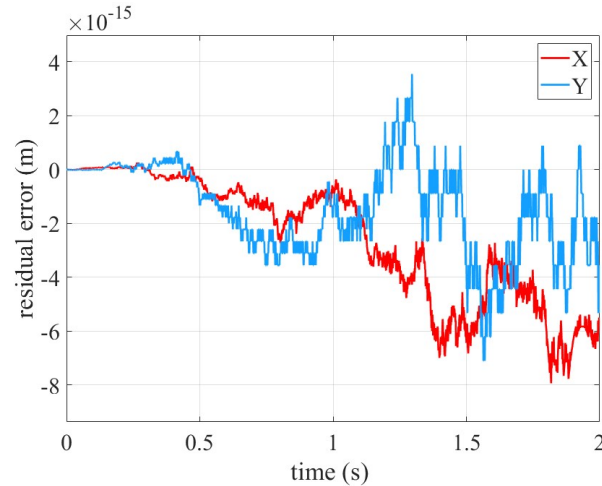


Figure 6.11 Comparison of the residual error between the open-loop control with a measured system disturbance and calculated disturbance for case I.

The differences in the resulting residual error for each form of disturbance is minimal, on the order of 10^{-15} m. This can be regarded as a negligible error in most applications.

6.2.2. Case II – Simple Movement II

Comparing the calculated position error to the simulated position error that is provided by the Simscape Multibody platform, the results shown in Figure 6.12 demonstrate similar plots for each result.

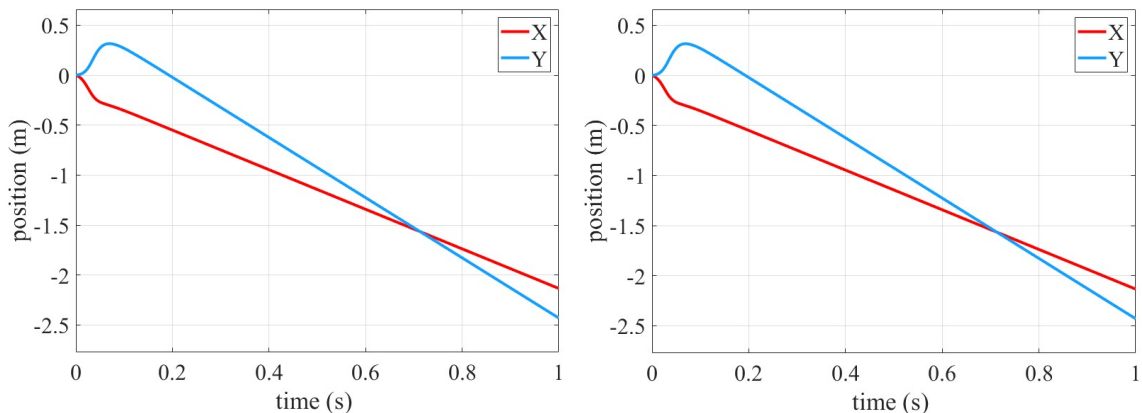


Figure 6.12 Calculated displacement (left) and measured displacement (right) of the XY-table pedestal center in case II.

When the open-loop control is applied based on the calculated disturbance the control pulleys follow a path opposite that of the applied disturbance, which results in the pedestal attempting controlled motion in the opposite direction along each axis as shown in Figure 6.13.

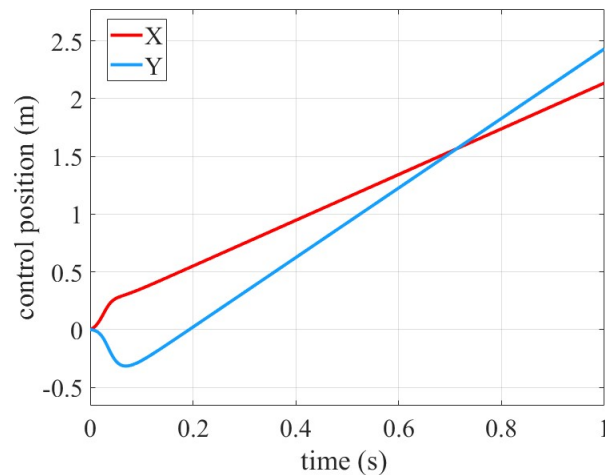


Figure 6.13 Open-loop control output converted to the XY-table pedestal position, case II.

When the control is applied the reaction forces of the robotic arm are cancelled out and the final position goes to zero.

For comparison, the controller is then compared using the measured disturbance with the open-loop control. The two results are compared in Figure 6.14 to observe differences between the measured displacement error of the simulated Simscape Multibody system and calculated system disturbances within the controller.

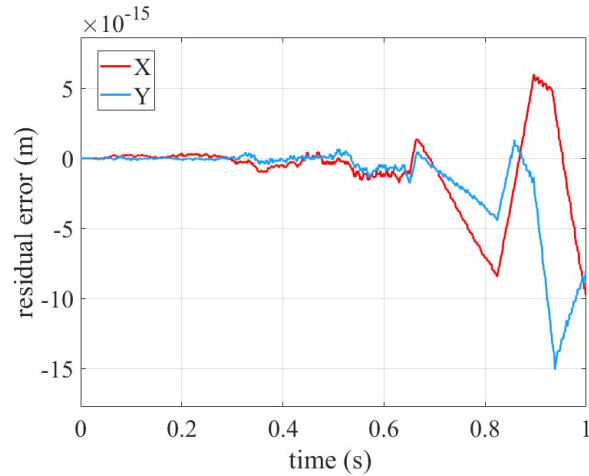


Figure 6.14 Comparison of the residual error between the open-loop control with a measured system disturbance and a calculated disturbance for case II.

As with case I, the differences in the resulting residual error for each form of disturbance is on the order of 10^{-15} m.

6.2.3. Case III – 7-DOF Movement I

Comparing the calculated position error to the simulated position error that is provided by the Simscape Multibody platform, the results shown in Figure 6.15 demonstrate similar plots for each result.

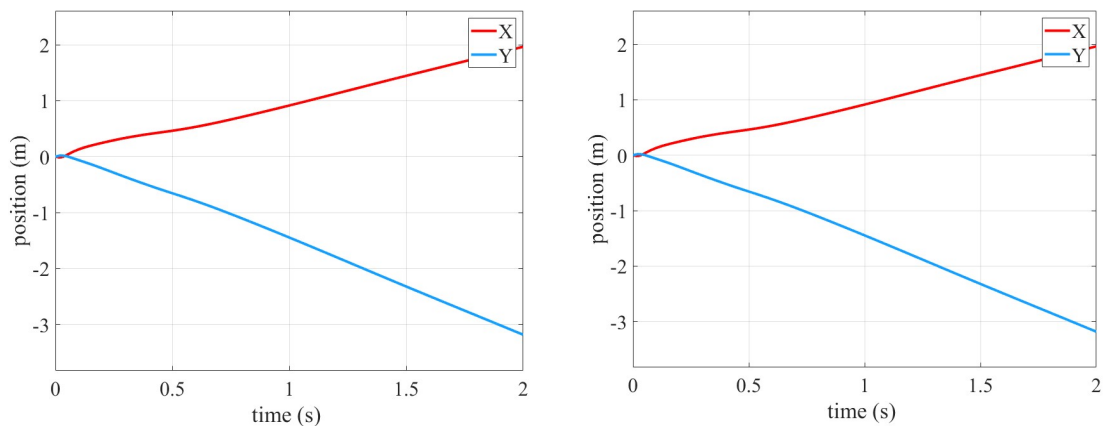


Figure 6.15 Calculated displacement (left) and measured displacement (right) of the XY-table pedestal center in case III.

When the open-loop control is applied based on the calculated disturbance the control pulleys follow a path opposite that of the applied disturbance which results in the pedestal attempting controlled motion in the opposite direction along each axis, Figure 6.16.

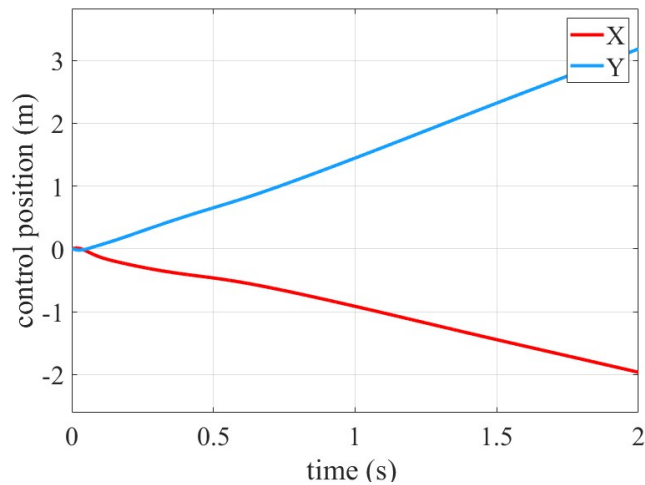


Figure 6.16 Open-loop control output converted to the XY-table pedestal position, case III.

When the control is applied the reaction forces of the robotic arm are cancelled out and the final position becomes zero throughout the simulation.

Comparing the results of the calculated displacement error to the measured displacement error with the open-loop control in Figure 6.17 a similar result as with previous test cases with a noticeable error propagation occurring the longer the displacement continues at constant velocity.

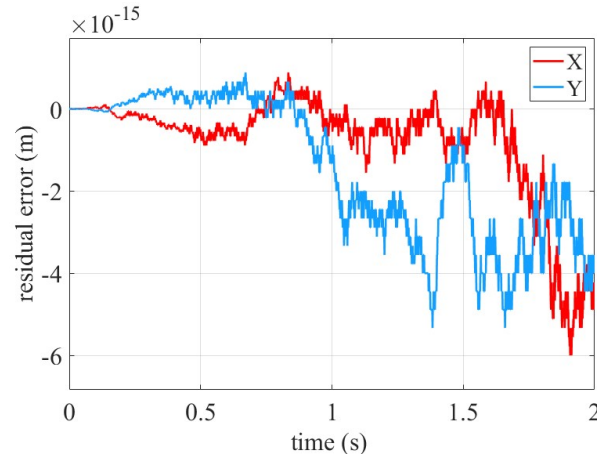


Figure 6.17 Comparison of the residual error between the open-loop control with a measured system disturbance and a calculated disturbance for case III.

The differences in the resulting residual error for each form of disturbance is minimal, on the order of 10^{-15} m. This can be regarded as a negligible error in most applications.

6.2.4. Case IV – 7-DOF Movement II

For case IV, the same process for analyzing the performance of the open-loop control was applied, this time with disturbances from the movements of the case IV robotic arm. Comparing the calculated position error to the simulated position error that is provided by the simulation within the Simscape Multibody platform, the results shown in Figure 6.18.

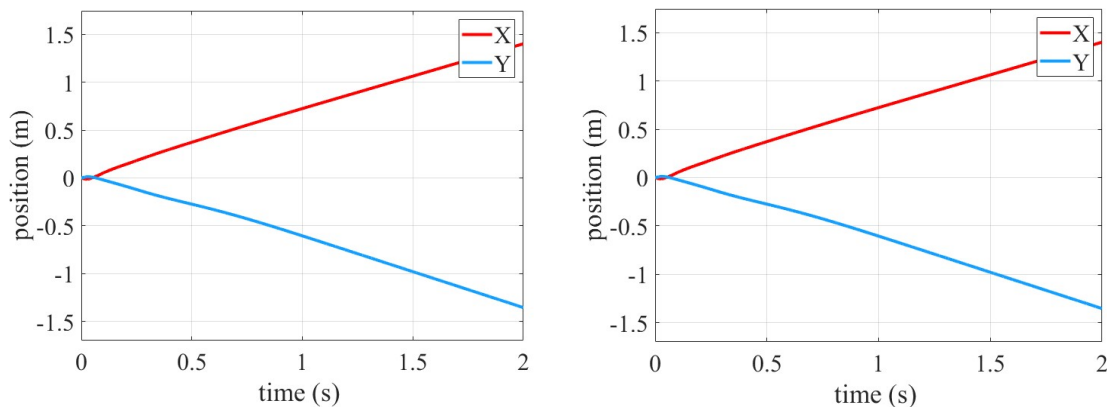


Figure 6.18 Calculated displacement (left) and measured displacement (right) of the XY-table pedestal center in case IV.

The open-loop control is again applied based on the calculated disturbance which results in the pedestal attempting controlled motion in the opposite direction along each axis, Figure 6.19.

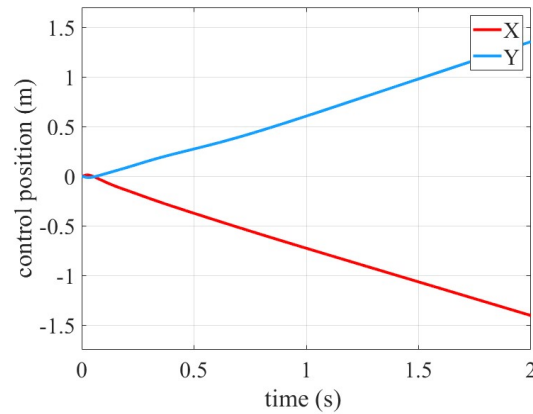


Figure 6.19 Open-loop control output converted to the XY-table pedestal position, case IV.

The results are compared with the measured disturbance in Figure 6.20 to observe the differences between the measured Simscape Multibody system and calculated system disturbances and how these results differ for case IV.

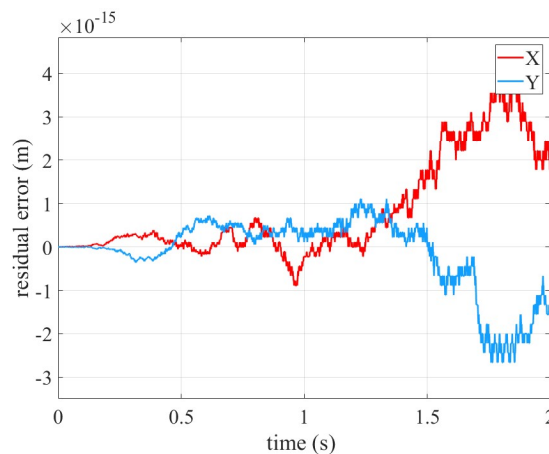


Figure 6.20 Comparison of the residual error between the open-loop control with a measured system disturbance and a calculated disturbance for case IV.

The differences in the resulting residual error for each form of disturbance is minimal; however, case IV shows the deviations between the measured and calculated residual error increasing as the system continues to displace. The error still remains on the order of 10^{-15} m. Based on the continued observation of propagated error; continuous movement could potentially cause an accumulation of error that could be significant over time.

6.3. Closed-Loop Control

Based on the performance of the open-loop controlled system, the closed-loop control simulation results would be redundant because a position feedback could not reduce the steady-state error any lower than the machine precision observed in all cases of section 6.2. For the closed-loop control cases, an external force that represents an external disturbance not accounted for in the model is applied to the robotic arm's base causing a residual position error from the unintended displacement to be introduced through the feedback loop. The external force is comprised of two pulses along the x and y axes of the inertial frame shown in Figure 6.21. These pulses are applied at 150 N and 50 N for 0.05 seconds at 0.05 seconds into the simulation.

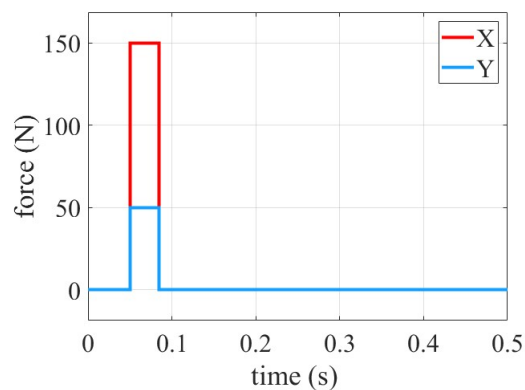


Figure 6.21 Externally applied forces measured at the pedestal.

The external forces shown in Figure 6.21 are arbitrary selections. The force magnitude is large enough to cause observable error for the closed-loop feedback. The forces are applied along two axes to provide residual position error along both x and y axes. The result of an unexpected external force is an increased acceleration of the pedestal and robotic arm which leads to a residual position error that is measured by a simulated position sensor at the pedestal center in the inertial frame. The sensor input will be utilized to provide the necessary feedback to stabilize the pedestal at the origin.

6.3.1. Case I – Simple Movement I with External Force

For case I, Figure 6.22 shows the comparison between the calculated position error being applied as the projected disturbance for the controller and the measured disturbance with the unanticipated external forces applied. The displacement at two seconds of simulation time shows a significant difference between the two with the position error reaching four meters difference along the y-axis and a half meter difference for the x-axis.

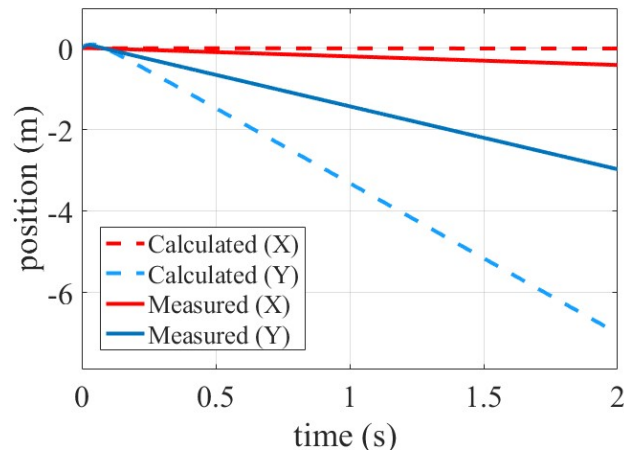


Figure 6.22 Residual position error measured by the feedback of the closed-loop control system for case I.

The position error sensor input is added through the closed-loop feedback, allowing the controller to recognize the residual error not accounted for by the calculated disturbance block of the controller. Without the sensor feedback, the controller would fail to recognize the error and over or under compensates for the reaction forces at the pedestal. Figure 6.23 displays the controller response when the sensor feedback is added to the controller. When the control pulleys are applied, the reaction forces of the robotic arm are cancelled out and the final position becomes zero throughout the simulation.

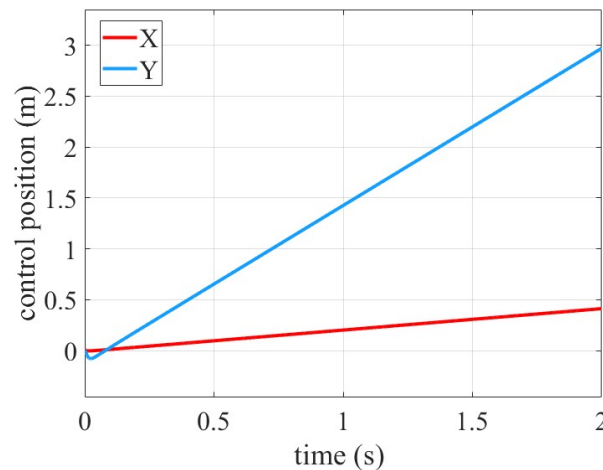


Figure 6.23 Closed-loop control output converted to the XY-table pedestal position, case I.

The differences in the resulting residual error for each form of disturbance is minimal. Figure 6.24 shows the deviations between the measured and calculated residual error increasing as the position error increases. The error still remains on the order of 10^{-15} m, but the closed-loop control is still limited by machine precision. This can be regarded as a negligible error in most applications.

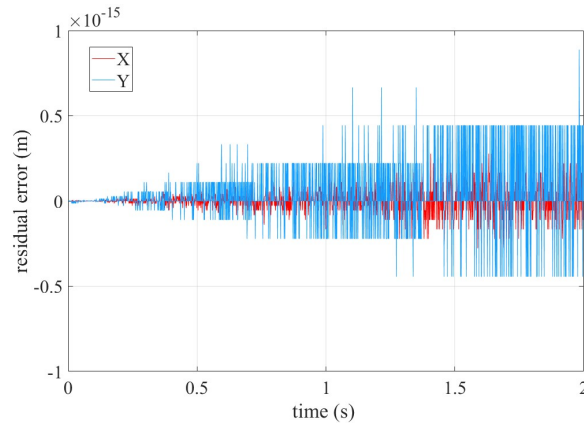


Figure 6.24 Steady-state error of the pedestal position for closed-loop control case I.

6.3.2 Case II – Simple Movement II with External Force

For case II, Figure 6.25 shows the comparison between the calculated position error being applied as the projected disturbance for the controller and the measured disturbance with the unanticipated external forces applied. The position error is significantly impacted along the y-axis, but only a minor change is noticed for the x-axis. The similarity is coincidental and not due to differences in controller performance between case I and II. Identical closed-loop control systems are used throughout testing to maintain consistency in results for all cases.

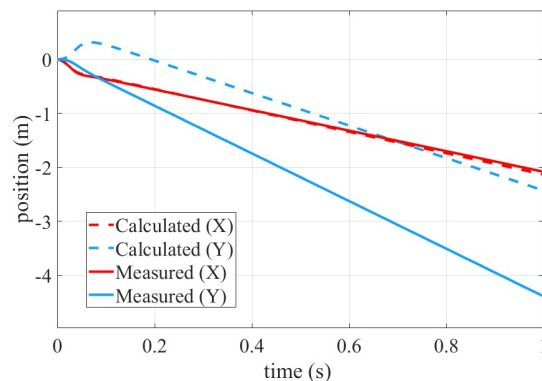


Figure 6.25 Residual position error measured by the feedback of the closed-loop control system for case II.

The position error sensor input added through the closed-loop feedback allows the controller to recognize the residual error not accounted for by the calculated disturbance block of the controller. As a result, Figure 6.26 displays the controller response with displacement error sensor feedback provided. When the XY-table control pulleys are applied the reaction forces of the robotic arm are cancelled out and the XY-table maintains equilibrium throughout the simulation.

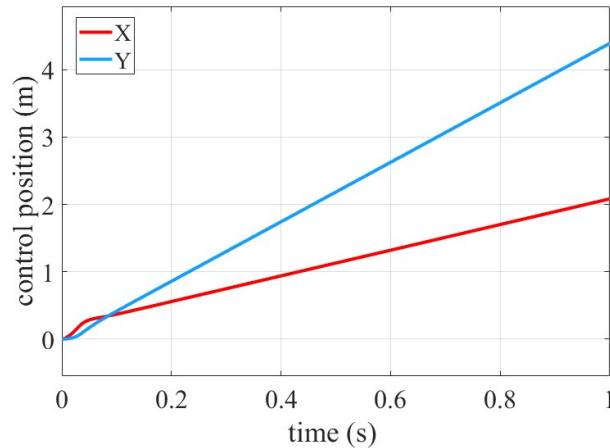


Figure 6.26 Closed-loop control output converted to the XY-table pedestal position, case II.

The differences in the resulting residual error for each form of disturbance is minimal. Figure 6.27 shows the deviations between the measured and calculated residual error increasing as the position error increases. The error still remains on the order of 10^{-15} m, but is reduced compared to the open-loop residual error plot in Figure 6.14. It is also noticeable that the error propagation is symmetric about the equilibrium and has been reduced during continued displacement.

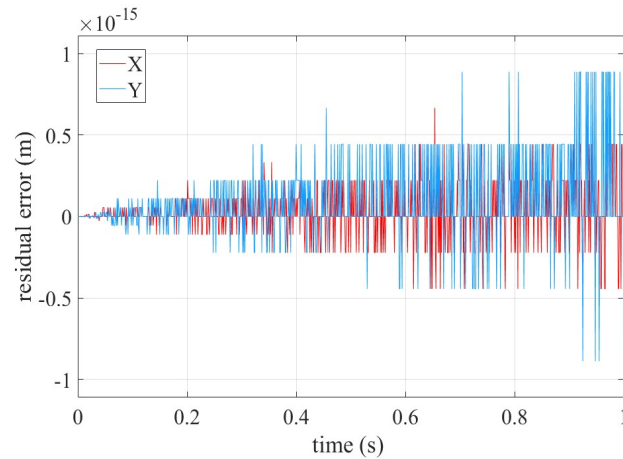


Figure 6.27 Steady-state error of the pedestal position for closed-loop control case II.

6.3.3. Case III – 7-DOF Movement I with External Force

For case III, Figure 5.28 shows the comparison between the calculated position error being applied as the projected disturbance for the controller and the measured disturbance with the unanticipated external forces applied. The displacement at two seconds of simulation time shows a significant different.

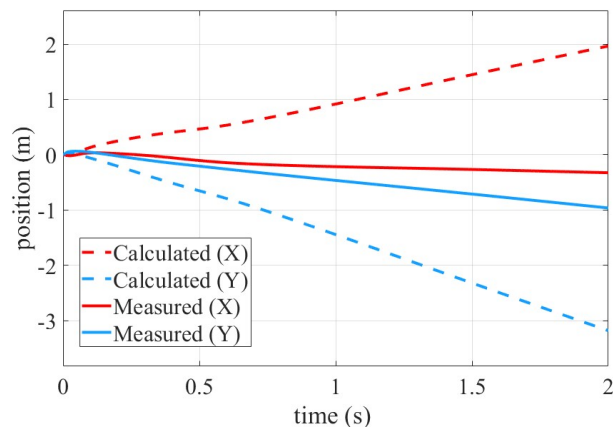


Figure 6.28 Residual position error measured by the feedback of the closed-loop control system for case III.

Figure 5.29 displays the controller response when the residual error in the feedback is provided. When the control pulleys are applied the reaction forces of the robotic arm are cancelled out and the final position is maintained at equilibrium throughout the simulation.

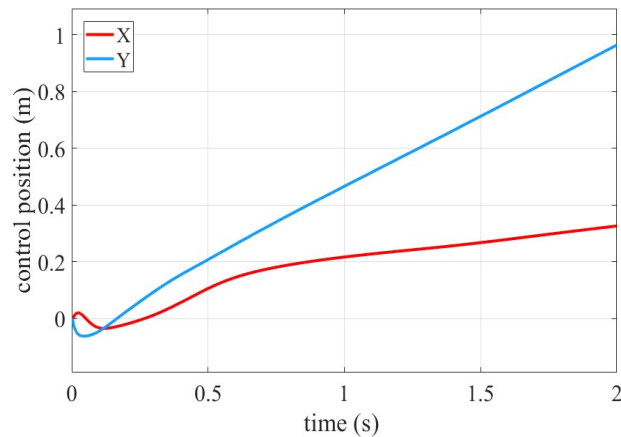


Figure 6.29 Closed-loop control output converted to the XY-table pedestal position, case III.

The differences in the resulting residual position error for each form of disturbance is minimal. Figure 5.30 shows the deviations between the measured and calculated residual error increasing as the position error increases. The error still remains on the order of 10^{-15} m, but the closed-loop control is still limited by machine precision.

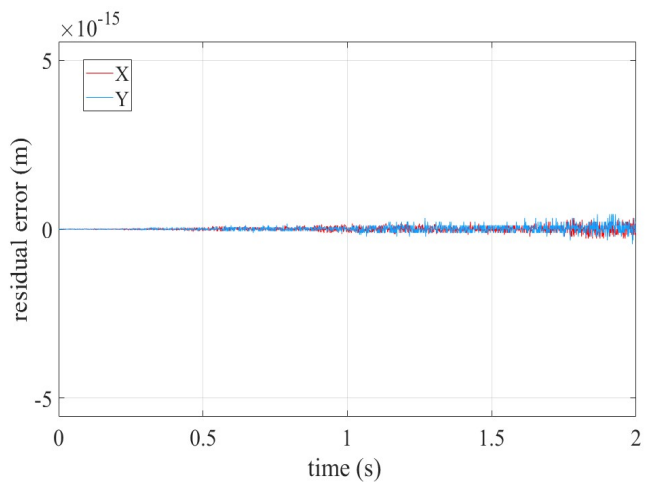


Figure 6.30 Steady-state error of the pedestal position for closed-loop control case III.

6.3.4. Case IV – 7-DOF Movement II with External Force

For case IV, Figure 6.31 shows the comparison between the calculated position error being applied as the projected disturbance for the controller and the measured disturbance with the unanticipated external forces applied.

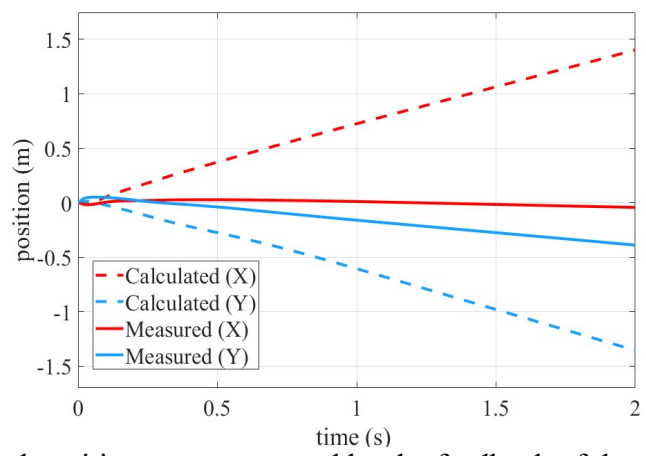


Figure 6.31 Residual position error measured by the feedback of the closed-loop control system for case IV.

Figure 6.32 displays the controller response when the residual error in the feedback is provided. When the control pulleys are applied the reaction forces of the robotic arm are cancelled out and the final position becomes zero throughout the simulation.

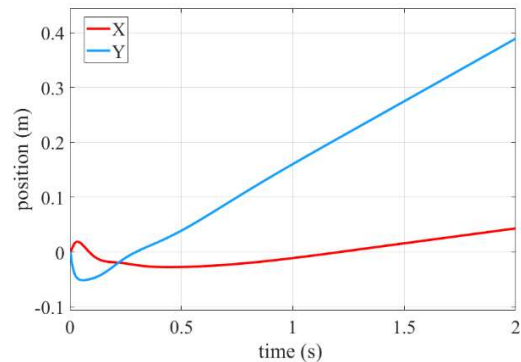


Figure 6.32 Closed-loop control output converted to the XY-table pedestal position, case IV.

The differences in the resulting residual error for each form of disturbance is minimal. Figure 6.33 shows the deviations between the measured and calculated residual error increasing as the position error increases. The error still remains on the order of 10^{-15} m, despite being small is still reduced compared to its open-loop control results in Figure 6.20.

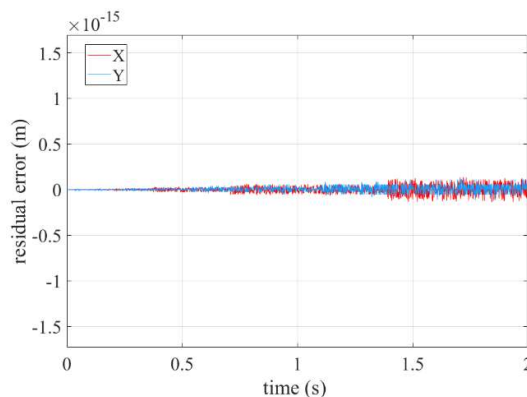


Figure 6.33 Steady-state error of the pedestal position for closed-loop control case IV.

7. Conclusions and Recommendations

The reaction forces generated by a robotic arm's movement to a variety of end-effector positions on a simulated zero-friction surface can be calculated and controlled utilizing a simple open-loop control system. For a variety of end-effector positions and joint movements the stability of its mount can be achieved with an accurate determination of the robotic arm's center of mass kinematics and a form of control force exerted at the base to arrest any displacement error that occurs.

For a non-ideal case involving dampening or viscous forces, the open-loop control is only as capable as the user's ability to define these forces and relate them to the reaction force at the robot arm base. Examples include joint friction, torque limits to control and robotic arm stepper motors, step precision, slippage of the pulley system, signal noise to the controller from sensors and from the controller to motors and propagated error during continuous operation. Engaging with a physical Sawyer robotic arm will require the controller to be developed with these sources of error accounted for in order to reduce the steady-state position error. Continued research can be applied by taking the results of the ideal case outlined in this thesis and incorporating a series of comprehensive viscous forces, which can then be applied to the actual Sawyer robotic arm open-loop stability controller.

Finally, one item that cannot be accounted for is the potential of failure within the system. In open-loop control, the lack of feedback from sensor systems reduces the open-loop controller performance. Failure scenarios include sudden failure of a joint motor, sudden increase in friction within the joints due to lubrication failure or an external force applied to the system during operation as shown in closed-loop cases I, II, III and IV. For

these scenarios, an open-loop control will not be enough and sensor feedback must be applied to mitigate risks to system performance. For these reasons adopting a closed-loop control design with a sensor feedback to determine the residual displacement error in the physical system becomes a necessity for the successful system performance in any non-idealized system applications.

REFERENCES

- AbdelHamid, A. Y., Abdeldayem, M., & Mabrouk, M. H. (2018, April). Low Cost XY Core Positioning System Using Stepper Motor. In *The International Conference on Applied Mechanics and Mechanical Engineering* (Vol. 18, No. 18th International Conference on Applied Mechanics and Mechanical Engineering., p. 65). Military Technical College.
- Ali, M. A., Park, H. A., & Lee, C. G. (2010, October). Closed-form inverse kinematic joint solution for humanoid robots. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (p. 704). IEEE.
- Briot, S., & Khalil, W. (2015). Homogeneous transformation matrix. In *Dynamics of Parallel Robots* (p. 20). Springer, Cham. Brueckner, Andre (2018, July 2). SAWYER 3D MODELS 4. Message posted to <https://rethinkrobotics.interaforum.com/files/file/23-sawyer-3d-models/>.
- Buss, S. R. (2004). Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19), 16.
- Chen, Q., Zhu, S., & Zhang, X. (2015). Improved inverse kinematics algorithm using screw theory for a six-DOF robot manipulator. *International Journal of Advanced Robotic Systems*, 12(10), pp. 2-6.
- Davidson, J. K., Hunt, K. H., & Pennock, G. R. (2004). Robots and screw theory: applications of kinematics and statics to robotics. *J. Mech. Des.*, 126(4), pp. 763-764.
- Hartenberg, R. S., & Denavit, J. (1955). A kinematic notation for lower pair mechanisms based on matrices, n.d., pp. 215-221.
- Hayat, A. A., Chittawadigi, R. G., Udai, A. D., & Saha, S. K. (2013, July). Identification of Denavit-Hartenberg parameters of an industrial robot. In *Proceedings of Conference on Advances In Robotics* (pp. 1-6).
- Junkins, J. L., & Schaub, H. (2009). *Analytical mechanics of space systems*. American Institute of Aeronautics and Astronautics, p. 28.
- Layeghi, Daniel. "4. Dynamic and Kinematic Modelling of the Sawyer Arm - Daniellayeghi." *Google Sites*, 20 Nov. 2017, sites.google.com/site/daniellayeghi/daily-work-and-writing/major-project-4.

- MathWorks. (2019, September). *Cable-Driven XY Table with Cross Base* (r2019b). Retrieved January 10, 2019 from <https://www.mathworks.com/help/phymod/sm/examples/cable-driven-XY-table-with-cross-base.html>
- MathWorks. (2019, September). MATLAB 2017b. [Computer Software]. Natick, Massachusetts. Retrieved from <https://www.mathworks.com/products/matlab.html>
- Rocha, C. R., Tonetto, C. P., & Dias, A. (2011). A comparison between the Denavit–Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators. *Robotics and Computer-Integrated Manufacturing*, 27(4), 723-728.
- Siciliano, B. (1990). Kinematic control of redundant robot manipulators: A tutorial. *Journal of intelligent and robotic systems*, 3(3), 202-204.
- Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: modelling, planning and control*. Springer Science & Business Media, p. 133.
- Spear, Don (2017, October 3). SAWYER URDF 1.0.0. Message posted to <https://rethinkrobotics.interaforum.com/files/file/31-sawyer-urdf/>
- Stramigioli, S., & Bruyninckx, H. (2001). Geometry and screw theory for robotics. *Tutorial during ICRA, 2001*, p. 39.
- Tsai, L. W. (1999). *Robot analysis: the mechanics of serial and parallel manipulators*. John Wiley & Sons, p. 42.
- “TTU Advanced Robotics.” (2010, April 12). Retrieved January 2019, from <https://www.alexia.com/siteinfo/ttuadvancedrobotics.wikidot.com>