# Analysis of a 5-DOF Robot Manipulator for Agriculture Applications

A Thesis Submitted
to the College of Graduate and Postdoctoral Studies
in Partially Fulfillment of the Requirements
for the Degree of Master of Science
in the Department of Mechanical Engineering
University of Saskatchewan

By

**Joshua Coté**

Saskatoon, Saskatchewan, Canada

# Permission to use

In presenting this thesis/dissertation in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis/dissertation in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis/dissertation work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis/dissertation or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis/dissertation.

Requests for permission to copy or to make other uses of materials in this thesis/dissertation in whole or part should be addressed to:

Head of the Department of Mechanical Engineering
University of Saskatchewan, College of Engineering
3B48 Engineering Building, 57 Campus Drive
Saskatoon, Saskatchewan, S7N 5A9, Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan  S7N 5C9 Canada

# Abstract

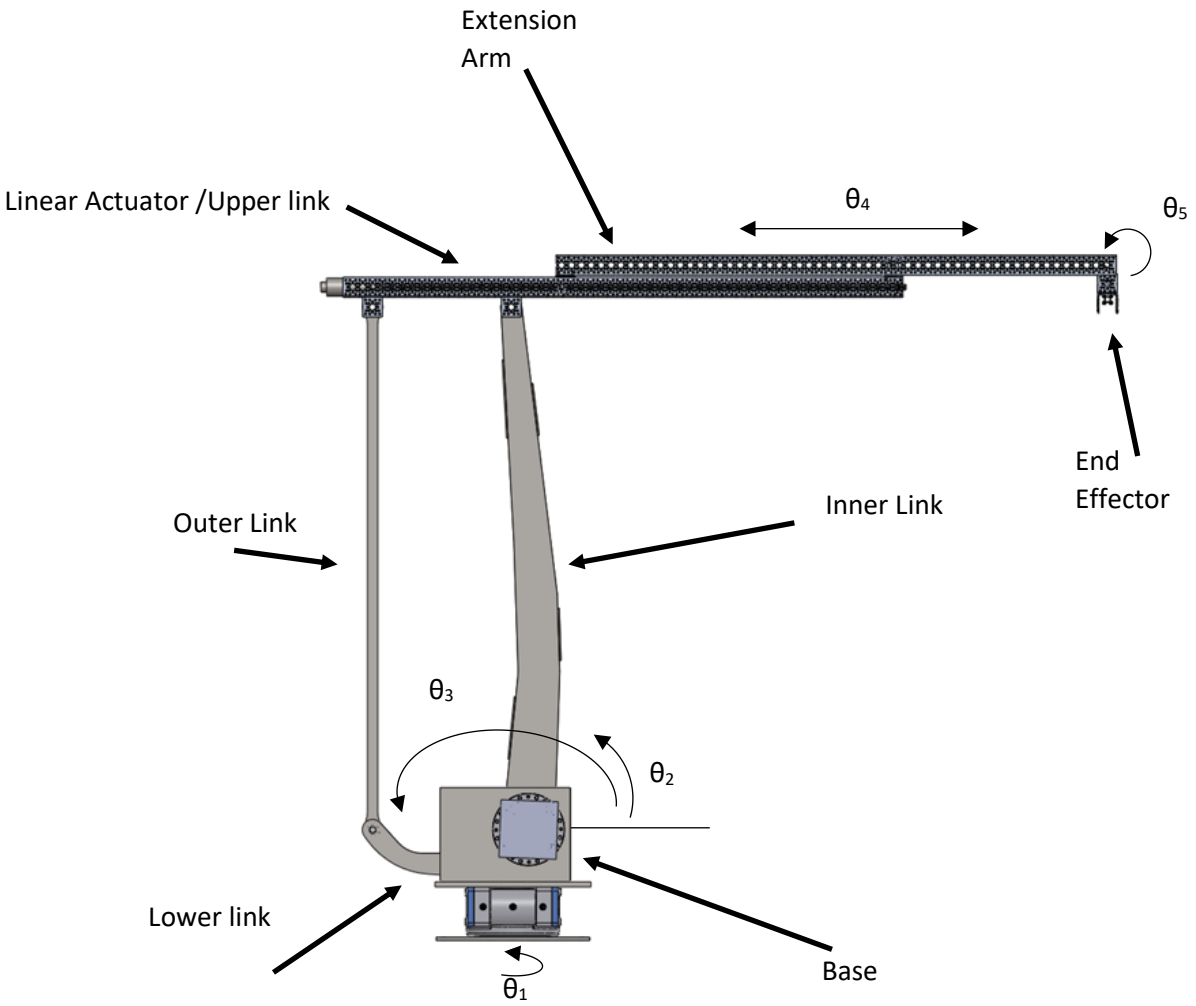



*Figure: Abstract -1*

A 5 degree of freedom (shown by $\theta_{1-5}$) robot was analyzed in this thesis for a better understanding of forward kinematic, inverse kinematic, and forward dynamic analyses. Using the forward and inverse kinematics, position of end effector (tip) of the robot can be predicted relative to base of the robot at any time, including while the robot is in motion. The analytical forward kinematic equations were derived using the Denavit-Hartenburg (DH) parameters by creating coordinate transformation matrices from the base of the robot to the tip. The inverse kinematic equations were then derived from the forward kinematic equations. Both sets of kinematic equations were then compared to a simulated model of the robot arm which was created in ANSYS rigid dynamics module.

Forward dynamic equations of the robot are used to determine forces required at each joints to produce motion. original method for determining the forward dynamics is a set of equations where the Newton-Euler equations for motion are applied to individual robot links for a serial robot (Craig 2005). As the robot arm concerned in this thesis is not a serial robot (each link attaches to only 1 following link), additional equations were created by analyzing free body diagram of outer link and upper link of the robot arm. After including additional equations, the analytical dynamic equations of the robot were complete. The same simulated robot arm as above was used to compare analytical results with simulated results.

The last section of this thesis presents a gradient optimization technique. Changing weight of the robot arm to a minimum will reduce torque requirements for joint motors, minimize power consumption, and reduce risk of soil compaction. Using a gradient search optimization, shape of the inner link is optimized subject to maximum deflection and maximum stress constraints. Once the optimization was completed, overall mass of the robot arm was reduced by about 49%. When the old inner link was replaced by the new optimized link the torque required of the inner link motor was reduced by about 11% when following a given trajectory.

# Acknowledgements

This thesis would not be possible without the help of various individuals.

I would like to thank my supervisor Dr. Reza Fotouhi for his time and guidance. I would also like to thank my committee members Dr. Walerian Szyszkowski and Dr. Travis Wiens.

I would also like to extend my gratitude to Robert Peace and Douglas Bitner for advising me on data collection and measurement methods.

I appreciate the help of the other members of the Robotics Lab: Majid Khak-Pour, Hedieh Badkoobehhezavah, Amir Khozeimeh for all of their support. Additionally I would like to thank the summer students that I worked with, Keegan Wegner, Danny Wu, Colby Neald, and Oluwatoyin Famure.

A special thanks goes to Tyler Zhang, who has given me an extrodinary amount of guidance and patience.

Dr. Allan Dolovich deserves a special mention for his insight, time, and mentorship. His guidance and friendship have been invaluable.

My final thanks goes to my family for all of the support and encouragement that they provided while I was completeing my studies. I could not have completed this endeavor without the love and support that I received.

The 3rd generation boom is a project that has been contributed to by many people. Credit is due to the following:

Original design: Tyler Zhang

Sensors and Electronics: Majid Khakpour

Vibration analysis: Hedieh Badkoobehhezavah

Analysis of materials: Amir Khozeimeh

Supervision: Dr. Reza Fotouhi

# Table of Contents

# List of Figures

# List of Tables

# Nomenclature

Listed below are the abbreviations used most frequently in this thesis

| | |
|---|---|
| 3-D | 3 Dimensional |
| CAD | Computer Aided Design |
| D-H | Denavit-Hartenburg |
| DOF | Degrees of Freedom |
| FAO | Food and Agriculture Organization |
| FEA | Finite Element Analysis |
| FEM | Finite Element Method |
| GPS | Global Positioning System |
| LHS | Left Hand Side (of an equation) |
| LiDAR | Light Detection And Ranging |
| RHS | Right Hand Side (of an equation) |
| S.F. | Safety Factor |

# List of symbols used in this thesis

| | |
|---|---|
| $a_i$ | A DH parameter. Measures the distance between links.* |
| $c_0$ | A constant used for cubic polynomial trajectory generation. |
| $c_1$ | A constant used for cubic polynomial trajectory generation. |
| $c_2$ | A constant used for cubic polynomial trajectory generation. |
| $c_3$ | A constant used for cubic polynomial trajectory generation. |
| $\mathrm{d}_i$ | A DH parameter. Measures the perpendicular offset between adjacent robot links.* |
| $\dot{d}_i$ | Linear velocity of the offset between adjacent links. |
| $\ddot{d}_i$ | Linear acceleration of the offset between adjacent links. |
| E | Modulus of elasticity |
| ${}^{a}\bar{F}_b$ | A force vector describing the forces acting in frame b relative to frame a. Used to specify forces acting on the center of mass of a robotic link to produce acceleration. |
| ${}^{a}\bar{f}_b$ | A force vector describing the forces acting in frame b relative to frame a. Used to specify forces exerted on a robot link by a neighboring link. |
| H | Height |
| $I_{bending}$ | area moment of inertia |
| ${}^{c\,i}I_i$ | Matrix of mass moment of inertia of the robot link i, measured relative to the orientation of frame i, with the origin located at the center of mass of the link. |
| $\mathrm{L}_i$ | Length of robot link i. |
| $\mathrm{L_r}$ | Distance to end of the beam |
| M | Internal bending moment |
| $m_i$ | Mass of the robot link that frame i is attached to. |
| ${}^{a}\bar{N}_b$ | A moment vector describing the moments acting in frame b relative to frame a. Used to specify moments acting on the center of mass of a robotic link to produce acceleration. |
| ${}^{a}\bar{n}_b$ | A moment vector describing the moments acting in frame b relative to frame a. Used to specify moments exerted on a robot link by a neighboring link. |
| ${}^{a}P_b$ | Position vector describing the location of frame b relative to frame a. |
| ${}^{a}P_{c_b}$ | Position vector describing the location of the center of mass of the robot link attached to frame b relative to frame a. |
| ${}^{a}_{b}R$ | A rotational matrix describing the orientation of frame b relative to frame a. |
| S.F. | Safety Factor |
| $t_{final}$ | The final duration of a path traveled. Used in cubic polynomial trajectory generation. |
| V | Deflection |
| ${}^{a}v_b$ | Linear velocity of frame b with respect to frame a. |
| ${}^{a}\dot{v}_b$ | Linear acceleration of frame b with respect to frame a. |
| ${}^{a}\dot{v}_{c_b}$ | Linear acceleration of the center of mass of the link attached to frame b with respect to frame a. |
| W | thickness |

| | |
|---|---|
| $x_i, y_i, z_i$ | Unit vector of frame i. |
| $\alpha_i$ | A DH parameter. Measures the twist of adjacent robotic links* |
| $\gamma_{i+1/i}$ or $\gamma_i$ | Angle between slices i+1 and i |
| $\delta$ | Displacement |
| $\theta_i$ | A DH parameter. Represents angle between adjacent links. |
| $\dot{\theta}_i$ | Rotational velocity of the angle between adjacent links. |
| $\ddot{\theta}_i$ | Rotational acceleration of the angle between adjacent links |
| $\rho$ | Density |
| $\sigma$ | Stress |
| $\tau_i$ | The torque produced at joint i. |
| ${}^a\omega_b$ | Rotational velocity of frame b with respect to frame a. |
| ${}^a\dot{\omega}_b$ | Rotational acceleration of frame b with respect to frame a. |

* see Appendix 1 for more information on DH parameters.

# Chapter 1: Introduction

## 1.1: Background and Motivation

According to the *State of Food Security and Nutrition in the World* report published by the Food and Agriculture Organizations of the United Nations, the number of people who face food insecurity is increasing and has been for the past 5 years; from 2014 to 2019 the number of people who have faced food insecurity has increased by 327.5 million. Severe food insecurity, defined as people who have gone without food for 1 or more days was experienced by 717.5 million people in 2019. Additionally, moderate food insecurity, people who have uncertainties about their ability to obtain food or must compromise on quantity and/or quality of food they consume has affected over 1280 million people in 2019. Combined, over 25% of the population of the world have faced food insecurity in 2019 and the number is expected to increase in the following years (FAO 2020). Methods to improve food availability include ensuring crop resilience to climate shocks such as flood or drought, improved disease and pest resistance and increased yield.

A nuance that complicates efforts to ensure that the world population receives the adequate food it needs is malnutrition and undernutrition. Communities may face barriers even when food is available, but the food does not satisfy the nutritional needs of the members. Healthy food that is available may be unaffordable, or the only food available is highly processed and does not contain significant nutrients. Malnutrition causes 1 in 3 children under 5 years of age across the world to suffer from stunting, wasting or being overweight (UNICEF 2019). In adults a poor diet is the highest risk factor for non-communicable diseases. To complicate matters, healthy diets are not always environmentally sustainable, and diets focused on sustainability are not always nutritious or healthy. Methods to improve nutritional quality in foods align closely with methods to increase food availability. Nutritional quality in food may be increased by ensuring nutritious food may be grown in wider ranges of environment (increased climate tolerance), improving pest resistance of the crops, and increasing nutrient values of native crops. One important tool in the fight against world hunger is the science of crop breeding.

Genetic improvements in common cereal crops such as wheat have contributed to increased global production.

One of the largest challenges that crop breeding faces is the difficulty of data acquisition (Furbank and Tester 2011). When choosing genetic lines to continue improving crop scientists need to measure the phenotypes of the crop. Whereas genotype refers to an organism's genetic makeup; the exact genes inherited from the parents, phenotypes are the physical manifestations of the plants which are determined by the interaction between the genetic code of the plants and the environment that the plants grow in (Schurr and Pieruschka 2019). Some examples of phenotypes are height, temperature, colour and yield. When developing new crop varieties plant scientists must measure traits of the progeny and a reference crop to track developments and changes of new varieties.

Traditional methods of measuring plant phenotypes require researchers to manually measure plant traits. Height measurement is taken using a tape measure or a meterstick (Mano 2017). The measuring instrument is placed into the plot and the average height of the plot is estimated by the researcher. Biomass estimates can require destructive sampling where the plants are uprooted and weighed (Munns, et al. 2010). The temperature of the plant canopy is correlated to the heat stress and water deficit that a plant undergoes (Yu, et al. 2015) and can be measured using an infrared thermometer, but depending on the location of the sample taken can produce inconsistent results even within the same plot. The ability to measure the temperature of large areas over short intervals is labor intensive. Colour and crop stage estimates rely on the researchers own senses and vary between individuals causing calibration and consistency to be near impossible to verify (Sadeghi-Tehran, et al. 2017). These methods are labor intensive, time consuming and unsuitable for large scale operations.

*Figure 1-1 Manual height measurement Picture Credit: Robotics Lab, U of S*

*Figure 1-2 Manual temperature measurement Picture Credit: Robotics Lab, U of S*

The ability to measure phenotypic expressions of plants is valuable to plant scientists so that the scientists may identify positive or negative improvements of genetic lines. With increased capacity for measuring phenotypes plant breeding programs may expand their scope and not be inhibited by the bottlenecks of intensive labor, time consuming and expensive nature of manual measurement. Computers and automation provide the opportunity for improving the capacity for measuring plant phenotypes. By integrating equipment such as cameras, thermometers, LiDAR and other such sensors with agriculture machinery it is possible to develop new methods and techniques for collecting plant data at increased rates. Using these new technologies to assist crop scientists and further the research in plant breeding will contribute to increased nutrition, food availability and will help alleviate world food insecurity.

## 1.2: Literature review

The use of machines and robots to collect phenotypic data is a modern trend with most earnest efforts beginning circa 2010. There are many different methods and platforms available to carry phenotyping sensors, each with their advantages and disadvantages. The following three tables attempt to highlight the advantages and disadvantages of the many different phenotyping platforms. Adapted and modified from (Dreery, et al. 2014), (White, et al. 2012), and (Yang, Feng and Zhang, et al. 2020).

*Table 1-1: Comparison of immobile platforms*

| Immobile Platforms | Advantages | Disadvantages | References |
|---|---|---|---|
| Immobile platforms (common) | • High repeatability<br>• Unsupervised operation | • Difficulty moving from site to site | |
| Fixed robot | • Mobile power source not required<br>• High sensor payload<br>• Individual plant measurements | • Generally expensive<br>• Can only measure a limited number of plots<br>• Slow data collection speed when compared to ground or aerial based platforms<br>• Suitable for individually potted greenhouse or laboratory plants; not outside plots. | (Dengyu, et al. 2016)<br><br>(Jansen, et al. 2009) |
| Towers | • Continuous monitoring<br>• High sensor payload<br>• Large field of view<br>• High rate of data collection compared to ground-based platforms | • High setup costs<br>• Distance from crop restricts types of sensors.<br>• May require multiple towers for large areas<br>• Only surface of the crop can be observed. | (Naito, et al. 2017)<br><br>(Shafiekhani, et al. 2017) |
| Cable robots and gantry systems | • Consistent point of view<br>• Unaffected by field/plot dimensions and layout<br>• Commercial availability | • Expensive to install<br>• Only covers small crop area | (LemnaTec 2020) |

*Table 1-2: Comparison of ground-based platforms*

| Ground-based Platforms | Advantages | Disadvantages | References |
|---|---|---|---|
| Ground-based (common) | • Medium rate of phenotyping<br>• Easily moved between research sites | • Unable to operate if the field is too muddy or wet.<br>• Lower rate of phenotyping compared to aerial platforms | |
| Manual measurement | • Widely available<br>• Easy to implement and teach | • Very slow<br>• High labour cost<br>• Unreliable: different operators give different results | (Schurr and Pieruschka 2019)<br><br>(Furbank and Tester 2011) |
| Pushcarts | • Inexpensive to manufacture<br>• Lightweight | • Hard to operate over irregular terrain<br>• Pushing the cart is tiring for the operator<br>• Low speed of travel limits throughput | (White and Conley 2013) |
| Modified Agricultural vehicles | • Widely available<br>• High payload<br>• Power source available<br>• Moderate speed of travel | • Skilled operator required<br>• Concerns of soil compaction<br>• High cost if not already owned.<br>• Large booms and attached arms can suffer from vibrations | (Kicherer, et al. 2017)<br><br>(Andrade-Sanchez, et al. 2013) |
| Manned buggies | • Widely available<br>• Easy to operate<br>• Many different design styles to accommodate different environments | • Requires dedicated vehicle<br>• Soil compaction concerns | (Dreery, et al. 2014) |
| Autonomous robots | • Continuous unsupervised data collection<br>• Low costs after purchase | • Expensive<br>• Safety mechanisms required | (Underwood, et al. 2017)<br>(Ruckelshausen, et al. 2009) |

*Table 1-3: Comparison of aerial platforms*

| Aerial platforms | Advantages | Disadvantages | References |
|---|---|---|---|
| Aerial platforms (common) | • High rate of data collection compared to ground-based platforms<br>• Easily moved between research sites<br>• Can collect data when the field is muddy/wet. | • Sensitivity to weather<br>• Except for manned aircraft: low sensor/power payload<br>• Cannot obtain measurements below the canopy | |
| UAV (unmanned aerial vehicle) | • Large field of view<br>• Inexpensive compared to ground-based platforms | • Short flight times<br>• Operator required<br>• Moderate sensitivity to wind<br>• Regulatory concerns | (Maes and Steppe 2019)<br><br>(Zhang and Kovacs 2012) |
| Manned aircraft (helicopters/ airplanes) | • High sensor payload<br>• Very high data collection rate<br>• Access to large power source<br>• Low sensitivity to wind | • Expensive<br>• Skilled operator required<br>• Low availability<br>• Regulatory concerns<br>• Downward forced air from rotors (downwash) can affect data collected | (Dreery, et al. 2014) |
| Blimps | • Inexpensive compared to other aerial platforms<br>• No risk of downwash | • High sensitivity to wind<br>• Unable for location to be set precisely | (White, et al. 2012) |

Each different style of platform provides advantages and disadvantages and can operated in different environments. The use of multiple platforms working synergistically has been explored by Shafiekhani et al. with vinobot and vinocular (discussed below) shows that when multiple systems are used in tandem the advantages of individual systems can diminish the disadvantages of the others at the cost of increased complexities and expenses. As this thesis focuses on the use of a ground-based platform to carry a robotic arm with sensors, further detail will be given on ground-based phenotyping platforms.

The use of a ground based mobile phenotyping device has been approached from many different designs. As one of the least complex designs, the proximal sensing cart or "phenocart" was designed to be easy and inexpensive to manufacture (White and Conley 2013). The phenocart is built from two bicycle frames, and a cage made from steel beams welded together. The cart has a 1m clearance from the ground and sensors can be mounted on supporting arms which are attached directly to the cage. The cart is moved manually. One person is required to push or pull the cart on level ground, but two people are needed if the terrain is irregular. The simple design of the cart results in a low cost of production and the cart itself is easily transported with a pickup truck trailer. Additionally, due to the low weight of the system (40kg) the risk of soil compaction would be greatly reduced. While autonomous ground vehicles are one method for a phenotyping platform, another option is to modify existing agricultural vehicles. In a study conducted at the University of Arizona, a high clearance sprayer tractor was modified to include a large boom on which sensors could be mounted (Andrade-Sanchez, et al. 2013). The boom was placed on the front of the vehicle to ensure that the crop was not affected by the sprayer itself. Two vertical lift arms combined with horizontally and vertically adjustable sensor brackets were used to ensure that the sensors continued to be orientated downward when the height of the boom was adjusted. The large distance between lift arms had the additional effect of increasing the stability of the boom while the prayer was in use. A 12V battery supplied by the tractor was used as the power source. The use of an agricultural vehicle is a popular idea as a platform for phenotyping sensors. Agricultural vehicles are widespread, can carry large payloads, and have the power capabilities to power multiple sensors. In contrast large farm vehicles require a trained operator, risk soil compaction if

multiple phenotyping sessions are required and would not be a suitable option if there has been recent rainfall.

The robotanist is a ground-based autonomous robot which was designed to operate in crops of corn and sorghum (Mueller-Sim, et al. 2017). Consisting of a 3  robotic arm and a GPS navigation system mounted on a mobile base, this unit operates underneath the canopy of the crop. Traveling underneath the canopy of the crop allows the robotanist to interact and manipulate individual plants; collecting data that is difficult or impossible for proximal sensors to collect. One of the difficulties faced by the robotanist occurs when the crop canopy has grown too tall and obscures the antenna, removing the ability of the robot to navigate. The robotanist is better suited for corn and sorghum with their wide seeded rows. Canola and wheat crops are seeded much more closely together and would not have the clearance for the robotanist to maneuver.

Vinobot and Vinocular are two distinct robots that work together to collect phenotypic data. Vinocular is an immobile tower, while Vinobot is an autonomous ground vehicle (Shafiekhani, et al. 2017). Equipped with camera sensors and able to rotate 360 degrees, Vinocular provides a high-level view of the crop below, identifying areas of stress or abnormalities.  The companion robot, Vinobot is a ground-based autonomous robot equipped with a robot arm. Vinobot is responsible for phenotyping individual plants and by communicating with Vinocular, Vinobot can prioritize and investigate areas of stress which were identified by the tower unit. Advantages of this system stem from having two different robot platforms working together. Vinocular has a large sensor and equipment payload and can provide perspectives that would be impossible for a ground-based robot to obtain. Vinobot can use Vinocular's information to target certain areas of the field for inspection. Having two units has drawbacks as well. The added complexity of communication between the two robots, longer setup times and increased manufacturing and maintenance costs of having two distinct platforms are to be kept in mind.

The robotic platform known as PhenoField (PF) is an attachment to a large agricultural tractor that houses very precise instruments for measuring plants (Svensgaard, Roitsch and Christensen 2014). A hydraulic crane holds a large box which is positioned over the crop. Wind and sunlight can vary throughout the day while phenotype measurements are being taken. PF is designed to compensate for changing environmental conditions such as wind and shade by lowering a box around the plants in the field creating a consistent environment for measurement. Sensors such as multispectral cameras are very sensitive to changing light sources. Changes in ambient light caused by cloud cover or shade caused by wind moving the plants are addressed by using the large box to ensure a consistent and known environment is used. Drawbacks of the PhenoField system mirror those of other agricultural vehicles. Difficulty of maneuvering in the field and soil compaction stem from using a large tractor. Additionally, when the box is lowered for measurements several plants are damaged. Throughput is limited as the procedure to take measurements is much more complex and time consuming.

A modified grape harvester was stripped of all the components usually required for harvest and instead was mounted with sensors. The vehicle itself drives overtop of each row of grapevines and the grapevines pass through a tunnel that was originally designed to shake the stem of the grape plant for harvest. Metal plates and a curtain were installed to block off any gaps in the machine to prevent direct sunlight interference with the sensors. Two different power sources are available for powering the electronics: the vehicle itself runs from a diesel motor which powers a generator while running, while the motor is off battery packs have been installed so that the computers and sensors can run without the harvester having to be running (Kicherer, et al. 2017). This system has a couple useful ideas; first the use of a tunnel with curtains to ensure lighting consistency is a less expensive and quicker procedure compared to the PhenoField's box apparatus, although the lighting might not be as precise. Secondly the use of two sources of power can increase the flexibility of use of the machine. However, as the system was adapted from a grape harvester it would be difficult to accommodate the dimensions of a wheat or canola plot. Additionally, the need to have a specific vehicle to modify is more stringent than having a robotic arm or boom to install on agricultural vehicles that are already owned by the researchers.

Ladybird is an autonomous ground-based phenotyping robot designed by the Australian Center for Field Robotics and the South Australian Research and Development Institute. While initially designed for vegetable crops, Ladybird has been adapted to work with commercial grain crops. Ladybird is powered by a bank of batteries which can be charged both manually and receive power from the solar panels on the back of the robot. The solar power source is sufficient for typical weeklong field trials without having to re-charge. With the current design, the Ladybird has a top speed of 1.2m/s. Four wheels on two decoupled axles helps reduce soil shear and power consumption, and the body of the robot is designed to make easy adjustments to clearance height and track width (Underwood, et al. 2017). The use of solar panels provides two benefits: the robot would require less charging resulting in longer active data collection times, as well as providing a shaded environment for consistent measurement. Currently two operators are required for data collection. With improvements to the navigation and data collection equipment lowering the human labour required may be possible.

The robotic platform for plant phenotyping known as Bonirob is a fully autonomous 4 wheeled mobile robot. Each of the wheels can be steered separately and adjusted hydraulically to change the track width or clearance height to collect data from different crop types. To ensure the safety of the humans in the operation site, each wheel contains an emergency shutdown button. Any malfunctions in the software also cause the robot to shut down (Ruckelshausen, et al. 2009). The ability to adjust the dimensions of the vehicle make Bonirob a very flexible option for phenotyping different crops and different seeding configurations. However, because Bonirob must straddle the plants it is phenotyping there are certain plot configurations (width greater than 2m) that Bonirob would not be suitable for.

### 1.2.1 Research at the U of S

Research at the University of Saskatchewan is driven by the need to collect phenotypic data from wheat and canola breeder plots. It was decided with consultation of plant scientists that height measurement, canopy temperature measurement and images of the crops are the initial scope of the research project, and a platform is to be created and tested for high throughput data collection.

The robotics lab research team focused on the design of 2 booms that could be attached to agricultural vehicles. The first-generation boom was a 6m length of steel beam that was mounted on the back of a swather (Zhang and Fotouhi, 2018). Sensors were installed along the boom and cables powering the sensors and transmitting data were connected to the datalogger and laptop secured in the center of the boom. Challenges in transporting the robotic arm were the catalyst for the second-generation boom, as the 6m long boom was too wide to travel down Saskatchewan roads.

*Figure 1-3: 1st Generation Boom. Picture Credit: U of S Robotics Lab*

The development of the second-generation boom added two degrees of freedom along the beam to allow the boom to be folded into a transport mode (Q. Zhang 2018). Additionally, new brackets were designed to hold sensors that could be adjusted to change the height of the sensor from the crop or the angle of the sensor. The 2nd generation boom accomplished the requirements of accuracy and throughput of the plant scientist consultant but there were still drawbacks. Adjustment of the boom height still required a minimum of two people to accomplish, and the robotic boom was not suitable for installation on an autonomous mobile robotic base. The use of a smaller vehicle to carry the robot arm will also address the problem of having heavy agricultural booms and machinery drive the same path multiple times during a growing season which contributes to soil compaction reducing the fertility of the soil.

*Figure 1-4: 2nd Generation Boom. Left: Folded Form, Right: Open Form. Picture Credit: Robotics Lab, U of S*

Current research at the robotics lab at the University of Saskatchewan revolves around the 5-DOF robotic arm known as the 3rd generation boom. This robotic arm is designed to be carried by an agricultural vehicle such as a tractor or a swather and has a 2.5m reach from its

base to the end effector. Sensors will be powered by batteries, negating the requirement to draw power from the agricultural vehicle and keeping the selection of acceptable mounting vehicles open. Sensor payloads of up to 10kg are supported. The robot arm can be folded into a closed position for travelling and can be opened and extended when collecting data. Made of structural steel and aluminum estimates put the entire weight of the robot at approximately 70kg without any payload. The 3rd generation boom is unique from other 5-DOF robotic arms due to the extensive horizontal reach and the lightweight structure, allowing for phenotyping of large breeder plots where other platforms struggle. Below Figure 1-5 shows a 3d model of the robotic arm and the specific names of each part. Specific degrees of freedom are described in the following chapter. Additional information on the development of the 3rd generation boom can be found in (Q. Zhang, et al. 2019).

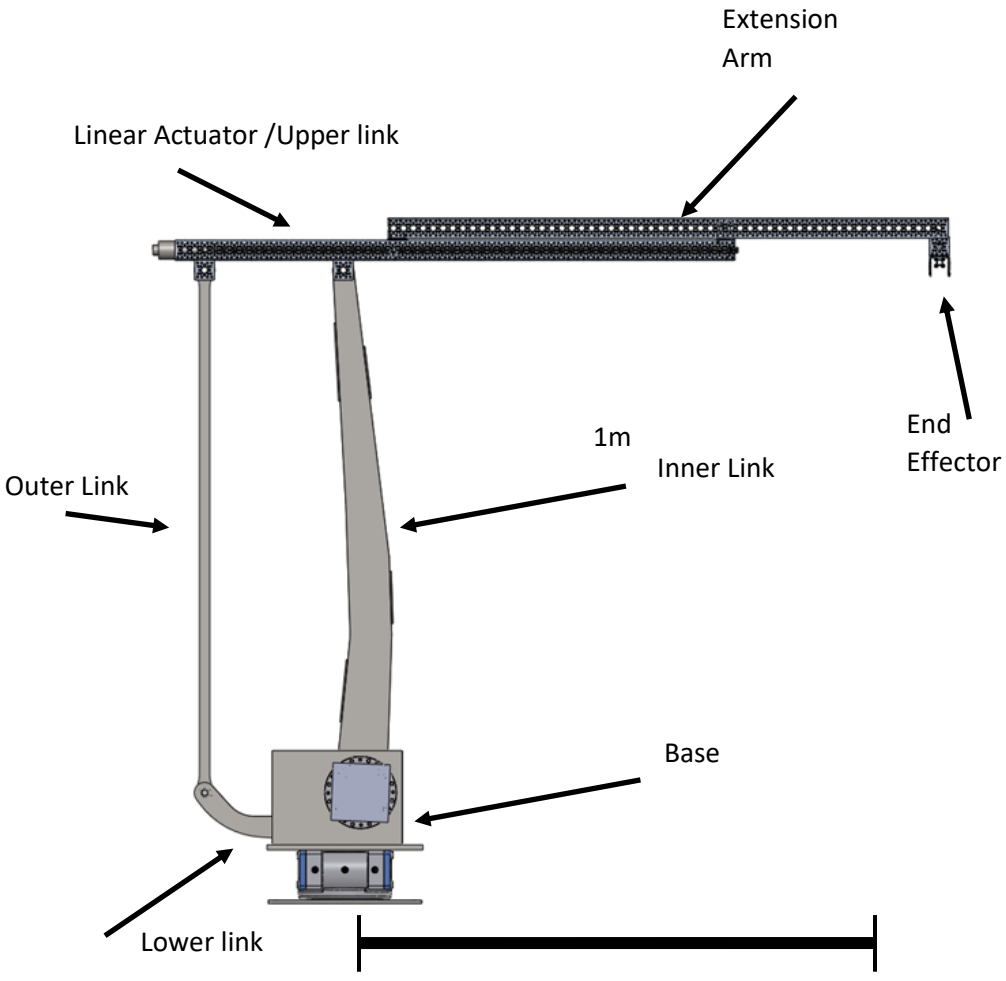



*Figure 1-5: 3rd generation boom in profile with link names*

## 1.3:Problem statement

The success of the 3rd generation boom as a viable high throughput phenotyping platform depends on being able to solve 3 distinct challenges. The location of the robot end effector must always be known, and the robot must have accurate, repeatable positioning and the robot must be lightweight.

Knowing the location of the end effector is extremely important. When data is collected by the sensors held by the robot the data must be geotagged; linking the GPS coordinates of the plants being measured to the data collected. As the GPS navigation system is not located at the end effector, the accurate transformations between the end effector and the GPS location must be calculated. Complicating this problem is the dynamic nature of the 3rd generation boom, the location of the end effector relative to the base of the robot is not static and changes with the movement of the robot. Additionally, the ability to specify trajectories for the end effector to follow is vital for use of the robot in an industrial environment.

The total mass of the robotic arm is also critical. By minimizing the mass of the robotic arm, the torque and force requirements of the joint motors and actuators can be decreased resulting in smaller motors required which in turn require smaller power sources. Additionally, the risk of soil compaction from having heavy equipment follow the same path through the field is reduced by having a lighter robot.

## 1.4: Objectives and Methodology

There are 2 major research questions addressed in this thesis:

1. What kind of analysis is appropriate for the development of a 5 DOF robot manipulator?

   Solving the kinematic and dynamic equations of the 5-DOF robotic arm will allow control of the robot arm in both joint space and task space. The objective to control the robot arm will be achieved by:

   a. Kinematic equations will be developed using a geometric method based off the Denavit-Hartenburg parameters and verified using simulation results from FEA results.
   b. The forces required at each joint motor or actuator will be solved using the analytical dynamic equations. The analytical dynamic equations for the motion of the robotic arm will be derived using Newton-Euler method. As the robot is not a linear chain of links like the Newton-Euler method was designed for, modifications to the Newton-Euler method to account for the parallel links will be required. Simulation results from the rigid dynamics module of ANSYS workbench will be compared with the analytical Newton-Euler method results.

2. Can the weight of individual links be minimized to reduce the overall weight of the robot arm?

   One of the challenges of the 3$^{rd}$ generation boom is for the robot arm to be as lightweight as possible. This has the benefits of reducing power requirements of joint motors and decreasing the risk of soil compaction. The objective to reduce the weight of the robot arm will be achieved by:

   a. Using the results of the dynamic analysis of the 5-DOF robotic arm as boundary conditions the inner link of the robot will be subjected to an optimization algorithm to minimize the weight of the link while respecting constraints of deflection and maximum stress.

b. To verify the results of the optimization, finite element analysis of the inner link will be completed to ensure that the maximum stress and maximum deflection constraints of the optimization were not violated.

## 1.5: Thesis Outline

Chapter 1 of this thesis will explore the background, motivation and previous research on the topic of high throughput phenotyping and the challenges facing the current robotic arm developed by the Robotics lab at the University of Saskatchewan.

Chapter 2 of this thesis will cover the kinematics, inverse kinematics, and dynamics of the robot arm. These analytical results computed in MATLAB will be compared to simulation results solved by ANSYS workbench.

Chapter 3 of this thesis will use the dynamic equations developed in chapter 2 to apply boundry conditions on a weight optimization algorithm created to minimize the weight of the inner link subject to deflection and stress constraints. Stress and deflection analysis will be completed using finite element analysis to verify the solutions.

Chapter 4 of this thesis will summarize the main contributions of this thesis and closing remarks.

# Chapter 2: Kinematics and Dynamics

## 2.1: Introduction

The 5 degree of freedom (5DOF) robotic arm has been designed by the robotic research lab at the University of Saskatchewan to be used in a farm setting, particularly to collect data from breeder plots. The detailed design was first reported by Tyler Zhang (Q. Zhang, Design and Analysis of Booms for Wheeled Mobile Platform for Crop Phenotyping 2018). In this section the forward kinematics, inverse kinematics, and forward dynamics of this robot manipulator will be presented.

Analysing robots requires the definition of task-space and joint-space. Joint-space is the term for the variables associated with the joints of a robot. Angle, angular velocity, and angular acceleration are parameters used to describe rotational joints, while position, velocity, and acceleration are used for prismatic joints. In contrast, task space, or cartesian space measure angular and linear positions, velocities, and accelerations relative to a fixed point in cartesian space. Task space variables are used to describe where the end effector or tool of a robot hand is.

Forward kinematics is the analysis of a robot to determine the equations which when solved, reveal the location of the end effector in task-space when the joint-space parameters are known. The forward kinematics are useful when the positions of the robot are predefined and constant, making forward kinematics the preferred for specifying robot paths when the robotic job tasks are low complexity, and the environment is known. When specifying trajectories in the joint-space, paths are defined by start and end locations of the end effector in task-space, but a task-space defined path of travel is not possible when only the forward kinematics are known.

Inverse kinematics is the technique used to solve for the joint-space parameters when the location of the end effector is given in task-space. This method allows the end effector to be placed in a location that is much more easily specified by the operator. However, solving for the joint-space parameters is more complex. The task space location specified may not have a

unique joint-space solution which will require additional constraints to solve. The end-effector locations may be outside of the work space of the robot or may be at singularities which are unsolvable. The trade-off for the complexities in calculation are offset by the ease of defining specific task-space trajectories that may be necessary when the robot is in complex environments or completing complex tasks.

Forward dynamics is transforming the joint-space parameters to task-space, but considering mass, mass moment of inertia of the manipulator, it specifically deals with forces and torques of the motors and joints that are required to perform job tasks. While forward kinematics deals with the position of each robot link and joint, the dynamic equations determine how much torque or force is required at each joint to produce motion. Dynamic analysis takes into consideration gravitational, inertial, and applied forces so that motors and actuators may be sized for robot design.

Physical experimental results for these equations are not obtained in this thesis as the robotic arm was still in design phase and has not been fabricated. Verification of the kinematics and dynamics has been done using simulated results using ANSYS workbench.

If physical experiments were to be performed to verify findings in this thesis a variety of sensors would be employed. The position of the robot arm joints would potentially be measured using encoders or optical sensors. deflection of the end effector could be recorded using accelerometers.

Key differences include new frame locations to reflect changes in the mechanical design, different methods of verification, and 5 degrees of freedom included in the inverse kinematics.

## 2.2: Methodologies

Figure 2-1 shows the robotic arm in a 2-d plane view. Each of the dimensions are labeled as $L_{1\text{-}9}$, while the degrees of freedom are $\theta_{1\text{-}5}$.

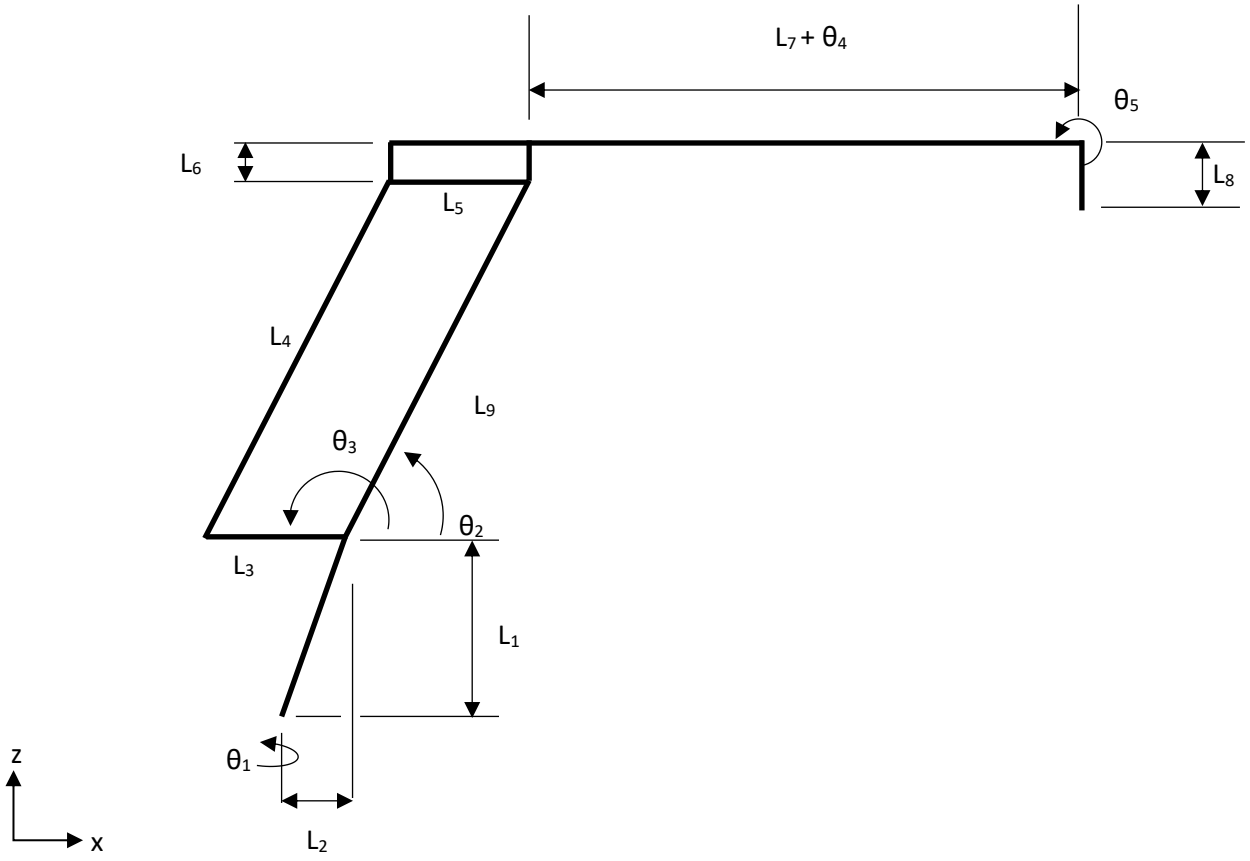



*Figure 2-1: Link variables and degrees of freedom of the 3rd generation boom*

To understand the motion of the 3rd generation robotic arm the equations for dynamics must be derived. Using the Denavit-Hartenburg parameters, or "DH" parameters, the 3rd generation boom may be described by 4 variables to denote each link's length, orientation, and degrees of freedom. Table 1-1 shows the DH table for the robotic arm. Coordinate frames were assigned according to the DH algorithm, where frame origins are located at each joint, and axis are determined through the location of the subsequent joint and the degrees of freedom available to the current joint. DH parameters were designed to be used in serial robots; robots whose links connect to a single link before, and a single link after.

The 5 DOF robot arm is a parallel manipulator, thus there are two different paths that one may take to reach the end effector,

1: base - inner link - end effector

or

2: base – outer link - end effector

If only one path is followed, information about the other path (position, velocity and acceleration, or joint forces) will not be known. To analyze the robot two separate sets of frames and DH parameters were required for the 3rd generation boom as shown in Figures 2-2 and 2-3. Appendix A gives an explanation and more information on DH parameters.

In the following Figures of this thesis, when indicating the position and orientation of coordinate frames, the origin of the coordinate frame will be marked with a red star: ✸. The orientation of the coordinate frames will be marked with 2 axis.

Assumptions and limitations

When deriving the kinematic and dynamic equations several assumptions are made:

1. During this analysis the effects of flexible links are ignored and the links of the robot are considered to be rigid bodies.

2. Perfectly aligned joints.
   it is assumed during this analysis that each of the joints of the robot are perfectly installed with no manufacturing errors, so that the motion of the robot is exactly as described.

3. Frictionless joints
   When calculating the forces necessary at each joint for the motion of the robot it is assumed that each of the joints are frictionless.

4. In this experiment the analytical kinematic and dynamic equations are derived and are programmed into MATLAB to be solved. As a verification a rigid dynamics ANSYS workbench model is also programmed and the simulated dynamics and kinematics are compared to the analytical equations. Comparing the analytical equations to the simulated results will only verify that the models are identical, but does not verify that the models are correct. It is ideal to perform Physical experiments to validate both the analytical equations and the simulated model.

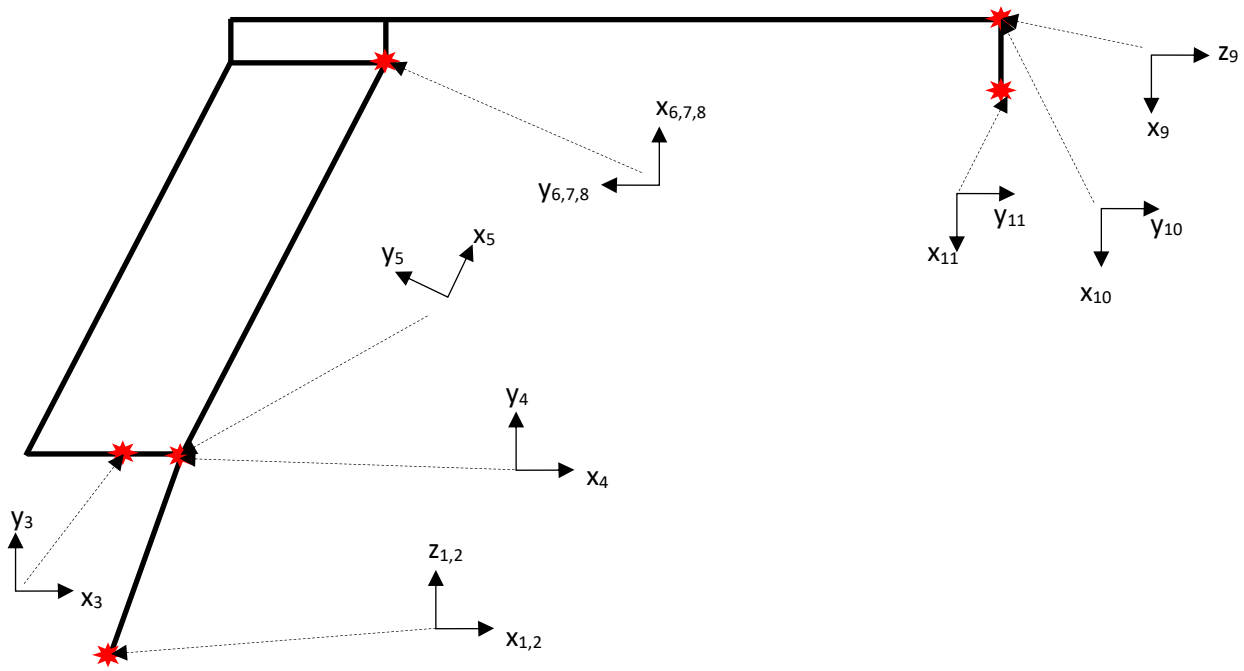



*Figure 2-2: Inner loop frame locations and orientations*

*Table 2-1: D-H Parameters of the inner loop*

| i | $\alpha_{i-1}$ (radians) | $a_{i-1}$ (m) | $d_i$ (m) | $\theta_i$ (radians) |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | $\theta_1$ |
| 2 | 0 | 0 | $L_1$ | 0 |
| 3 | 0 | $L_2$ | 0 | 0 |
| 4 | $\pi/2$ | 0 | 0 | $\theta_2$ |
| 5 | 0 | $L_9$ | 0 | $\theta_3 - \theta_2 - \pi/2$ |
| 6 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 |
| 8 | $\pi/2$ | $L_6$ | $L_7 + \theta_4$ | 0 |
| 9 | $\pi/2$ | 0 | 0 | $\theta_5$ |
| 10 | 0 | $L_8$ | 0 | 0 |

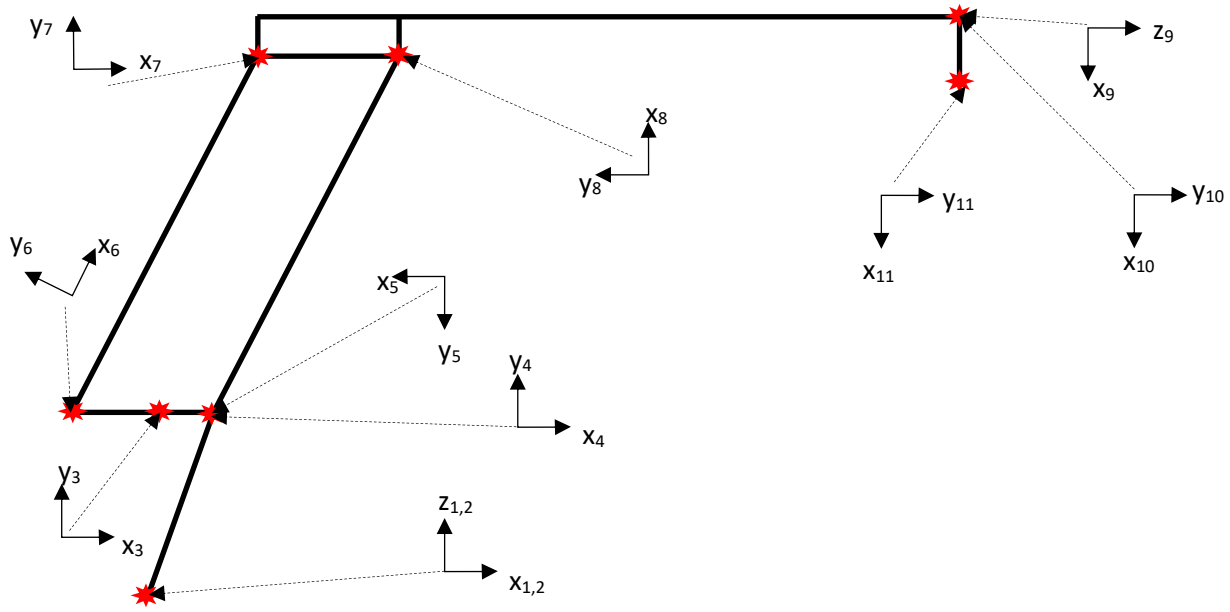



*Figure 2-3: outer loop frame locations and orientations*

*Table 2-2: D-H parameters for the outer loop*

| i | $\alpha_{i-1}$ (radians) | $a_{i-1}$ (m) | $d_i$ (m) | $\theta_i$ (radians) |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | $\theta_1$ |
| 2 | 0 | 0 | $L_1$ | 0 |
| 3 | 0 | $L_2$ | 0 | 0 |
| 4 | $\pi/2$ | 0 | 0 | $\theta_3$ |
| 5 | 0 | $L_3$ | 0 | $2\pi - \theta_3 + \theta_2$ |
| 6 | 0 | $L_4$ | 0 | $\pi + \theta_3 + \theta_2$ |
| 7 | 0 | $L_5$ | 0 | $\pi/2$ |
| 8 | $\pi/2$ | $L_6$ | $L_7 + \theta_4$ | $\pi$ |
| 9 | $\pi/2$ | 0 | 0 | $\theta_5$ |
| 10 | 0 | $L_8$ | 0 | 0 |

## 2.2.1 Kinematics methodologies

The forward kinematic equations are derived using the D-H parameters to create a series of coordinate transformation matrices that when multiplied together will create the total coordinate transformation from the base of the robot to the end effector. These coordinate transformations can be used by MATLAB to convert the joint space geometry of the robot into cartesian coordinates of the end effector.

A transformation matrix which relates frame i-1 to frame i can be derived using the following formula, which has the form of

$$ {}^{i-1}_{i}T = \left[\begin{array}{c|c} {}^{i-1}_{i}R & {}^{i-1}P_i \\ \hline 0\ 0\ 0 & 1 \end{array}\right] \qquad 2.1 $$

Where ${}^{i-1}_{i}R$ is the rotation matrix from i-1 to i, and ${}^{i-1}P_i$ is the location vector describing the origin of i-1 relative to the origin of i. The transformation matrix can be written in the terms of the DH parameters to obtain the transformation from each frame to the next as shown below in 2.2. Note that c, and s are shorthand for sin() and cos() respectively.

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 2.2 $$

A complete explanation of the transformation matrix and derivation can be found in the textbook “Introduction to Robotics” by John Craig, chapter 2. (Craig 2005).

Using the DH parameter tables above and formula 2-2 the following transformation matrices may be derived. For the outer loop:

$$ {}^{1}_{2}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.3 $$

$$ {}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.4 $$

$$ {}^{3}_{4}T = \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.5 $$

$$ {}^{4}_{5}T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.6 $$

$$ {}^{5}_{6}T = \begin{bmatrix} c(\theta_2 - \theta_3 - 360) & -s(\theta_2 - \theta_3 - 360) & 0 & L_3 \\ s(\theta_2 - \theta_3 - 360) & c(\theta_2 - \theta_3 - 360) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.7 $$

$$ {}^{6}_{7}T = \begin{bmatrix} c(\theta_3 - \theta_2 - 180) & -s(\theta_3 - \theta_2 - 180) & 0 & L_4 \\ s(\theta_3 - \theta_2 - 180) & c(\theta_3 - \theta_2 - 180) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.8 $$

$$ {}^{7}_{8}T = \begin{bmatrix} 0 & -1 & 0 & L_5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.9 $$

$$ {}^{8}_{9}T = \begin{bmatrix} -1 & 0 & 0 & L_6 \\ 0 & 0 & -1 & -(L_7 + \theta_4) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.10 $$

$$ {}^{9}_{10}T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.11 $$

$$ {}^{10}_{11}T = \begin{bmatrix} 1 & 0 & 0 & L_8 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.12 $$

For the inner loop, only 4 transformation matrices are different, ${}^{4}_{5}T,\ {}^{5}_{6}T,\ {}^{6}_{7}T, and\ {}^{7}_{8}T$.

$$ {}^{4}_{5}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.13 $$

$$ {}^{5}_{6}T = \begin{bmatrix} c(\theta_2 - \theta_3 + 90) & -s(\theta_2 - \theta_3 + 90) & 0 & L_3 \\ s(\theta_2 - \theta_3 + 90) & c(\theta_2 - \theta_3 + 90) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.14 $$

$$ {}^{6}_{7}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.15 $$

$$ {}^{7}_{8}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.16 $$

The forward transformation from the base to robot end effector can be found by multiplying the transformation matrices together.

$$ {}^{1}_{11}T = {}^{1}_{2}T \cdot {}^{2}_{3}T \cdot {}^{3}_{4}T \cdot {}^{4}_{5}T \cdot {}^{5}_{6}T \cdot {}^{6}_{7}T \cdot {}^{7}_{8}T \cdot {}^{8}_{9}T \cdot {}^{9}_{10}T \cdot {}^{10}_{11}T \qquad 2.17 $$

Which results in the following rotation and position matrices.

To improve clarity of ${}^{1}_{11}T$, as the matrix would be too large for readability the matrix has been split into ${}^{1}_{11}R$, and ${}^{1}_{11}P$ and may be combined as per Equation 2-1 to obtain the full matrix ${}^{1}_{11}T$.

$$ {}^{1}_{11}R = \begin{bmatrix} -C\theta_1 S(\theta_3+\theta_5) & -C\theta_1 C(\theta_3+\theta_5) & -S\theta_1 \\ -S\theta_1 S(\theta_3+\theta_5) & -S\theta_1 C(\theta_3+\theta_5) & -C\theta_1 \\ C(\theta_3+\theta_5) & -S(\theta_3+\theta_5) & 0 \end{bmatrix} \qquad 2.18 $$

$$ {}^{1}_{11}P = \begin{bmatrix} C\theta_1[L_2 + L_9 C\theta_2 - (L_7+\theta_4)C\theta_3 \ - L_8 S(\theta_3+\theta_5) + L_6 S(\theta_3)] \\ S\theta_1[L_2 + L_9 C\theta_2 - (L_7+\theta_4)C\theta_3 \ - L_8 S(\theta_3+\theta_5) + L_6 S(\theta_3)] \\ L_1 + L_9 S\theta_2 - (L_7+\theta_4)S\theta_3 + L_8 C(\theta_3+\theta_5) \ - L_6 C(\theta_3) \end{bmatrix} \qquad 2.19 $$

To solve for the inverse kinematics, the transformation matrix ${}^{1}_{11}T$ is given, and the equations must be solved for the joint parameters rather than the task space parameters that are already solved for.

Once the inverse kinematic equations are derived, MATLAB can solve the equations to convert cartesian coordinates of the end effector into joint space positions.

$$ \theta_1 = atan2(y, x) \qquad 2.20 $$

$$ \theta_2 = \sin^{-1}\left(\frac{G + \varphi S\theta_3 + L_6 C\theta_3}{L_9}\right) \qquad 2.21 $$

$$ H = \varphi^2 + 2(F\varphi + GL_6) * Cos(\theta_3) + 2(\varphi G - FL_6) * sin(\theta_3) \qquad 2.22 $$

Equation 2-22 can be solved for either $\varphi \; or \; \theta_3$ depending on which additional constraints are given by the operator. If the linear actuator is to be horizontal, $\theta_3 = 0$ and then Equation 2-23 is used to solve for $\varphi, then \; for \; \theta_4$. If the linear actuator is to be extended a certain length then $\varphi$ is known and $\theta_3$ can be solved for using Equation 2-24 .

$$\varphi = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad 2.23$$

$$\theta_3 = 2n\pi \pm \cos^{-1}\left(\frac{f}{\sqrt{d^2 + e^2}}\right) + \tan^{-1}\left(\frac{d}{e}\right) \qquad 2.24$$

$$\theta_4 = \varphi - L_7 \qquad 2.25$$

$$\theta_5 = sin^{-1}(-A) - \theta_3 \qquad 2.26$$

Where

$$A = T_{11}^{1}(3,2) \qquad 2.27$$

$$B = T_{11}^{1}(3,1) \qquad 2.28$$

$$D = AL_8 + L_2 \qquad 2.29$$

$$E = L_8 B + L_1 \qquad 2.30$$

$$F = \frac{x}{Cos(\theta_1)} - D \qquad \text{or if x=0:} \qquad F = \frac{y}{sin(\theta_1)} - D \qquad 2.31$$

$$G = z - E \qquad 2.32$$

$$a = 1 \tag{2.33}$$

$$b = 2F \cdot Cos(\theta_3) + 2G \cdot Sin(\theta_3) - H \tag{2.34}$$

$$c = 2GL_6\, Cos(\theta_3) - 2FL_6\, Sin(\theta_3) \tag{2.35}$$

$$d = 2(\varphi G + FL_6) \tag{2.36}$$

$$e = 2(F\varphi - GL_6) \tag{2.37}$$

$$f = H - \varphi^2 \tag{2.38}$$

These equations are only valid when the joints are within the appropriate limits of operation as shown in Table 2-3. These limitations are chosen to reflect the mechanical restrictions of the robot. Positions of the end effector which cause the degrees of freedom limits to be violated will produce errors in the inverse kinematic equations.

*Table 2-3 Range of motion for each joint in the robotic arm*

| Degree of freedom | Lower limit | Upper limit |
|---|---|---|
| $\theta_1$ | 0° | 360° |
| $\theta_2$ | 0° | 120° |
| $\theta_3$ | $\leq (\theta_2 + 5^o)$ | 180° |
| $\theta_4$ | 0mm | 1000mm |
| $\theta_5$ | -90° | 90° |

One challenge of the inverse kinematics is that the solution for any given task-space position is not unique. That is: there are 5 degrees of freedom ($\theta_{1-5}$) to solve for. Even if the complete position and orientation of the end effector is known, there are multiple joint combinations that could produce the exact position of the end effector. As there are multiple solutions the above equations must be solved with additional restrictions, such as keeping the linear actuator extended as little as possible or keeping the linear actuator as parallel to the ground as possible. Which criteria is used as extra constraints is dependant on the environment the robot is operating in as well as the tasks it is assigned.

### 2.2.2 Dynamics Methodology

Understanding the dynamics of the robotic arm is imperative to choosing the proper motors and actuators. In this section the Newton-Euler equations of motion are applied to each link of the robot arm to determine the forces required at each of the motors to produce a specified motion of the robot arm.

First, the outward iterations work from the base of the robot to the end effector converting the joint parameters ( positions, velocities and accelerations) to task space parameters so that the positions, velocities and accelerations of the centers of mass of each link are known in cartesian coordinates. After calculating the linear and rotational accelerations and knowing the mass distribution of each link the Newton-Euler equations are used to determine the forces and

moments acting on each of the robot links. Next, force and moment equilibrium equations are written for each link working from the end effector back to the base of the robot in order to calculate the joint forces required for motion of the robot.

The Newton-Euler equations of motion written in D-H parameter form are as follows. (Craig 2005)

For $0$ to $i-1$

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} \qquad 2.39$$

$$\text{For rotational: } {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\,{}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\,{}^{i}\omega_i \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} \qquad 2.40$$

$$\text{For rotational: } {}^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R\left({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times \left({}^{i}\omega_i \times {}^{i}P_{i+1}\right) + {}^{i}\dot{v}_i\right) \qquad 2.41$$

$$\begin{aligned}\text{For prismatic: } {}^{i+1}\dot{v}_{i+1} \\ &= {}^{i+1}_{i}R\left({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times \left({}^{i}\omega_i \times {}^{i}P_{i+1}\right) + {}^{i}\dot{v}_i\right) \\ &+ 2\,{}^{i+1}\omega_{i+1} \times \dot{d}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{d}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}\end{aligned} \qquad 2.42$$

$$\text{For prismatic: } {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\,{}^{i}\omega_i \qquad 2.43$$

$${}^{i+1}\dot{v}_{C_{i+1}} = {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C\,i+1} + {}^{i+1}\omega_{i+1} \times \left({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{C\,i+1}\right) + {}^{i+1}\dot{v}_{i+1} \qquad 2.44$$

$${}^{i+1}F_{i+1} = m_{i+1}\,{}^{i+1}\dot{v}_{C_{i+1}} \qquad 2.45$$

$${}^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\omega_{i+1} \qquad 2.46$$

for $i$ to $1$

$$ {}^{i}f_i = {}_{i+1}^{i}R\ {}^{i+1}f_{i+1} + {}^{i}F_i \qquad 2.47 $$

$$ {}^{i}n_i = {}^{i}N_i + {}_{i+1}^{i}R\ {}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i + {}^{i}P_{i+1} \times {}_{i+1}^{i}R\ {}^{i+1}f_{i+1} \qquad 2.48 $$

$$ \tau_i = {}^{i}n_i^T\ {}^{i}\hat{Z}_i \qquad 2.49 $$

These equations for dynamic forces were derived for serial manipulators, where each link forms a continuous chain. However, the top link in the 3rd generation boom is connected to both the inner link and the outer link rendering the above equations unusable in their current form. Individual free body diagrams must be drawn to determine the correct internal joint forces; ${}^{i}f_i$ and ${}^{i}n_i$; for each of the connections. The free body diagram is shown below in Figure 2-4.

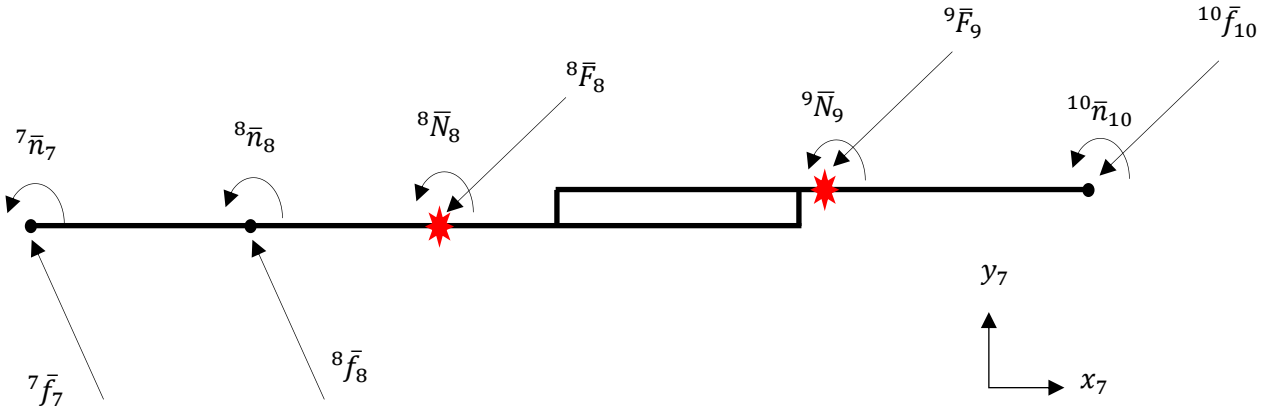



*Figure 2-4: Free body diagram of the actuator and extension link*

Applying the static equations for sum of moments and forces in 3 dimensions results in the following 2 equations.

$$ \sum moments_8 = 0 \qquad 2.50 $$

$$\begin{bmatrix}0\\0\\0\end{bmatrix} = {}^{7}\bar{n}_7 + {}^{8}\bar{n}_8 + {}^{10}\bar{n}_{10} + {}^{8}\bar{N}_8 + {}^{9}\bar{N}_9 + ({}^{8}\bar{P}_7 \times {}^{7}\bar{f}_7) + ({}^{8}\bar{P}_{C_8} \times {}^{8}\bar{F}_8) + ({}^{8}\bar{P}_{C_9} \times {}^{9}\bar{F}_9) + ({}^{8}\bar{P}_7 \times {}^{10}\bar{f}_{10}) \quad 2.51$$

$$\sum forces = 0 \quad 2.52$$

$$\begin{bmatrix}0\\0\\0\end{bmatrix} = {}^{7}\bar{f}_7 + {}^{8}\bar{f}_8 + {}^{8}\bar{F}_8 + {}^{9}\bar{F}_9 + {}^{10}\bar{f}_{10} \quad 2.53$$

The purpose of this analysis is to determine the torques acting at the joints where motors are located. As such it is known that the joints which actuate $\theta_2$and $\theta_3$ only produce torque in the z direction of their respective frames. Thus, only components of ${}^{8}\bar{f}_8$that can contribute to the motor forces are in the x-y plane of frame 8. The problem is reduced to neglect the forces acting in the z direction and only solve for the forces at frames 7 and 8 in the x and y directions.

$$\sum moments_{8z} = 0 \quad 2.54$$

$$0 = {}^{7}n_{z_7} + {}^{8}n_{z_8} + {}^{10}n_{z_{10}} + {}^{8}N_{z_8} + {}^{9}N_{z_9} + ({}^{8}\bar{P}_7 \times {}^{7}\bar{f}_7) + ({}^{8}\bar{P}_{C_8} \times {}^{8}\bar{F}_8) + ({}^{8}\bar{P}_{C_9} \times {}^{9}\bar{F}_9) + ({}^{8}\bar{P}_7 \times {}^{10}\bar{f}_{10}) \quad 2.55$$

Due to pin connections at 7 and 8

$${}^{7}n_{z_7} = {}^{8}n_{z_8} = 0 \quad 2.56$$

Equation 2-54 can be then solved as

$$-\left({}^{8}\bar{P}_{7} \times {}^{7}\bar{f}_{7}\right) = {}^{7}n_{z_7} + {}^{8}n_{z_8} + {}^{10}n_{z_{10}} + {}^{8}N_{z_8} + {}^{9}N_{z_9} + ({}^{8}\bar{P}_{C_8} \times {}^{8}\bar{F}_{8}) + ({}^{8}\bar{P}_{C_9} \times {}^{9}\bar{F}_{9}) + ({}^{8}\bar{P}_{7} \times {}^{10}\bar{f}_{10}) \quad \textit{2.57}$$

Where $-\left({}^{8}\bar{P}_{7} \times {}^{7}\bar{f}_{7}\right)$ is the moment required at frame 7, developed by ${}^{7}\bar{f}_{7}$ for equilibrium.

$$\sum forces = 0 \quad \textit{2.58}$$

$$-{}^{8}\bar{f}_{8} = {}^{7}\bar{f}_{7} + {}^{8}\bar{F}_{8} + {}^{9}\bar{F}_{9} + {}^{10}\bar{f}_{10} \quad \textit{2.59}$$

All of the quantities on the RHS of the equation are known, as well as the value of ${}^{8}\bar{P}_{7}$from the kinematics described above, However the value of ${}^{7}\bar{f}_{7}$ is unknown and the inverse of a cross product is not unique. To solve Equation 2-58 a free body diagram of the outer link must be drawn Once solved, the equation for sum of forces may be solved for the value of ${}^{8}\bar{f}_{8}$.

$$\sum forces = 0 \quad \textit{2.60}$$

$$-{}^{8}\bar{f}_{8} = {}^{7}\bar{f}_{7} + {}^{8}\bar{F}_{8} + {}^{9}\bar{F}_{9} + {}^{10}\bar{f}_{10} \quad \textit{2.61}$$

A free body diagram of the outer link is shown below in Figure 2-5

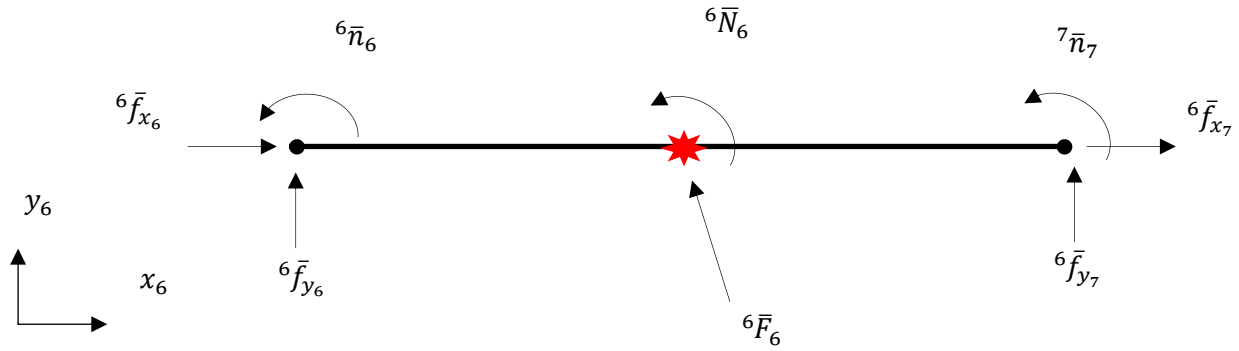



*Figure 2-5: Outer link Free body diagram*

The static equilibrium equations for 2 dimensions are shown below.

Due to pin connections at frames 6 and 7;

$$ {}^6n_{z_6} = {}^7n_{z_7} = 0 \tag{2.62} $$

$$ \sum moments_{6z} = 0 \tag{2.63} $$

$$ 0 = {}^7n_{z_7} + {}^6n_{z_6} + {}^6N_{z_6} + ({}^6\bar{P}_{c_6} \times {}^6\bar{F}_6) + ({}^6\bar{P}_7 \times {}^7\bar{f}_7) \tag{2.64} $$

As ${}^6f_{x_7}$ cannot contribute any moment in the frame 6 $\hat{z}$ direction, the moment equation may be simplified to

$$ 0 = +{}^6N_{z_6} + ({}^6\bar{P}_{c_6} \times {}^6\bar{F}_6) + {}^6P_7{}^6f_{y_7}) \tag{2.65} $$

Which can be solved for ${}^6f_{y_7}$

Performing the transformation from ${}^{6}\bar{f}_{7}$ to ${}^{7}\bar{f}_{7}$

$$ {}^{6}\bar{f}_{7} = \begin{bmatrix} {}^{6}f_{x_7} \\ {}^{6}f_{y_7} \\ {}^{6}f_{z_7} \end{bmatrix} \qquad 2.66 $$

$$ {}^{7}\bar{f}_{7} = {}^{7}_{6}R\,{}^{6}\bar{f}_{7} = {}^{7}_{6}R \begin{bmatrix} {}^{6}f_{x_7} \\ {}^{6}f_{y_7} \\ {}^{6}f_{z_7} \end{bmatrix} \qquad 2.67 $$

As ${}^{6}f_{y_7}$ is known, and ${}^{6}f_{z_7}$ can be neglected, the LHS of 2-56

$$ \left({}^{8}\bar{P}_{7} \times {}^{7}\bar{f}_{7}\right) \qquad 2.68 $$

can be written as

$$ {}^{8}\bar{P}_{7} \times {}^{7}_{6}R\,{}^{6}f_{y_7} + {}^{8}\bar{P}_{7} \times {}^{7}_{6}R\,{}^{6}f_{x_7} \qquad 2.69 $$

Equation 2-58 can be solved for ${}^{6}f_{x_7}$which can then be substituted into Equation 2-66 to solve for ${}^{7}\bar{f}_{7}$ and finally ${}^{7}\bar{f}_{7}$ can be substituted into sum of forces Equation 2-58 to solve for ${}^{8}\bar{f}_{8}$ (x-y) components. These equations are then used instead of the generic dynamic equations for the inner and outer links. This is possible because the question can be reduced into 2 dimensions, as only the moment in the frame 5 z direction is relevant to determining the joint torques.

As the z direction forces and x,y direction moments of the top link are not determined, when calculating the forces acting on the base joint ($\theta_1$) a free body diagram of the entire arm must be considered. Shown in Figure 2-6. Sum of moments only needs to occur in the $1\hat{z}$ direction as that is the only degree of freedom available to joint 1. Forces and that are duplicate or that do not impact the value of torque at joint 1 are omitted.

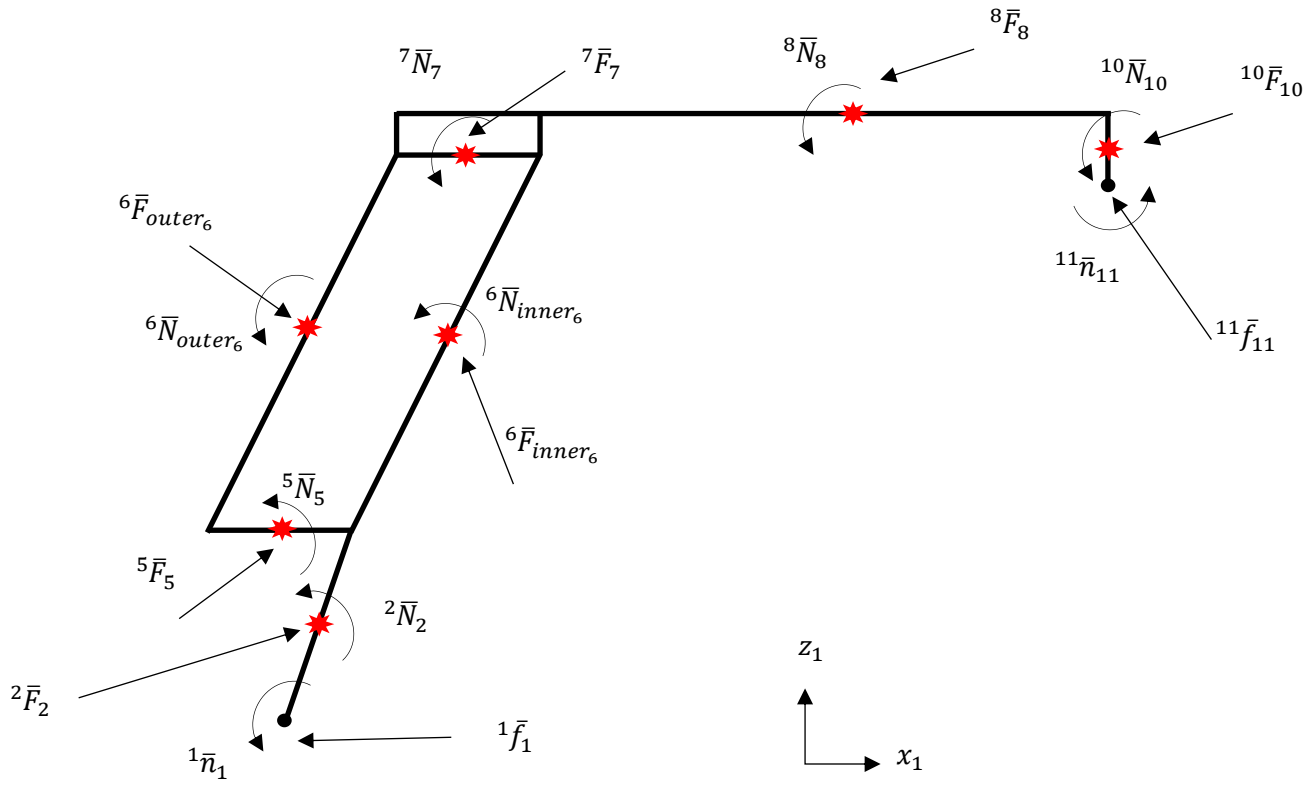



*Figure 2-6: Free Body Diagram of the Entire Arm to Solve for Joint 1 Torques*

$$\sum moments_{1z} = 0 \qquad 2.70$$

$$\begin{aligned} 0 = {}^{1}\bar{n}_1 + {}^{2}\bar{N}_2 + {}^{5}\bar{N}_5 + {}^{6}\bar{N}_{inner_6} + {}^{6}\bar{N}^{+}_{outer_6} + {}^{7}\bar{N}_7 + {}^{8}\bar{N}_8 + {}^{10}\bar{N}_{10} + {}^{11}\bar{n}_{11} \\ + ({}^{1}\bar{P}_{c_2} \times {}^{2}\bar{F}_2) + ({}^{1}\bar{P}_{c_5} \times {}^{5}\bar{F}_5) + \left({}^{1}\bar{P}_{inner_{C6}} \times {}^{6}\bar{F}_{inner_6}\right) \\ + \left({}^{1}\bar{P}_{outer_{C6}} \times {}^{6}\bar{F}_{outer_6}\right) + ({}^{1}\bar{P}_{c_7} \times {}^{7}\bar{F}_7) + ({}^{1}\bar{P}_{c_8} \times {}^{8}\bar{F}_{8}) \\ + ({}^{1}\bar{P}_{c_{10}} \times {}^{10}\bar{F}_{10}) + ({}^{1}\bar{P}_{11} \times {}^{11}\bar{f}_{11}) \end{aligned} \qquad 2.71$$

As each of the terms on the RHS of the equation aside from ${}^{1}\bar{n}_1$ are known values, ${}^{1}\bar{n}_{z1}$may be solved for. Having solved for the last unknown, the equations of motion may be solved for any trajectory of the robot to reveal the end effector position and orientation, along with the required torques at each of the joint motors required for the prescribed motion. Solving the

derived equations gives the analytical solution for the forward dynamics of the 3rd generation arm. That is; given joint parameters of: acceleration, velocity and position the required torques and forces to be produced by the motors can be specified.

The MATLAB code used for solving the above equations can be found in Appendix C

## 2.3: Comparing results

The equations derived above will be compared to a simulated model of the robot arm that was created in ANSYS workbench (more information on the ANSYS workbench model can be found in appendix F). A cubic polynomial trajectory for the robot arm to follow will be defined in the joint space and used for the comparison. MATLAB will use the forward kinematic and dynamic equations to calculate the task space position of the end effector and the forces/moments required at each of the motors using the respective equations. The ANSYS workbench model will be programmed with the joint motion of the trajectory and the position of the end effector, and forces/moments required at the joints will be measured by an inserted probe. The measured results from the simulated experiment will then be compared to the calculated solutions to the analytical equations.

The method to compare results is as follows:

1. Create a cubic polynomial trajectory function:
   Cubic polynomial trajectory generation is a method used to create a trajectory for a robot manipulator to follow. The start and end joint space parameters are specified, along with the time of travel. The cubic polynomial function then solves for the position, velocity and acceleration of each joint variable along the path.

2. Testing the forward kinematics:

   2.1 Using the cubic polynomial function the forward kinematic equations can be tested. Rather than testing only a handful of situations with the analytic forward kinematics

using the generated trajectory to specify the joint angles, the position of the end effector can be evaluated for the full duration of the trajectory.

2.2 Using a rigid dynamic simulation of the robot arm, the same trajectory can be programmed into the robot arm and the end effector can be tracked for the full duration of the trajectory.

2.3 Once the calculated end effector position results are obtained, they can be compared to the simulated end effector position results.

3. Testing the inverse kinematics:
   To test the inverse kinematics end effector positions in cartesian space will be selected and used as inputs for the analytical inverse kinematic equations, producing the joint angles required. To verify the results, the joint angles obtained will then be used as the inputs for the forward kinematic equations. If the inverse kinematic equations are correct, the forward kinematics will produce the original cartesian end effector location.

4. Testing forward dynamics
   4.1 Derive the forward analytical dynamic equations using the Newton-Euler equations for motion.
   4.2 Use the joint motion parameters created by the cubic trajectory as inputs for the derived dynamic equations, record the calculated forces and torques at each of the joints.
   4.3 Program the joint variables determined by the trajectory into the rigid body module of ANSYS workbench to simulate the trajectory. By placing probes at each of the joints where motors will be located the corresponding moments and forces required for motion can be measured.
   4.4 Compare the calculated analytical results with the measured simulated results.

Once the analytic equations have been evaluated and the results have been gathered from the rigid dynamic model, the simulated results and the analytical results can be compared. The following sections go into more detail.

## 2.4: Trajectory generation

An arbitrary cubic polynomial trajectory for the 3rd generation boom to follow was generated with the parameters listed in Table 2-4 below. This trajectory allows each of the joints to be actuated and tested. Each of the trajectories starts and finishes with zero velocity and zero acceleration of the joints. This trajectory will be used again in the following section discussing dynamic testing.

*Table 2-4: Parameters for Test Path*

| Parameter | Variable | Initial Value | Final Value |
|---|---|---|---|
| **Time** | **Time (s)** | 0 | 10 |
| **Position (θ)** | **Joint 1 (degrees)** | 0 | 90 |
| | **Joint 2 (degrees)** | 90 | 45 |
| | **Joint 3 (degrees)** | 180 | 150 |
| | **Joint 4 (m)** | 0 | 0.5 |
| | **Joint 5 (degrees)** | 0 | 90 |
| **Velocity ($\dot{\theta}$)** | $\mathbf{Joint\ 1\ (degrees/}s\mathbf{)}$ | 0 | 0 |
| | $\mathbf{Joint\ 2\ (degrees/}s\mathbf{)}$ | | |
| | $\mathbf{Joint\ 3\ (degrees/}s\mathbf{)}$ | | |
| | $\mathbf{Joint\ 4\ (m/}s\mathbf{)}$ | | |
| | $\mathbf{Joint\ 5\ (degrees/}s\mathbf{)}$ | | |
| **Acceleration ($\ddot{\theta}$)** | $\mathbf{Joint\ 1\ (degrees/}s^2\mathbf{)}$ | 0 | 0 |
| | $\mathbf{Joint\ 2\ (degrees/}s^2\mathbf{)}$ | | |
| | $\mathbf{Joint\ 3\ (degrees/}s^2\mathbf{)}$ | | |
| | $\mathbf{Joint\ 4\ (m/}s^2\mathbf{)}$ | | |
| | $\mathbf{Joint\ 5\ (degrees/}s^2\mathbf{)}$ | | |

The cubic polynomials trajectories for the joint space variables have the following forms (Craig 2005):

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 \quad 2.72$$

$$\dot{\theta}(t) = c_1 + 2c_2 t + 3c_3 t^2 \quad 2.73$$

$$\ddot{\theta}(t) = 2c_2 + 6c_3 t \quad 2.74$$

Using the initial and final values from Table 2-4, Equations 2-71 to 2-73 can be simplified and solved for the constants:

$$c_0 = \theta_{initial} \quad 2.75$$

$$c_1 = 0 \quad 2.76$$

$$c_2 = \frac{3}{t_{final}^2}(\theta_{final} - \theta_{initial}) \quad 2.77$$

$$c_3 = -\frac{2}{t_{final}^3}(\theta_{final} - \theta_{initial}) \quad 2.78$$

Each joint will have an equation for position, velocity, and speed. Table 2-5 below shows all the constants for each of the joints during the trajectory which can be placed into Equations 2-71 to 2-73 to determine the task space variables at any time during the trajectory.

*Table 2-5: Parameters for the cubic polynomial path*

| Joint | $c_0$ | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|---|
| **1** | 0 | 0 | 2.7 | -0.18 |
| **2** | 90 | 0 | -1.35 | 0.09 |
| **3** | 180 | 0 | -0.9 | 0.06 |
| **4** | 0 | 0 | 0.015 | -0.001 |
| **5** | 0 | 0 | 2.7 | -0.18 |

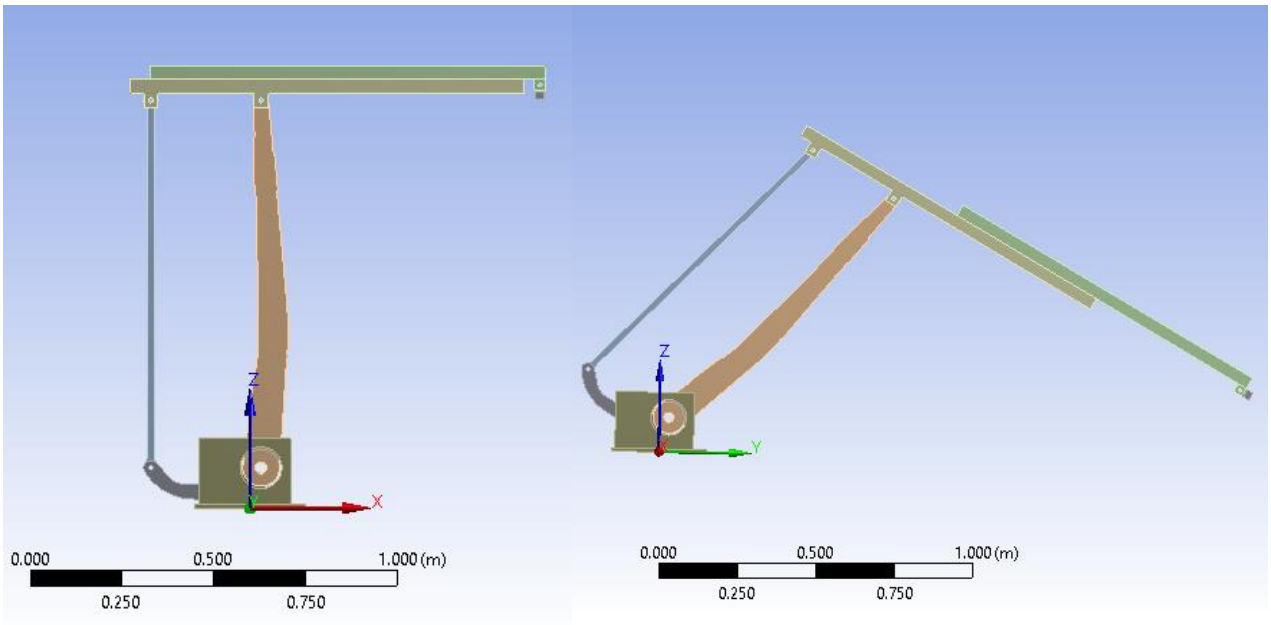



*Figure 2-7: Position of the robot arm during the trajectory. Left: initial position, Right: final position. Note the rotation about z axis*

## 2.5: Kinematic testing

Testing the forward kinematic equations is much less complex than deriving the equations, as the joint angles are specified; basic geometry can be used to calculate the position of the end effector. The accuracy of the derived forward kinematic equations is verified by comparing the position of the end effector calculated by the forward kinematic equations to the simulated position of the end effector. To calculate the position of the end effector throughout the trajectory, the analytical forward kinematic equations were evaluated at each 0.1s interval of the 10 second path.

To track the position of the end effector as it travels through the trajectory, a 3D model of the robot arm was imported into the rigid dynamics module of ANSYS workbench. The rigid dynamics module uses a Runge- Kutta differential equation solver (Ansys Support 2011) to simulate the motion of the robotic arm, however in this scenario only basic trigonometric functions are required to solve the forward kinematic equations. After inputting the trajectory for the robot arm to follow, the cartesian position of the end effector is tracked and recorded at every 0.1s of the path.

Below, Figure 2-8, Figure 2-9, and Figure 2-10 show the results of the two methods.

The following inputs were used during the experiment. These values were obtained by inspecting and measuring the SolidWorks model of the 3rd generation boom. Refer to Figure 2-1 at the beginning of the section.

$L_1 = 110\ mm$

$L_2 = 30\ mm$

$L_3 = 300\ mm$

$L_4 = 1000\ mm$

$L_5 = 300\ mm$

$L_6 = 42\ mm$

$L_7 = 760\ mm$

$L_8 = 40\ mm$

$L_9 = 1000\ mm$

## 2.6: Kinematic results

When comparing the position of the end effector relative to frame 0, the following results were calculated

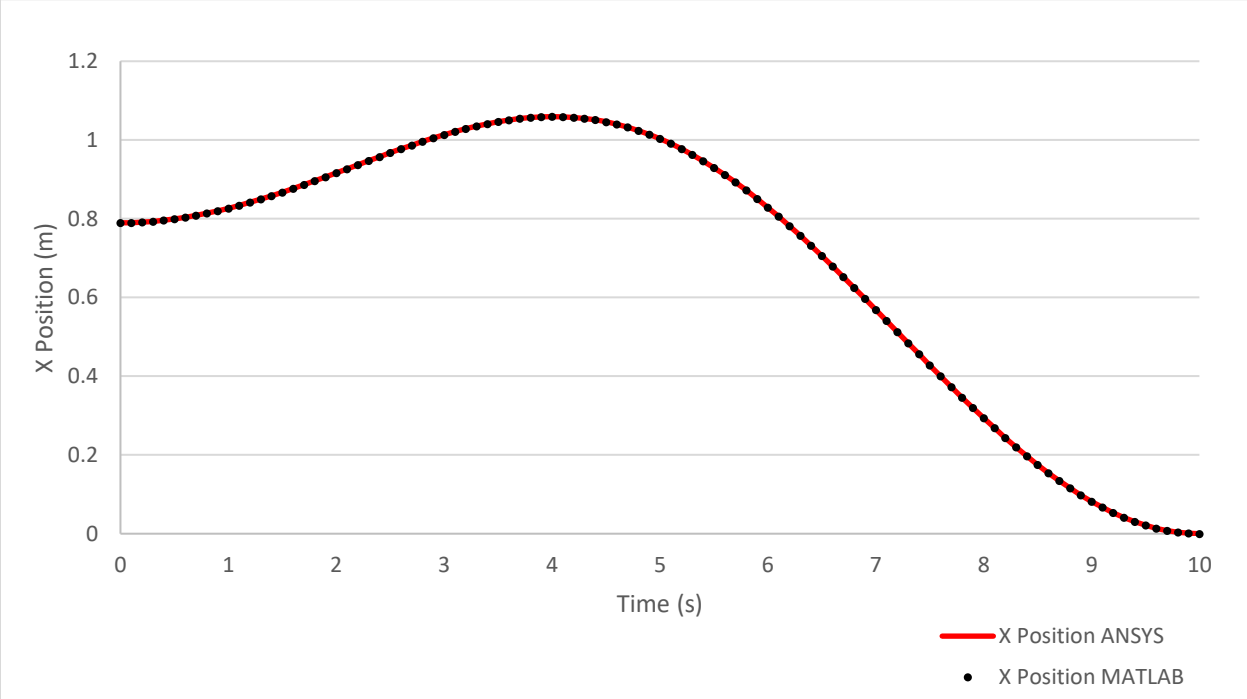



*Figure 2-8: X position of the end effector while traveling*

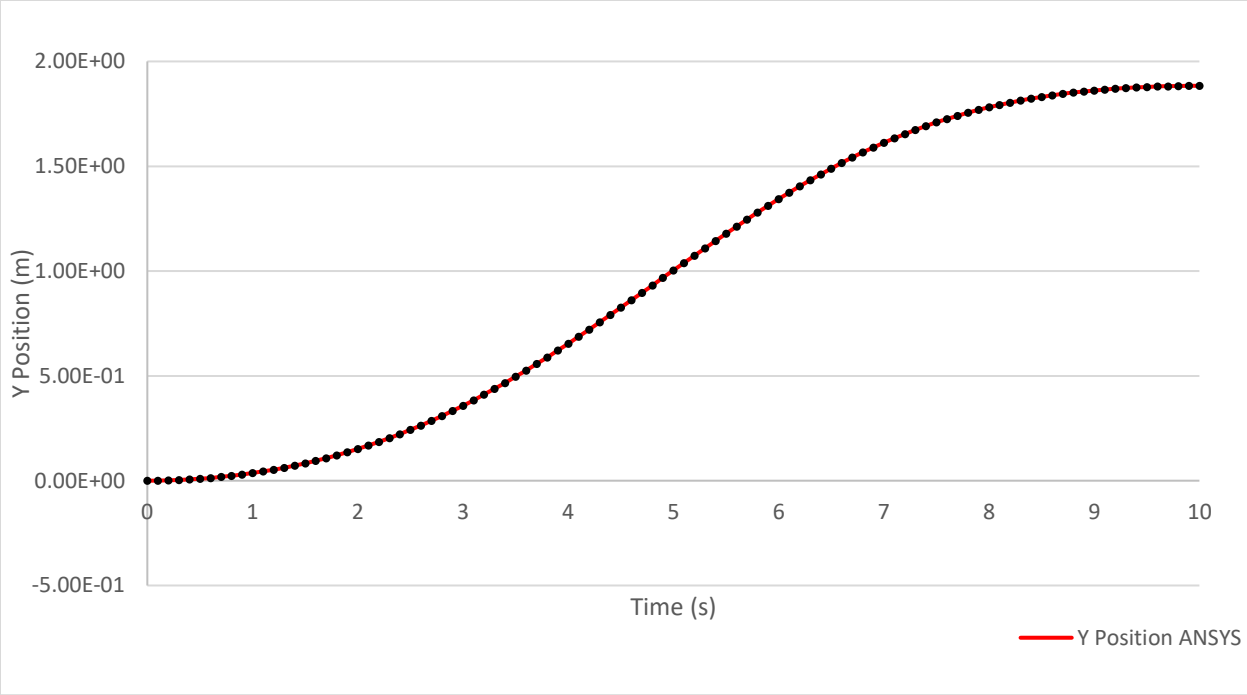



*Figure 2-9: Y position of the end effector while traveling*

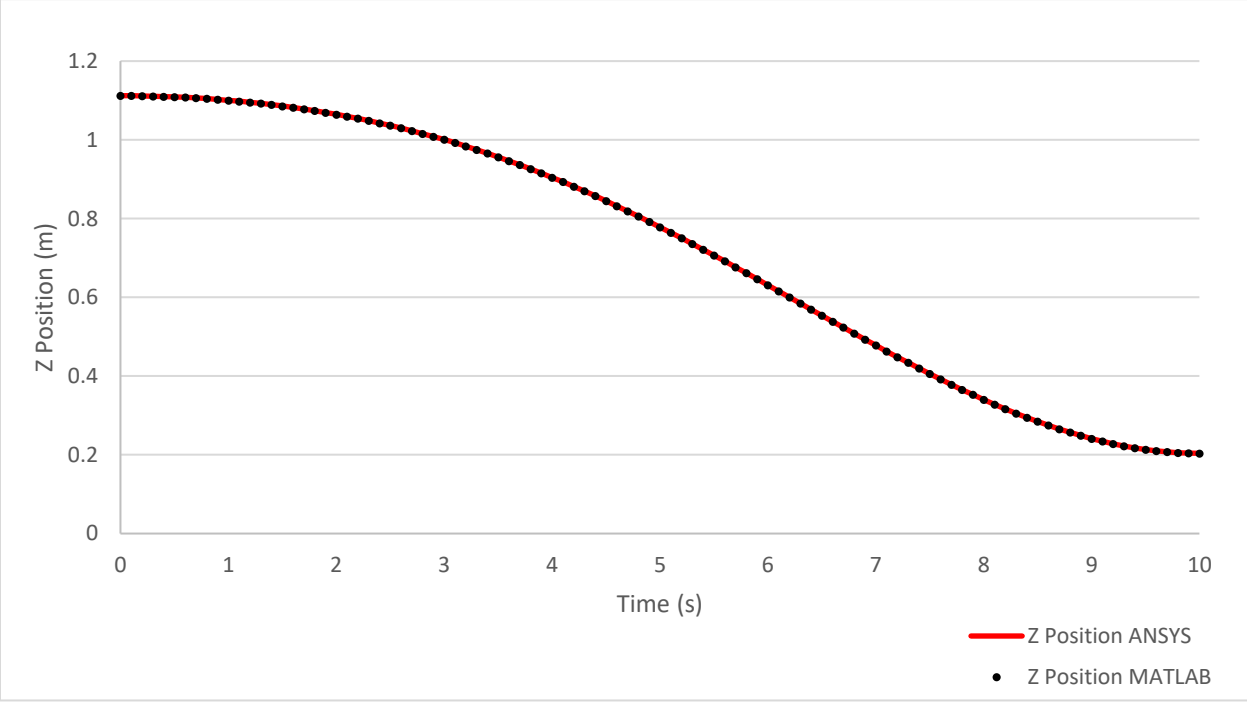



*Figure 2-10: Z position of the end effector while traveling*

*Table 2-6: Position of end effector while traveling*

| | Ansys Workbench Position (measured) | | | MATLAB Position (Calculated) | | |
|---|---|---|---|---|---|---|
| Time (s) | x (m) | y (m) | z (m) | x (m) | y (m) | z (m) |
| 0.0 | 0.790 | 0.000 | 1.112 | 0.790 | 0.000 | 1.112 |
| 1.0 | 0.826 | 0.036 | 1.101 | 0.826 | 0.036 | 1.101 |
| 2.0 | 0.916 | 0.151 | 1.065 | 0.916 | 0.151 | 1.065 |
| 3.0 | 1.013 | 0.358 | 1.001 | 1.013 | 0.358 | 1.001 |
| 4.0 | 1.059 | 0.654 | 0.905 | 1.059 | 0.654 | 0.905 |
| 5.0 | 1.003 | 1.003 | 0.779 | 1.003 | 1.003 | 0.779 |
| 6.0 | 0.829 | 1.343 | 0.631 | 0.829 | 1.343 | 0.631 |
| 7.0 | 0.569 | 1.612 | 0.478 | 0.569 | 1.612 | 0.478 |
| 8.0 | 0.294 | 1.781 | 0.340 | 0.294 | 1.781 | 0.340 |
| 9.0 | 0.082 | 1.862 | 0.241 | 0.082 | 1.862 | 0.241 |
| 10.0 | 0.000 | 1.884 | 0.204 | 0.000 | 1.884 | 0.204 |

*Table 2.6: Comparison of ANSYS and MATLAB positions*

| | % difference | | | Absolute difference (mm) | | |
|---|---|---|---|---|---|---|
| Time (s) | x | y | Z | x | y | Z |
| 0.0 | 0.00 | 0.00 | 0.00 | 0.005 | 0.000 | 0.020 |
| 1.0 | 0.00 | 0.00 | 0.00 | 0.001 | 0.000 | 0.011 |
| 2.0 | 0.00 | 0.00 | 0.00 | 0.002 | 0.004 | 0.048 |
| 3.0 | 0.00 | 0.00 | 0.00 | 0.042 | 0.000 | 0.029 |
| 4.0 | 0.00 | 0.00 | 0.00 | 0.035 | 0.005 | 0.004 |
| 5.0 | 0.00 | 0.00 | 0.00 | 0.041 | 0.041 | 0.006 |
| 6.0 | 0.00 | 0.00 | 0.00 | 0.003 | 0.047 | 0.003 |
| 7.0 | 0.00 | 0.00 | 0.00 | 0.002 | 0.021 | 0.004 |
| 8.0 | 0.00 | 0.00 | 0.00 | 0.003 | 0.047 | 0.008 |
| 9.0 | 0.00 | 0.00 | 0.00 | 0.000 | 0.019 | 0.002 |
| 10.0 | 0.00 | 0.00 | 0.00 | 0.000 | 0.051 | 0.007 |

The % difference is calculated according to Equation 2-78 below. Which compares the absolute value of the difference between the two types of results to the average of the two types of results.

$$\%\, difference\ = |\frac{(Simulated\ result\ -\ Analytical\ result)}{((simulated\ result\ +\ Analytical\ result)/2)}| \qquad 2.79$$

When comparing the cartesian position of the end effector the above results were available. As the method used compares the results at a time interval of 0.1s over a 10s interval, 100 position comparisons were made. Showing the complete data set would decrease the clarity of the results, so only 11 data points are presented. Maximum differences between the measured ANSYS workbench position and the calculated MATLAB position were found at 10.0 seconds in the y direction.

## 2.7: Inverse kinematic testing and results

The inverse kinematic equations developed earlier in the chapter were tested by specifying a cartesian coordinate position for the end effector. MATLAB was then used to solve the inverse kinematic equations, converting the cartesian space coordinates of the end effector into joint space . Once the values for the joint angles are known they are used as the joint angles in the ANSYS rigid dynamic simulation, and the position of the end effector is recorded. Additional constraints were needed such as a horizontal linear actuator, position of the end effector, or length of the prismatic joint in order to solve for all 5 degrees of freedom. The results of the inverse kinematic testing are found on the next page in Table 2-7. The inverse kinematic equations agreed to millimetre precision. Following the table, Figures 2-11, 2-12 and 2-13 visually show the position of the robot arm in the tested positions.

*Table 2-7: results of inverse kinematic equation testing*

| Test # | x,y,z Desired position (m) | Extra constraints | Inverse kinematic solution | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Joint 1 (Rad) | Joint 2 (Rad) | Joint 3 (Rad) | Joint 4 (m) | Joint 5 (Rad) | Simulated position, given joint positions. (m) |
| 1 | $\begin{vmatrix}1.40\\0.00\\1.00\end{vmatrix}$ | $\theta_4$=0, end effector facing downward | 0 | 0.897 | 3.283 | 0 | -0.141 | $\begin{vmatrix}1.40\\0.00\\1.00\end{vmatrix}$ |
| 2 | $\begin{vmatrix}1.40\\0.00\\1.00\end{vmatrix}$ | Linear actuator Horizontal, end effector facing downward | 0 | 1.093 | 3.141 | 0.151 | 0 | $\begin{vmatrix}1.40\\0.00\\1.00\end{vmatrix}$ |
| 3 | $\begin{vmatrix}1.40\\0.00\\1.00\end{vmatrix}$ | $\theta_4$=0.25, end effector facing downward | 0 | 1.203 | 3.097 | 0.250 | 0.045 | $\begin{vmatrix}1.40\\0.00\\1.00\end{vmatrix}$ |
| 4 | $\begin{vmatrix}2.00\\1.00\\1.00\end{vmatrix}$ | Linear actuator horizontal, end effector facing downwards | 0.464 | 1.093 | 3.141 | 0.987 | 0 | $\begin{vmatrix}2.00\\1.00\\1.00\end{vmatrix}$ |
| 5 | $\begin{vmatrix}0.00\\2.50\\0.80\end{vmatrix}$ | Linear actuator horizontal, end effector facing upwards | 0.785 | 0.653 | 3.141 | 0.916 | 3.141 | $\begin{vmatrix}0.00\\2.50\\0.80\end{vmatrix}$ |

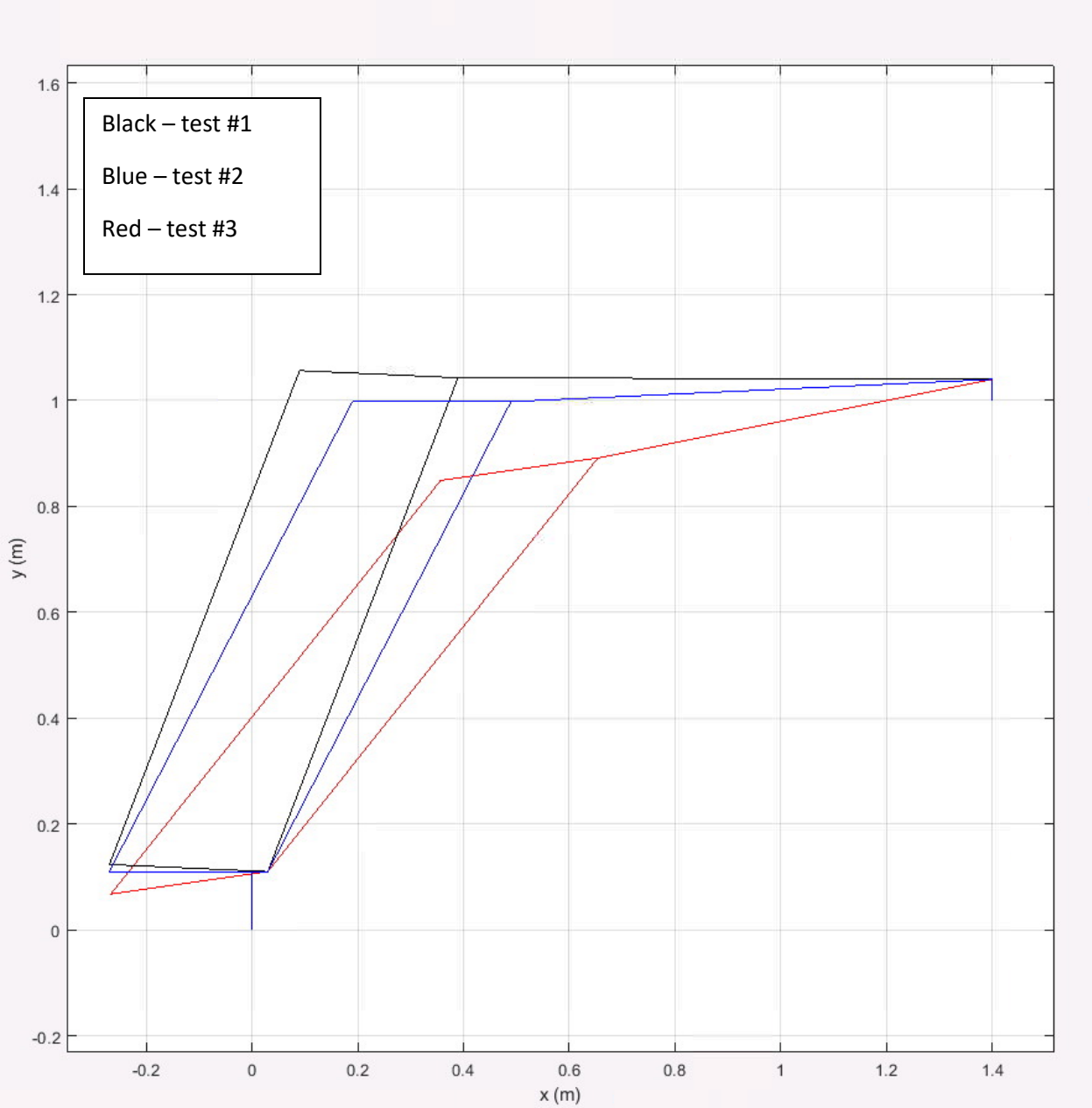



*Figure 2-11: Results of 1,2 and 3 inverse kinematic tests, highlighting the property of the robot to have multiple solutions.*

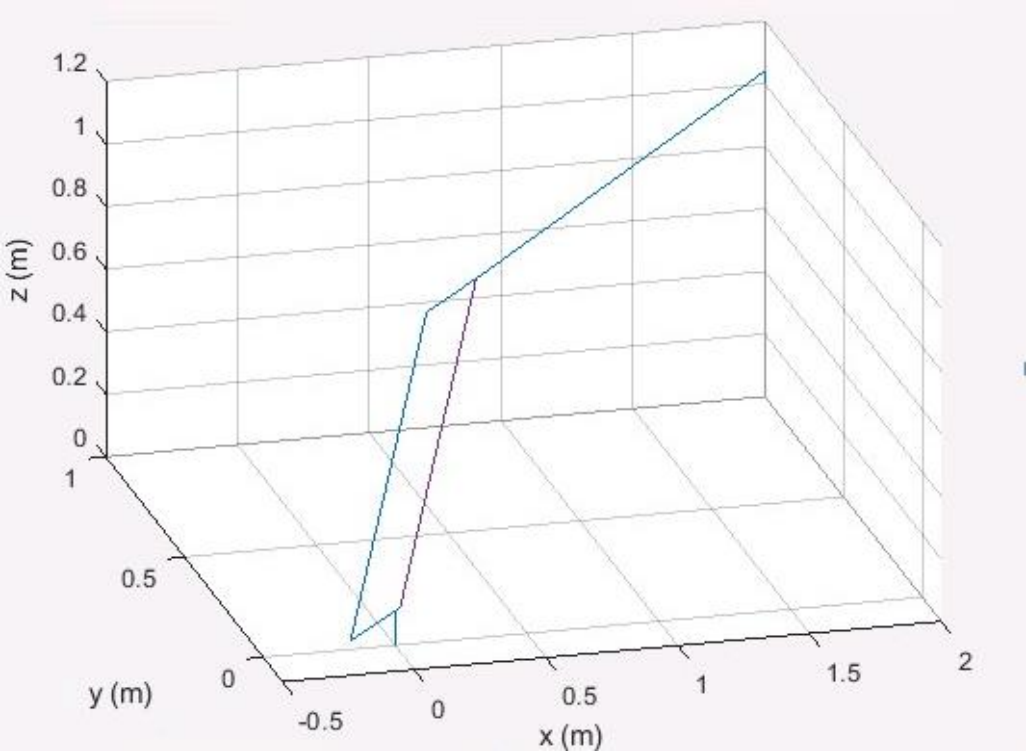



*Figure 2-12: test # 4 showing the 3d position of the arm.*

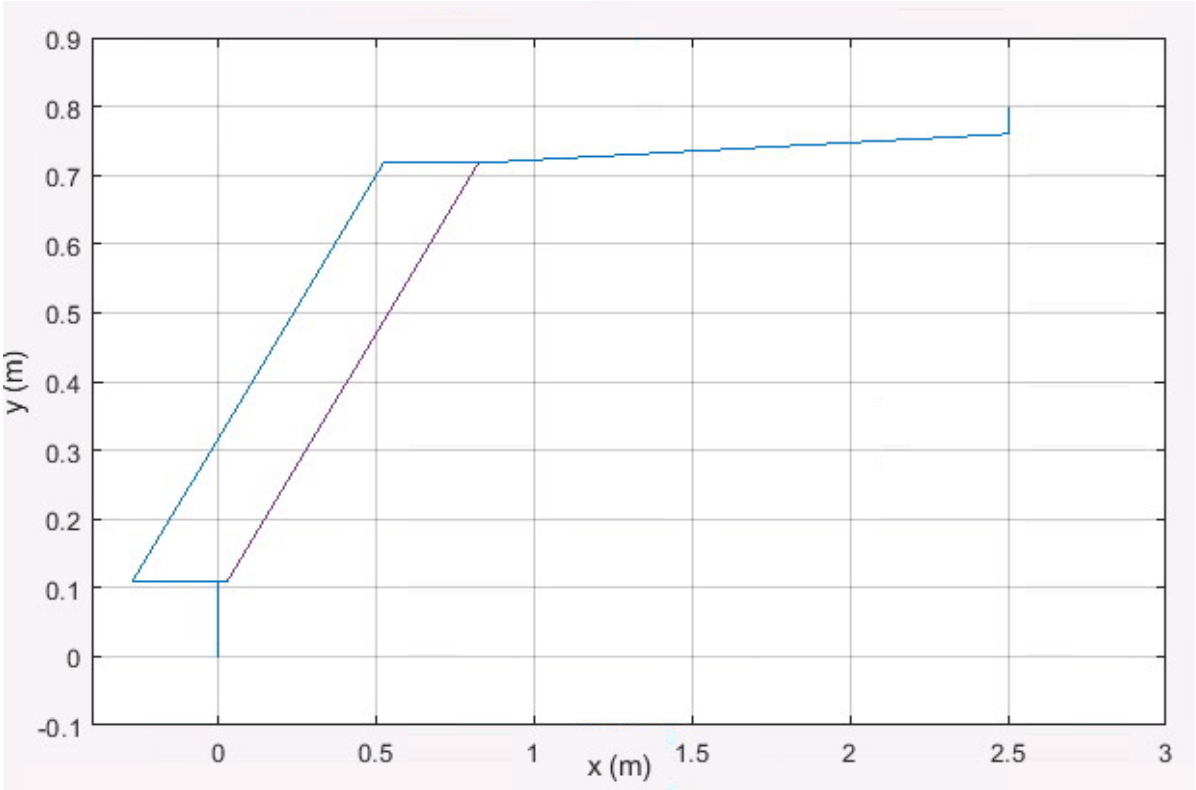



*Figure 2-13: Test# 5 solution, showing the end effector in a non-downward position.*

While these results are promising the accuracy of a physical robot is expected to be lower due to several contributing factors. Inaccuracies in the position of the end effector can be increased by imperfect joints; where the backlash of the motors will not allow precise position values, fabrication errors; if the links do not match the theoretical values listed above the position the end effector will not match the simulated values and difficulties of measuring millimetre precision of the end effector on a physical model. Additionally manufacturing errors may cause the D-H parameters to be incorrect contributing to errors in the kinematic equations. Experimental results were not completed as the robotic arm has not yet been fabricated.

## 2.8: Dynamics testing

To test the dynamics component of the MATLAB code the joint torques were compared with the joint torques in ANSYS workbench rigid dynamic module when the robot followed the same trajectory as decribed above.

In addition to the lengths of the robotic links specified earlier, the following inputs were used during the experiment. These values were obtained by inspecting and measuring the SolidWorks model of the 3rd generation boom.

$mass_{Base\ link} = 28.404\ kg$

$mass_{Lower\ arm} = 2.494\ kg$

$mass_{Inner\ link} = 10.902\ kg$

$mass_{Outer\ link} = 3.880\ kg$

$mass_{Upper\ link} = 2.978\ kg$

$mass_{Extension\ arm} = 2.125\ kg$

$mass_{End\ effector} = 0.783\ kg$

The location of the center of gravity for each link, ${}^{i}P_{c}$ relative to frame i is listed below. The location of centers of mass relative to other frames can be found using the transformation discussed earlier.

${}^{1}P_{C\ Base\ link} = [5.6; 0; 69.4]\ mm$

${}^{5}P_{C\ Lower\ link} = [115.7; 37.7; 0]\ mm$

${}^{5}P_{C\ Inner\ link} = [407.3; -10.3; 0]\ mm$

${}^{6}P_{C\ Outer\ link} = [500.0; 0; 0]\ mm$

${}^{8}P_{C\ Upper\ link} = [33.1; -137.3; 0]\ mm$

${}^{9}P_{C\ Extension\ arm} = [-15.1; 0; -590.4]\ mm$

${}^{10}P_{C\ End\ effector} = [8.2; 0; 75.1]\ mm$

The mass moment of inertia matrix , ${}^{i}I$ , measured relative relative to the location and orientation of frame i .

${}^{1}I_{C\ Base\ link} = [0.587,0,0; 0,0.251,0; 0,0,0.593]\ kg \cdot m^2$

${}^{5}I_{C\ Lower\ link} = [0.008,0,0; 0,0.031,0; 0,0,0.029]\ kg \cdot m^2$

${}^{6}I_{Outer\ link} = [0.001,0,0; 0,0.334,0; 0,0,0.334]\ kg \cdot m^2$

${}^{5}I_{Inner\ link} = [0.034,0,0; 0,0.964,0; 0,0,0.954]\ kg \cdot m^2$

${}^{8}I_{Upper\ link} = [0.468,0,0; 0,0.003,0; 0,0,0.465]\ kg \cdot m^2$

${}^{9}I_{Extension\ link} = [0.258,0,0; 0,0.255,0; 0,0,0.003]\ kg \cdot m^2$

${}^{10}I_{End\ effector} = [0.014,0,0; 0,0.014,0; 0,0,0]\ kg \cdot m^2$

The boundery conditions for the analysis are given below.

$$^{1}f_{11} = \begin{bmatrix} 0 \\ 0 \\ 9.81 * 10 \end{bmatrix} N$$

$$^{1}n_{11} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} Nm$$

$$^{1}\omega_{1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \frac{rad}{s}$$

$$^{1}\dot{\omega}_{1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \frac{rad}{s^2}$$

$$^{1}a_{1} = \begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix} \frac{m}{s^2}$$

$$^{1}v_{1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \frac{m}{s^2}$$

The explicit acceleration formulas (rads/s$^2$ except for $\ddot{\theta}_4$, which is in m/s$^2$) used in the rigid dynamics simulation are as follows

$$\ddot{\theta}_1 = 0.094 + (-0.018) * (time) \qquad 2.79$$

$$\ddot{\theta}_2 = -0.048 + (0.012) * (time) \qquad 2.80$$

$$\ddot{\theta}_3 = -0.032 + (0.006) * (time) \qquad 2.81$$

$$\ddot{\theta}_4 = 0.03 + (-0.006) * (time) \qquad 2.82$$

$$\ddot{\theta}_5 = 0.094 + (-0.018) * (time) \qquad 2.83$$

The values of angular acceleration, angular velocity, linear acceleration and linear velocity refer to the movement of the base of the robot. By having an acceleration of 9.81 $m/s^2$ in the positive z direction of frame 1 the robot experiences gravitational acceleration. All of the other parameters are chosen to be 0 for this simulation however this leaves the analysis open to the option of mounting the robot on a mobile vehicle which could experience it's own accelerations and velocities.

Following the methods described for the kinematic testing, the Newton-Euler equations developed above were evaluated for the test trajectory by programming the analytical equations into MATLAB. The test trajectory was also entered into ANSYS workbench using the rigid dynamic program to obtain simulated results of the forces and torques required for the robot to follow the test trajectory. Results were obtained for every 0.1s over the 10s trajectory duration and compared with eachother. The following charts compare the analytical and experimental results.

## 2.9: Dynamics results

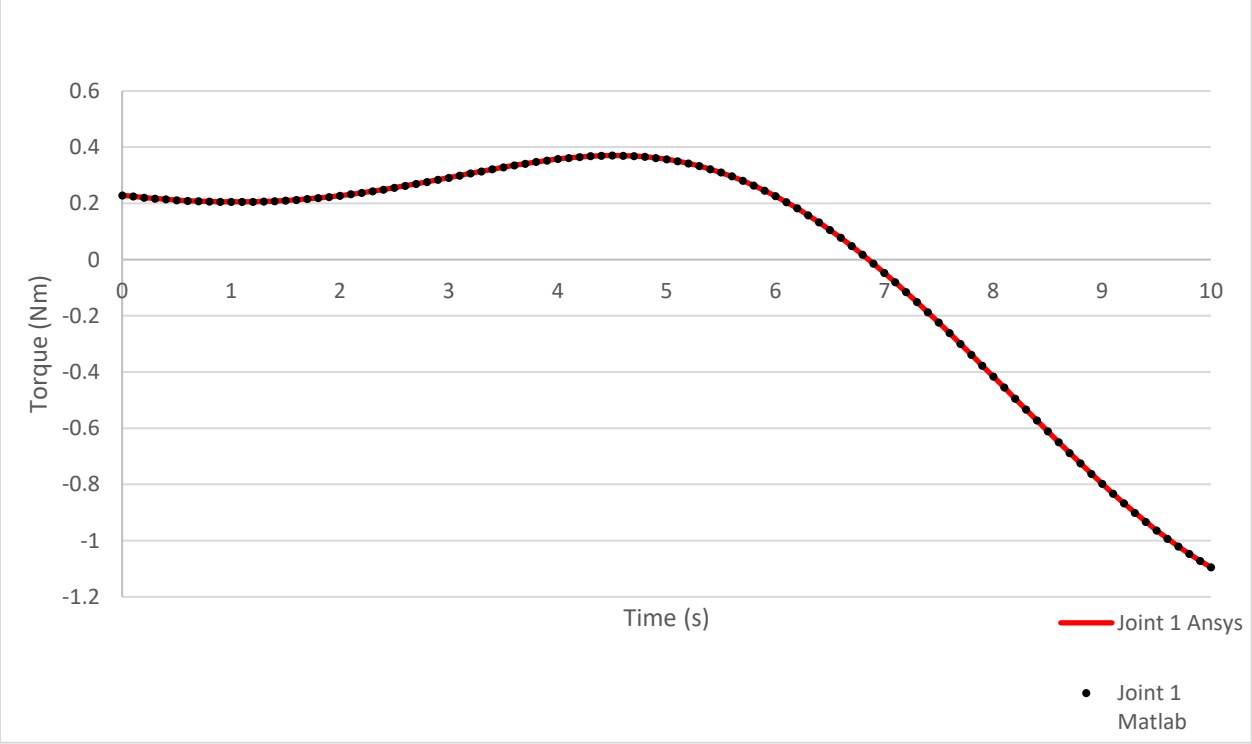



*Figure 2-14: MATLAB computed torque requirements of joint 1 compared with ANSYS simulation*

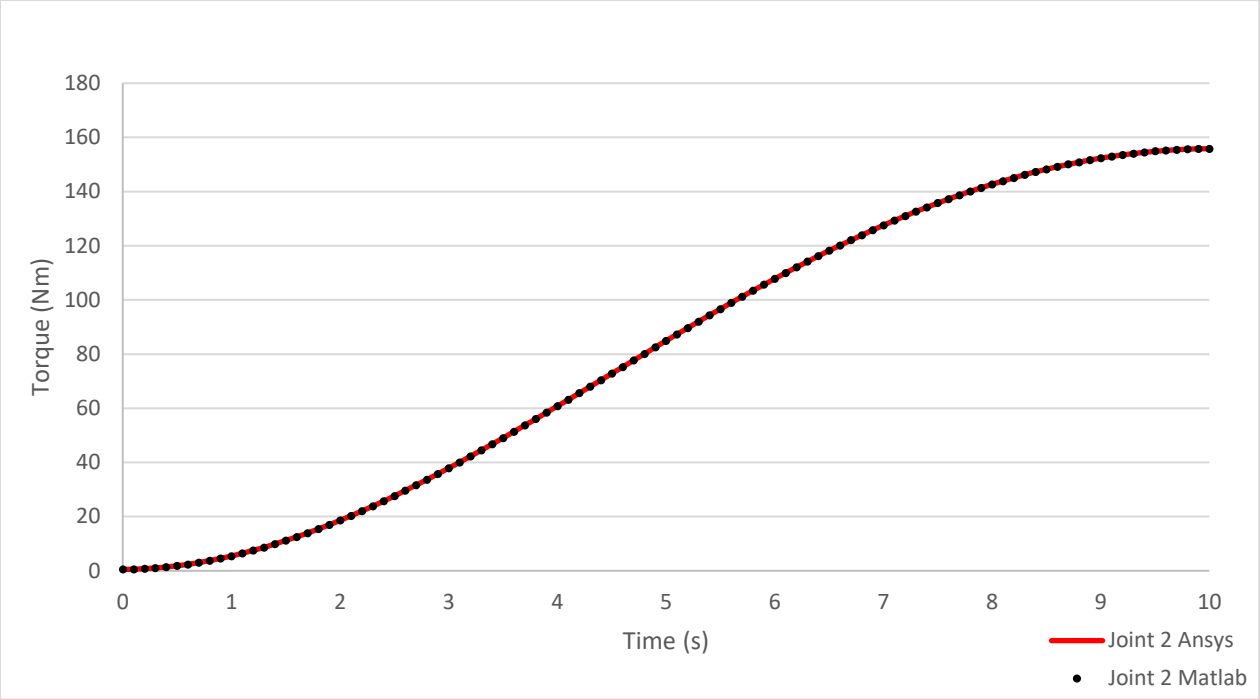



*Figure 2-15: MATLAB computed torque requirements of joint 2 compared with ANSYS simulation*

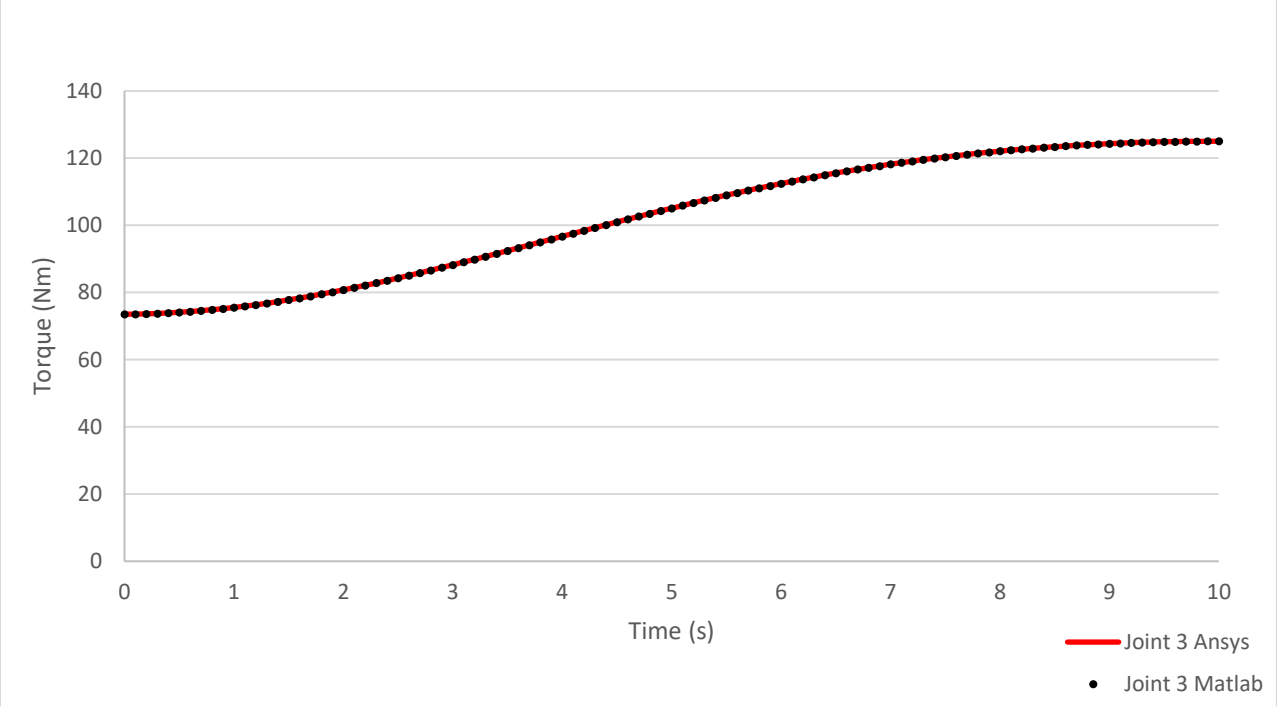



*Figure 2-16: MATLAB computed torque requirements of joint 3 compared with ANSYS simulation*

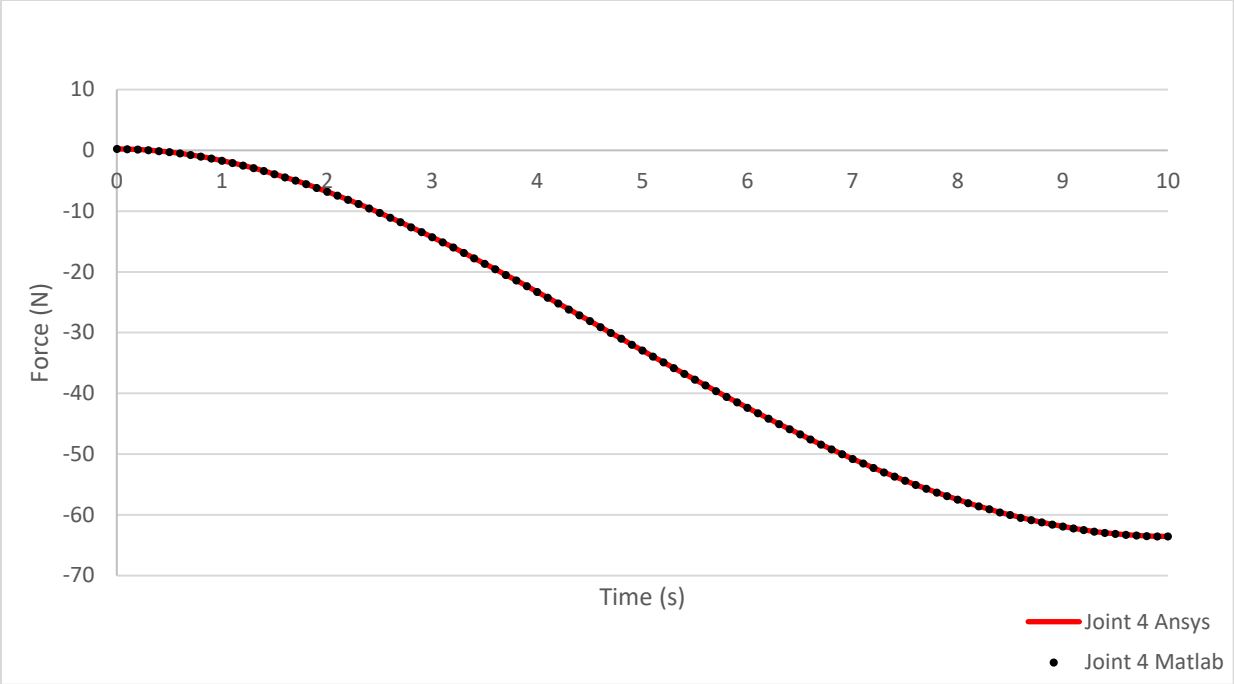



*Figure 2-17: MATLAB computed force requirements of joint 4 compared with ANSYS simulation*

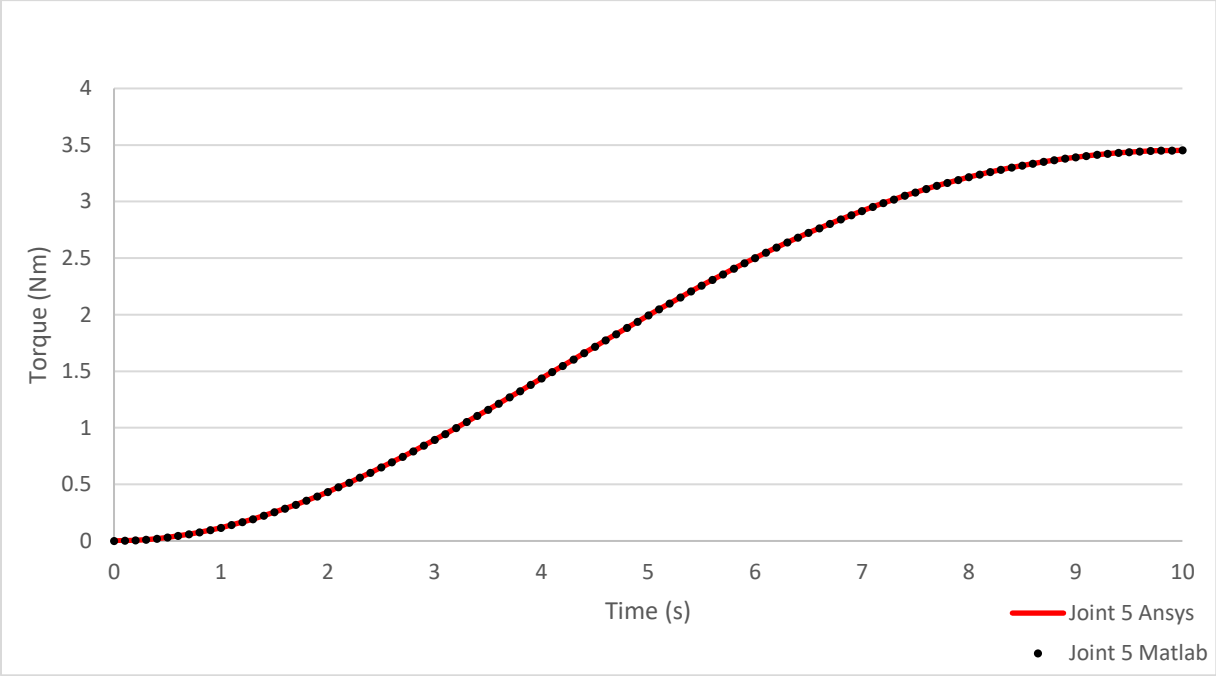



*Figure 2-18: MATLAB computed torque requirements of joint 5 compared with ANSYS simulation*

*Table 2-8: Select joint torque results*

| | ANSYS Measured Joint Torques (Nm) | | | | | MATLAB Calculated Joint Torques (Nm) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Time (s) | Joint 1 (Nm) | Joint 2 (Nm) | Joint 3 (Nm) | Joint 4 (N) | Joint 5 (Nm) | Joint 1 (Nm) | Joint 2 (Nm) | Joint 3 (Nm) | Joint 4 (N) | Joint 5 (Nm) |
| 0.0 | 0.229 | 0.527 | 73.564 | 0.235 | 0.001 | 0.229 | 0.527 | 73.563 | 0.235 | 0.001 |
| 1.0 | 0.206 | 5.454 | 75.539 | -1.677 | 0.117 | 0.206 | 5.455 | 75.539 | -1.678 | 0.117 |
| 2.0 | 0.228 | 18.623 | 80.787 | -6.787 | 0.433 | 0.228 | 18.626 | 80.788 | -6.788 | 0.434 |
| 3.0 | 0.292 | 37.856 | 88.230 | -14.277 | 0.894 | 0.292 | 37.863 | 88.232 | -14.278 | 0.894 |
| 4.0 | 0.358 | 60.781 | 96.710 | -23.291 | 1.436 | 0.358 | 60.792 | 96.713 | -23.293 | 1.436 |
| 5.0 | 0.357 | 84.897 | 105.100 | -32.945 | 1.993 | 0.357 | 84.912 | 105.101 | -32.947 | 1.993 |
| 6.0 | 0.226 | 107.810 | 112.450 | -42.372 | 2.502 | 0.226 | 107.832 | 112.459 | -42.375 | 2.502 |
| 7.0 | -0.046 | 127.540 | 118.180 | -50.779 | 2.917 | -0.046 | 127.561 | 118.187 | -50.782 | 2.917 |
| 8.0 | -0.416 | 142.670 | 122.080 | -57.485 | 3.215 | -0.416 | 142.694 | 122.093 | -57.489 | 3.215 |
| 9.0 | -0.797 | 152.320 | 124.300 | -61.916 | 3.393 | -0.798 | 152.347 | 124.308 | -61.921 | 3.393 |
| 10.0 | -1.094 | 155.810 | 125.040 | -63.547 | 3.453 | -1.095 | 155.832 | 125.051 | -63.551 | 3.453 |

*Table 2-9: Joint torque comparison*

| % difference | | | | | | Absolute difference | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Time (s)** | **Joint 1** | **Joint 2** | **Joint 3** | **Joint 4** | **Joint 5** | **Joint 1 (Nm)** | **Joint 2 (Nm)** | **Joint 3 (Nm)** | **Joint 4 (N)** | **Joint 5 (Nm)** |
| 0.0 | 0.04 | 0.07 | 0.00 | 0.00 | 0.23 | 8.81E-05 | 3.64E-04 | 8.48E-04 | 2.58E-06 | 1.34E-06 |
| 1.0 | 0.01 | 0.02 | 0.00 | 0.01 | 0.01 | 2.91E-05 | 1.25E-03 | 8.93E-05 | 1.60E-04 | 5.89E-06 |
| 2.0 | 0.03 | 0.02 | 0.00 | 0.01 | 0.01 | 7.30E-05 | 3.21E-03 | 5.24E-04 | 4.95E-04 | 2.26E-05 |
| 3.0 | 0.06 | 0.0 | 0.00 | 0.01 | 0.00 | 1.66E-04 | 6.89E-03 | 1.72E-03 | 1.06E-03 | 3.60E-05 |
| 4.0 | 0.06 | 0.02 | 0.0 | 0.01 | 0.0 | 2.07E-04 | 1.13E-02 | 2.57E-03 | 1.69E-03 | 4.40E-05 |
| 5.0 | 0.05 | 0.02 | 0.00 | 0.01 | 0.00 | 1.78E-04 | 1.54E-02 | 1.31E-03 | 2.36E-03 | 5.16E-05 |
| 6.0 | 0.04 | 0.02 | 0.01 | 0.01 | 0.01 | 8.60E-05 | 2.19E-02 | 9.03E-03 | 2.81E-03 | 1.54E-04 |
| 7.0 | 0.08 | 0.02 | 0.0 | 0.01 | 0.00 | 3.86E-05 | 2.14E-02 | 7.35E-03 | 3.10E-03 | 1.25E-04 |
| 8.0 | 0.04 | 0.02 | 0.01 | 0.01 | 0.01 | 1.63E-04 | 2.39E-02 | 1.28E-02 | 3.78E-03 | 1.92E-04 |
| 9.0 | 0.03 | 0.02 | 0.01 | 0.01 | 0.0 | 2.68E-04 | 2.66E-02 | 7.78E-03 | 4.79E-03 | 1.54E-04 |
| 10.0 | 0.03 | 0.01 | 0.01 | 0.01 | 0.00 | 3.50E-04 | 2.22E-02 | 1.06E-02 | 4.18E-03 | 1.26E-04 |

Figures 2-14 to 2-18 above show the comparison between the results of the analytical dynamic solution found using the Newton-Euler equations and the simulated measurements. Tables 2-8 and 2-9 respectively show the torque required at points during the trajectory and the relative difference between the analytical and simulated results. The maximum difference between the two results is 0.23% in joint 5 shows that the two methods agree with each other. However these results do not show that the two methods are correct. Physical experiments need to be performed with the robot arm to prove that these derived equations and simulated model are indeed correct. The tables above show individual results at 1 second intervals, as the volume of data collected during the experiments would impair clarity of the results.

The maximum torque experienced by the robot arm during this trajectory is in joint 2, which reaches a maximum of 155.8 Nm. This is quite a high level of torque, and can be attributed to the fact that the end effector has a payload of 10kg, and the extreme reach of the robot means that the end effector can be extended up to 2.5m from the base. In this specific trajectory the end effector is 2.08m away from the joint 2 motor at t=10 seconds.

## 2.10: Discussion and Summary

In this chapter the analytical equations of motion of the robotic arm are derived. In the first half, D-H parameters are used to create coordinate transformations between the different coordinate frames of the robot. These transformations, or forward kinematic equations, convert the joint space parameters of the robot to cartesian space or task space coordinates.

The inverse kinematic equations are derived from the forward kinematic equations. These equations convert the task space coordinates of the end effector into the joint space to determine the position of each of the degrees of freedom.

The dynamic equations of the robot arm are derived by applying the Newton-Euler equations of motion to the individual links of the robot. First the task space positions, velocities and accelerations are determined from the joint space parameters. Once the cartesian space velocities and accelerations are known for each of the robot links, along with the mass properties, the Newton-Euler equations of motion can be applied to determine the inertial forces and torques acting on each link. Using the force and moment equilibrium equations, the joint forces supplied by each of the joint motors can be calculated.

The analytical equations of motion were then compared to an ANSYS rigid body dynamics simulation. A cubic polynomial trajectory was created to specify the joint space parameters of position, velocity and acceleration. Using the joint space trajectory as the inputs, the analytical equations were solved using MATLAB to determine the position of the end effector relative to the base of the robot, and the forces required at each of the motors to create the motion.

The results of the forward kinematic analytical equations and simulated results agreed within millimetre accuracy showing that the analytical equations and the simulation agree with each other.

When the analytical dynamic equation results and the simulated results were compared the difference between the two sets of results was a maximum of 0.23% in joint 1 and otherwise less than 0.1% error which may be attributed to rounding errors. The results of the dynamic simulation agreed well with each other.

While the analytical and simulated results agree with each other for both the kinematic and dynamic equations these results do not show that the equations derived are correct. Physical experiments must be performed and compared to the equation results to prove the analytical equations do accurately reflect the motion of the robot arm.

Additionally, there may be some difference between the analytical and simulation results compared to the results obtained from physical experiments. Backlash from the motors and gearboxes will reduce the accuracy of the end effector, manufacturing defects will affect the alignment of the degrees of freedom affecting the motion and end effector position. Manufacturing error can also affect the center of mass values and symmetries that existed in the analytical solution may not exist in the physical robotic arm. Friction forces were neglected during the analysis thus the dynamic forces will be different than the values projected from the analytical and simulated experiments.

Objective 1 of this thesis states that appropriate methods of analysis of a 5 DOF parallel robotic arm must be identified. The transformation matrix from the base of the robot to the end effector allows the data collected by the end effector to be geotagged accurately despite the GPS unit not being attached to the end effector. These ability for the arm to follow a path is critical for the operation of the robot arm and will allow complex tasks to be programmed into the arm. The results of this chapter show that the kinematic and dynamic equations that were derived are accurate and can be used to position the robot and to have the robot follow an arbitrary trajectory thus the Newton – Euler equations of motion are an appropriate method for the robotic arm.

# Chapter 3: A gradient optimization technique of a cantilever beam

## 3.1: Introduction

The original weight of the robotic arm is approximately 50kg. Due to the payload of 10kg, and the extreme reach of the robot (2.5m) over half of the weight (28.4kg) is contributed by the base plate, motors and gearboxes for joints 1,2 and 3. Keeping the weight of the robotic arm as small as possible reduces the power requirements from the motors and reduces the risk of soil compaction when the arm is collecting data in a field. However, by reducing the mass of the robot arm it may become weaker and increase the deflection under loading. In this chapter the inner link is subjected to a gradient optimization technique to reduce the weight of the link while ensuring that maximum stress and maximum deflection constraints are not violated.

Performing the mass optimization of the inner link will have two effects on the mass of the overall robot arm. First and most directly the mass of the individual link will be reduced. Secondly, as the masses of the links are reduced the motors required to actuate the joints can become smaller as well. The actual motors for joints 2 and 3 the 5 DOF robot arm are MST342C02 which weigh about 3.8kg. The motors are paired with a CBG320160 harmonic gear which weighs 3.3kg. Combined, both motors and gearboxes contribute 14.2 kg to the arm. By reducing the torque requirements these motors and gearboxes may be replaced with lighter models.

The datasheet for the motor and gearbox models can be found in Appendix E.

The original inner link is made of structural steel and has a mass of 10.9 kg. The effective length is 1 meter and has connections at the top and bottom to connect to the base and the upper links. The inner link is constructed of 0.25in thick steel plates that were assembled through welding. Figure 3-2 shows the link as originally designed.

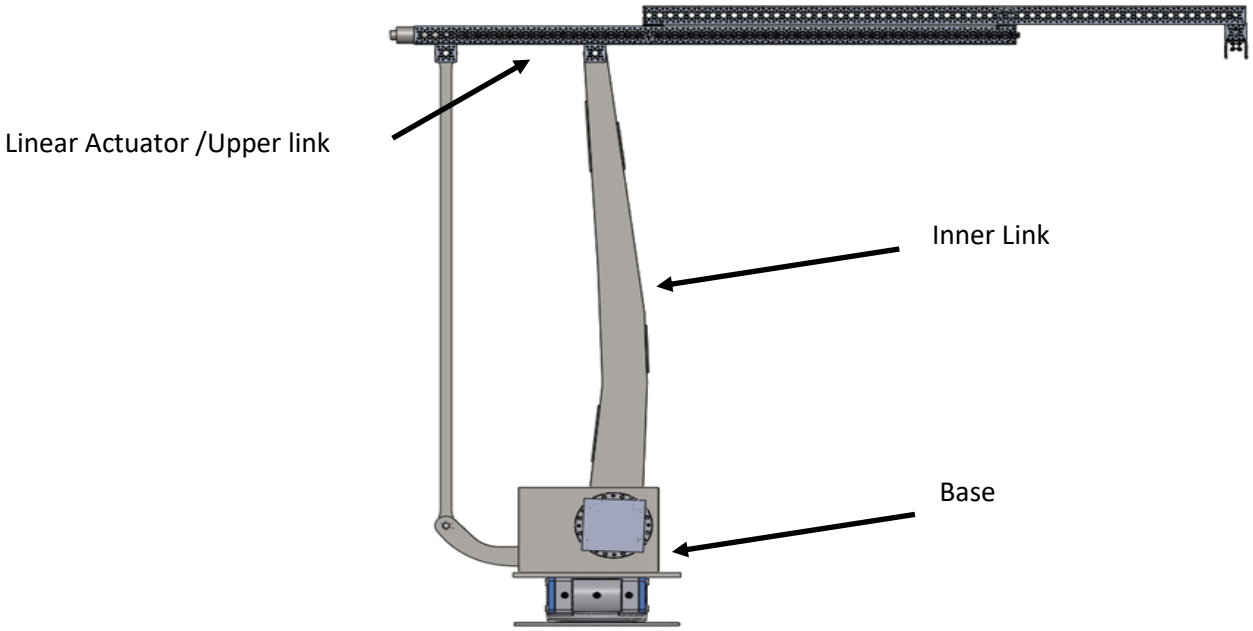
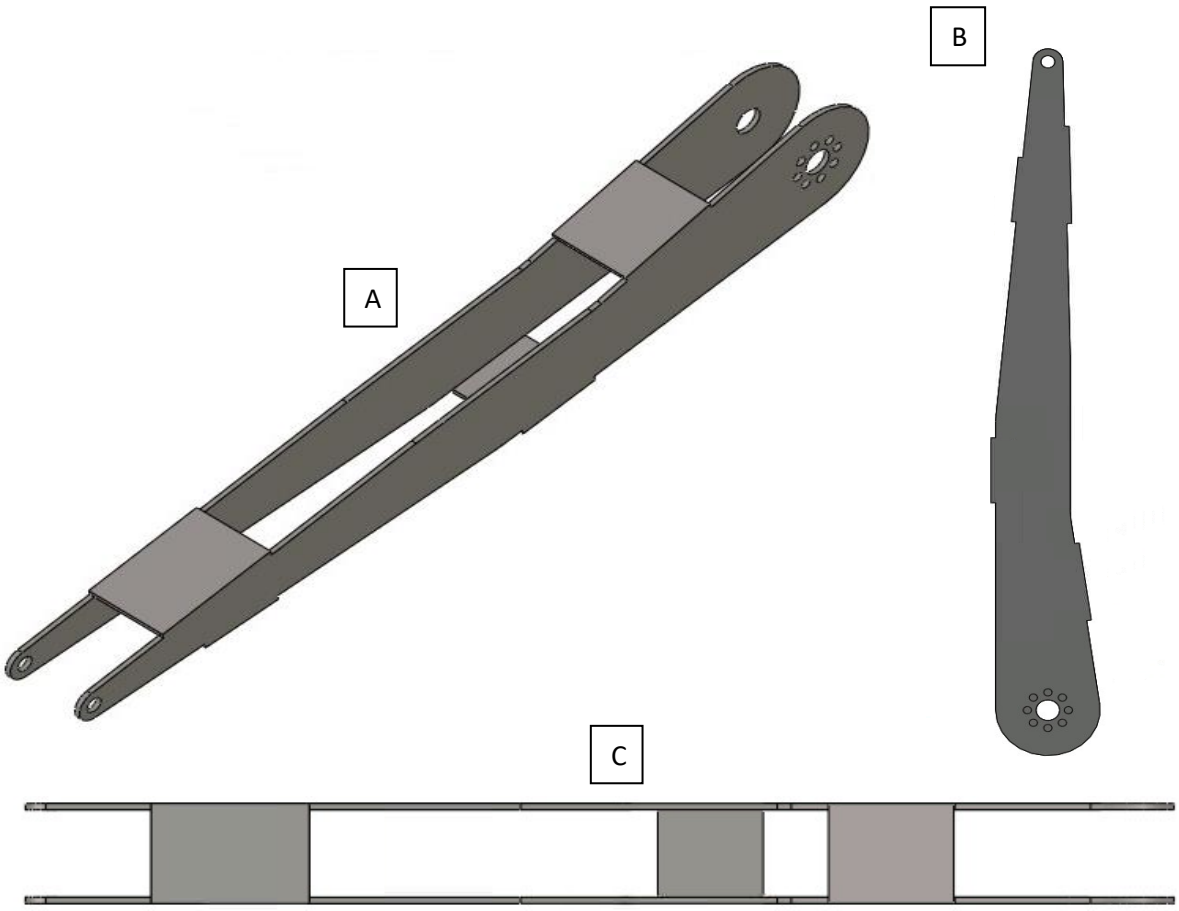



*Figure 3-1: The location of the inner link, which is the largest individual link of the robot.*



## 3.2: Stress Optimization

The purpose of this optimization is to determine the shape of a robotic link with minimum mass that can resist the forces applied by the other links in the mechanism. The objective function can be written as

$$\min \quad m = \rho LHw \tag{3.1}$$

Where m is the mass of the robot link.

$\rho$ is the density of the beam material.

H is the height of the beam.

W is the thickness of the beam.

$$\sigma_{maxiumum} \leq \sigma_{constraint} \tag{3.2}$$

And

$$\delta_{maximum} \leq \delta_{constraint} \tag{3.3}$$

Where

$\sigma_{maxiumum}$ is the maximum measured stress that occurs in the link during loading.

$\sigma_{constraint}$ is the specified maximum stress allowable during loading.

$\delta_{maximum}$ is the maximum measured deflection of the link during loading.

$\delta_{constraint}$ is the specified maximum deflection allowable during loading.

An additional constraint for this optimization is that the optimized beam must maintain the same connections to the linear actuator and base parts as the original inner link design.

The optimization variable is H, the height of the beam at any given location.

When looking at the beam in a 2-dimension plane the inner link can be simplified as shown below. The connection between the inner link and the base of the robot is labeled A and is located at frame 5 according to Figure 2-2. The connection between the inner link and the upper link is labeled B, and is located at frame 8 according to Figure 2-3.


*Figure 3-3: A simplified inner link*

The load of this link depends on the position and motion of the robotic arm and is not constant while the arm is moving along a trajectory. A free body diagram of the beam is shown below in Figure 3-4.

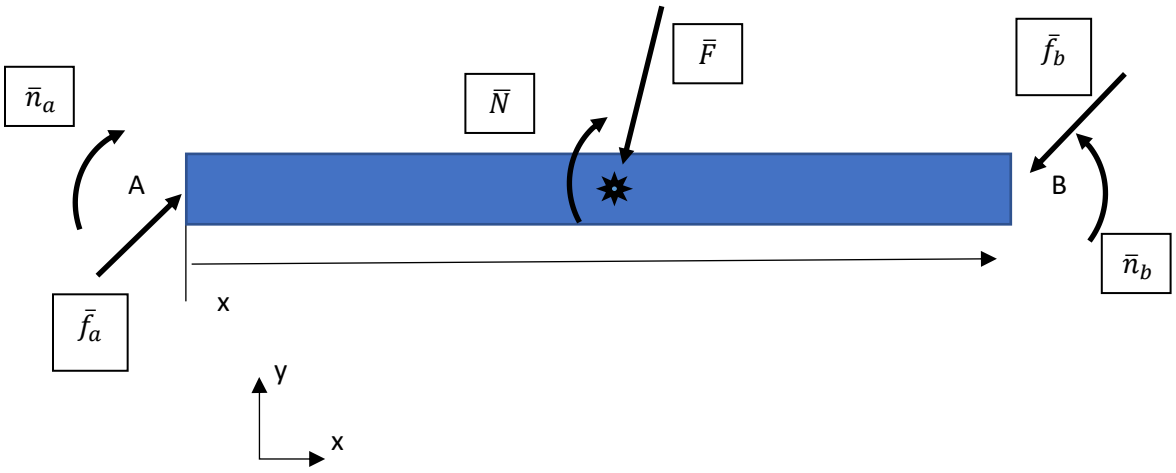



*Figure 3-4: Free body diagram of the inner link*

It is known that A behaves as a fixed support (it is a pin support with a moment applied), while B is a pin support. For this analysis it is assumed that the forces in the $\hat{z}$ axis and moments about the $\hat{x}$ and $\hat{y}$ axes are negligible.

The applied forces at B are a combination of static forces and dynamic forces developed by motion of the robot arm. Using the dynamic equations from the previous chapter the forces acting at B can be solved for, while the robot arm is in motion. One complication arises due to the forces at B not being constant during motion of the robot arm, due to changing configuration of the arm. To account for the changing forces at B, the boundary condition that is applied at will be the maximum force in each direction that occurs during the chosen trajectory. More explanation can be found below. The beam has a depth of w.

Once the boundary conditions are established, the bar is divided into n slices and the height of each of the slices is optimized.  A beam with 4 slices has been illustrated below in Figure 3-5.

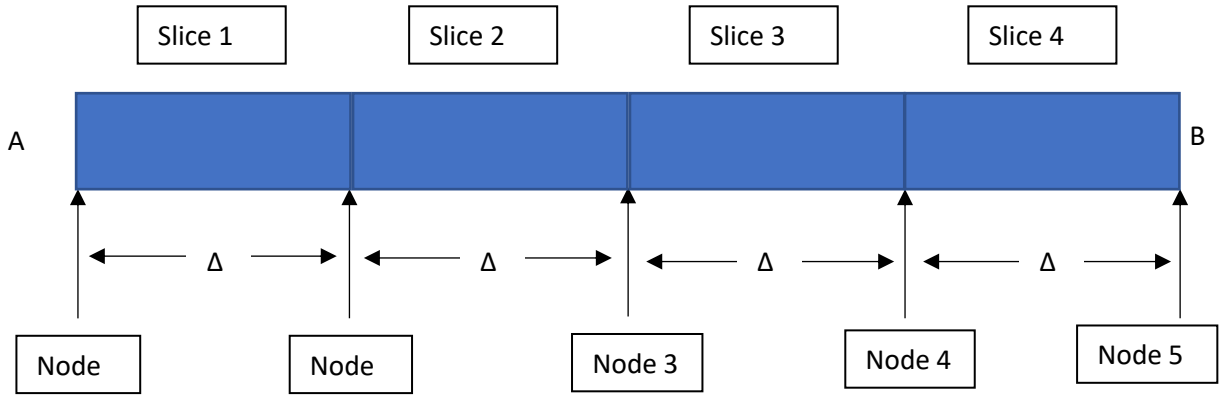



*Figure 3-5: a beam split into 4 slices*

Each of the elements has a unique loading resulting from a combination of a distributed loading and point loading.

Breaking the beam down to show the internal forces gives the following Figure.

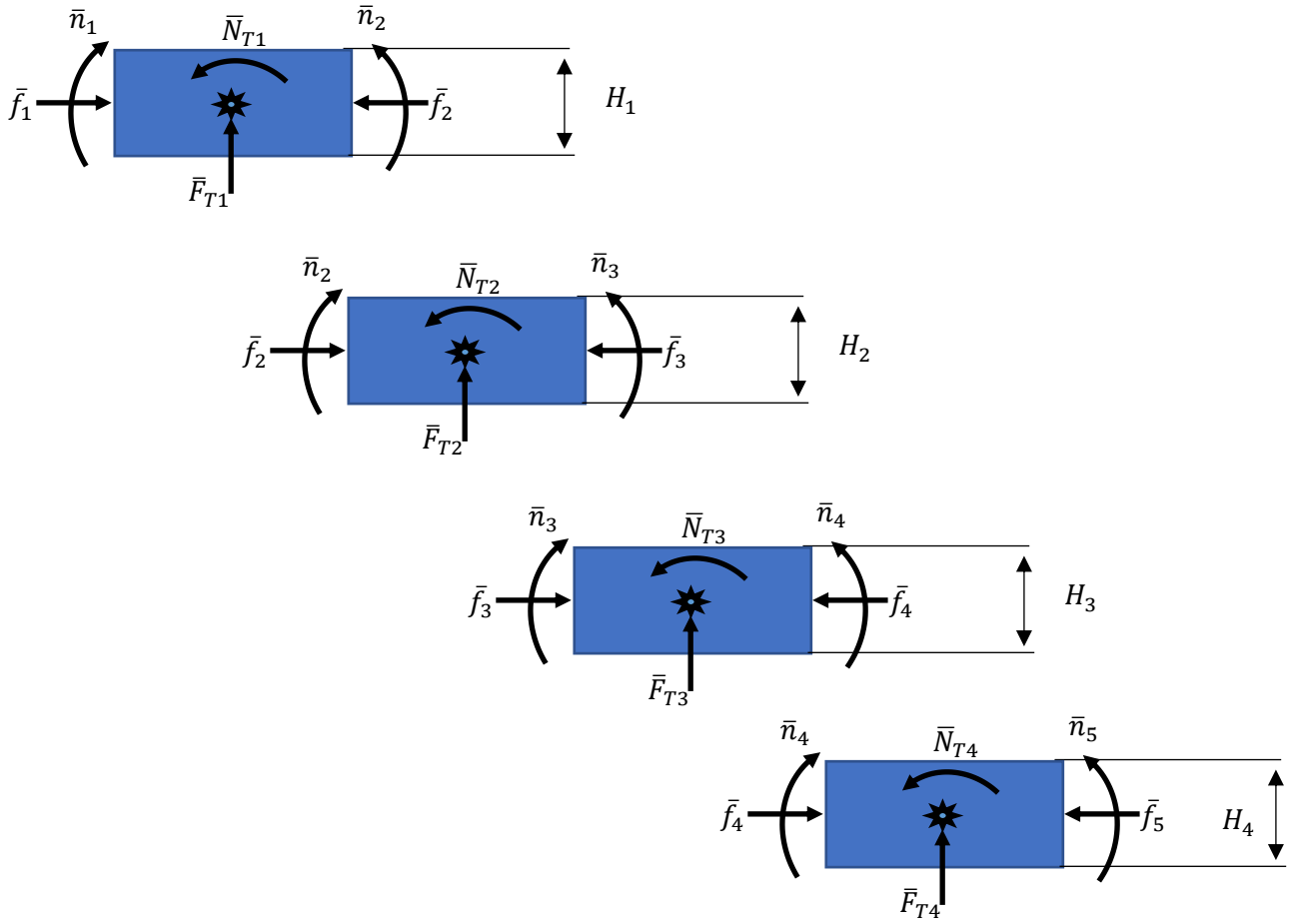



*Figure 3-6: Internal forces and free body diagram (FBD)*

$\bar{f}$ *and* $\bar{n}$ are the external force/moment vectors applied on each side of individual slices.

Where $\bar{F}_e$ and $\bar{N}_e$ are the external force/moment vectors. $\bar{F}_T$ and $\bar{N}_T$ are the total combinations of inertial forces ($\bar{F}$ and $\bar{N}$), and any other external applied distributed loads or point loads ($\bar{F}_e$ and $\bar{N}_e$).

$$\bar{F}_T = \bar{F}_e + \bar{F} \qquad 3.4$$

$$\bar{N}_T = \bar{N}_e + \bar{N} \qquad 3.5$$

The inertial forces/moments, $\bar{F}$ and $\bar{N}$, can be calculated from known values using:

$$\bar{F}_i = m\dot{v}_{c_i} \quad 3.6$$

$$\bar{N}_i = {}^{c_i}I_i\ {}^{i}\dot{\omega}_i + \omega_i \times {}^{c_i}I_i\,{}^{i}\omega_i \quad 3.7$$

Where $\dot{v}_{c_i}$ can be calculated with Equation 2-40.

${}^{i}\dot{\omega}_i\ and\ {}^{i}\omega_i$ can be calculated from Equations 2-38 and 2-39.

and

$$m = \rho * \Delta x * w * H \quad 3.8$$

A general description of a slice is shown in Figure 3-7 below. Each slice i can be shown as having two sides, which are referred to as nodes during the optimization.

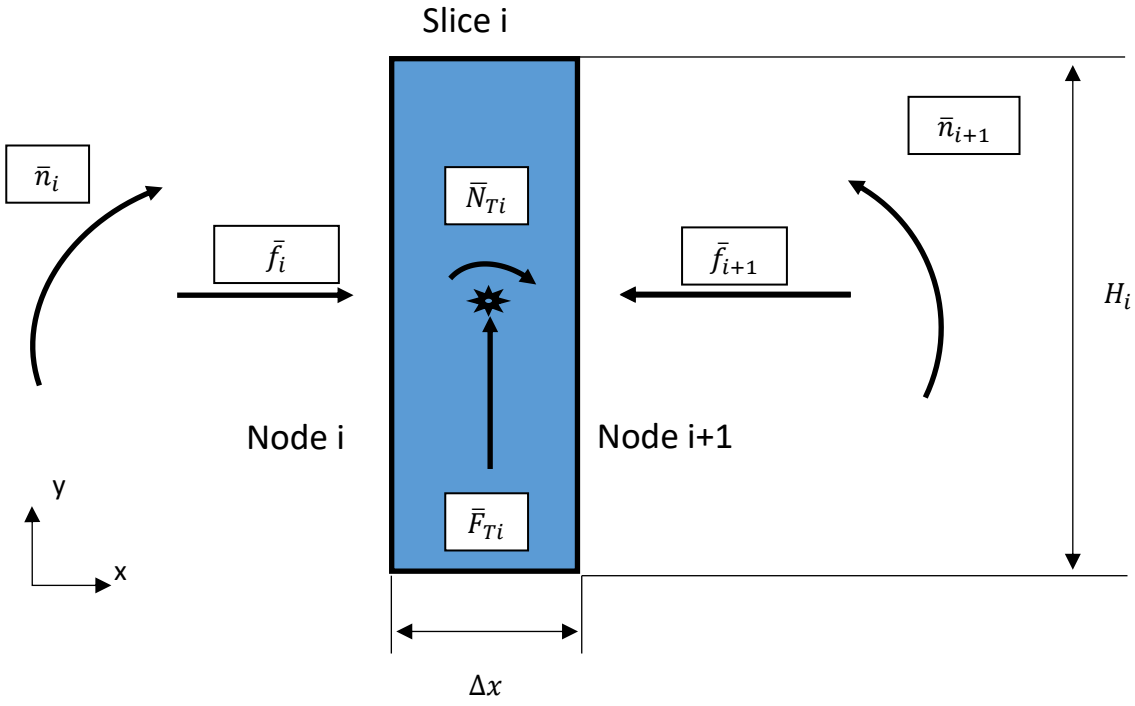



*Figure 3-7: A free body diagram of an individual slice of the bar*

A general description of a slice is shown in Figure 3-7 above. Each slice i can be shown as having two sides, which are referred to as nodes during the optimization. As this optimization progresses from B to A (the boundary conditions of B are known), the forces/moments at node i will be referred to as the reaction forces.

As the forces at B are known, statics equations are solved for the unknown forces at node i.

$$\sum Forces = 0 \tag{3.9}$$

$$\sum Moments = 0 \tag{3.10}$$

Once all the unknown forces are solved, stress calculations may begin on each surface to determine the optimal height at each node. The stresses on face i are as follows:

$$\sigma_{Maximum\ Normal} = \frac{F_x}{A} \tag{3.11}$$

$$A = w * H \tag{3.12}$$

$$\sigma_{Maximum\ Bending} = \frac{n_z(H/2)}{I_{zbending}} \tag{3.13}$$

$$I_{zbending} = \frac{w * H^3}{12} \tag{3.14}$$

The maximum stress of the beam will be found at the surface and will be a combination of normal and bending stress.

$$\sigma_{maximum} = \sigma_{maximum\ bending} + \sigma_{maximum\ normal} \tag{3.15}$$

A safety factor of 2 was also included in the analysis, such that

$$\sigma_{constraint} = \frac{\sigma_{yield}}{S.F.} \tag{3.16}$$

$$\sigma_{constraint} = \frac{\sigma_{yield}}{2} \quad 3.17$$

Assumptions and limitations

Several assumptions were made during the optimization process.

1. Transverse forces (z direction in Figure 3-4) acting on the beam are considered to be zero.

   Under normal loading circumstances and a static structure, there will be no transverse forces acting on the arm. However, due to the motion of joint 1, there can be some inertial forces acting in the transverse direction. These forces are neglected during the optimization process.

2. The pin connection at frame 8 is assumed to be frictionless.

   A frictionless pin will allow the assumption of no moment in the Z direction at point B of Figure 3-4.

3. Stresses and deflections caused by vibrations are not considered.

4. This method will look at the optimized contour created from a solid beam of material. There are additional ways to lower the mass of the arm which include the use of lightning holes, or different configurations of material other than two parallel plates. These methods are not explored in this thesis, but may be investigated in future work.

5. Different materials are not explored in this thesis. Manufacturing the robot arm out of different materials is also a method for reducing the mass of the robot arm while maintaining performance requirements. The use of different materials is being investigated by another member of the robotics lab and is outside the scope of this thesis.

6. While the method presented is developed to account for dynamic forces due to the motion of the robot, the specific trajectory described in this chapter does not produce high dynamic forces (the average speed of the end effector while traveling the trajectory described in Table 3-1 is 0.82m/s). This speed however, is a reasonable estimate for the expected performance of the robot arm.

7. The gradient optimization technique presented in this thesis is not the only method available to solve the optimization problem. Additional optimization techniques exist, in addition to the analytical solution to the appropriate differential equations.

## 3.3: Stress Optimization Process

At the beginning of the optimization the following are known:

The number of slices,

An arbitrary initial height of each slice,

The boundary conditions existing at A,

The linear acceleration, linear velocity, angular acceleration, and angular velocity of every point on the robot link, as can be calculated from the dynamic equations in chapter 2,

Any exterior distributed or point loads acting on the beam.

The procedure for the stress optimization can be listed as:

1) start at slice i= number of slices
2) Solve Equations 3-4 and 3-5 to determine the reaction forces acting on a slice.
3) Use the forces to determine the stresses acting on the slice using Equations 3-9 to 3-13.
4) Determine the Von Mises stress ($\sigma_{maximum}$), and compare it to the maximum allowable stress ($\sigma_{constraint}$)

5a) If the Von Mises stress exceeds the constraint maximum stress. Increase the height of slice i by a small amount (ΔH). Repeat steps 1-4

5b) if the Von Mises stress is below the maximum stress, record the height of slice i and repeat steps 1-4 with the slice i-1 until all slices have been considered.

Figure 3-11 at the end of the next section shows a flowchart of the process.

In summary: a beam is spilt into many slices. Full knowledge of the forces applied to each slice is known. Boundary conditions and inertial forces/moments are known by solving the dynamic equations presented in chapter 2. Reaction forces acting on each node can be calculated using sum of forces/moments. As the full breakdown of forces is known the maximum Von Mises stress that occurs within a slice can be calculated. The calculated maximum stress is compared to the specified maximum stress constraint. The height of each slice is then optimized so that the smallest contour of the beam that does not violate the maximum stress criteria.

However, the deflection constraint of the beam has not yet been considered.

## 3.4: Deflection optimization

Due to the nature of the optimization completed earlier it is impossible to predict the deflection of the beam before the height (and therefore mass) of each section is determined. Stage 2 of the optimization adjusts the height of each slice again to ensure that the total deflection of the beam criteria is met.

Deflection of a beam is given by the differential equation.

$$\frac{d^2V}{dx^2} = \frac{M}{EI_{bending}} \qquad 3.18$$

Where V is the deflection of the beam.

E is the modulus of elasticity.

$I_{bending}$ is the area moment of inertia.

M is the internal bending moment of the beam.

Integrating the differential equation once gives the first moment area theorem. The moment-area method uses the graph of M/EI vs x of the beam, and says that the area under the curve of the M/EI line from a to b is the change in angle between node i and i+1 in radians. This method is particularly useful as values for M, E, and I are known at each node on the beam from the earlier optimization. This method can be expressed as

$$\int_i^{i+1} \frac{d^2V}{dx^2}\,dx = \int_i^{i+1} \frac{M}{EI}\,dx = \frac{dV}{dx} = \gamma_{i/_{i-1}} \qquad 3.19$$

Where $\gamma_{i+1/_i}$ is the angle between slice i and slice i+1.

For ease of notation $\gamma_{i/_{i-1}}$ shall be referred to as $\gamma_i$. (see Fig 3-9)

Neither the cross-sectional area of the beam, nor the loading is constant across the entire beam. This disallows the use of the closed form solutions to the displacement integral. Additionally, the analytical solution to the integral is complex. To solve the equation a numerical integration technique is employed.

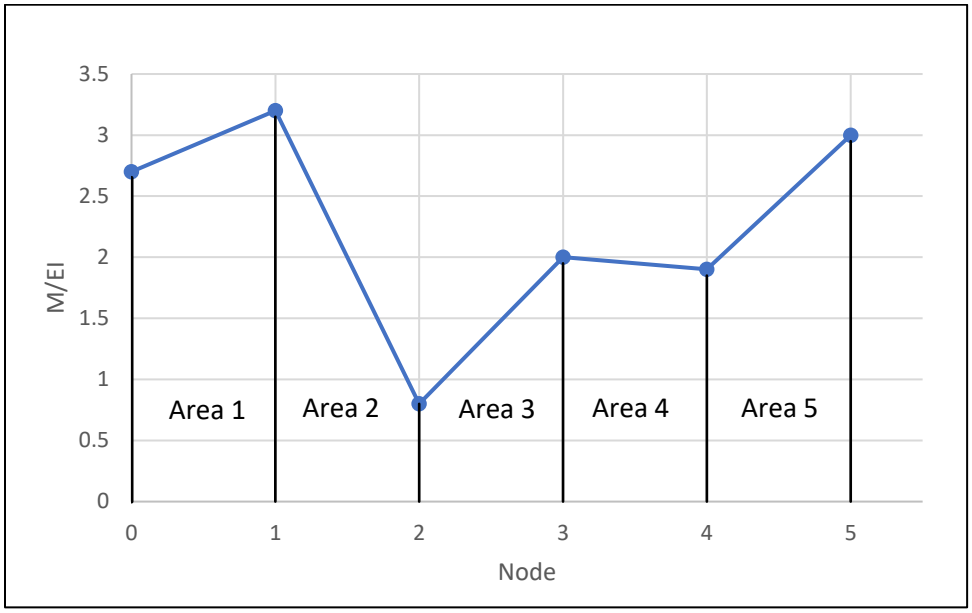



*Figure 3-8: A demonstration of the area underneath a M/EI graph*

The area under the curve between each node represents the change in angle of an element between each node ($\gamma_i$).

An approximation of the area is made by using the calculated values of M/EI at each node and assuming a linear change. The change in angle over an element i is:

$$\gamma_i = \left(\frac{M_i}{E_i I_i} + \frac{M_{i+1}}{E_{i+1} I_{i+1}}\right)\frac{\Delta x}{2} \qquad 3.20$$

Deflection of the beam is explored in Figures 3-9 and 3-10 below, with expressions for element contribution to tip deflection, and total tip deflection derivation following.

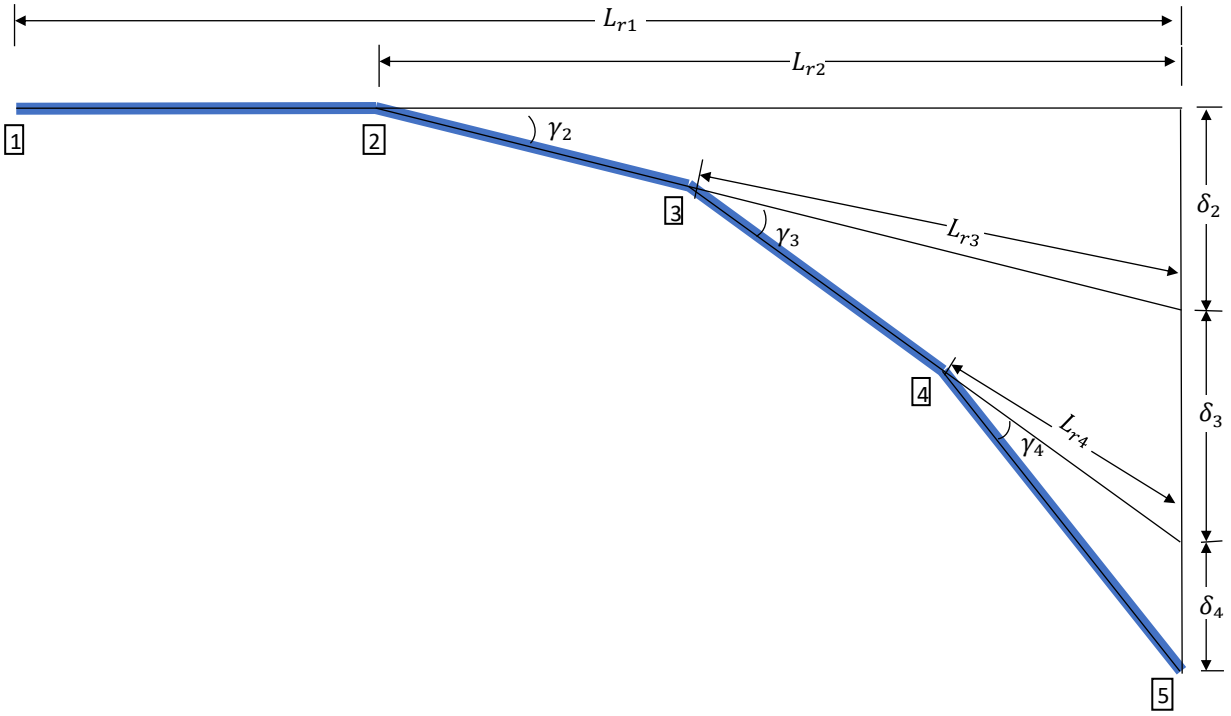



*Figure 3-9 Locations of lengths and angles of a segmented beam in deflection*

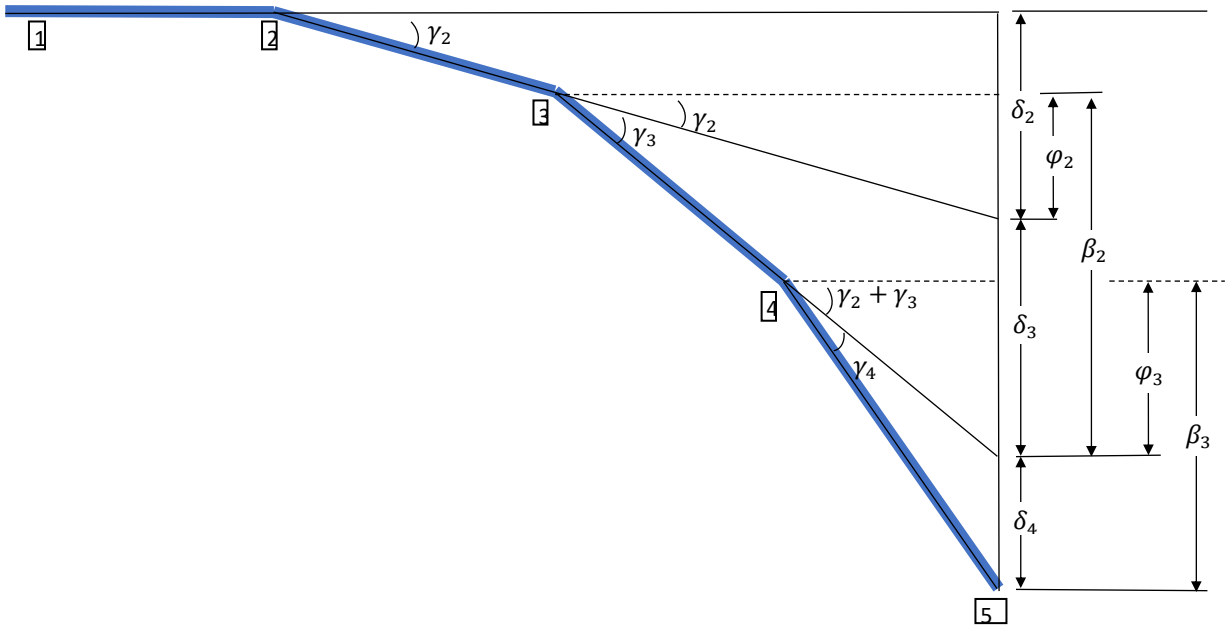



*Figure 3-10: Angles and measurements of a deflecting segmented beam*

To determine the contribution of overall deflection attributed to a single $\gamma_i$ Figures $3-9$ and $3-10$ are analyzed.

Where $L_{ri}$ is the remaining distance from node i to the tip of the beam and calculated as

$$L_{ri} = [L_{total} - \Delta x * (i-1)] \quad 3.21$$

For example

$$\delta_2 = L_{ri} sin(\gamma_2) \quad 3.22$$

$$\delta_2 = \beta_2 - \varphi_2 \quad 3.23$$

Where

$$\beta_2 = L_{r2} * sin(\gamma_2 + \gamma_3) \quad 3.24$$

$$\varphi_2 = L_{r2} * sin(\gamma_2) \quad 3.25$$

As the pattern continues, it can be shown that

$$\delta_i = L_{ri}[sin(\sum_{j=1}^{i} \gamma_j) - (sin(\sum_{j=1}^{i-1} \gamma_j)] \quad 3.26$$

Knowing that

$$\sum_{j=1}^{i} \gamma_j = \sum_{j=1}^{i-1} \gamma_j + \gamma_j \quad 3.27$$

3-24 can be rewritten as:

$$\delta_i = L_{ri}[sin\left\{(\sum_{j=1}^{i-1} \gamma_j) + (\gamma_j)\right\} - sin\left\{\sum_{j=1}^{i-1} \gamma_j\right\}] \quad 3.28$$

A substitution for A and B can be made

$$A = \sum_{j=1}^{i-1} \gamma_j \qquad 3.29$$

$$B = \gamma_i \qquad 3.30$$

3-26 then can be written as

$$\delta_i = L_{ri}[sin\{A + B\} - sin\{A\}] \qquad 3.31$$

Using the approximation when angles are small (this approximation can be made as each angle is < 3$^o$ )

$$\sin(A + B) \approx sin(A) + sin(B) \qquad 3.32$$

Then

$$sin(A + B) - sin(A) \approx sin(B) \qquad 3.33$$

Substituting back into 3-29

$$\delta_i \approx L_{ri}[sin(B)] \qquad 3.34$$

Gives the contribution for each node

$$\delta_i = L_{ri}[sin(\gamma_i)] \qquad 3.35$$

Total deflection can be written as

$$\delta_t = \sum_{j=1}^{i} \delta_j = \sum_{j=1}^{i} L_{rj}[sin(\gamma_j)] \qquad 3.36$$

Now that the total deflection of the beam is known, it can be compared to the deflection constraint specified. If the beam deflection exceeds the limit, the deflection of the beam must be reduced.

Returning to the M/EI graph; to lower the overall deflection of the beam the area under the curve must be reduced.

As E is constant throughout the entire beam, either moment acting on a slice can be reduced, or the bending moment of inertia can be increased.

The height H of each slice has been chosen to be modified along the length of the beam. As H increases, $I_{bending}$ will increase, additionally M too will increase but at a lower rate. This will cause M/EI to be lowered and the deflection of the beam overall decreased.

Applying a blanket ΔH increase to the entire bar is an option, however this solution is not effective as areas of the bar that do not require it may be over-reinforced, increasing the mass of the beam without cause, countering the objective to minimize the mass of the link.

Instead, the height of each slice of the beam can be adjusted independently of the others.

The new height of node i can be referred to as $H_{new\ i}$

As the (M/EI) is not constant for each of the slices, it is difficult to predict which slices' heights should be adjusted. The effects must be calculated.

1) Record previously calculated deflection of slice i. This is $\delta_{old,i}$.
2) Record the new dimensions of slice i. ($H_{new} = H_i + \Delta H$)
3) Calculate the forces/moments
4) Plot new (M/EI)
5) Determine $\gamma_{new\ i}$
6) Calculate the new deflection contribution using Equation 3-41, this is $\delta_{new,i}$.
7) Calculate $\Delta\delta_i$.
   where

$$\Delta\delta_i = \delta_{new,i} - \delta_{old,i} \qquad 3.37$$

8) Complete 1-7 for remaining slices
9) Identify slice that produces the greatest $\Delta\delta_i$.
10) For the slice that creates the largest change in deflection, make the change in height permanent ($H_i = H_i + \Delta H$)

11) Calculate total deflection of the beam, if the maximum deflection constraint is met, end optimization. If not continue to step 12.
12) Repeat steps 1-10 until deflection constraint is met.

As the optimization progresses, each node will experience a change in height independent of each other node and the change in deflection of the tip of the beam is recorded. The node that resulted in the greatest change in tip deflection is selected and the change in height is made permanent. The algorithm is then repeated to find the next node which has the greatest effect on tip deflection.

In essence, a beam is being built by adding small amounts of mass to individual slices until the stress and deflection criteria are met.

The original beam has two parallel beams connected by plates at the top and bottom. During the optimization one beam will be analyzed. When constructing a model from the results collected, the parallel plates will be half the width of the single beam analyzed during the optimization.

After applying the optimization each individual slice will have the least amount of mass possible while still satisfying both maximum stress and maximum deflection constraints.

A final verification of the completed beam will test for buckling failure using Equation 3-37 below. As the critical load formula for buckling assumes a consistent cross section, the critical load will be evaluated using the smallest cross section of the fully optimized beam.

$$P_{cr} = \frac{\pi^2 EI}{(KL)^2} \qquad 3.38$$

Where $P_{cr}$ is the critical force that will cause buckling in a column.

K is the effective length factor which depends on the boundary conditions of the beam. (0.7) in this case.

The overall optimization is shown graphically in Figure 3-11 below.

The MATLAB code for this optimization is shown in Appendix D

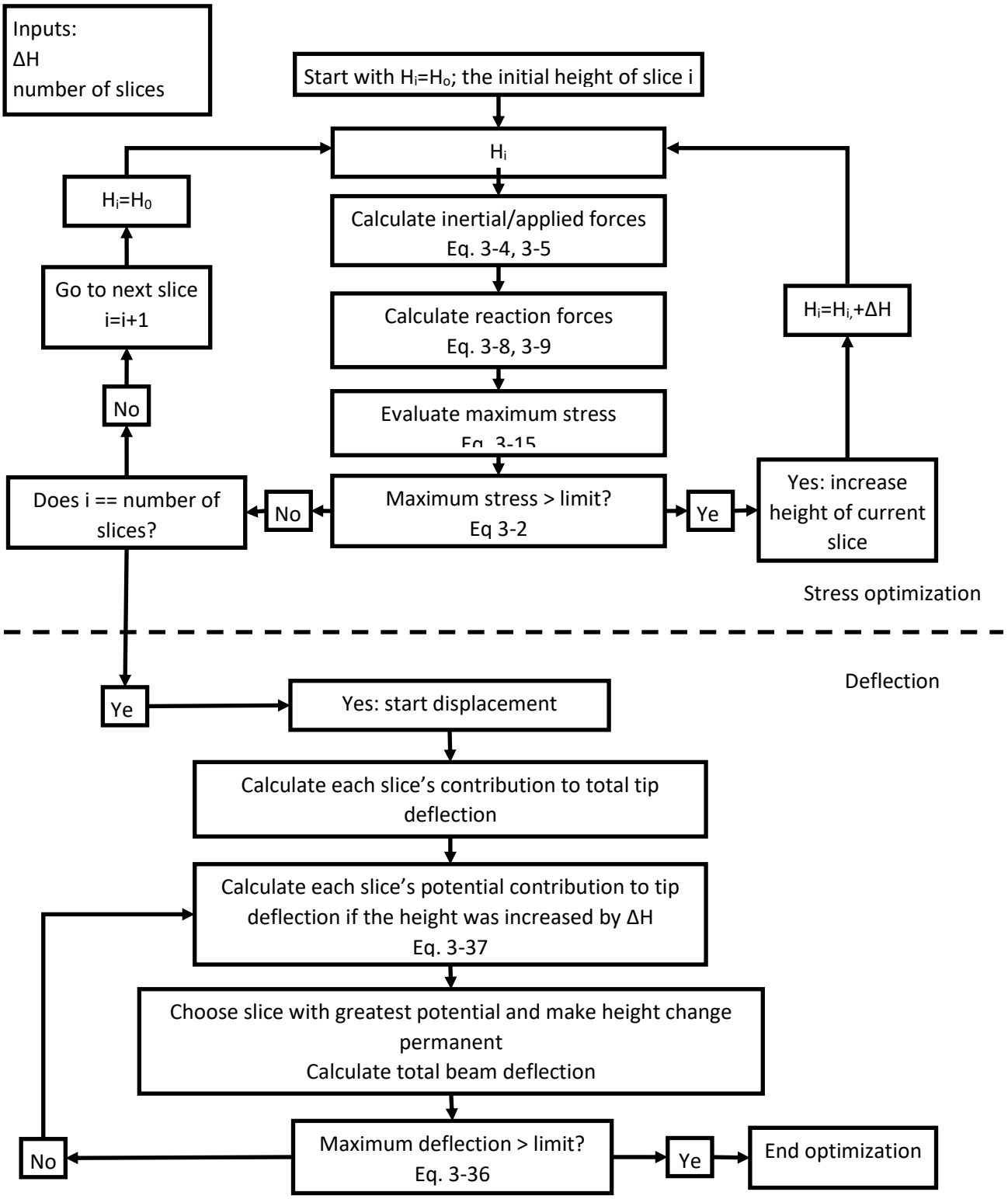





*Figure 3-11: A flowchart showing the method of optimization*

## 3.5: Optimization boundary conditions

Using the method described earlier for path planning a sample cubic polynomial trajectory for the robot arm was constructed with details as follows. This trajectory is a realistic expectation of the robotic arm to be extended from a closed "transport" position to an open data collection position.

*Table 3-1: A trajectory expected to be performed by the robotic arm*

| Parameter | Variable | Initial Value | Final Value |
|---|---|---|---|
| **Time** | **Time (s)** | 0 | 5 |
| **Position** | **Joint 1 (degrees)** | 0 | 90 |
| | **Joint 2 (degrees)** | 0 | 60 |
| | **Joint 3 (degrees)** | 30 | 180 |
| | **Joint 4 (m)** | 0 | 1 |
| | **Joint 5 (degrees)** | -45 | 0 |
| **Velocity** | $\mathbf{Joint\ 1\ (degrees/}s\mathbf{)}$ | 0 | 0 |
| | $\mathbf{Joint\ 2\ (degrees/}s\mathbf{)}$ | | |
| | $\mathbf{Joint\ 3\ (degrees/}s\mathbf{)}$ | | |
| | $\mathbf{Joint\ 4\ (m/}s\mathbf{)}$ | | |
| | $\mathbf{Joint\ 5\ (degrees/}s\mathbf{)}$ | | |
| **Acceleration** | $\mathbf{Joint\ 1\ (degrees/}s^2\mathbf{)}$ | 0 | 0 |
| | $\mathbf{Joint\ 2\ (degrees/}s^2\mathbf{)}$ | | |
| | $\mathbf{Joint\ 3\ (degrees/}s^2\mathbf{)}$ | | |
| | $\mathbf{Joint\ 4\ (m/}s^2\mathbf{)}$ | | |
| | $\mathbf{Joint\ 5\ (degrees/}s^2\mathbf{)}$ | | |

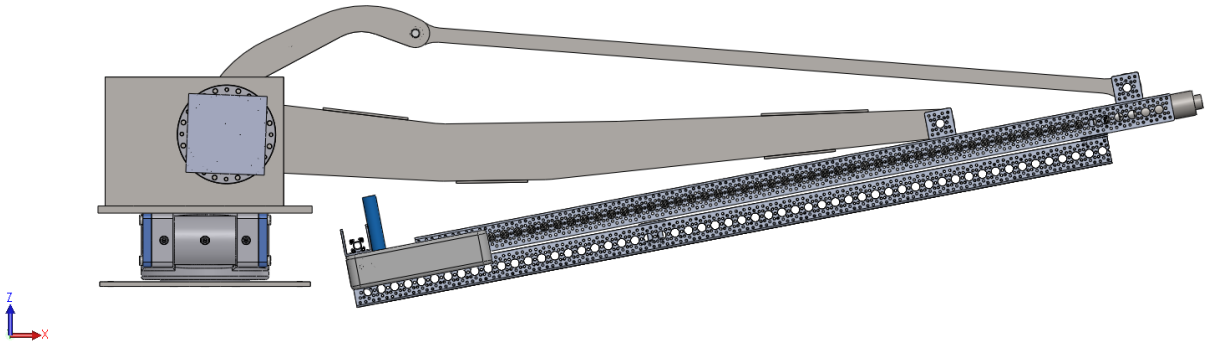


*Figure 3-12: 3rd generation boom in "closed" or "transport" position*

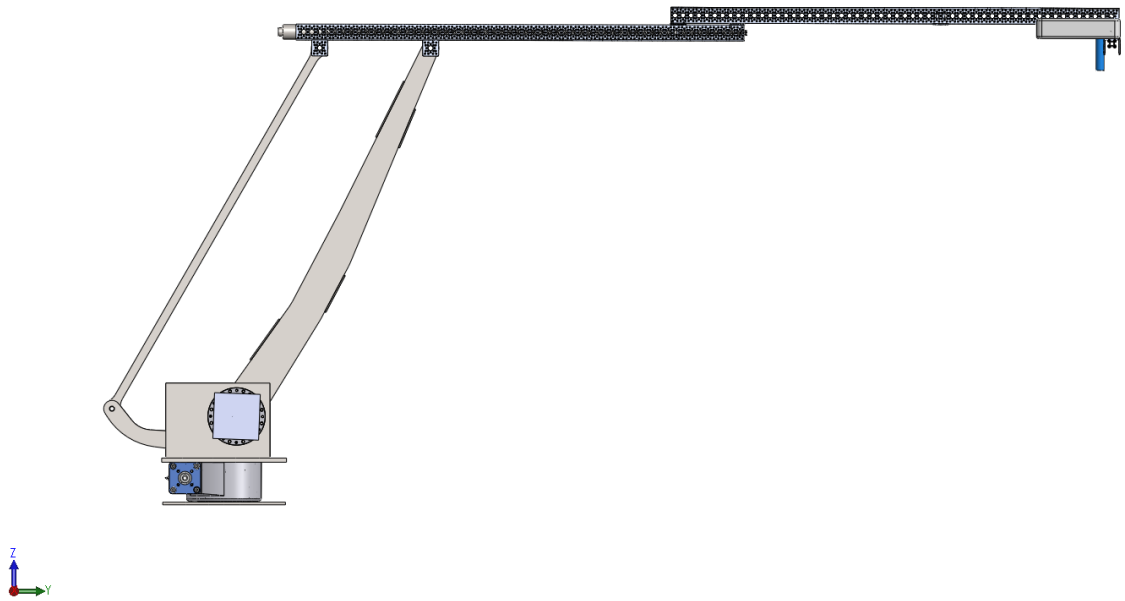


*Figure 3-13: 3rd Generation Boom in "open" or "phenotyping" position*

Between the two positions above, note the 90 degree rotation of theta 1 as well.

To determine the boundary conditions applied at B, the trajectory described above was entered into the MATLAB dynamic code derived in chapter 2 and the values of rotational acceleration, rotational velocity, force, moment and linear velocity were recorded at each $\Delta t=0.01$ second of the trajectory.

While the robot travels throughout the trajectory each joint of the robot will experience a range of forces and moments as the links change orientations and accelerations. This makes it difficult to choose 1 set of boundary conditions for the robot link, and because of this, an optimization run at different stages of the trajectory would produce different results.

To address this challenge the maximum values of each boundary condition were chosen to be applied, even if the values did not occur concurrently. For example, if the maximum x force value was found at T=1s. And the maximum acceleration $\ddot{z}$ was found at T=2.5, both values would be used in the optimization.

The following boundary conditions were calculated and used during the optimization. All values are relative to frame 5.

Several forces were neglected in this analysis, frictional forces frame 8 at the pin connection are assumed to be zero additionally the arm is assumed to be loaded symmetrically at the end effector, therefore all components of $\bar{n}_B$ are considered to be 0. Force $f_{8z}$ was found to be 3.0N and has been neglected as it is greater than two magnitudes smaller than $f_{8y}\ and\ f_{8x}$.

$$\bar{f_b} = \begin{bmatrix} -758.1 \\ -859.9 \\ 0 \end{bmatrix} N$$

$$\bar{n}_B = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} N \cdot m$$

$$\omega_i = \begin{bmatrix} 0.293 \\ 0.423 \\ 0.314 \end{bmatrix} rad/s$$

$$\dot{\omega}_i = \begin{bmatrix} 0.327 \\ 0.377 \\ 0.251 \end{bmatrix} rad/s^2$$

$$\dot{v}_{c_i} = \begin{bmatrix} 8.493 \\ 9.912 \\ 0.165 \end{bmatrix} m/s^2$$

The initial parameters for the optimization are as follows:

$E = 200$ GPa

$\Delta H = 0.1\ mm$

$Slices\ =\ 1000$

$Slice\ width\ =\ 1\ mm$

$Safety\ Factor\ =\ 2$

$Yeild\ stress\ =$ 350 MPa

$Width\ =\ 1\ cm$

$Density\ =\ 7950\ \frac{kg}{m^3}$

*Specific FEA details are shown in Appendix B.*

The constraints for the optimization may be chosen arbitrarily and can be changed based on the desired performance of the robot arm. For the purpose of this thesis the following constraints have been chosen to reflect reasonable performance expectations of the robot arm as provided by Dr. Reza Fotouhi and Hedieh Badkoobehhezavah.

The constraints for the optimization are:

$\delta_{limit} =$ 1cm

$\sigma_{maximum} = \frac{\sigma_{yeild}}{2} =\ 175Mpa$

Before the optimization is performed, a convergence analysis will be performed on the parameters of the optimization. The convergence analysis will evaluate convergence of the optimization and determine an acceptable range of values for number of slices, and $\Delta$H. The input values reported above for number of slices and $\Delta H$ are chosen after the convergence

analysis was completed, although the results of the convergence analysis are reported in the next section of this chapter.

After the optimization is completed, the torque requirements of the robot arm with the original inner link design are compared with the torque requirements of the robot arm after the inner link has been replaced with the optimized inner link. The dynamic equations from chapter 2 will be used to calculate the torque required by joint 2 while traveling the path described above.

## 3.6: Results

The optimization program has two design variables which control the convergence and accuracy of the results. Number of slices, and size of the adjustment ($\Delta H$). Number of slices reflects the number of individual elements that the complete beam is divided into. As the number of slices increases the beam will become smoother, and the optimal shape of the beam will be more closely followed resulting in a lower total mass of the beam. Figure 3-14 below shows the total weight of the optimized beam meeting both stress and deflection criteria as the number of slices in the model is increased.

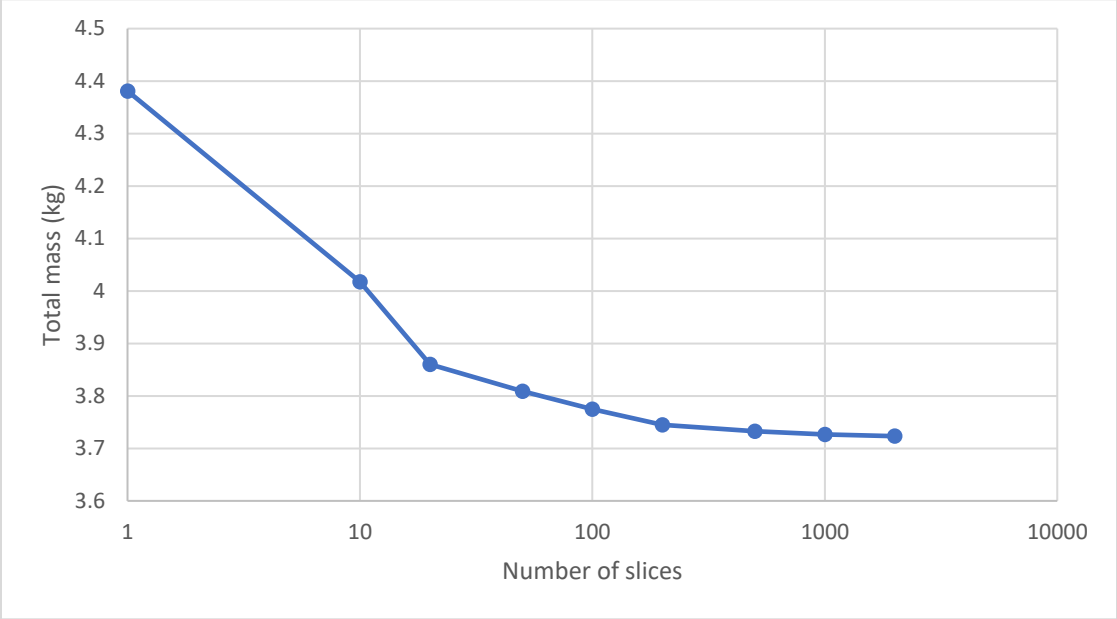



*Figure 3-14: Convergence of optimization based on number of slices.*

From convergence analysis shown in Figure 3-14, an acceptable range of slices for a 1m beam is approximately 600-1000. Too few elements do not reflect the complexity of the changing cross-sectional area of the beam and provide an inaccurate optimization. More than 1000 elements and the additional burden of computation do not improve the results to any meaningful level. Additionally, if too many elements are used the simulation will start to accumulate roundoff error, where the error between the exact values for very small numbers and the computer approximation for very small numbers will increasingly contribute to the optimization.

The second variable that determines the accuracy of the optimization is the $\Delta H$ variable. This input is the small change in height that is applied at each slice to determine if the slice may be any larger without violating any of the constraint equations. As the size of ΔH decreases the optimization improves by finding the minimum mass. However, similar to the number of slices, the consequences of decreasing ΔH leads to no additional optimization improvement and increased round-off error despite of a smaller step size. Based off the convergence analysis performed below in Figure 3-15 acceptable sizes for $\Delta H$ is within 0.01m and 0.001m.

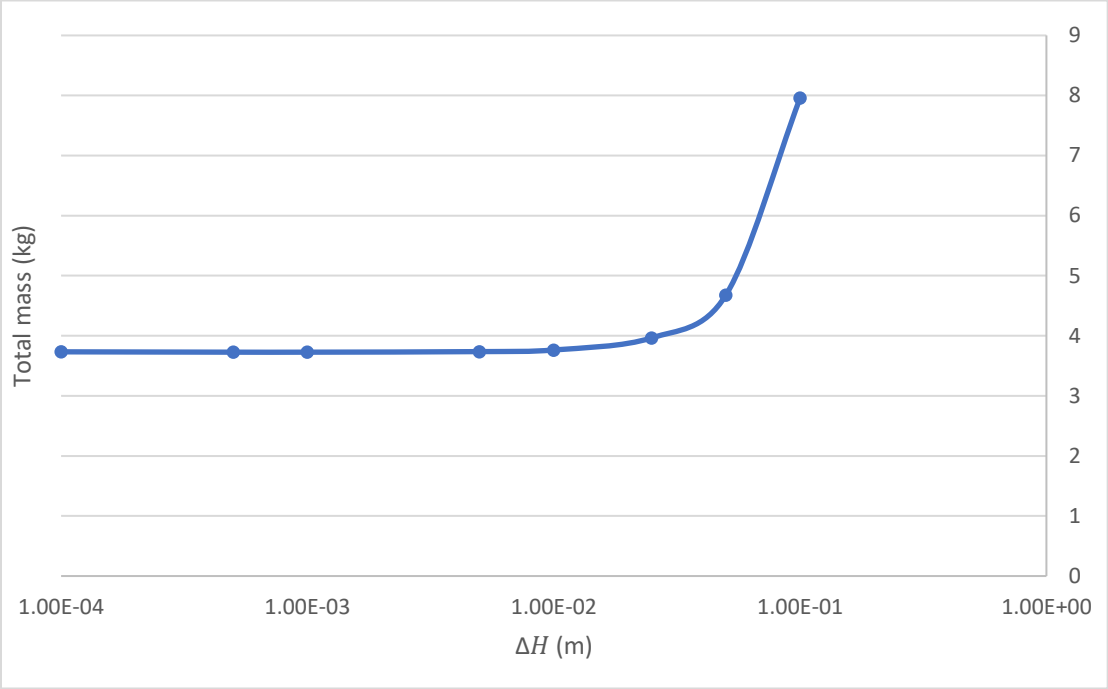



*Figure 3-15: Convergence analysis for ΔH shows convergence when ΔH is smaller than 1E-2m*

After running the optimization with the parameters stated in the previous section, the following results are available.

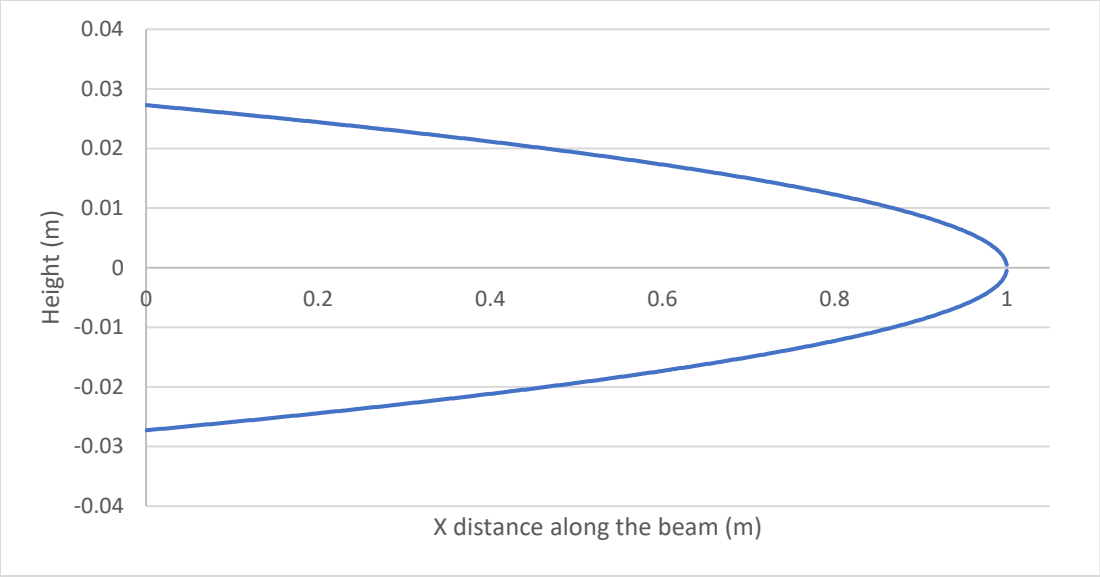



*Figure 3-16: Beam contour after optimization for stress*

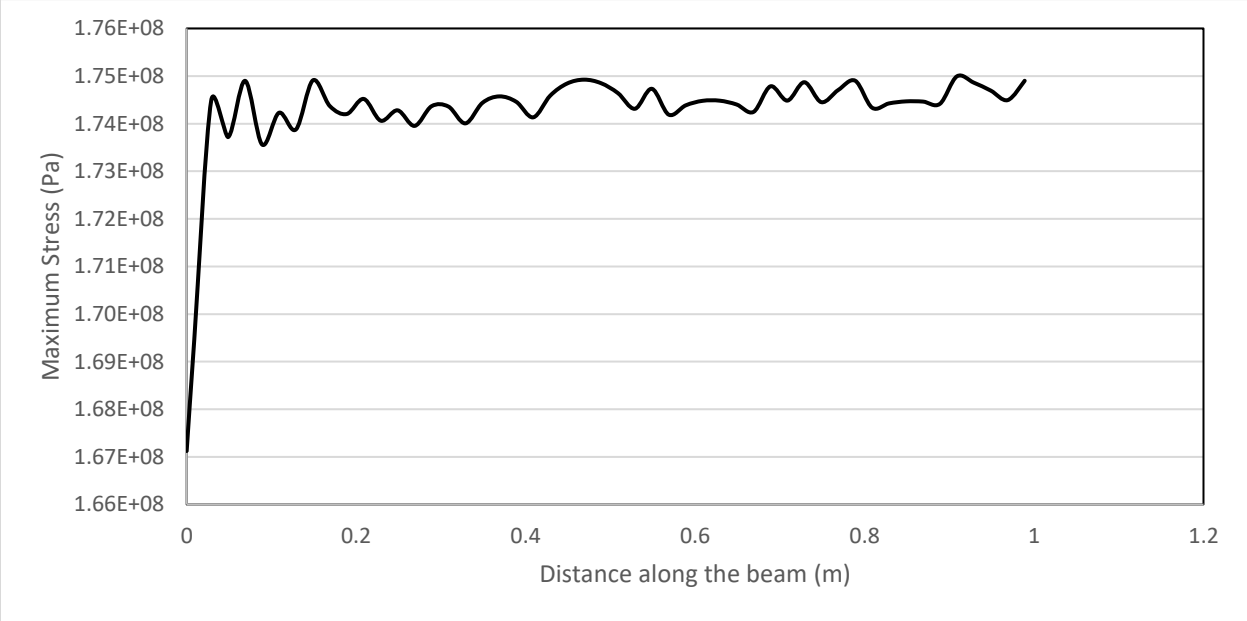



*Figure 3-17: Maximum stresses along the beam as calculated during the optimization.*

Figure 3-17 above shows the shape of the beam optimized for stress. After the first pass of the optimization the beam will be optimized for the least amount of mass to resist failure under the Von-Mises criteria. The maximum stress will occur at the surface of the beam due to a combination of bending, axial and shear stresses. The maximum stress constraint is not violated, as nowhere along the beam does the maximum stress exceed 175 MPa.

It is noticed that according to the graph shown, the maximum stress that occurs along the beam is not constant and can change quickly along the beam, specifically in the first 0.05m of the beam. This occurs due to the chosen value of ΔH. If the height of the slice was decreased by ΔH, the maximum stress of the slice would exceed the maximum stress constraint. As such, the stress value reported is the highest stress value that could be obtained without violating the maximum stress constraint. This could be solved by decreasing the value of ΔH to produce much smaller steps, and the maximum stress would approach the limit much more consistently, however the drawbacks as described above would become significant.

To verify that the optimization did not violate the maximum stress condition of the beam a FE analysis will be completed in ANSYS APDL. A convergence test to determine the proper amount of beam 188 elements for the FE stress analysis was completed and the results are displayed. Figure 3-18 below shows the results of the convergence analysis and indicates that an appropriate FEA of the optimized beam will use a minimum of 100 elements for analysis to be accurate. Normally when analysing a beam very few elements are required, but in this situation, the non constant cross-sectional area and nonconstant distributed loading require additional elements for accurate analysis.

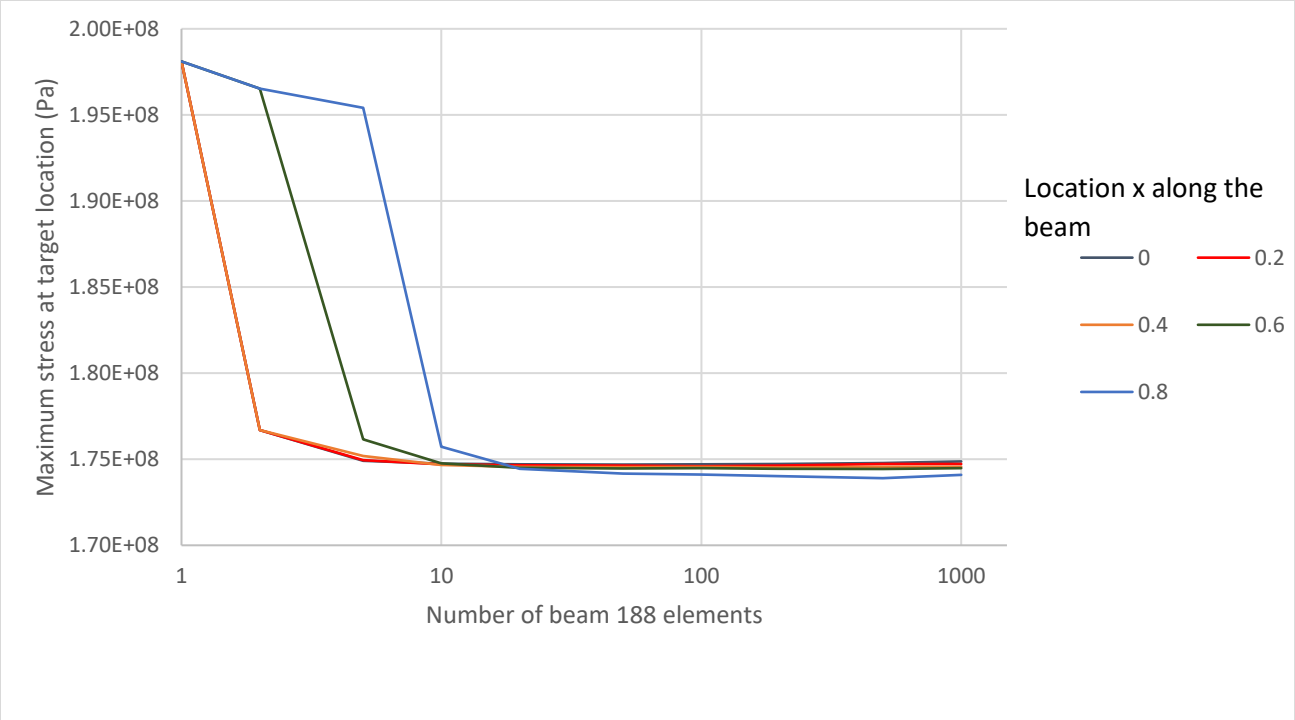



*Figure 3-18: Maximum stress values measured at specific distances along the beam. Compared as number of elements were increased.*

The shape results of the MATLAB optimization were used to create the FEA model. Each slice of the beam was converted into a beam 188 element where the cross section of the element was defined from the results of the optimization. 1000 elements were used in the analysis to mirror the 1000 slices of the optimization.

The results of the FEA simulation and the MATLAB optimization are shown in Table 3-2 below which compares the maximum stress at select locations of the beam, and the maximum stress that occurred overall. While only select locations are show for clarity, the maximum stress found on the entire beam did not exceed 175.0 MPa verifying the maximum allowed stress optimization constraint was not violated. However the deflection criteria was violated as the beam underwent a deformation of 2.1 cm which exceeds the 1cm deflection limit.

*Table 3-2: Maximum stress along the beam*

| X Location along the beam (m) | Maximum stress calculated by optimization program (Pa) | Maximum stress simulated by FEA (Pa) | Relative Difference (%) |
|---|---|---|---|
| 0.1 | 1.746E+08 | 1.745E+08 | 0.06% |
| 0.2 | 1.745E+08 | 1.744E+08 | 0.07% |
| 0.3 | 1.745E+08 | 1.744E+08 | 0.08% |
| 0.4 | 1.743E+08 | 1.742E+08 | 0.10% |
| 0.5 | 1.742E+08 | 1.740E+08 | 0.12% |
| 0.6 | 1.743E+08 | 1.740E+08 | 0.16% |
| 0.7 | 1.750E+08 | 1.746E+08 | 0.20% |
| 0.8 | 1.739E+08 | 1.734E+08 | 0.31% |
| 0.9 | 1.731E+08 | 1.720E+08 | 0.61% |
| Maximum | 1.749E+08 | 1.750E+08 | 0.06% |

The maximum stresses were found at a location specified along the x direction of the beam as described in Figure 3-18. The stresses occurred at the surface of the beam. The maximum stress simulated by FEA was found by probing the maximum bending, normal and shear stresses that occur at a node corresponding to the x location along the beam, and using Equation 3-15.

A second pass of optimization was performed, and the beam was stiffened by increasing the height of individual slices of the beam which would be most effective at reducing the maximum deflection. The results of the second pass compared to the first pass are shown in Figure 3-19 below. In both optimization the height at x=1m approaches 0 thickness but does not reach 0mm of height. The height at the end of both optimizations is 1mm.

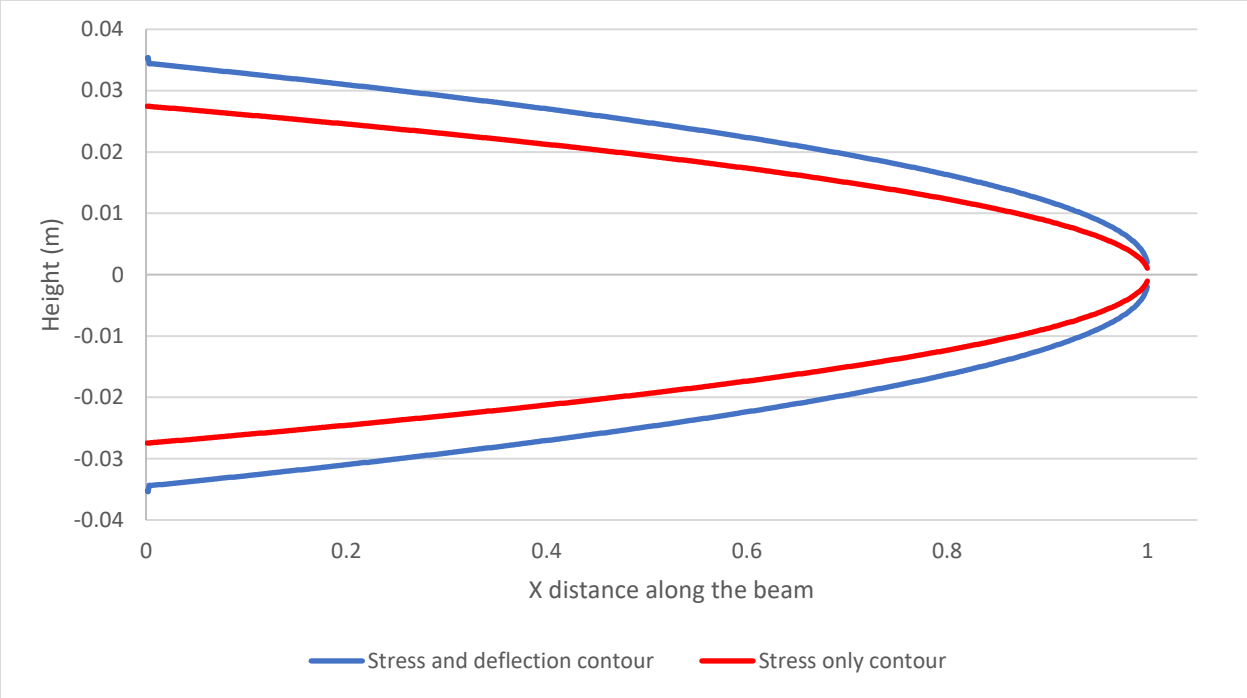



*Figure 3-19: Optimization for Stress and Deflection vs optimization for stress only*

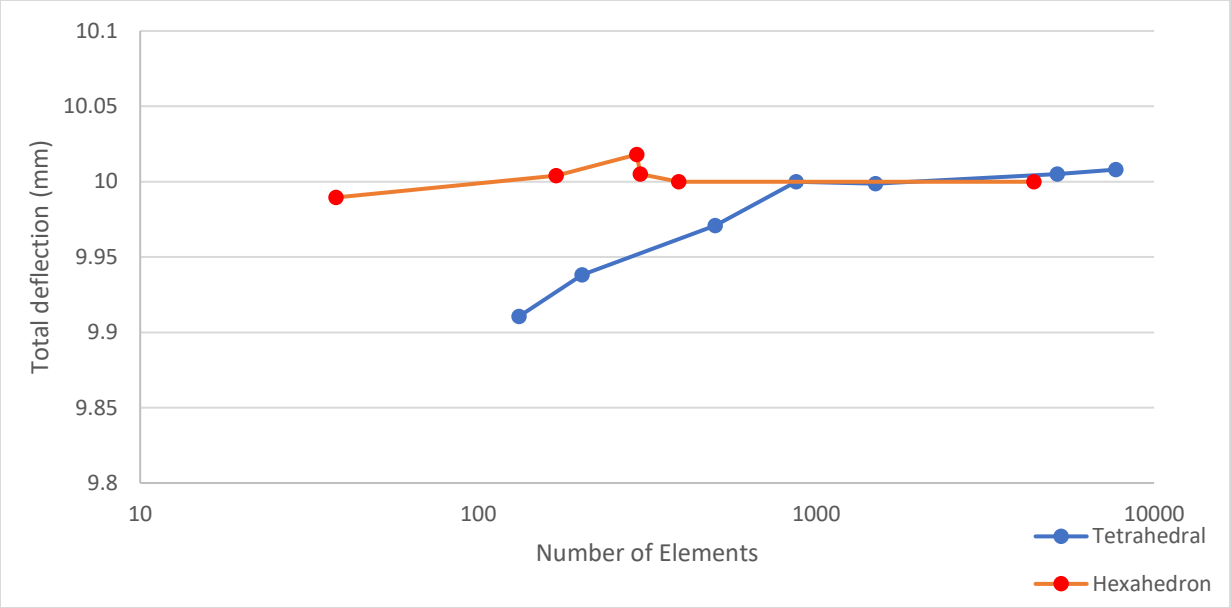



*Figure 3-20: Convergence analysis for maximum deflection*

To verify the results of the deflection optimization, a SolidWorks model of the beam was generated and imported into ANSYS Workbench where a static structural analysis was performed (see Appendix b for details). Figure 3-20 above shows the results of a convergence analysis. The Figure shows that a suitable FEA can be performed on the structure using approximately 100 to 2000 elements if a hexahedral element is used, and 2000-5000 elements if a tetrahedral element is used. Shown in Figure 3-21 below using a hexahedron mesh and 620

elements are the results of the maximum deflection. The overall deflection of the beam under loading was 1.0021cm which is above the deflection limit but within allowable error.

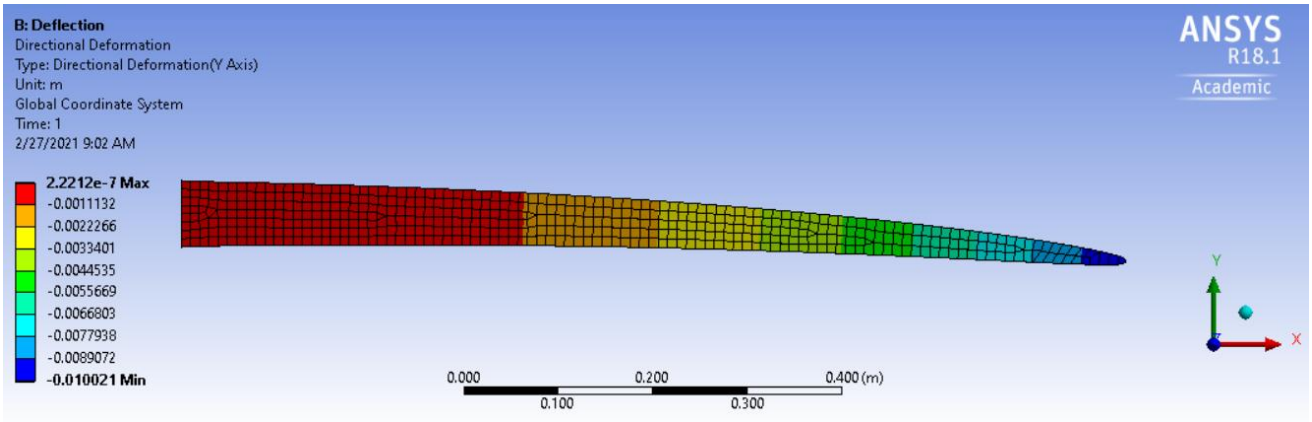



*Figure 3-21: Deformation determined by ANSYS: 3262 elements and inputs as described above*

After proving that the optimized shape of the beam met the constraints of the optimization the results gathered are used to create an improved inner link to replace the original inner link. It is necessary to preserve the pin connections to the base and to the linear actuator. Figure 3-22 below shows the superposition of the optimized link and the original link. To create a useable link that can be placed in a physical robot the optimized structure must be modified.

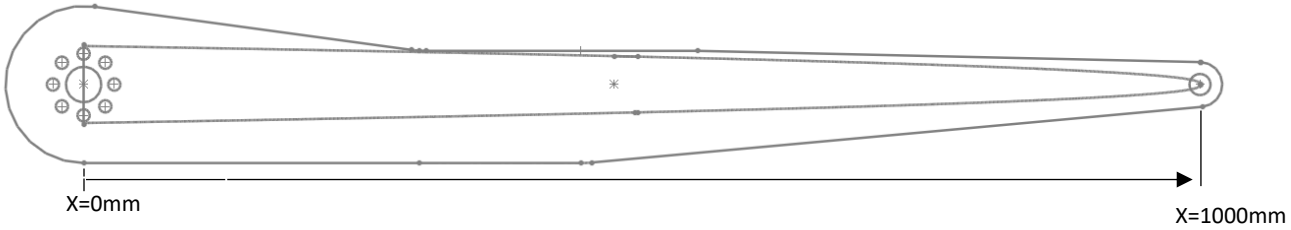



*Figure 3-22: Comparison between original link and optimized link*

Combining the profiles to adapt connections and the optimized structure results in Figure 3-23 below. Splines were added to connect the two designs from x=0 to 150mm, and x=900mm to 1000mm, leaving 750mm (75%) of the beam with the optimized structure. During the optimization 1 beam was considered with width of 1cm. For fabrication, the beam was divided

into two plates with width 0.5cm each. 3 plates were added to the top and bottom to prevent buckling, and to compensate for any twisting that occurs in the beams caused by unbalanced loading of the end effector which was neglected during the optimization.

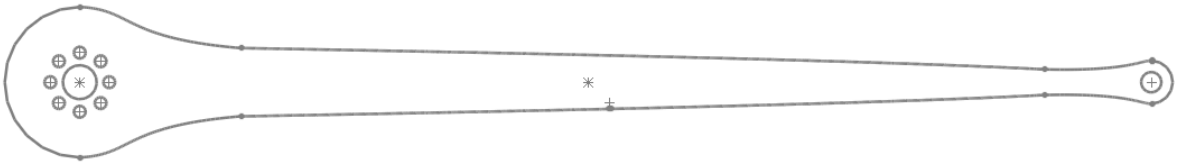


*Figure 3-23: Final shape of the inner link*

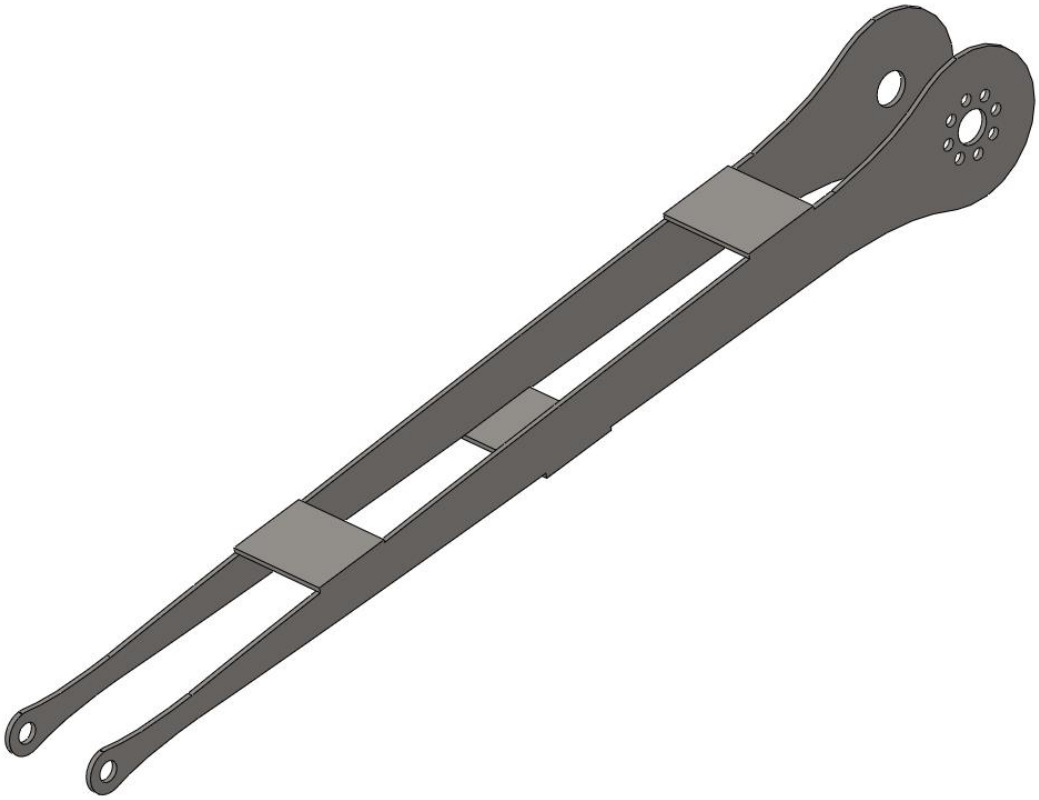


*Figure 3-24: 3-d model of the inner link after optimization and modification*

Figure 3-24 shows the completed 3d model of the inner link after optimization has been completed. The part is made from 5mm thick structural steel and has a total mass of 5.55kg. As the profile of the main beam structure is not easily dimensioned in engineering drawings the fabrication of this part would be based off of a CAD file provided to the manufacturer and be

cut using a plasma cutter. This optimized structure can easily replace the original inner link without any modifications to the base plate, or to the linear actuator.

After comparing with Equation 3-37 for buckling, the critical load for buckling will be 6,700N when choosing the minimal dimensions of the beam to be 1cm x 2cm, which is well above the forces experienced by the optimized beam.

Additional properties of the optimized link

$I_c = [0.014,0,-0.001;\ 0\ ,0.486,0;\ -0.001,0,0.477] kg \cdot m^2$

$P_c = [362.7; 1.4; 0]\ mm$

The torque requirements at joint 2 while the robot travels the path described above were calculated using the dynamic equations presented in chapter 2. Figure 3-25 below compares the torque requirements of the robot with the original inner link to the torque requirements of the robot with the optimized inner link. By using the optimized inner link the torque required is reduced by 12-11% over the duration of the trajectory.

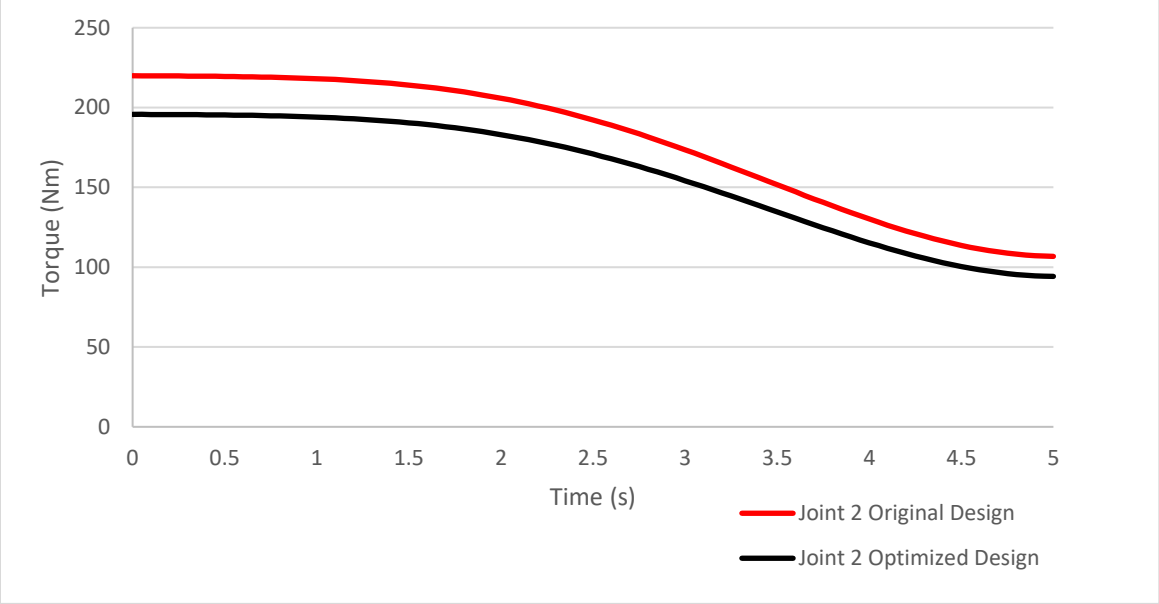



*Figure 3-25: Torque requirements of joint 2 with original and optimized link*

After the modifying the inner link were made to ensure that the connections to the linear actuator and the base plate were maintained the structure retained 75% of the optimized

shape. A final FE analysis was performed on the completed structure to ensure that neither of the optimization constraints were violated during the modification. The convergence analysis completed in ANSYS workbench compares different amounts of elements for tetrahedral and hexahedral element types. The model is loaded as described earlier in the chapter, with the base connection being fixed. Figure 3-26 shows that the maximum stress in the link converges to approximately 151 MPa when stress concentration are taken into account. This is below the 175.0 MPa constraint limit.

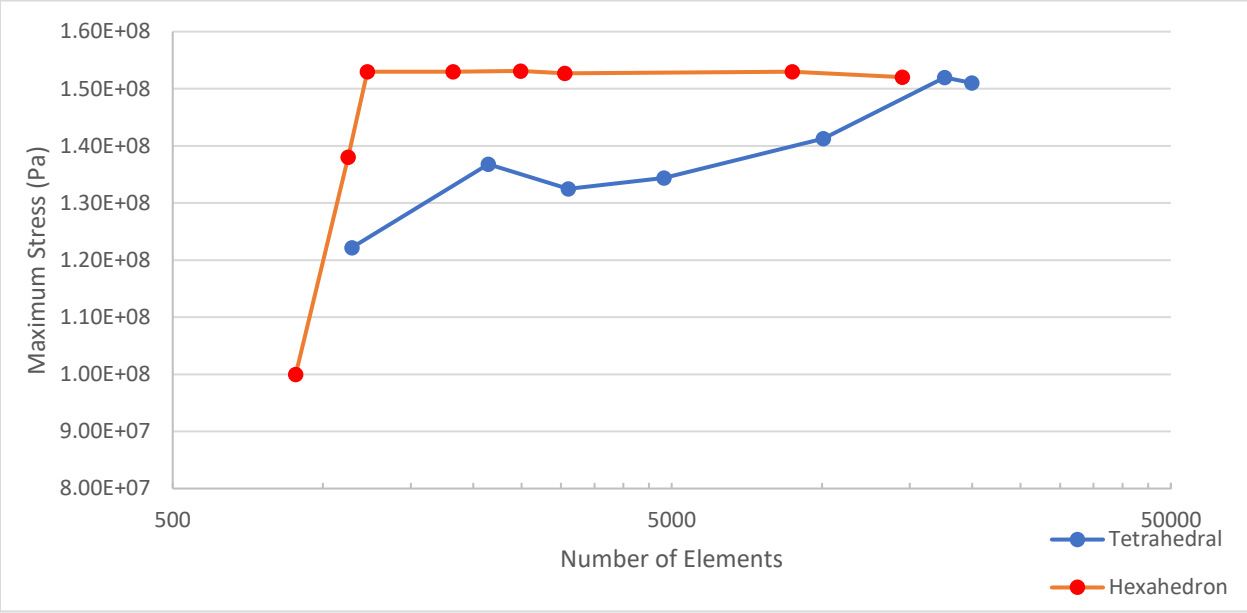



*Figure 3-26: Maximum stress found in final optimized link*

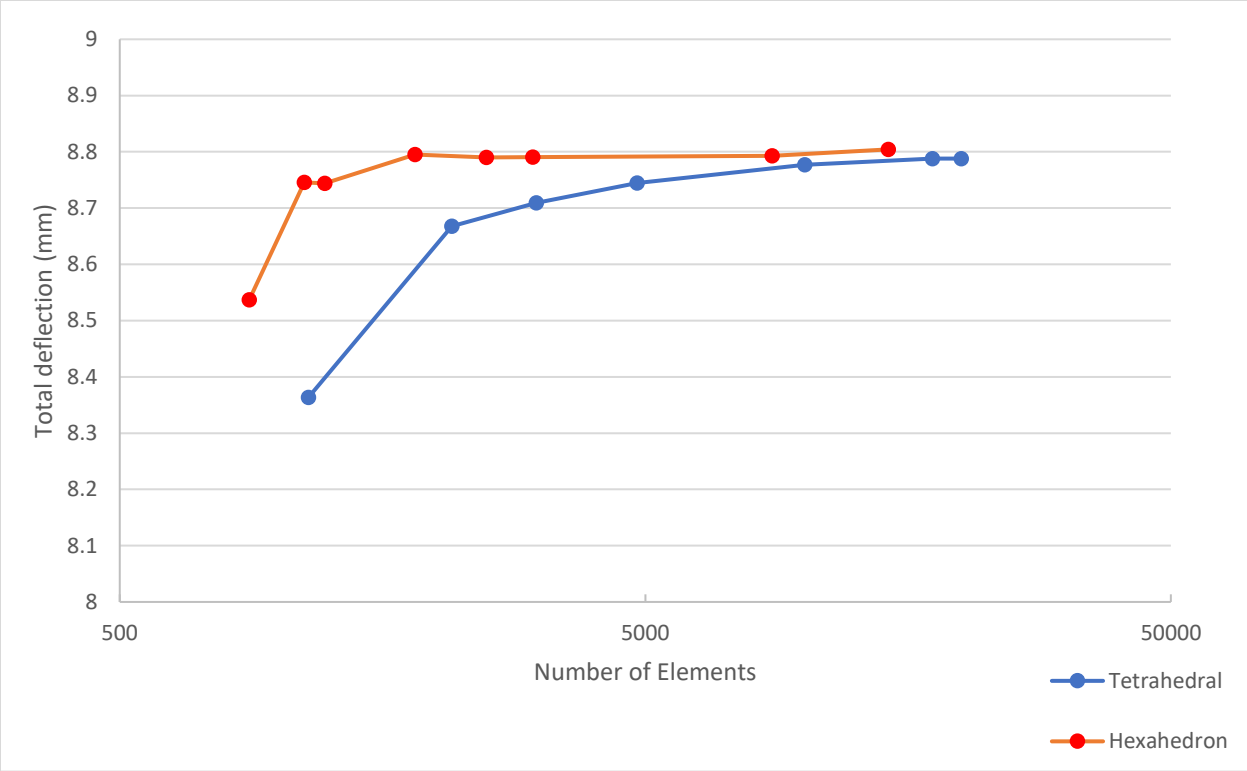



*Figure 3-27: Maximum tip deflection found in the final optimized link*

Figure 3-27 shows that the maximum deflection of the beam converges to approximately 0.9cm which is below the 1cm limit but does not violate the maximum deflection constraint. While different from the target constraints, the values for maximum stress reached and maximum deflection of the beam are acceptable because the link has been changed from the purely optimized structure by introducing modifications for fabrication. In future work on this topic, both of the ends of the inner link should be included in the optimization analysis so that the beam can be further optimized to reach the constraints.

## 3.7: Discussions and Summary

*Table 3-3 Summary of Optimization results*

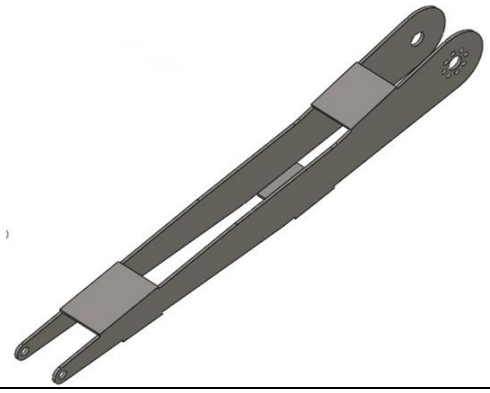
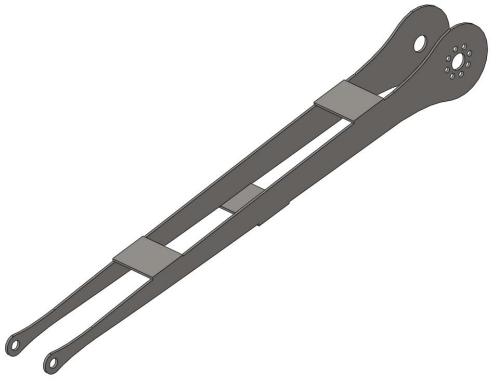
| | Original Beam | Optimized Beam |
|---|---|---|
| Structure |  |  |
| Mass | 10.90 Kg | 5.55Kg |
| Maximum deflection | 0.02mm | 10mm |
| Maximum stress | 87.8 MPa | 174.9 MPa |

Table 3-3 shows the results of the optimization. The first stage of the optimization computed the reaction forces and maximum stress of each slice and increased the height of the slice until the maximum stress was below the limit. After each slice was examined the resulting tapered beam was analysed in ANSYS APDL. The maximum stress found was 175.0 MPa, which satisfied the maximum stress constraint. However, during loading the beam deflected 2.1cm, over double the allowable deflection limit.

The second stage of the optimization examined each slice, and the contribution to total deflection of the tip of the beam was recorded. The sensitivity of the tip deflection was analysed for each slice. The slice that had the most potential to reduce tip deflection had it's size increased and the process repeated until the tip of the beam was within acceptable deflection limits. After the second stage of optimization was completed a 3-d model of the link was created in SOLIDWORKS and the link was analyzed in FE software. After convergence analysis of the model, the total deflection of the beam was 1.0014cm. While above the optimization constraint, the value is acceptable within error.

After verifying the results of the optimization did not violate any of the optimization constraints, the beam was modified for manufacturability. The pin connections of the original inner link were preserved, and splines were added in SOLIDWORKS to bridge the two designs. Ultimately 75% of the optimized structure was preserved, and the total mass of the redesigned inner link was 5.55kg, compared to the 10.9 kg of the original link. This represents a 49% reduction in weight. The reduction in weight of the inner link contributes to lowering the risk of soil compaction during data collection. Additionally, the reduced weight of the link allows a 11% reduction in torque required by the joint 2 motor.

Several sources of error can be identified in this chapter. The boundary conditions were calculated using the method described in chapter 2 of this thesis and suffer from the same sources of error described above. Several forces were not included in the boundary conditions as well. Frictional forces in the joints were assumed to be 0 in this analysis, and twisting moments at frame 8 were also neglected in the optimization program. The optimization neglects any unbalanced loading which can be caused by improper loading of the end effector. An unbalanced end effector will produce forces that are not symmetrical as assumed by this analysis.

After the optimization program was run in MATLAB the heights of each slice were exported to SOLIDWORKS and generated into a 3d solid. One source of error is the spline function in SolidWorks; the curve generating algorithm in SolidWorks does not perfectly fit all the data points generated by the optimization program, thus there is some small difference in the finalized optimization predicted by the MATLAB program, and the 3d model analysed in ANSYS workbench.

As this optimization was only verified by comparing analytical to simulated values, experimentation on the physical part may not produce exact values as predicted. Manufacture of the part can add stress concentrations to the structure and dimensions will not be precise.

# Chapter 4: Conclusions

The global increase of people who experience food insecurity over the past 5 years is a worrying trend. Crop scientists have made large contributions to plant genetics, producing plants that are more resistant to disease, pests, and climate uncertainty, helping produce sufficient nutrients to ensure a healthy lifestyle for a growing population. One of the challenges prohibiting more rapid results from plant breeders is the phenotyping bottleneck, where the act of measuring plants is expensive and slow. To alleviate the phenotyping bottleneck the robotics lab at the University of Saskatchewan has designed a robotic arm to be used with an agricultural vehicle. The arm will carry phenotyping sensors to rapidly collect data from wheat and canola crops. The robot arm was analysed in the chapters above to improve useability and capacity.

The first research question presented in this thesis was:
What kind of analysis is appropriate for the development of a 5 DOF robot manipulator?

In chapter 2 the kinematics and dynamics of the 3$^{rd}$ generation boom were analysed. Using the Denavit-Hartenburg parameters the analytical transformation matrices from the base of the robot to the end effector were created. A model of the 3$^{rd}$ generation boom was imported into the rigid dynamic module of ANSYS workbench and the model was simulated undergoing a 10s test trajectory that was developed. While the robot arm followed the motions of the test trajectory the position of the end effector was calculated by the analytical solutions in MATLAB and simulated in ANSYS workbench. The results of the experiment were promising; data was recorded at every 0.1s of the trajectory and the analytical solution and the simulation agreed within millimetre accuracy.
In the second half of chapter 2 the forward dynamics of the 3$^{rd}$ generation boom were analyzed. The original presentation of the Newton-Euler equations of motion in the D-H parameter form was only applicable to serial robots, not parallel robots such as the robotic arm. Additional analytical equations to describe the forward dynamics were found by applying the Newton-Euler equations to additional links. The analytical equations developed were then

programmed with the test trajectory and compared with a simulated model of the robot following the same trajectory. Both the simulated model and analytical equations predicted the required torques at the rotational joints and the forces at the prismatic joint to produce the specified motion. Values of moments and forces were recorded at every 0.1s of the 10s trajectory. When the results of the analytical equations and the simulated model were compared it was found that the two experiments agreed with each other, a maximum 0.23% difference was found during the test.

As the analytical results between the first and the second half of chapter 2 agree closely with the simulated results, it can be concluded that the analytical equations derived are appropriate for analyzing the motion of the robot arm. However, ideally it is better to perform physical experiments to validate the equations derived.

Appropriate methods for deriving the dynamic equations of a parallel robot allow the motion of the robot to be predicted and controlled. Operators of the robotic arm can specify paths and trajectories for the robot arm to undergo, which in turn allows for new tasks to be specified for the robot.

The second research question of this thesis was:
Can the weight of individual links be minimized to reduce the overall weight of the robot arm?

Chapter 3 focuses on the optimization of the inner link. The objective function of the optimization is the mass of the optimization while maximum stress and maximum tip deflection are the optimization constraints. The inner link is modeled as a beam and split into 1000 slices with each slice having an initial height of 0m. The maximum stress is calculated for each slice and the height of the slice is increased by $\Delta H$ until the maximum stress constraint is not violated. The results of the first stage of optimization gave a 3-d beam contour. When analyzed in ANSYS APDL using beam 188 elements the maximum stress experienced by the beam was 175.0 MPa, satisfying the maximum stress constraint, but the deflection of the beam was found to be 2.1 cm, well over the 1cm deflection limit.

During the second stage of optimization the beam the contribution of each slice to the deflection of the tip of the beam was calculated. The contribution of the slice was compared to the contribution of the slice if the height was changed by ΔH. The slice that is most sensitive to the change has the change made permanent and the process was repeated until the deflection was within allowable limits. This process produced a second 3-d beam contour which was analysed in ANSYS workbench. The analysis showed that the beam produced a maximum deflection of 1.0014cm while loaded. While above the 1cm limit, the result is within acceptable error. The final optimized structure was modified to ensure that the original pin connections of the inner link to the base plate and the linear actuator were maintained. After modification 25% of the beam was replaced to maintain the connections at either end leaving 75% of the optimized structure intact. The original inner link had a mass of 12.1kg, while the optimized link had an end weight of 5.55kg resulting in a 49% reduction in mass.

By reducing the weight of the inner link the 3$^{rd}$ generation boom contributes less to soil compaction during data collection than before. Additionally, the lower mass of the link reduced the power requirements for the joint 2 motor by 11% allowing for a smaller, lighter motor to replace the current chosen joint 2 motor. After completing the optimization verifying the results objective 2 of this thesis has been completed.

## 4.1: Future work

There are several areas where further work could be done in the future.

1. Physical experiments.

   As the robot arm has not been fabricated at the time of writing, no physical experiments have been conducted. Kinematic and dynamic experiments should be conducted against the analytical equations to test their accuracy.

2. Optimization of the connection points of the inner link

   The inner link was modified from the optimized design to accommodate the pin connections at either end. These were not included in the optimization due to the complexities of stress concentration effects. Further research in this area can reduce the mass of the inner link even further.

3. Further optimization of the lower link, outer link, and actuator.

   Using the optimization program developed in chapter 3 of this thesis additional links of the robot could be subjected to optimization and the mass of the robot could be further reduced.

4. Consider the effects of vibrations and flexible links.

   In this thesis each of the links of the robot were assumed to be rigid, however, as the robot links are quite long this assumption should be examined and investigated. Additionally the effects of vibrations should be included to create more accurate results of both the dynamic and kinematic equations.

5. Explore additional methods of optimization

   The use of lightning holes, different styles of beam other than two parallel plates and different materials are all methods to further reduce the mass of the inner link. Exploring these additional areas can provide additional mass optimization. Additionally, optimization that includes the ends of the link (connections) should be included.

# References


Andrade-Sanchez, Pedro, Michael Gore, John Heun, Kelly Thorp, Elizabete Carmo-Silva, Andrew French, Michal Salvucci, and Jeffery White. 2013. "Development and evaluation of a field-based high-throughput phenotyping platform." *Functional Plant Biology* 41 (1): 68-79.

Ansys Support. 2011. *Ansys rigid Body Dynamics & Simplorer.* October 12. Accessed September 23, 2020. https://support.ansys.com/staticassets/ANSYS/Conference/Irvine/downloads/ANSYS%20Rigid%20Body%20Dynamics%20and%20ANSYS%20Simplorer.pdf.

Borlaug, Norman. 1983. "Contributions of Conventional Plant Breeding to Food Production." *Science* (219): 689-693. doi:10.1126/science.219.4585.689.

Center for Biosafety and Sustainability. 1995. *Methods of Plant Breeding.* September 17. Accessed September 09, 2020. https://www.bats.ch/bats/publikationen/1995-1_TA/2-methods.php.

Collard, Bertrand, and David Mackill. 2007. "Marker-assisted selection: an approach for precision plant breeding in the twenty-first century." *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 557-572. doi: doi: 10.1098/rstb.2007.2170.

Craig, John J. 2005. *Introduction to Robotics: Mechanics and Control.* Upper Saddle River: Pearson Education International.

Dengyu, Xiao, Gong Liang, Liu Chengliang, and Huang Yixiang. 2016. "Phenotype-based Robotic Screening Platform for Leafy Plant Breeding." *IFAC-PapersOnLine* 237-241.

Dreery, David, Jose Jimenez-Berni, Hamlyn Jones, Xavier Sirault, and Robert Furbank. 2014. "Proximal Remote Sensing Buggies and Potential Applications for Field-Based Phenotyping." *Agronom* 349-379.

FAO, IFAD, UNICEF, WFP and WHO. 2020. *The State of Food Security and Nutrition in the World 2020.* Rome: FAO. doi:https://doi.org/10.4060/ca9692en.

Flowers, TJ, and AR Yeo. 1995. "Breeding for Salinity Resistance in Crop Plants: Where Next?" *Australian Journal of Plant Physiology* 22 (6): 875-884.

Furbank, Robert, and Mark Tester. 2011. "Phenomics – technologies to relieve the phenotyping bottleneck." *Trends in plant science* 16 (12): 635-644.

Jansen, Marcus, Frank Gilmer, Bernhard Biskup, Kerstin Nagel, Uwe Rascher, Andreas Fischbach, Sabine Briem, et al. 2009. "Simultaneous phenotyping of leaf growth and chlorophyll fluorescence via GROWSCREEN FLUORO allows detection of stress tolerance in Arabidopsis thaliana and other rosette plants." *Functional Plant Biology* 902-914.

Jobson, Emma, Rachel Johnston, Alanna Oiestad, John Martin, and Michael Giroux. 2019. "The Impact of the Wheat Rht-B1b Semi-Dwarfing Allele on Photosynthesis and Seed Development Under Field Conditions." *Frontiers in Plant Science* 10 (51). doi:https://doi.org/10.3389/fpls.2019.00051.

Kicherer, Anna, Katja Herzog, Nele Bendel, Hans-Christian Kluk, Andreas Backhaus, Markus Wieland, Johann Christian Rose, et al. 2017. "Phenoliner: A New Field Phenotyping Platform for Grapevine Research." *Sensors* 1625.

LemnaTec. n.d. *Field Scanalyzer.* Accessed September 16th, 2020. https://www.lemnatec.com/customized-solutions/field-scanalyzer/.

Maes, Wouter, and Kathy Steppe. 2019. "Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture." *Trends in Plant Science* 152-164.

Mano, Masayoshi. 2017. "Precise and continuous measurement of plant heights in an agricultural field using a time-lapse camera." *Journal of Agricultural Meteorology* 100-108.

Mueller-Sim, Tim, Merritt Jenkins, Justin Abel, and George Kantor. 2017. "The Robotanist: A Ground-Based Agricultural Robot for High-Throughput Crop Phenotyping." *2017 IEEE International Conference on Robotics and Automation (ICRA).* Singapore,: IEEE.

Munns, Rana, Richard James, Xavier Sirault, Robert Furbank, and Hamlyn Jones. 2010. "New phenotyping methods for screening wheat and barley for beneficial responses to water deficit." *Journal of Experimental Botany* 3499-3507.

Naito, Hiroki, Satoshi Ogawa, Milton Orlando Valencia, Hiroki Mohri, Yutaka Urano, Fumiki Hosoi, Yo Shimizu, et al. 2017. "Estimating rice yield related traits and quantitative trait loci analysis under different nitrogen treatments using a simple tower-based field phenotyping system with modified single-lens reflex cameras." *ISPRS Journal of Photogrammetry and Remote Sensing* 50-62.

Olivera, Pablo, Matthew Rouse, and Yue Jin. 2018. "Identification of New Sources of Resistance to Wheat Stem Rust in Aegilops spp. in the Tertiary Genepool of Wheat." *Plant Science* 9 (1719). doi:doi: 10.3389/fpls.2018.01719.

Purdue University. 2003. *Plant Breeding Reviews Volume 22.* Hoboken: John Wiley & Sons, Inc.

Reyolds, M, R Singh, A Ibrahim, O Ageeb, A Larque-Saavendra, and J Quick. 1998. "Evaluating physiological traits to complement empirical selection for wheat in warm environments." *Euphutica* (100): 85-94.

Ruckelshausen, A, P Biber, M Dorna, H Gremmes, R Klose, A Linz, R Rahe, et al. 2009. "Bonirob: an autonomous field robot platform for individual plant phenotyping." *Precision Agriculture* 841-847.

Sadeghi-Tehran, Pouria, Kasra Sabermanesh, Nicolas Virlet, and Malcolm Hawkesford. 2017. "Automated Method to Determine Two Critical Growth Stages of Wheat: Heading and Flowering." *Frontiers in Plant Science.* doi:https://doi.org/10.3389/fpls.2017.00252.

Schumann, G.L., and K.J. Leonard. 2000. "Stem rust of wheat (black rust)." *The Plant Health Instructor.* doi:DOI: 10.1094/PHI-I-2000-0721-01.

Schurr, Uli, and Roland Pieruschka. 2019. "Plant Phenotyping: Past, Present, and Future." *Plant Phenomics* 6 pages.

Shafiekhani, Ali, Suhas Kadam, Felix Fritschi, and Guilherme DeSouza. 2017. "Vinobot and Vinoculer: Two Robotic Platforms for High-Throughput Field Phenotyping." *Sensors* 214.

Svensgaard, Jesper, Thomas Roitsch, and Svend Christensen. 2014. "Development of a Mobile Multispectral Imaging Platform for Precise Field Phenotyping." *Agronomy* 322-336.

Trethowan, Richard, Maarten Van Ginkel, and Sanjaya Rajaram. 2002. "Progress in Breeding Wheat for Yield and Adaptation in Global Drought Affected Environments." *Crop Breeding, Genetics & Cytology* (42): 1441-1446. doi:https://doi-org.cyber.usask.ca/10.2135/cropsci2002.1441.

Underwood, James, Alexander Wendel, Brooke Schofield, Larn Mcmurray, and Rohan Kimber. 2017. "Efficient in-field plant phenomics for row-crops with an autonomous ground vehicle." *Journal of Field Robotics* 1061-1083.

UNICEF. 2019. *The State of the World's Children 2019. Children, Food and Nutrition: Growing well in a changing world.* New York: UNICEF.

White, Jeffery, Pedro Andrade-Sanchez, Michael Gore, Kevin Bronson, Terry Coffelt, Matthew Conley, Kenneth Feldmann, et al. 2012. "Field based phenomics for plant genetics research." *Field Crops Research* 101-112.

White, Jeffrey, and Matthew Conley. 2013. "A Flexible, Low Cost Cart for Proximal Sensing." *Crop Science* 1646-1649.

Yang, Wanneng, Hui Feng, Xuehai Zhang, Jian Zhang, John Doonan, William David Batchelor, Lizhong Xiong, and Jianbing Yan. 2020. "Crop Phenomics and High-Throughput Phenotyping: Past Decades, Current Challenges, and Future Perspectives." *Molecular Plant* 187-214.

Yu, Ming-Han, Guo-Dong Ding, Gao-Guang-Lei, Yuan-Yuan Zhao, Lei Yan, and Ke Sai. 2015. "Using Plant Temperature to Evaluate the Response of Stomatal Conductance to Soil Moisture Deficit." *Forests* 3748-3762.

Zhang, Chunhua, and John Kovacs. 2012. "The application of small unmanned aerial systems for precision agriculture: a review." *Precision Agriculture* 693-712.

Zhang, Qianwei. 2018. *Design and Analysis of Booms for Wheeled Mobile Platform for Crop Phenotyping.* Graduate Thesis, Saskatoon: University of Saskatchewan.

Zhang, Qianwei, and Reza Fotouhi. 2018. "vibration analysis of a long boom for a farm machine." *1 Copyright © 2018 by ASMEProceedings of the ASME 2018 International Design Engineering Technical Conferences & Computers and Information in Engineering Conferenc.* Quebec.

# Appendix A: DH parameters

The following is a direct excerpt from Introduction to Robotics by John Craig. (Craig 2005)

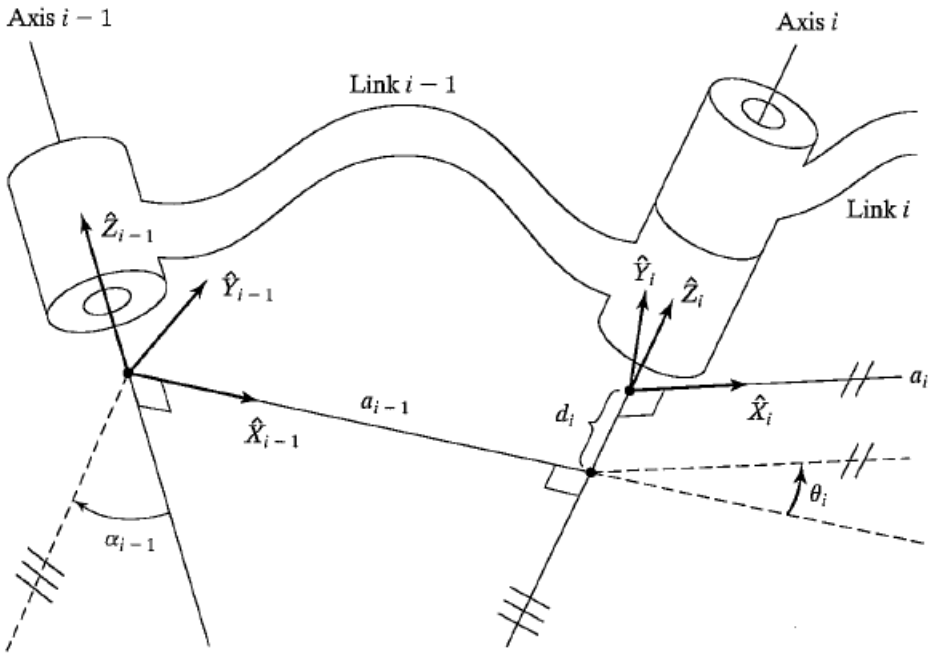



*Figure A.1 An diagram illustrating the measurements of DH parameters (Craig 2005)*

$$a_i = \text{the distance from } \hat{Z}_i \text{ to } \hat{Z}_{i+1} \text{ measured along } \hat{X}_i$$

$$\alpha_i = \text{the angle from } \hat{Z}_i \text{ to } \hat{Z}_{i+1} \text{ measured about } \hat{X}_i$$

$$d_i = \text{the distance from } \hat{X}_{i-1} \text{ to } \hat{X}_i \text{ measured along } \hat{Z}_i$$

$$\theta_i = \text{the angle from } \hat{X}_{i-1} \text{ to } \hat{X}_i \text{ measured along } \hat{Z}_i$$

The following are steps used to assign frames to joints:

1. Identify the joint axes and imagine (or draw) infinite lines along them. For steps 2 through 5 below, consider two of these neighboring lines (at axes *i* and *i*+1)
2. Identify the common perpendicular between them or point of intersection. At the point o intersection, or at the point where the common perpendicular meets the *i*th axis, assign the link-frame origin.
3. Assign the $\hat{Z}_i$ axis pointing along the *i*th joint axis.
4. Assign the $\hat{X}_i$ axis pointing along the common perpendicular, or , if the axes intersect, assign $\hat{X}_i$ to be normal to the plane containing the two axes.

5. Assign the $\hat{Y}_i$ Axis to complete a right-hand coordinate system.
6. Assign frame {0} to match frame {1} when the first joint variable is zero. For frame {N}, choose as origin location and $\hat{X}_N$ direction freely, but generally so as to cause as many linkage parameters as possible to become zero.

# Appendix B: FEA details

Below are images of Ansys Workbench detailing the inputs for the analysis.

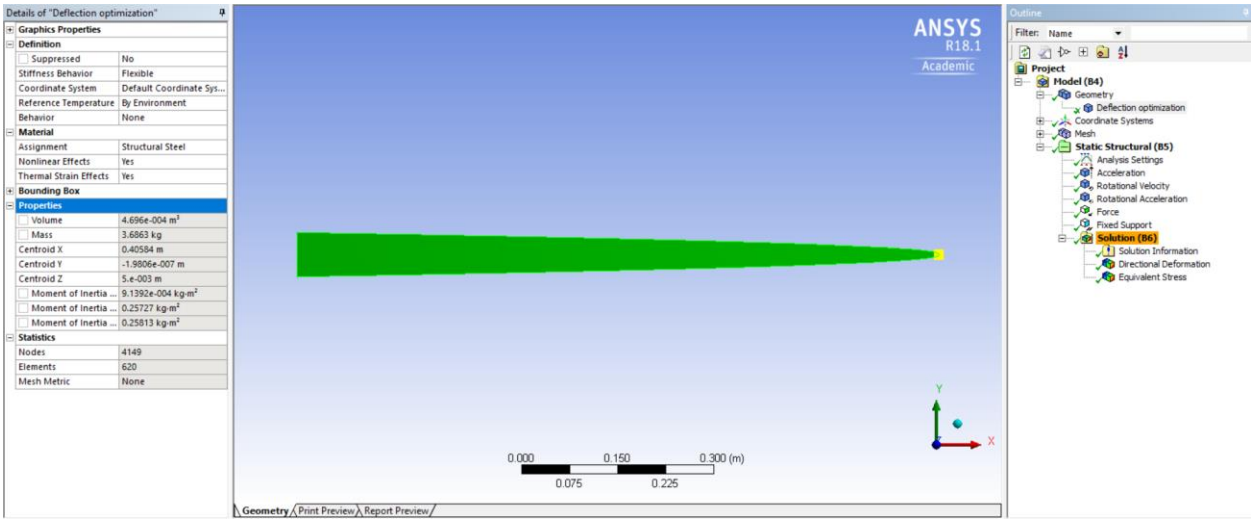



*Figure B.1: Material properties*

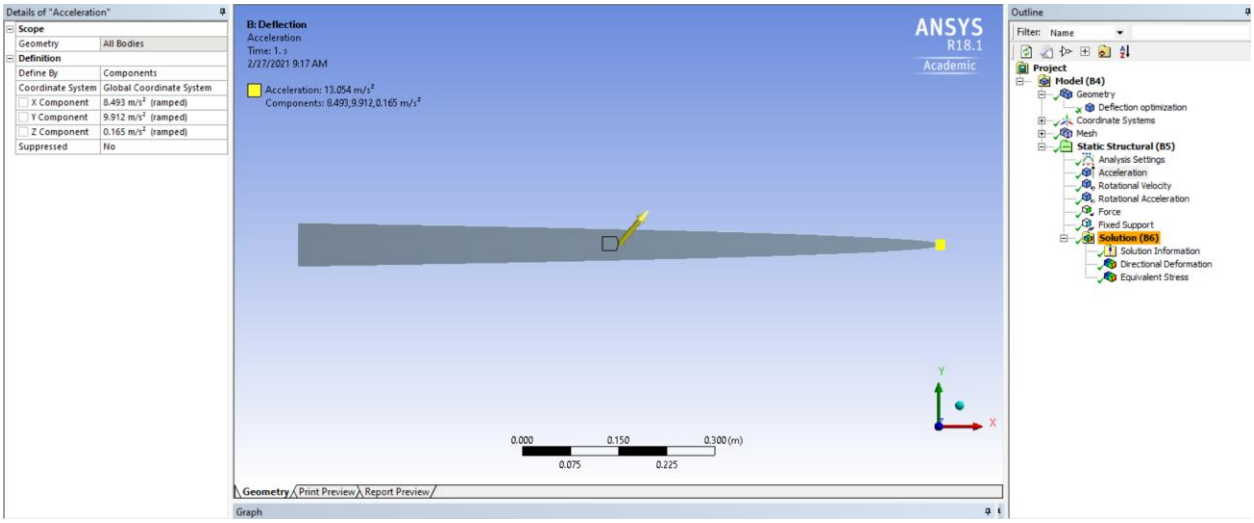



*Figure B.2: Accelerations*

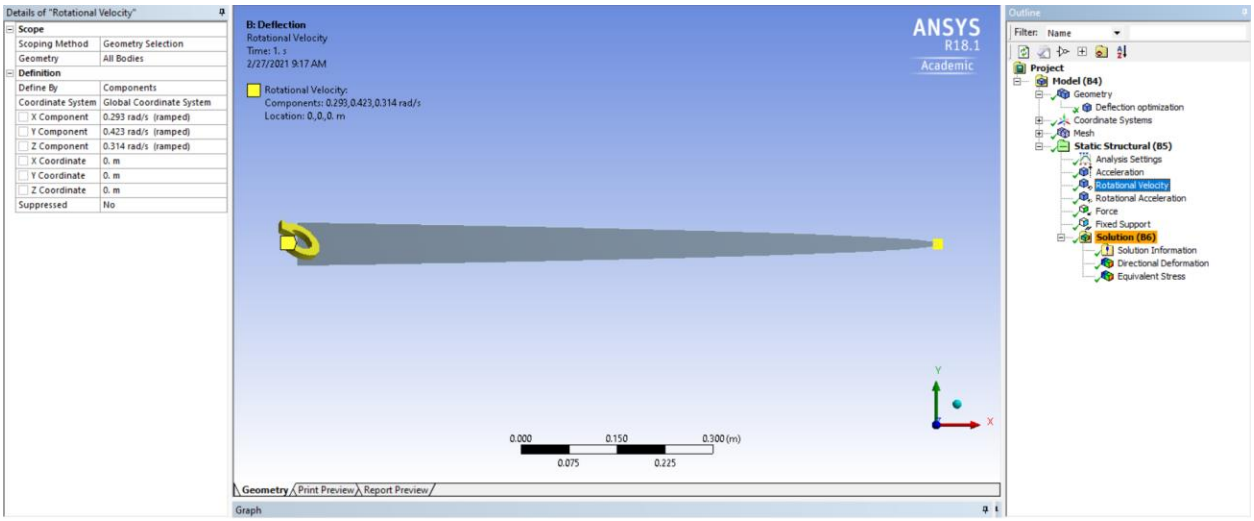



*Figure B.3: Rotational velocities*

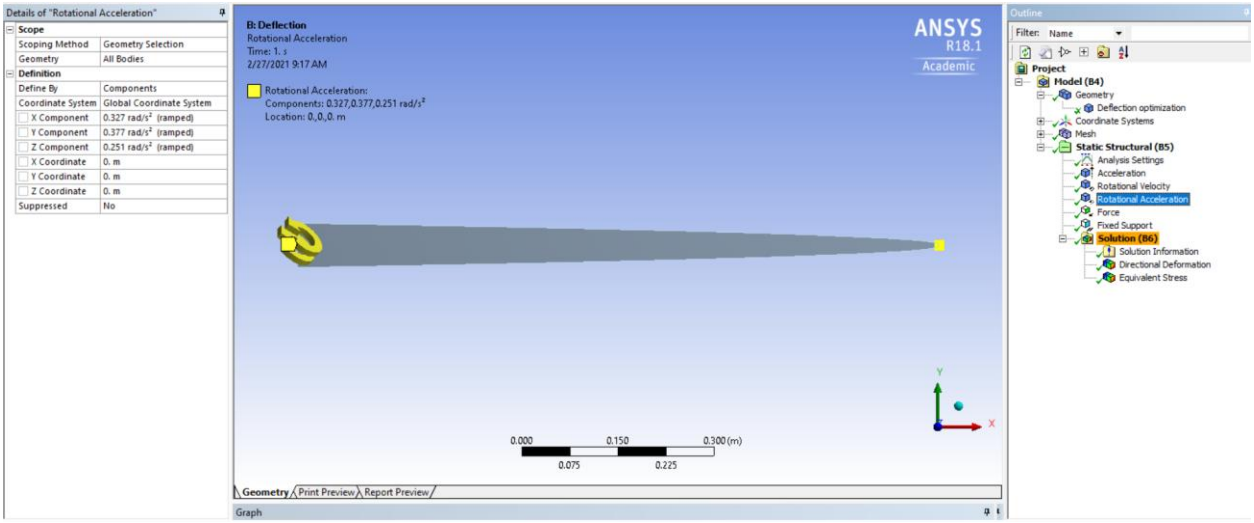



*Figure B.4: Rotational Accelerations*

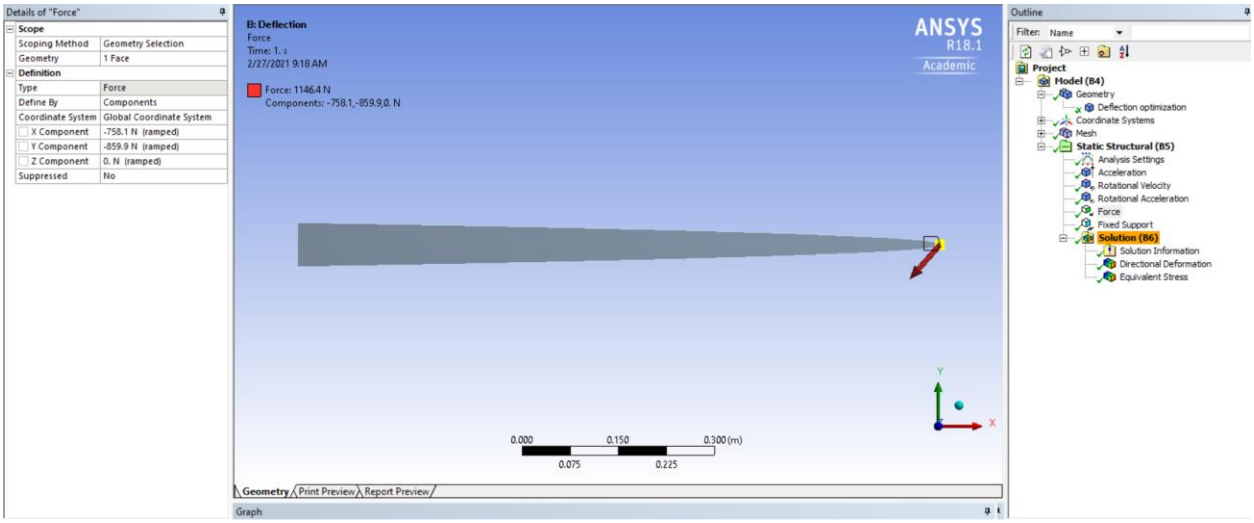



*Figure B.5: Applied Forces*

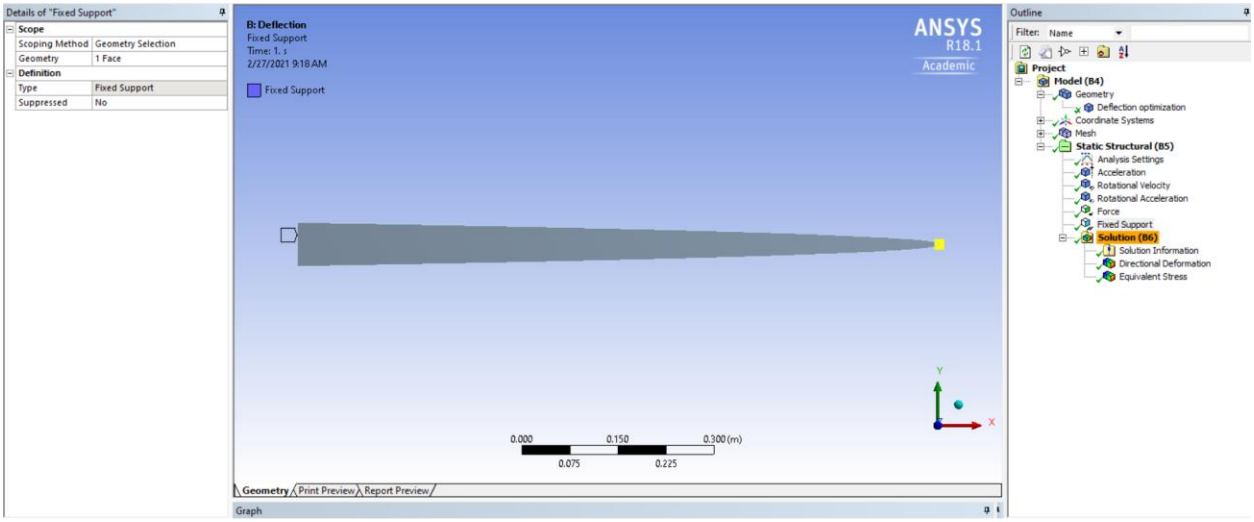



*Figure B.6: Fixed support*

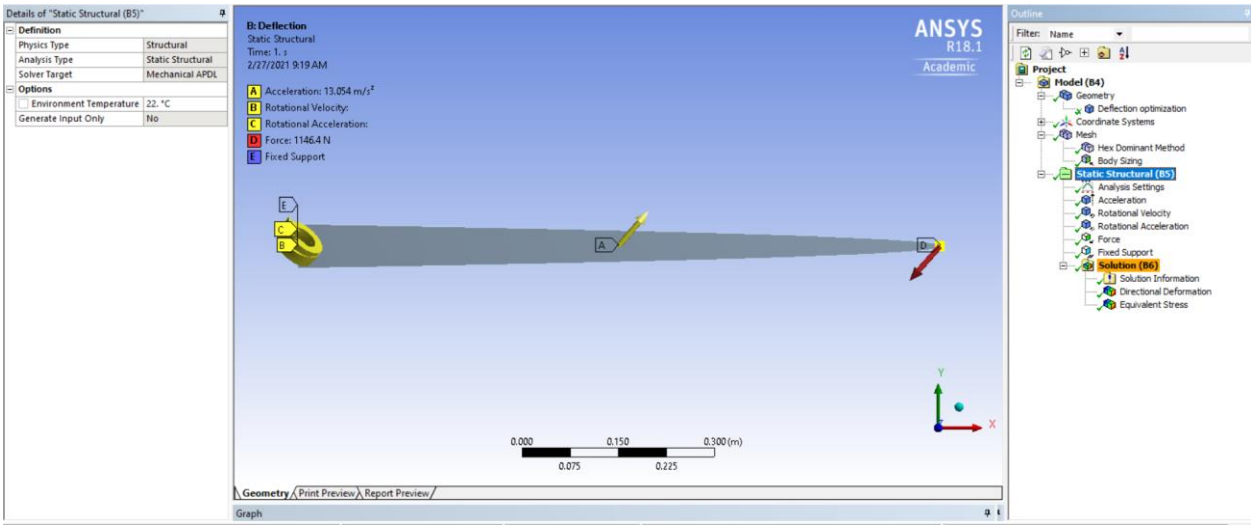

Details of "Static Structural (B5)"
Definition
Physics Type
Structural
Analysis Type
Static Structural
Solver Target
Mechanical APDL
Options
Environment Temperature
22. °C
Generate Input Only
No
B: Deflection
Static Structural
Time: 1. s
2/27/2021 9:19 AM
A Acceleration: 13.054 m/s²
B Rotational Velocity:
C Rotational Acceleration:
D Force: 1146.4 N
E Fixed Support
ANSYS
R18.1
Academic
0.000
0.150
0.300 (m)
0.075
0.225
Geometry
Print Preview
Report Preview
Graph
Outline
Filter:
Name
Project
Model (B4)
Geometry
Deflection optimization
Coordinate Systems
Mesh
Hex Dominant Method
Body Sizing
Static Structural (B5)
Analysis Settings
Acceleration
Rotational Velocity
Rotational Acceleration
Force
Fixed Support
Solution (B6)
Solution Information
Directional Deformation
Equivalent Stress

*Figure B.7*

# Appendix C – MATLAB code for kinematics and dynamics

** the development of this code can be found in section 2.2.1 and 2.2.2 above**

```
clc
clear
clf
%% intro/definitions
%V1_0 created DH parameters and transformation matrix
%V1_1 finished velocities and rotational velocities
%V1_2 redid dh parameters to have 2 separate transformation paths through
%both links
%V1_3 finished inwards forces and moments.
%V1_4 finished accelerations
%V1_5 performed velocities and accelerations for both links.
%V1_6 added the inertia subfunctions
%V2_0
%V2_1 removed inertia subfunctions
%--------------------------------------------------------------------------
%frame (i) to (i+1) is connected by link(i)
%--------------------------------------------------------------------------
%D-H parameters
%theta, and d have subscripts,i
%alpha and a have subscripts i-1
%Transforms from frame i-1 to i
%Transform subscript (i,i+1) = transformation from i to i+1
%--------------------------------------------------------------------------
%--------------------------------------------------------------------------
%lengths of links (mm/1m)
L1=110/1000;
L2=30/1000;
L3=300/1000;
```

```
L4=1000/1000;

L5=L3;%if the links are not equal, the links will not be parallel which will effect my derived expressions for theta.

L6=42.08/1000;

L7=759.505/1000;

L8=40/1000;

L9=L4;%if the links are not equal, the links will not be parallel which will effect my derived expressions for theta.

Lextension=1059.505/1000; % distance from frame 9 to end support

Lactuator=717.016/1000; % with respect to frame 8 to center of end support

%% Trajectory

thetainitial=[0,pi/2,pi,0,0];

thetafinal=[pi/2,pi/4,5*pi/6,0.5,pi/2];

totaltime=10

for i=1:5

a0(i)=thetainitial(i)

a1(i)=0

a2(i)=(3/(totaltime^2))*(thetafinal(i)-thetainitial(i));

a3(i)=(-2/(totaltime^3))*(thetafinal(i)-thetainitial(i));

end

%% INPUTS

%DOF inputs, thetas are measured from the original +x horizontal and are in radians, x is in m

steps=100

for currentstep=1:steps+1

 time=totaltime*(currentstep-1)/steps;

 for i=1:5

 pathposition(i)=a0(i)+a1(i)*(time)+a2(i)*(time^2)+a3(i)*(time^3);

 pathvelocity(i)=a1(i)+2*a2(i)*(time)+3*a3(i)*(time^2);

 pathaccel(i)=2*a2(i)+6*a3(i)*time;

 end

theta1=pathposition(1);
```

```
theta2=pathposition(2);
theta3=pathposition(3);
d1=pathposition(4);
theta5=pathposition(5);
theta1dot=pathvelocity(1);
theta2dot=pathvelocity(2);
theta3dot=pathvelocity(3);
d1dot=pathvelocity(4);
theta5dot=pathvelocity(5);
theta1ddot=pathaccel(1);
theta2ddot=pathaccel(2);
theta3ddot=pathaccel(3);
d1ddot=pathaccel(4);
theta5ddot=pathaccel(5);
% %positions in radians
% theta1=0;
% theta2=pi/2;
% theta3=pi;
% d1=0;
% theta5=0;

% % velocities in rad/s
% theta1dot=0;
% theta2dot=0;
% theta3dot=0;
% d1dot=0;
% theta5dot=0;
%
% % Accelerations in rad/s^2
% theta1ddot=0;
```

```
% theta2ddot=1;
% theta3ddot=0;
% d1ddot=0;%linear acceleration of the prismatic joint "d1 double dot"
% theta5ddot=0;
%inital frame accelerations: %% ignore the - sign infront of the vector,
%and place the values as if they were in frame 1, or 11.
rotvelocity1{1}=-[0;0;0]; % base link rotational velocity
velocity1{1}=-[0;0;0]; % Base frame velocity
rotaccel1{1}=-[0;0;0];
accel1{1}=-[0;0;-9.81];
f1{11}=-[0;0;-9.81*10]; % load on end effector relative to frame 1
n1{11}=-[0;0;0];
positionsummary=[theta1,theta2,theta3,d1,theta5]*180/pi;% output in degrees for ease of reading
velocitysummary=[theta1dot,theta2dot,theta3dot,d1dot,theta5dot];
accelerationsummary=[theta1ddot,theta2ddot,theta3ddot,d1ddot,theta5ddot];
%%
%outer loop transformations
%-------------------------------------------------------------------------
% DH Parameters (outer)
alphaLoop1=[0,0,0,90,0,0,0,90,90,0];
aLoop1=[0,0,L2,0,L3,L4,L5,L6,0,L8];
dLoop1=[0,L1,0,0,0,0,0,L7+d1,0,0];
thetaLoop1=[theta1*180/pi,0,0,theta3*180/pi,(360-theta3*180/pi+theta2*180/pi),(180+theta3*180/pi-
theta2*180/pi),90,180,theta5*180/pi,0];
%-------------------------------------------------------------------------
% creation of the transformation matrix for frame transformation (outer)
NFrame=11;
for i=1:NFrame
   for j=1:NFrame
      T1(i,j)={eye(4)};
```

```
    end
end
for i=1:NFrame-1
    T1(i,i+1)={[cosd(thetaLoop1(i)),-sind(thetaLoop1(i)),0,aLoop1(i);
        sind(thetaLoop1(i))*cosd(alphaLoop1(i)),cosd(thetaLoop1(i))*cosd(alphaLoop1(i)),-
sind(alphaLoop1(i)),-sind(alphaLoop1(i))*dLoop1(i);

sind(thetaLoop1(i))*sind(alphaLoop1(i)),cosd(thetaLoop1(i))*sind(alphaLoop1(i)),cosd(alphaLoop1(i)),co
sd(alphaLoop1(i))*dLoop1(i);
        0,0,0,1]};
    T1(i+1,i)={cell2mat(T1(i,i+1))^-1};
    T1(i,i)={eye(4)};
end
for i=1:NFrame
    for j=1:NFrame
        if abs(i-j)>=2
            if j>=i
                for k=i:j-1
                    T1(i,j)={cell2mat(T1(i,j))*cell2mat(T1(k,k+1))};
                    T1(j,i)={cell2mat(T1(i,j))^-1};
                end
            end
        end
    end
end
%Creating the Position and Rotation matricies
for i=1:NFrame
    for j=1:NFrame
        R1(i,j)={T1{i,j}(1:3,1:3)};
        P1(i,j)={T1{i,j}(1:3,4)};
    end
```

```
end
%-------------------------------------------------------------------------
%% Inner loop transformations
% there are 2 extra (invisible) frames added before frame 7 (5, and 6), this is to ensure that
% frames 2 and 7 on both inner and outer loops are identical
alphaLoop2=[0,0,0,90,0,0,0,90,90,0];
aLoop2=[0,0,L2,0,L9,0,0,L6,0,L8];
dLoop2=[0,L1,0,0,0,0,0,L7+d1,0,0];
thetaLoop2=[theta1*180/pi,0,0,theta2*180/pi,(theta3*180/pi-90-
theta2*180/pi),0,0,180,theta5*180/pi,0];
%-------------------------------------------------------------------------
% creation of the transformation matrix for frame transformation
for i=1:NFrame
   for j=1:NFrame
      T2(i,j)={eye(4)};
   end
end
for i=1:NFrame-1
   T2(i,i+1)={[cosd(thetaLoop2(i)),-sind(thetaLoop2(i)),0,aLoop2(i);
      sind(thetaLoop2(i))*cosd(alphaLoop2(i)),cosd(thetaLoop2(i))*cosd(alphaLoop2(i)),-
sind(alphaLoop2(i)),-sind(alphaLoop2(i))*dLoop2(i);

sind(thetaLoop2(i))*sind(alphaLoop2(i)),cosd(thetaLoop2(i))*sind(alphaLoop2(i)),cosd(alphaLoop2(i)),co
sd(alphaLoop2(i))*dLoop2(i);
      0,0,0,1]};
   T2(i+1,i)={cell2mat(T2(i,i+1))^-1};
   T2(i,i)={eye(4)};
end
for i=1:NFrame
   for j=1:NFrame
      if abs(i-j)>=2
         if j>=i
```

```
            for k=i:j-1
               T2(i,j)={cell2mat(T2(i,j))*cell2mat(T2(k,k+1))};
               T2(j,i)={cell2mat(T2(i,j))^-1};
            end
         end
      end
   end
end
%Creating the Position and Rotation matricies
for i=1:NFrame
   for j=1:NFrame
      R2(i,j)={T2{i,j}(1:3,1:3)};
      P2(i,j)={T2{i,j}(1:3,4)};
   end
end
%---------------------------------------------------------------------
%% both loop plotting for robot pose

for i=1:NFrame
   keypnts2(1:3,i)=P2{1,i};
   keypntsx2(1,i)=keypnts2(1,i);
   keypntsy2(1,i)=keypnts2(2,i);
   keypntsz2(1,i)=keypnts2(3,i);
end
for i=1:NFrame
   keypnts(1:3,i)=P1{1,i};
   keypntsx(1,i)=keypnts(1,i);
   keypntsy(1,i)=keypnts(2,i);
   keypntsz(1,i)=keypnts(3,i);
end
```

```
for i=1:NFrame
   plot3(keypntsx2,keypntsy2,keypntsz2);
   hold on
end
hold on
for i=1:NFrame
   plot3(keypntsx,keypntsy,keypntsz);
   hold on
end
% axis([-2 2 -2 2 -2 2]);
xlabel('x');
ylabel('y');
zlabel('z');

%% Kinematics
% through the outer link
%-------------------------------------------------------------------------
%inputs: joint angle velocities (thetadot)rad/s, and joint extension
%velocities(ddot)mm/s
%rotvel(i) is the rotational velocity of link from frame (i-1) to (i)
%velocity(i) is the velocity of frame (i) with respect to frame (i)
thetadot1=[theta1dot,0,0,theta3dot,-theta3dot+theta2dot, theta3dot-theta2dot, 0, 0,theta5dot,0];
%rotational joints
for i=1:7
   rotvelocity1{i+1}=R1{i+1,i}*rotvelocity1{i}+thetadot1(i)*[0;0;1]; %due to notation differences the
thetadot refered to is i, not i+1 like the textbook refers to.
   velocity1{i+1}=R1{i+1,i}*(velocity1{i}+cross(rotvelocity1{i},P1{i,i+1}));
end
%Prismatic joint
for i=8
```

```
   rotvelocity1{i+1}=R1{i+1,i}*rotvelocity1{i};
   velocity1{i+1}=R1{i+1,i}*(velocity1{i}+cross(rotvelocity1{i},P1{i,i+1}))+d1dot*[0;0;1];
end
%back to rotational joints
for i=9:10
   rotvelocity1{i+1}=R1{i+1,i}*rotvelocity1{i}+thetadot1(i)*[0;0;1];%similar note to above
   velocity1{i+1}=R1{i+1,i}*(velocity1{i}+cross(rotvelocity1{i},P1{i,i+1}));
end
%-------------------------------------------------------------------------
%Inner loop kinematics
thetadot2=[theta1dot,0,0,theta2dot,theta3dot-theta2dot,0,0,0, theta5dot,0];
 % base link rotational velocity
rotvelocity2{1}=rotvelocity1{1}; % base link rotational velocity
velocity2{1}=velocity1{1}; % base Frame velocity
%rotational joints
for i=1:7
   rotvelocity2{i+1}=R2{i+1,i}*rotvelocity2{i}+thetadot2(i)*[0;0;1]; %due to notation differences the
thetadot refered to is i, not i+1 like the textbook refers to.
   velocity2{i+1}=R2{i+1,i}*(velocity2{i}+cross(rotvelocity2{i},P2{i,i+1}));
end
%Prismatic joint
for i=8
   rotvelocity2{i+1}=R2{i+1,i}*rotvelocity2{i};
   velocity2{i+1}=R2{i+1,i}*(velocity2{i}+cross(rotvelocity2{i},P2{i,i+1}))+d1dot*[0;0;1];
end
%back to rotational joints
for i=9:10
   rotvelocity2{i+1}=R2{i+1,i}*rotvelocity2{i}+thetadot2(i)*[0;0;1];%similar note to above
   velocity2{i+1}=R2{i+1,i}*(velocity2{i}+cross(rotvelocity2{i},P2{i,i+1}));
end
```

```
%% centers of mass/ Inertai Values
%centers of mass from frame(i) to frame {i+1} through link (i), the dh frame is assumed to be
%located at the end of the link, but situated at the center of mass of the
%cross section.
PC1{1}=[5.59;0;69.39]/1000;
PC1{2}=[5.59;0;69.39]/1000;
PC1{3}=[5.59;0;-43.31]/1000;
PC1{4}=[-24.41;0;-43.31]/1000;
PC1{5}=[115.69;37.67;0]/1000;
PC1{6}=[500.04;0;0]/1000;
PC1{7}=[403.95,33.05,0]/1000;
PC1{8}=[33.05;-137.25;0]/1000;
PC1{9}=[-15.146;0;-590.44]/1000;
PC1{10}=[8.18;0;75.13]/1000;
PC2{1}=[5.59;0;69.39]/1000;
PC2{2}=[5.59;0;69.39]/1000;
PC2{3}=[5.59;0;-43.31]/1000;
PC2{4}=[-24.41;0;-43.31]/1000;
PC2{5}=[407.29;-10.26;0]/1000;
PC2{6}=[33.05;-137.25;0]/1000;
PC2{7}=[33.05;-137.25;0]/1000;
PC2{8}=[33.05;-137.25;0]/1000;
PC2{9}=[-15.146;0;-590.44]/1000;
PC2{10}=[8.18;0;75.13]/1000;
mass1(1)=28404.94/1000;
mass1(2)=28404.94/1000;
mass1(3)=28404.94/1000;
mass1(4)=28404.94/1000;
mass1(5)=2494.21/1000;
mass1(6)=3880.04/1000;
```

```
mass1(7)=2978.79/1000;
mass1(8)=2978.79/1000;
mass1(9)=2125.98/1000;
mass1(10)=783.66/1000;
mass2(1)=28404.94/1000;
mass2(2)=28404.94/1000;
mass2(3)=28404.94/1000;
mass2(4)=28404.94/1000;
mass2(5)=10902.05/1000;
mass2(6)=2978.79/1000;
mass2(7)=2978.79/1000;
mass2(8)=2978.79/1000;
mass2(9)=2125.98/1000;
mass2(10)=783.66/1000;
Imc1{1}=[587314994.58,0,0;0,251205614.05,0;0,0,592757975.96]/1E9;
Imc1{2}=[587314994.58,0,0;0,251205614.05,0;0,0,592757975.96]/1E9;
Imc1{3}=[587314994.27,0,0;0,251205613.8,0;0,0,592757976.03]/1E9;
Imc1{4}=[587314993.42,0,0;0,251205613.38,0;0,0,592757975.1]/1E9;
Imc1{5}=[7943614.13,0,0;0,31319008.31,0;0,0,28858113.96]/1E9;
Imc1{6}=[523671.74,0,0;0,333902698,0;0,0,334322068]/1E9;
Imc1{7}=[3443704.9,0,0;0,467859710.93,0;0,0,465385244.26]/1E9;
Imc1{8}=[467859710.93,0,0;0,3443704.9,0;0,0,465385244.26]/1E9;
Imc1{9}=[257782355.42,0,0;0,255434189.92,0;0,0,3059584.79]/1E9;
Imc1{10}=[13997366.02,0,0;0,14221111.08,0;0,0,376344.81]/1E9;
Imc2{1}=[587314994.58,0,0;0,251205614.05,0;0,0,592757975.96]/1E9;
Imc2{2}=[587314994.58,0,0;0,251205614.05,0;0,0,592757975.96]/1E9;
Imc2{3}=[587314994.27,0,0;0,251205613.8,0;0,0,592757976.03]/1E9;
Imc2{4}=[587314993.42,0,0;0,251205613.38,0;0,0,592757975.1]/1E9;
Imc2{5}=[33996691.35,0,0;0,964478373.4,0;0,0,953807977.46]/1E9;
Imc2{6}=[467859710.93,0,0;0,3443704.9,0;0,0,465385244.26]/1E9;
```

```
Imc2{7}=[467859710.93,0,0;0,3443704.9,0;0,0,465385244.26]/1E9;

Imc2{8}=[467859710.93,0,0;0,3443704.9,0;0,0,465385244.26]/1E9;

Imc2{9}=[257782355.42,0,0;0,255434189.92,0;0,0,3059584.79]/1E9;

Imc2{10}=[13997366.02,0,0;0,14221111.08,0;0,0,376344.81]/1E9;




%% Accelerations

% solved usindg the outer loop

%rotaccel(i) is the rotational acceleration of link from frame (i-1) to (i)

%accel(i) is the linear acceleration of frame (i)

% joint input accelerations

%outer loop accelerations

thetaddot1=[theta1ddot,0,0,theta3ddot,-theta3ddot+theta2ddot,theta3ddot-
theta2ddot,0,0,theta5ddot,0];

for i=1:7 % rotational joints


rotaccel1{i+1}=R1{i+1,i}*rotaccel1{i}+cross(R1{i+1,i}*rotvelocity1{i},thetadot1(i)*[0;0;1])+thetaddot1(i)*[
0;0;1];

   temp=cross(rotvelocity1{i},P1{i,i+1});

    temp2=cross(rotvelocity1{i},temp);

    temp3=cross(rotaccel1{i},P1{i,i+1});

    temp4=accel1{i};

   accel1{i+1}=R1{i+1,i}*(temp2+temp3+temp4);

end

for i=8% prismatic joint

   rotaccel1{i+1}=R1{i+1,i}*rotaccel1{i};

    temp=cross(rotvelocity1{i},P1{i,i+1});

    temp2=cross(rotvelocity1{i},temp);

    temp3=cross(rotaccel1{i},P1{i,i+1});
```

```
    temp4=accel1{i};
    temp5=cross(2*rotvelocity1{i+1},d1dot*[0;0;1]);
    temp6=d1ddot*[0;0;1];
   accel1{i+1}=R1{i+1,i}*(temp2+temp3+temp4)+temp5+temp6;
end

for i=9:10 % rotational joints

rotaccel1{i+1}=R1{i+1,i}*rotaccel1{i}+cross(R1{i+1,i}*rotvelocity1{i},thetadot1(i)*[0;0;1])+thetaddot1(i)*[
0;0;1];
    temp=cross(rotvelocity1{i},P1{i,i+1});
    temp2=cross(rotvelocity1{i},temp);
    temp3=cross(rotaccel1{i},P1{i,i+1});
    temp4=accel1{i};
   accel1{i+1}=R1{i+1,i}*(temp2+temp3+temp4);
end
%accelerations of centers of mass
for i=0:NFrame-2
   temp=cross(rotaccel1{i+1},PC1{i+1});
   temp1=cross(rotvelocity1{i+1},PC1{i+1});
   temp2=cross(rotvelocity1{i+1},temp1);
   accelc1{i+1}=temp+temp2+accel1{i+1};
end
for i=0:NFrame-2
   F1{i+1}=mass1(i+1)*accelc1{i+1};
end
for i=0:NFrame-2
  temp=Imc1{i+1}*rotaccel1{i+1};
  temp2=Imc1{i+1}*rotvelocity1{i+1};
  temp3=cross(rotvelocity1{i+1},temp2);
  N1{i+1}=temp+temp3;
```

```
end
%inner loop accelerations----------------------------------------------------
thetaddot2=[theta1ddot,0,0,theta2ddot,theta3ddot-theta2ddot,0,0,0,theta5ddot,0];
f1{11}=R1{11,1}*f1{11};
%inital frame accelerations:
rotaccel2{1}=rotaccel1{1};
accel2{1}=accel1{1};
for i=1:7 % rotational joints

rotaccel2{i+1}=R2{i+1,i}*rotaccel2{i}+cross(R2{i+1,i}*rotvelocity2{i},thetadot2(i)*[0;0;1])+thetaddot2(i)*[
0;0;1];
    temp=cross(rotvelocity2{i},P2{i,i+1});
    temp2=cross(rotvelocity2{i},temp);
    temp3=cross(rotaccel2{i},P2{i,i+1});
    temp4=accel2{i};
   accel2{i+1}=R2{i+1,i}*(temp2+temp3+temp4);
end
for i=8% prismatic joint
   rotaccel2{i+1}=R2{i+1,i}*rotaccel2{i};
    temp=cross(rotvelocity2{i},P2{i,i+1});
    temp2=cross(rotvelocity2{i},temp);
    temp3=cross(rotaccel2{i},P2{i,i+1});
    temp4=accel2{i};
    temp5=cross(2*rotvelocity2{i+1},d1dot*[0;0;1]);
    temp6=d1ddot*[0;0;1];
   accel2{i+1}=R2{i+1,i}*(temp2+temp3+temp4)+temp5+temp6;
end
for i=9:10 % rotational joints

rotaccel2{i+1}=R2{i+1,i}*rotaccel2{i}+cross(R2{i+1,i}*rotvelocity2{i},thetadot2(i)*[0;0;1])+thetaddot2(i)*[
0;0;1];
    temp=cross(rotvelocity2{i},P2{i,i+1});
```

```
    temp2=cross(rotvelocity2{i},temp);
    temp3=cross(rotaccel2{i},P2{i,i+1});
    temp4=accel2{i};
   accel2{i+1}=R2{i+1,i}*(temp2+temp3+temp4);
end
%accelerations of centers of mass
for i=0:NFrame-2
   temp=cross(rotaccel2{i+1},PC2{i+1});
   temp1=cross(rotvelocity2{i+1},PC2{i+1});
   temp2=cross(rotvelocity2{i+1},temp1);
   accelc2{i+1}=temp+temp2+accel2{i+1};
end
for i=0:NFrame-2
   F2{i+1}=mass2(i+1)*accelc2{i+1};
end
for i=0:NFrame-2
  temp=Imc2{i+1}*rotaccel2{i+1};
  temp2=Imc2{i+1}*rotvelocity2{i+1};
  temp3=cross(rotvelocity2{i+1},temp2);
  N2{i+1}=temp+temp3;
end
%inward iterations
for i= NFrame-1:-1:9
   f1{i}=R1{i,i+1}*f1{i+1}+F1{i};
   temp=N1{i}+R1{i,i+1}*n1{i+1};
   temp2=cross(PC1{i},F1{i});
   temp3=cross(P1{i,i+1},R1{i,i+1}*f1{i+1});
   n1{i}=temp+temp2+temp3;
end
% because of the parallel connections between the actuator and the
```

```
% extension it is difficult to determine the individual internal reactions
% of f9a and f9b. However, considering an FBD of the extension link the
% only reaction force in the frame9 z direction is located at f9a, due to
% the bearing connection showing f9bz=0. Only the z component (joint load)
% is determined.
f9=(F1{9}+R1{9,10}*f1{10});
f1{9}=[0;0;f9(3,1)]; % only the joint load and not the total forces acting at f9a
% fbd of the actuator and extension give the following equillibrium
% equations
%sum of moments( all relative to frame 8):
component1=cross(PC1{8},F1{8}); % moment about frame 8 due to F8
component2=cross(P1{8,10},R1{8,10}*f1{10}); % moment about frame 8 due to f10
   PC9from8=(P1{8,9}+R1{8,9}*PC1{9}); % distance from frame 8 to center of gravity of frame 9
component3=cross(PC9from8,R1{8,9}*F1{9}); % moment about frame 8 due to F9
component4=N1{8}+R1{8,9}*N1{9}+R1{8,10}*n1{10}; % moments about 8 due to other moments
temp=component1+component2+component3+component4; % total moments about frame 8 that must be balanced by f7,
f7in8x=temp(3,1)/(P1{8,7}(2,1));% force acting at f7 in direction of x in frame 8
f7y=R1{7,8}*[f7in8x;0;0];
% outer arm fbd
temp=cross(PC1{6},F1{6})+N1{6}; % moments about z in frame 6 to be counteracted by f7 all relative to frame 6
temp1=cross(P1{6,7},R1{6,7}*-f7y); % moment about 6 due to f7y, now all that is left to find is f7x
temp2=temp+temp1;% moment needed to be generated by P1{6,7} cross f7x
syms f7x
direction= R1{6,7}*[f7x;0;0]; % f7x relative to 6
momentf7x=cross(P1{6,7},direction);% amount of moment generated by f7x
A=momentf7x(3,1);
B=temp2(3,1);
C=solve(A+B==0,f7x); %solve for f7x in frame 6
f1{7}=double(f7y-[C;0;0]);
```

```
% sum of forces to find f1{8}
temp1=R1{8,7}*f1{7};% f7 in 8
temp2=F1{8};
temp3=R1{8,9}*F1{9}; %F9 in 8
temp4=R1{8,10}*f1{10};% force f10 in 8
temp5=temp1+temp2+temp3+temp4;% sum of all forces
f1{8}=-temp5;
syms n7y n7x n8y n8x f7z f8z
n1{7}=[n7x;n7y;0];
n1{8}=[n8x;n8y;0];
A=double(f1{8}(1,1));
B=double(f1{8}(2,1));
f1{8}=-([A;B;0]+[0;0;f8z]);
C=double(f1{7}(1,1));
D=double(f1{7}(2,1));
f1{7}=-([C;D;0]+[0;0;f7z]);
for i= 6:-1:4
   f1{i}=R1{i,i+1}*f1{i+1}+F1{i};
   temp=N1{i}+R1{i,i+1}*n1{i+1};
   temp2=cross(PC1{i},F1{i});
   temp3=cross(P1{i,i+1},R1{i,i+1}*f1{i+1});
   n1{i}=temp+temp2+temp3;
end
%% inner links forces
for i=8:11
   f2{i}=f1{i};
   n2{i}=n1{i};
end
for i=7:-1:6
   f2{i}=f2{8};
```

```
   n2{i}=n2{8};
end
%
for i= 5:-1:4
   f2{i}=R2{i,i+1}*f2{i+1}+F2{i};
   temp=N2{i}+R2{i,i+1}*n2{i+1};
   temp2=cross(PC2{i},F2{i});
   temp3=cross(P2{i,i+1},R2{i,i+1}*f2{i+1});
   n2{i}=temp+temp2+temp3;
end
%% suturing the f2 and f1 and n2 and n1 back together
ftot{4}=f1{4}+f2{4};
ntot{4}=n1{4}+n2{4};
% chekc this part again
for i= 3
   f1{i}=R1{i,i+1}*ftot{4}+F1{i};
   temp=N1{i}+R1{i,i+1}*ntot{4};
   temp2=cross(PC1{i},F1{i});
   temp3=cross(P1{i,i+1},R1{i,i+1}*ftot{4});
   n1{i}=temp+temp2+temp3;
end
for i= 2:-1:1
   f1{i}=R1{i,i+1}*f1{i+1}+F1{i};
   temp=N1{i}+R1{i,i+1}*n1{i+1};
   temp2=cross(PC1{i},F1{i});
   temp3=cross(P1{i,i+1},R1{i,i+1}*f1{i+1});
   n1{i}=temp+temp2+temp3;
end
% determination of n1{1} to avoid unknown internal reactions, total sum of
% moments of the inertial forces for the entire structure
```

```
for i=1:9% initializing my matricies
   totalmoment(i)={[0;0;0]};
   linktotalmoment(i)={[0;0;0]};
end
for i=[2,5,6,8,9]
   linkmoment(i)={R1{1,i}*N1{i}}; % Inertial moments
   distance(i)={P1{1,i}+R1{1,i}*PC1{i}};% direction to center of mass
   linkforcemoment(i)={cross(distance{i},R1{1,i}*F1{i})}; % moments caused by inertial forces
   linktotalmoment(i)={linkforcemoment{i}+linkmoment{i}}; % total moments in each link
end
for i=1:9
   totalmoment(i+1)={totalmoment{i}+linktotalmoment{i}}; % total moments
end
% then for 10 and F2
temp6=R2{1,5}*N2{5}; % moment at the base due to F2{8}
distance6=P2{1,5}+R2{1,5}*PC2{5};
temp8=cross(distance6,R2{1,6}*F2{6});
temp7=R1{1,10}*n1{10};% moment about the base due to f1{10}
distance7=P1{1,10};
temp9=cross(distance7,R1{1,10}*f1{10});
totalmoment{11}=totalmoment{10}+temp6+temp8+temp7+temp9;
n1(1)={totalmoment{11}};
for i=1:3
   f2{i}=f1{i};
   n2{i}=n1{i};
end
for i=1:NFrame
   tau1{i}=(n1{i}.')*[0;0;1];
end
% special for prismatic joint, force in actuator not joint torque
```

```
tau1{9}= (f1{9}.')*[0;0;1];
tau2{5}=(f2{5}.')*[0;0;1];
jointforces=double([n1{1}(3,1),n2{5}(3,1),n1{5}(3,1),f1{9}(3,1),n1{10}(3,1)]);
endposition(currentstep,1)=T1{1,11}(1,4);
endposition(currentstep,2)=T1{1,11}(2,4);
endposition(currentstep,3)=T1{1,11}(3,4);
endposition(currentstep,4)=time;
torqueend(currentstep,1)=jointforces(1,1);
torqueend(currentstep,2)=jointforces(1,2);
torqueend(currentstep,3)=jointforces(1,3);
torqueend(currentstep,4)=jointforces(1,4);
torqueend(currentstep,5)=jointforces(1,5);
% FF=f2{8};% What are the forces at 8 relative to 5.
% results(currentstep,1)=time;
% results(currentstep,2)=FF(1,1);
% results(currentstep,3)=FF(2,1);
% %results(currentstep,4)=f2{8}(3,1);
% Forceresults=[max(results(:,2)),max(results(:,3)),0]
% Forceresults2=[min(results(:,2)),min(results(:,3)),0]
% rotaccelresults(currentstep,1)=time;
% rotaccelresults(currentstep,2)=rotaccel2{5}(1,1);
% rotaccelresults(currentstep,3)=rotaccel2{5}(2,1);
% rotaccelresults(currentstep,4)=rotaccel2{5}(3,1);
% rotaccelresults1= [max(rotaccelresults(:,2)),max(rotaccelresults(:,3)),max(rotaccelresults(:,4))]
% rotaccelresults2=[min(rotaccelresults(:,2)),min(rotaccelresults(:,3)),min(rotaccelresults(:,4))]
% accelresults(currentstep,1)=time;
% accelresults(currentstep,2)=accel2{5}(1,1);
% accelresults(currentstep,3)=accel2{5}(2,1);
% accelresults(currentstep,4)=accel2{5}(3,1);
% accelresults1=[max(accelresults(:,2)),max(accelresults(:,3)),max(accelresults(:,4))]
```

```
% accelresults2=[min(accelresults(:,2)),min(accelresults(:,3)),min(accelresults(:,4))]
% velocityresults(currentstep,1)=time;
% velocityresults(currentstep,2)=rotvelocity2{5}(1,1);
% velocityresults(currentstep,3)=rotvelocity2{5}(2,1);
% velocityresults(currentstep,4)=rotvelocity2{5}(3,1);
% velocityresults1=[max(velocityresults(:,2)),max(velocityresults(:,3)),max(velocityresults(:,4))]
% velocityresults2=[min(velocityresults(:,2)),min(velocityresults(:,3)),min(velocityresults(:,4))]
currentstep/steps*100
end
jointforces(1,5);
plot3(endposition(:,1),endposition(:,2),endposition(:,3))
xlabel('x (m)')
ylabel('y (m)')
zlabel('z (m)')
xlim([0 1.5])
ylim([0 1.5])
zlim([-1.5 1.5])
```

# Appendix D – MATLAB code for optimization

** a flow diagram for this code can be found in Figure 3-11**

```
clc

clear

%%  Optimization

Length=1;

slices=1000;

adjustsize=1e-4

delx=Length/slices;

deflection_lim=10/1000

density=7950;

E=200E9;

w=10/1000;

accel2(5)={[8.4931;9.9124;0.1649]};

rotvelocity2(5)={[0.2933;0.4231;0.314]};

rotaccel2(5)={[0.3265;0.3770;0.2513]};

fapplied=[758.1;859.9;0];% maximum forces that occur on this link from the dynamics code,  frame 8 forces relative to frame 5

for i=1:slices+1

   x(i)=(i-1)*delx;

end

%% initial height (H(0)) because my iterations start at i=1 and not i=0

 for i=1:1

   k=1;

   flag=0;

   H(i,k)=adjustsize;

 while flag == 0

   L(i)=delx;

   A(i,k)=H(i,k)*w;

   Iz_bending(i)=(w*H(i,k)^3)/12;
```

```
%% cumulative forces
%fb is the boundry force applied at frame8 as seen from the inner link in
%the new* coordinate system.
f_b=-fapplied;% -f2{8} because f2{8} sees the force on the linear actuator, we want the force at 8 seen from the bar. then R2{5,8} the forces at 8 seen from the bar in the 5 coordinate system, then multiplied by R so that it is in the new coordinate symstem as seen from the bar.
n_b=[0;0;0]; % no moment in z due to pin connection, others are chosen to be zero
%stress equations
stress_normal(i)=double(f_b(1,1)/A(i,k));
stress_bending(i)=double((n_b(3,1)*(H(i,k)/2))/Iz_bending(i));
stress_shear(i)=double(f_b(2,1)/A(i,k));
% maximum normal stress is a combination of bending stress and normal stress
stress_normalmax(i,1)=stress_normal(i)+stress_bending(i);
stress_normalmax(i,2)=stress_normal(i)-stress_bending(i);
stress_normalmax(i,3)=-stress_normal(i)+stress_bending(i);
stress_normalmax(i,4)=-stress_normal(i)-stress_bending(i);
stress_normalmax=max(stress_normalmax,[],2);
stress_vm(i,k)=sqrt(stress_normalmax(i,1)^2+3*stress_shear(i)^2);
%constraints
if (stress_vm(i,k)>=((350e6)/2));
    H(i,k+1)=H(i,k)+adjustsize;
 else flag = 1;
 end
k=k+1;
 end
 Hstress(1)=H(i,k-1)
 stress_vm0=stress_vm(i,k-1);
 end
%% rest of the bar
for i=1:slices
   k=1;
```

```
   flag=0;
   H(i,k)=adjustsize;
 while flag == 0
      temp1000(i)=H(i,k)*delx; % area of each slice
   xbar_slice(i)=i*delx-0.5*delx;
   temp1100(i)=xbar_slice(i)*temp1000(i);
   numerator=sum(temp1100);
   denominator=sum(temp1000);
   xbar_partial(i)=numerator/denominator;% center of gravity of the cumulative slices
   mass_slice(i)=delx*w*density*H(i,k);
   phi(i)=(delx*(i)/Length);
   A(i,k)=H(i,k)*w;
   Iz_bending(i)=(w*H(i,k)^3)/12;
   cog_slice(i)={[xbar_slice(i);0;0]};% center of gravity is always along the x axis
   mass_partial(i)=sum(mass_slice); % cumulative mass from each of the previous slices
   location_cog(i)={[xbar_partial(i);0;0]}; %3d cog for moment cross below
   location_slice(i)={[delx*i;0;0]};% location of the rxn forces for moment balance below
   % mass moments of inertia for each individual slice
   Ixx_slice(i)=((mass_slice(i))/12)*(H(i,k)^2+w^2);
   Iyy_slice(i)=((mass_slice(i))/12)*(delx^2+w^2);
   Izz_slice(i)=((mass_slice(i))/12)*(delx^2+H(i,k)^2);
   I_slice(i)={[Ixx_slice,0,0;0,Iyy_slice,0;0,0,Izz_slice]};
   for j=1:i% parallel axis theorum, to add each individual slices' inertia to the center of gravity of all the
slices (cumulative)
      paralleldistance(j)=xbar_slice(j)-xbar_partial(i); % only for 2d, w and H are symmetric about the axis
      sliceIxxcontribution(j)=Ixx_slice(j);
      sliceIyycontribution(j)=Iyy_slice(j)+mass_slice(j)*paralleldistance(j)^2;
      sliceIzzcontribution(j)=Izz_slice(j)+mass_slice(j)*paralleldistance(j)^2;
   end
partial_Ixx(i)=sum(sliceIxxcontribution);% cumulative inertia values
```

```
partial_Iyy(i)=sum(sliceIyycontribution);

partial_Izz(i)=sum(sliceIzzcontribution);

I_partial(i)={[partial_Ixx(i),0,0;0,partial_Iyy(i),0;0,0,partial_Izz(i)]};

%% cumulative forces

%fb is the boundry force applied at frame8 as seen from the inner link in

%the new* coordinate system.

f_b=-fapplied;% -f2{8} because f2{8} sees the force on the linear actuator, we want the force at 8 seen from the bar. then R2{5,8} the forces at 8 seen from the bar in the 5 coordinate system, then multiplied by R so that it is in the new coordinate symstem as seen from the bar.

n_b=[0;0;0]; % no moment in z due to pin connection, others are chosen to be zero

%acceleration at centroid of partial portion

temp=cross(rotaccel2{5},[Length-xbar_partial(i);0;0]);

temp1=cross(rotvelocity2{5},[Length-xbar_partial(i);0;0]);

temp2=cross(rotvelocity2{5},temp1);

accel_partial(i)={temp+temp2+accel2{5}};

%internal forces

F_partial(i)={-mass_partial(i)*accel_partial{i}}; % negative because the inertial force acts in the opposite direction to motion (cumulative)

N_partial(i)={-(I_partial{i}*rotaccel2{5}+cross(rotvelocity2{5},I_partial{i}*rotvelocity2{5}))};

F_slice(i)={-mass_slice(i)*accel_partial{i}};

%reaction forces (f_r)

f_r(i)={-(f_b+F_partial{i})};

n_r(i)={-(n_b+N_partial{i}+(cross(location_cog{i},F_partial{i}))+(cross(location_slice{i},f_r{i})))};

%stress equations

stress_normal(i)=double(f_r{i}(1,1)/A(i,k));

stress_bending(i)=double((n_r{i}(3,1)*(H(i,k)/2))/Iz_bending(i));

stress_shear(i)=double(f_r{i}(2,1)/A(i,k));

% maximum normal stress is a combination of bending stress and normal stress

stress_normalmax(i,1)=stress_normal(i)+stress_bending(i);

stress_normalmax(i,2)=stress_normal(i)-stress_bending(i);

stress_normalmax(i,3)=-stress_normal(i)+stress_bending(i);
```

```
stress_normalmax(i,4)=-stress_normal(i)-stress_bending(i);
stress_normalmax=max(stress_normalmax,[],2);
stress_vm(i,k)=sqrt(stress_normalmax(i,1)^2+3*stress_shear(i)^2);
%% constraints
if (stress_vm(i,k)>=((350e6)/2));
    H(i,k+1)=H(i,k)+adjustsize;
 else flag = 1;
   stress_vm_result(i+1)=stress_vm(i,k);
end
 k=k+1;
 end
 Hstress(i+1)=H(i,k-1);
 mass_total=sum(mass_slice)
  i/slices*100
end
 stress_vm_result(1)=stress_vm0;
for i=1:slices+1
   Hend(slices-i+2)=Hstress(i);
end
results(:,1)=x(:);
results(:,2)=Hend(:)/2; % use mirror in solidworks
results(:,3)=0; %solidworks curve needs z component
ansysresults(:,1)=x(:);% for apdl
ansysresults(:,2)=Hend(:);
ansysresults(slices+2,2)=accel2{5}(1,1);
ansysresults(slices+3,2)=accel2{5}(2,1);
ansysresults(slices+4,2)=accel2{5}(3,1);
ansysresults(slices+5,2)=rotvelocity2{5}(1,1);
ansysresults(slices+6,2)=rotvelocity2{5}(2,1);
ansysresults(slices+7,2)=rotvelocity2{5}(3,1);
```

```
ansysresults(slices+8,2)=rotaccel2{5}(1,1);
ansysresults(slices+9,2)=rotaccel2{5}(2,1);
ansysresults(slices+10,2)=rotaccel2{5}(3,1);
ansysresults(slices+11,2)=fapplied(1,1);
ansysresults(slices+12,2)=fapplied(2,1);
ansysresults(slices+13,2)=fapplied(3,1);
for i=1:slices+1
   Hend(slices-i+2)=Hstress(i);
end
%% going forwards now
%changing coordinate systems for my sanity
% 0 is now the large end
m_r(slices+1)={n_b};
for i=1:slices
m_r(slices-i+1)=n_r(i);
end
momentcheck(slices+1)=n_b(3,1);
for i=1:slices
momentcheck(slices-i+1)=n_r{i}(3,1);
end
flag=0;
counter=0;
H=Hend;
for i=1:slices+1
I_bending(i)=(w*H(i)^3)/12;
mei(i)=m_r{i}(3,1)/(I_bending(i)*E);
end
area_to_node(1)=0;
for i=1:slices
areamei(i)=((mei(i)+mei(i+1))/2)*delx;
```

```
area_to_node(i+1)=sum(areamei);
effect_of_area(i)=tan(areamei(i))*(Length-x(i));
end
original_displacement=sum(effect_of_area);
areamei_old=areamei;
H_old=H;
mei_old=mei;
while flag==0
for i=1:slices+1
   areamei_new=areamei_old;
   H_new(i)=H_old(i)+adjustsize;

   I_bending_new(i)=(w*H_new(i)^3)/12;
   mei_new(i)=m_r{i}(3,1)/(I_bending_new(i)*E);
   %if I increase node i then the mei of both area i and i-1 will change slightly
   if i==1
      effect_old(i)=tan(areamei_old(i))*(Length-x(i));%displacement effect of node (i)
      areamei_new(i)=(mei_new(i)+mei_old(i+1))*(delx/2);
      effect_new(i)=tan(areamei_new(i))*(Length-x(i));%displacement effect of node (i)
   end
   if i>=2 & i<=slices
      effect_old(i)=tan(areamei_old(i))*(Length-x(i))+tan(areamei_old(i-1))*(Length-x(i-1));
      areamei_new(i)=(mei_new(i)+mei_old(i+1))*(delx/2);
      areamei_new(i-1)=(mei_new(i)+mei_old(i-1))*(delx/2);
      effect_new(i)=(tan(areamei_new(i))*(Length-x(i)))+(tan(areamei_new(i-1))*(Length-x(i)));
   end
   if i==slices+1
      effect_old(i)=tan(areamei_old(i-1))*(Length-x(i-1));
      areamei_new(i-1)=(mei_new(i)+mei_old(i-1))*(delx/2);
      effect_new(i)=tan(areamei_new(i-1))*(Length-x(i-1));
```

```
    end
   improvement(i)=abs(effect_old(i)-effect_new(i));
end
[M,index]=max(improvement);
H_old(index)=H_new(index);
mei_old(index)=mei_new(index);
for i=1:slices
    areamei_old(i)=(mei_old(i)+mei_old(i+1))*(delx/2);

    effect_of_area_old(i)=tan(areamei_old(i))*(Length-x(i));
end
for i=1:slices+1
    Itest(i)=((1/12)*w*H_old(i)^3);
    meitest(i)=m_r{i}(3,1)/(Itest(i)*E);
end
% for i=1:slices
%     meiareatest(i)=(meitest(i)+meitest(i+1))*delx/2;
%     cumulativeareatest(i)=sum(meiareatest(1:i));
%     sectiondisplacementtest(i)=delx*tan(cumulativeareatest(i));
% end
% displacementtest=abs(sum(sectiondisplacementtest(:)));
total_improved_displacement=sum(abs(effect_of_area_old));
if total_improved_displacement<=deflection_lim
    flag=1
end
counter=counter+1
end
results2(:,1)=x(:);
results2(:,2)=((H_old(:)/2));
results2(:,3)=0; % for input into solidworks curve generator
```

```
ansysresults2(:,1)=x(:);

ansysresults2(:,2)=(Hend(i)); % for ansys apdl

mass_total2=mass_total+adjustsize*delx*w*counter*density;

%

% writematrix(results2(:,1:3),'\\cabinet.usask.ca\work$\joc358\Desktop\Research consolidated\Curves for ansys\Ansys deflection.txt')

% writematrix(results(:,1:3),'\\cabinet.usask.ca\work$\joc358\Desktop\Research consolidated\Curves for ansys\Ansys stress.txt')

writematrix(ansysresults,'\\cabinet.usask.ca\work$\joc358\Desktop\APDL\Take 1 stress\stress_opti_apdl.csv')

writematrix(ansysresults2,'\\cabinet.usask.ca\work$\joc358\Desktop\APDL\Take 1 stress\stress_and_deflection_opti_apdl.csv')

% writematrix(results(:,1:3),'\\cabinet.usask.ca\work$\joc358\Desktop\Research consolidated\Solidworks for beams\stress_opti_workbench.txt')

% writematrix(results2,'\\cabinet.usask.ca\work$\joc358\Desktop\Research consolidated\Solidworks for beams\stress_and_deflection_opti_workbench.txt')
```

# Appendix E - Datasheets

https://www.conetools.com/Harmonic/Harmonic/HarmonicInfo?MN=CBG320160--&Units=Metric&Mode=Direct

Cone Drive BY TIMKEN

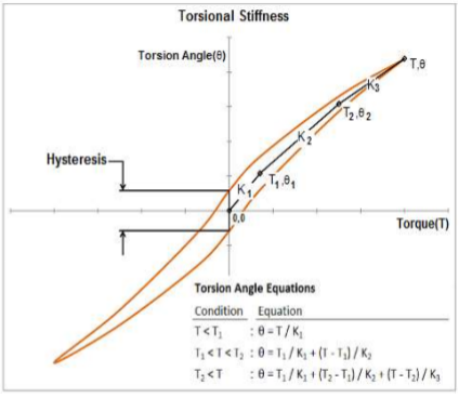
Mon Aug 23 2021

**Model - CBG320160--**

| Series | Size | Ratio |
|---|---|---|
| CBG | 32 | 160 |

| Specifications | | |
|---|---|---|
| Model Number | CBG320160-- | |
| Configuration | Cup Big Gearhead | |
| Size | 32 | |
| Ratio | 160 | |
| Continuous Torque | 172 Nm | 1522 lb-in |
| Start - Stop Torque | 460 Nm | 4071 lb-in |
| Max Average Torque | 275 Nm | 2434 lb-in |
| Peak Torque | 751 Nm | 6647 lb-in |
| Efficiency | 65 % | |
| No Load Input Torque | 0.42 Nm | 3.7 lb-in |
| Starting Torque | 0.15 Nm | 1.3 lb-in |
| Backdrive Starting Torque | 29 Nm | 257 lb-in |
| Mass | 3.3 Kg | 7.3 lb |
| Drivetrain Inertia at input | 176 10-6 kg-m2 | 1558 lb-in-s2 |

| Torsional Stiffness - Lost Motion | | |
|---|---|---|
| T1 | 29 Nm | 257 lb-in |
| T2 | 108 Nm | 956 lb-in |
| K1 | 67 Nm/mRad | 593 lb-in |
| K2 | 110 Nm/mRad | 974 lb-in |
| K3 | 120 Nm/mRad | 1062 lb-in |
| Max Backlash | 0.024 mRad | |
| Hysteresis | 0.29 mRad | |




* If you plan to place an order or obtain a quote for this unit, please email or fax this specification document along with your request.
E-Mail : orders@conedrive.com
Fax : 1-888-907-2663

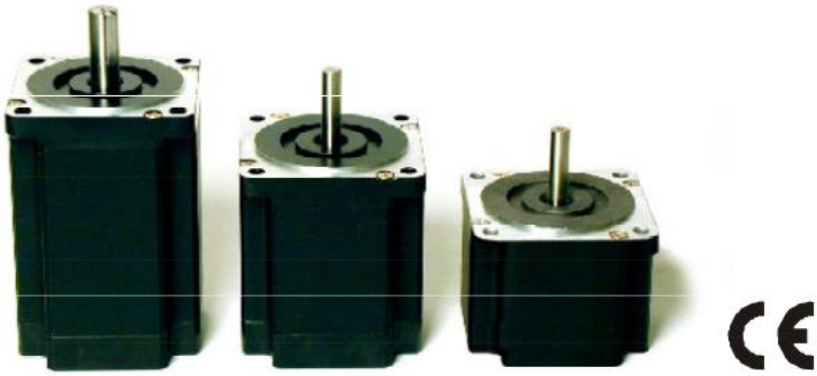
**Cone Drive Operations, Inc. (www.conedrive.com)**
240 E. 12th Street,Traverse City, Michigan USA 49684 - Tel: 231-946-8410, Sales: 888-994-2663

https://www.jvl.dk/files/_2011clean/pdf/ld042gb.pdf

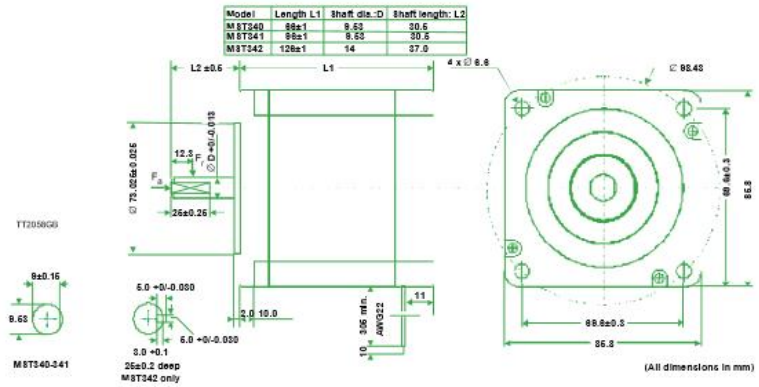
Product Data

JVL...integration in motion

# High Quality, High Torque Step Motors MST340, 341 and 342



- Highest torque density rating in the industry
- High torque-to-inertia for faster start and stop
- Rugged design and long life bearings
- High power, cooler running, rare-earth magnet design
- Exposed-lamination housing, optimized for high torque and smooth, accurate microstepping
- Standard NEMA34 mounting
- Facilities for encoders, double shaft, different shaft types, etc.
- High axial and radial shaft load
- Cost-effective alternative to servo motors
- Low noise
- Option for planetary gearhead




LD0042-06GB Date: 23-1-08

# High Quality, High Torque Step Motors MST340-342

## Specifications

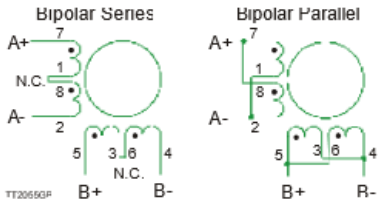
Step motors MST340, 341, 342 have been designed by JVL for use with the entire range of JVL Step Motor Drivers and Controllers. MST340/41/42 are high torque step motors especially made for mini- and microstep operation.

| General Specifications | |
|---|---|
| Step angle | 1.8 Deg/200 steps/rev. |
| Number of phases | 4 (8 leads) |
| Fullstep angle accuracy. No load | ±2% of 1.8° = 0.036° |
| Microstep angle accuracy. No load | ±5% of 1.8° = 0.09° |
| Ambient temperature | -25°C to 40°C |
| Operating temperature | Max. 130°C |
| Insulation resistance | 100 MOhm Min. (500V DC) |
| Insulation class | Class B=130 |
| Cable Length | 305mm |
| Protection | IP41 Option IP54 with oil seal |
| Radial play | Max. 0.025mm (0.5kg load) |
| End play | Max. 0.075mm (1kg load) |
| Radial load ($F_r$) | Max. 18kg (applied 12.3mm from shaft end) |
| Thrust load ($F_t$) | Max. 11.5kg |
| Max. drive voltage | 165 VDC |

Nominal data at 25°C. Actual data will vary ±10%

| Electrical and Mechanical Specifications | MST340B02 | | MST341B02 | | MST342B02 | | MST342C02 | |
|---|---|---|---|---|---|---|---|---|
| | Parl. | Ser. | Parl. | Ser. | Parl. | Ser. | Parl. | Ser. |
| Holding Torque (Nm) | 3.0 | | 6.1 | | 9.0 | | 9.0 | |
| Running Torque (low speed )(Nm) | 2.5 | | 5.1 | | 7.2 | | 7.2 | |
| Phase Current (A) | 6.6 | 3.3 | 6.5 | 3.3 | 6.8 | 3.4 | 9.5 | 4.7 |
| Phase Resistance (Ohm) | 0.2 | 0.8 | 0.2 | 0.9 | 0.4 | 1.9 | 0.2 | 0.9 |
| Phase Inductance (mH) | 0.9 | 3.4 | 1.3 | 5.2 | 2.2 | 8.8 | 1.4 | 5.4 |
| Phase Voltage (V) | 1.9 | 3.8 | 3.0 | 6.0 | 3.5 | 7.0 | 3.0 | 6.0 |
| Weight (kg) | 1.8 | | 2.8 | | 3.8 | | 3.8 | |
| Rotor Inertia (kgcm²) | 1.4 | | 2.7 | | 4.0 | | 4.0 | |

## Connections

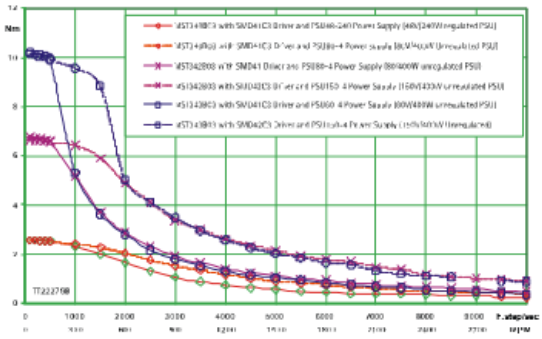



| Color | Pin no. | Bipolar Series | Bipolar Parallel |
|---|---|---|---|
| Black | 7 | A+ | A+ |
| White/Black | 1 | Pin 8 and 1 connected together | A- |
| White/Orange | 8 | | A+ |
| Orange | 2 | A- | A- |
| Red | 5 | B+ | B+ |
| White/Red | 3 | Pin 3 and 6 connected together | B- |
| White/Yellow | 6 | | B+ |
| Yellow | 4 | B- | B- |

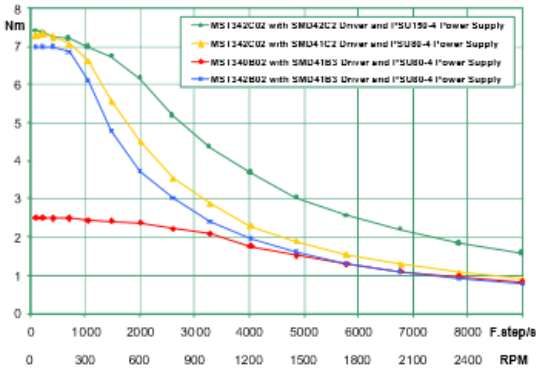
Motor direction can be reversed by changing A+ and A-

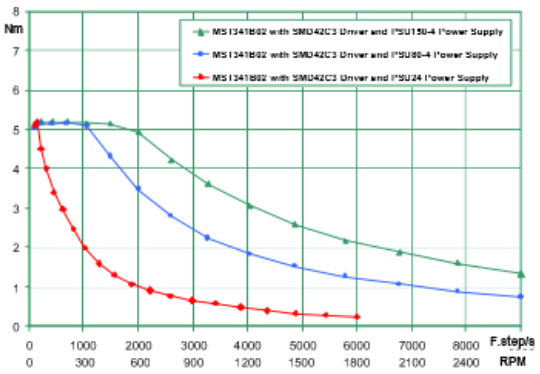
**Note:** N.C. not to be connected to other places in Bipolar Series.

**Option:** IP54 Oil Seal MST340-SMI1



## Torque-Velocity Profiles





JVL Industri Elektronik A/S
Blokken 42
DK-3460 Birkerød, Denmark
Tel: +45 4582 4440
Fax: +45 4582 5550
E-mail: jvl@jvl.dk www.jvl.dk

# Appendix F – Ansys workbench rigid dynamics module

The following screenshots and steps will explain how the rigid dynamics model was created for testing the kinematic and dynamic equations presented in chapter 2.

First the model is imported from Solidworks. Several simplifications are made at this step, motors and gearboxes are not imported, but rather added in during the next step as point masses. Sensors held by the end effector are replaced by a distributed load, and the upper link and extension arms are shown as two solid pieces instead of the complex pattern of holes the physical links contain.

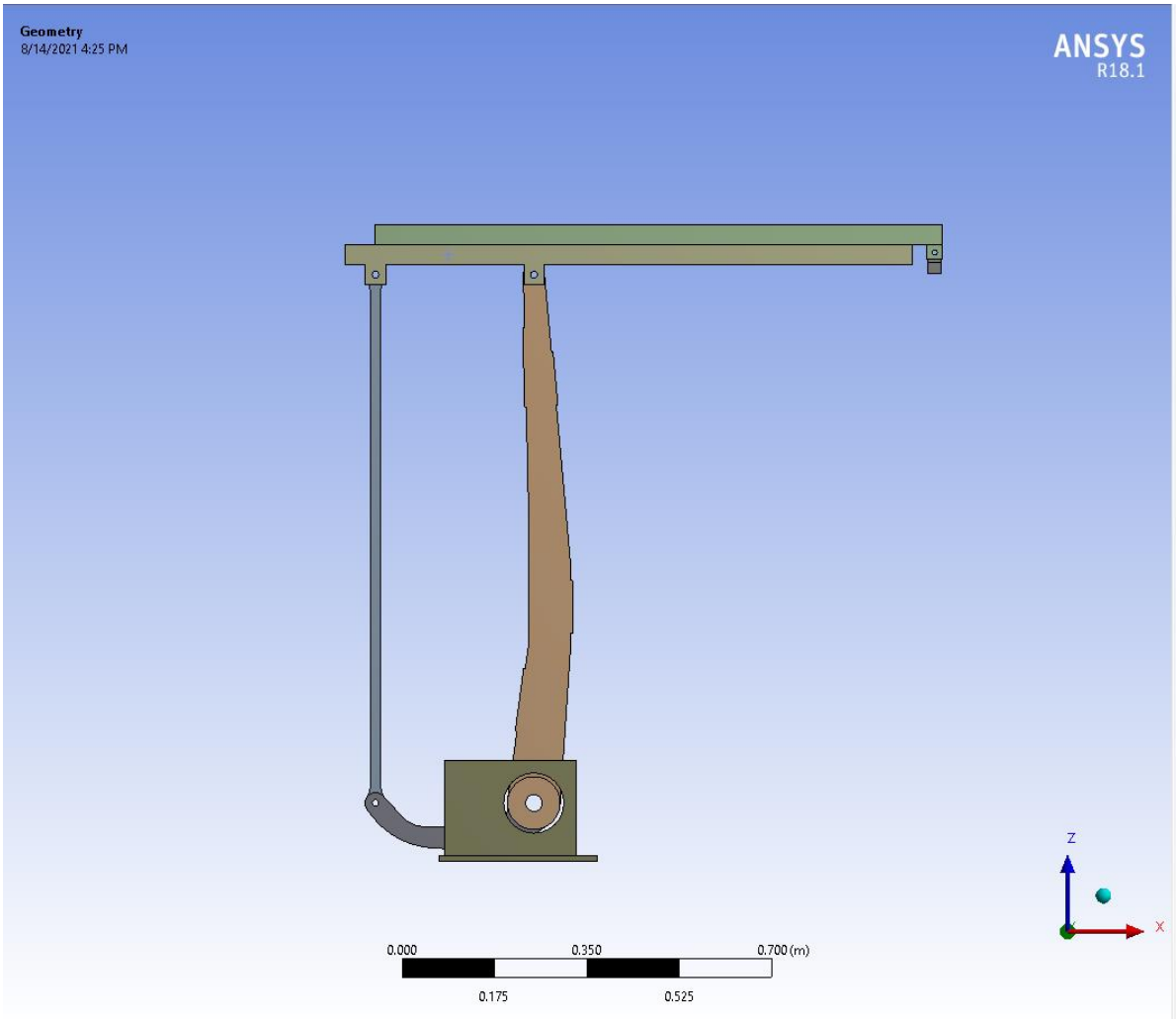



*Figure F.1 the geometry of the imported model*

Next, the centers of mass are defined relative to the global coordinate system and attached to each link of the robot (shown in ANSYS workbench by spheres). At this stage the position of the center of mass,

and mass moments of inertia are defined manually. These values are recorded in section 2 of this thesis and include/account for the simplified components (motors, gearboxes and any geometric simplification)

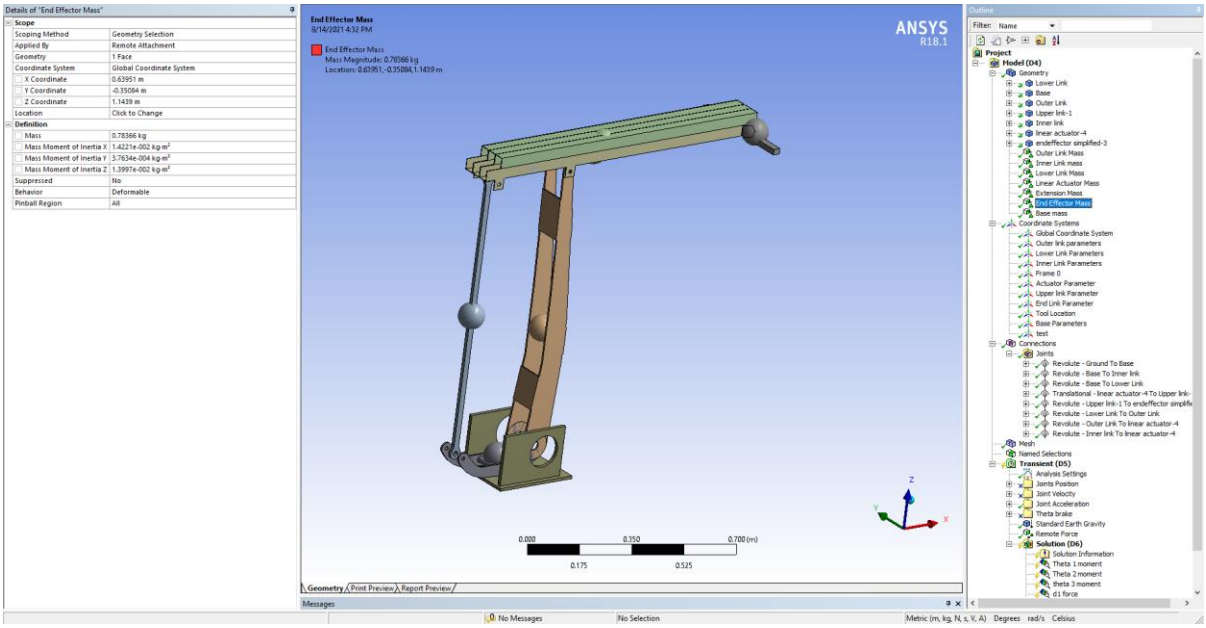



*Figure F.2: Placement of center of mass according to reported values in Chapter 2*

After the links properties have been input, joints are created at link intersections to define and restrict the motion of the robot arm. The image below shows the joint connection between the base plate and the inner link.  There are many different types of joints to choose from, and a full description can be found in the ANSYS help menu.

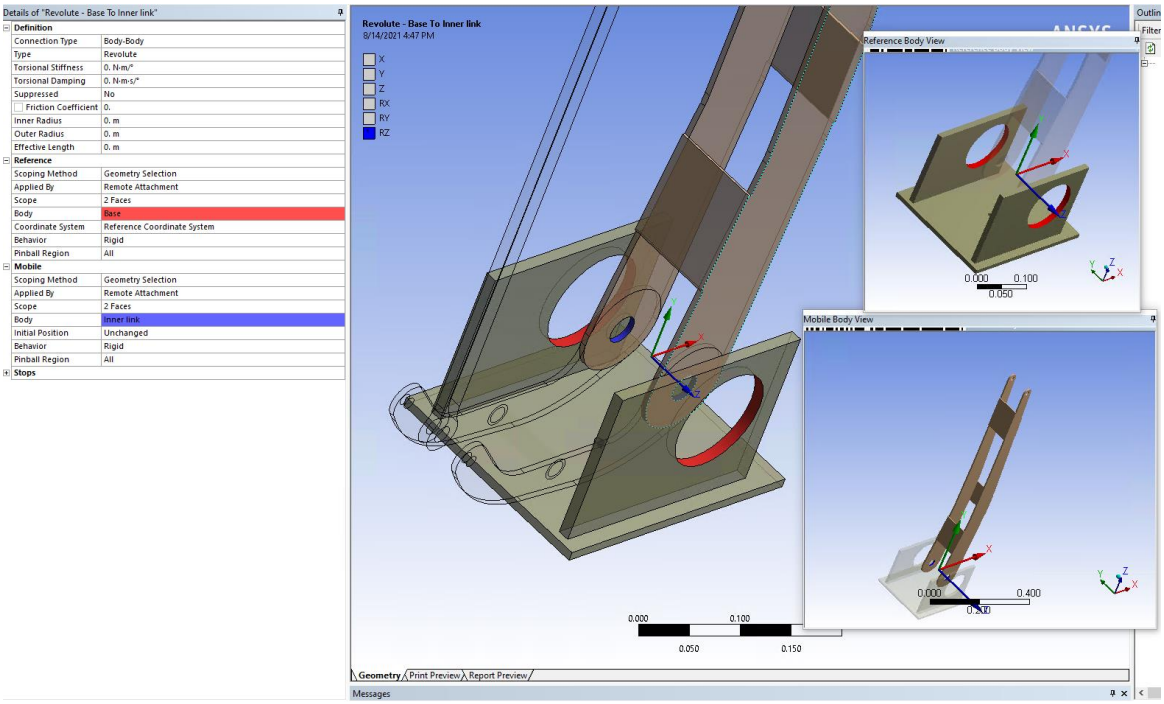



*Figure F.3: Example of joint connection being specified*

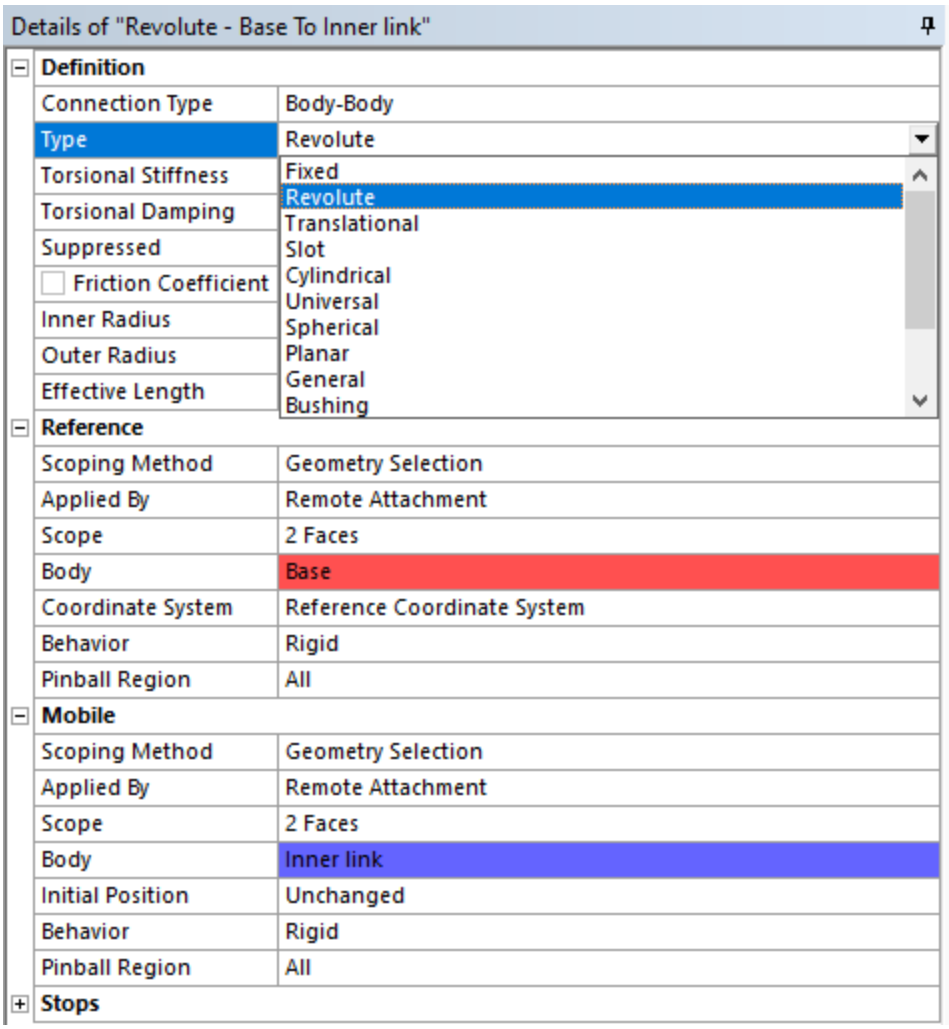



*Figure F.4: Showing different types of joints available*

Meshing the robot arm is very basic, as each link is modeled as 1 element in the rigid body dynamics module.

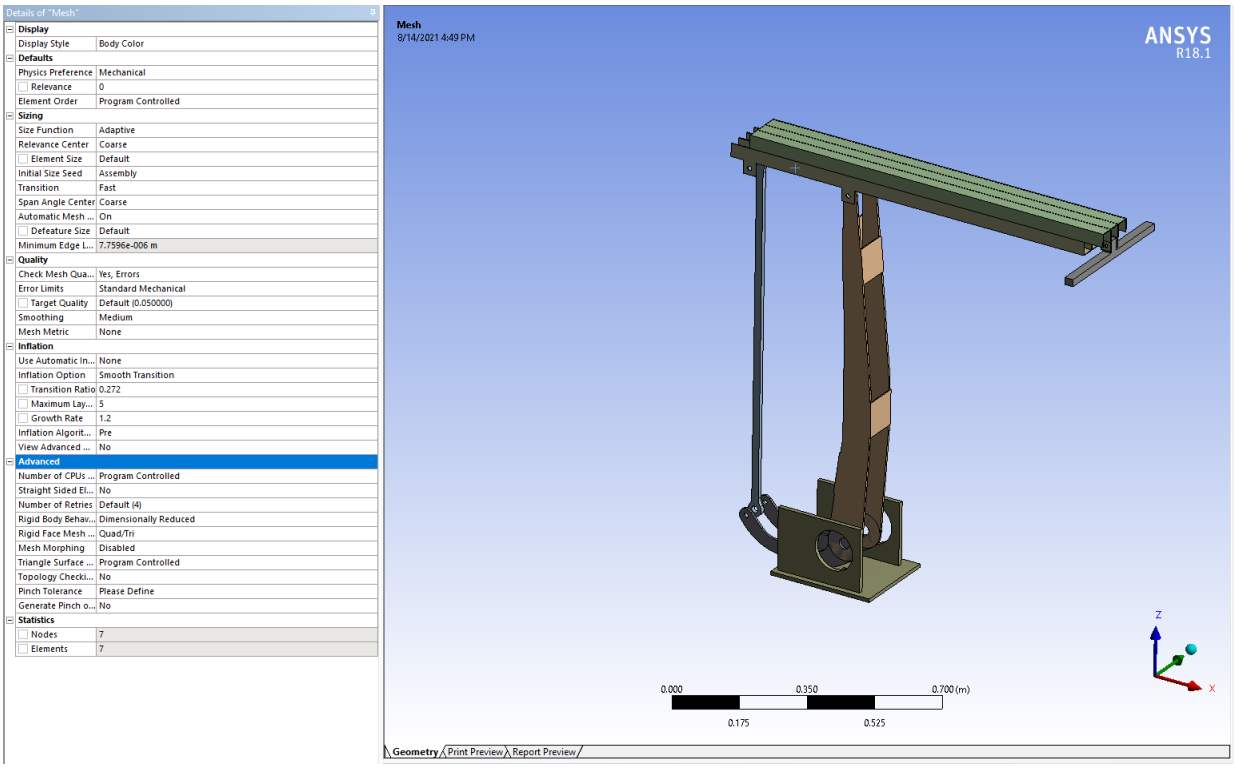



*Figure F.5: Mesh details of the Workbench model*

Inputting the motion of the robot is done by applying accelerations and velocities to the joints that have been defined. The inputs are based off of the trajectory created in chapter 2. As ANSYS can derive the position and velocity equations from the acceleration equation, only the acceleration Equations 2-78, 2-82 are needed. The initial positions of each of the joints is defined by the position of the robot when it is imported from SOLIDWORKS.

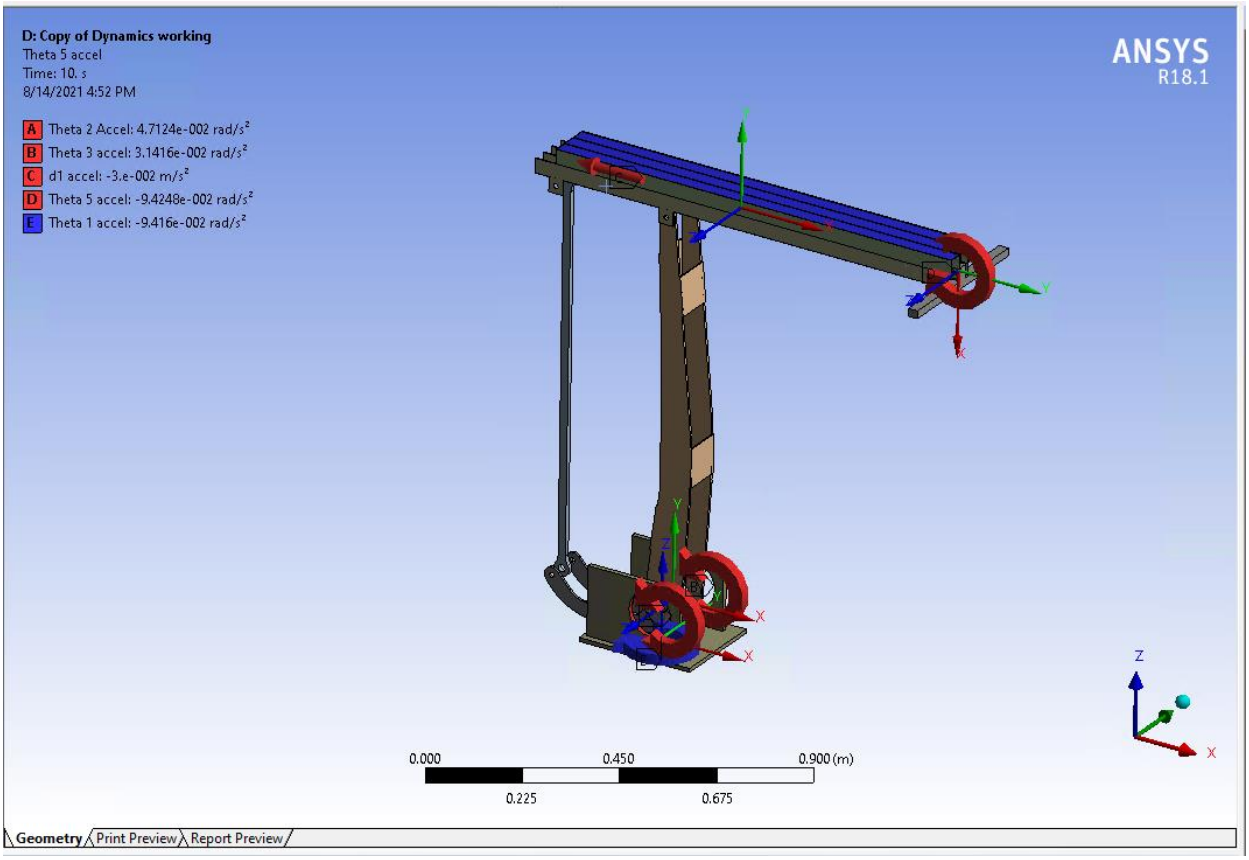



*Figure F.6: Overview of the different accelerations of each joint*

| Details of "Theta 1 accel" | |
|---|---|
| **Scope** | |
| Joint | Revolute - Ground To Base |
| **Definition** | |
| DOF | Rotation Z |
| Type | Rotational Acceleration |
| Magnitude | = 0.09424+(-0.01884)*(time) |
| Suppressed | No |
| **Function** | |
| Unit System | Metric (m, kg, N, s, V, A) |
| Angular Measure | Degrees |
| **Graph Controls** | |
| Number Of Segments | 200. |

*Figure F.7: details of the joint acceleration inputs*

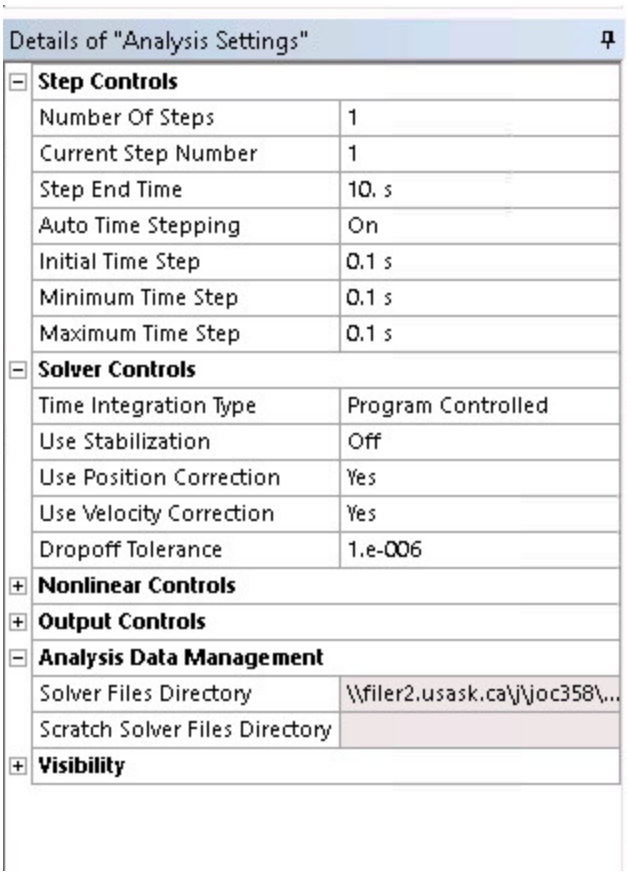



*Figure F.8: Analysis settings*

Under the “analysis settings” panel the time step is set at 0.1s. The step end time is set as 10 seconds ( the duration of the trajectory).

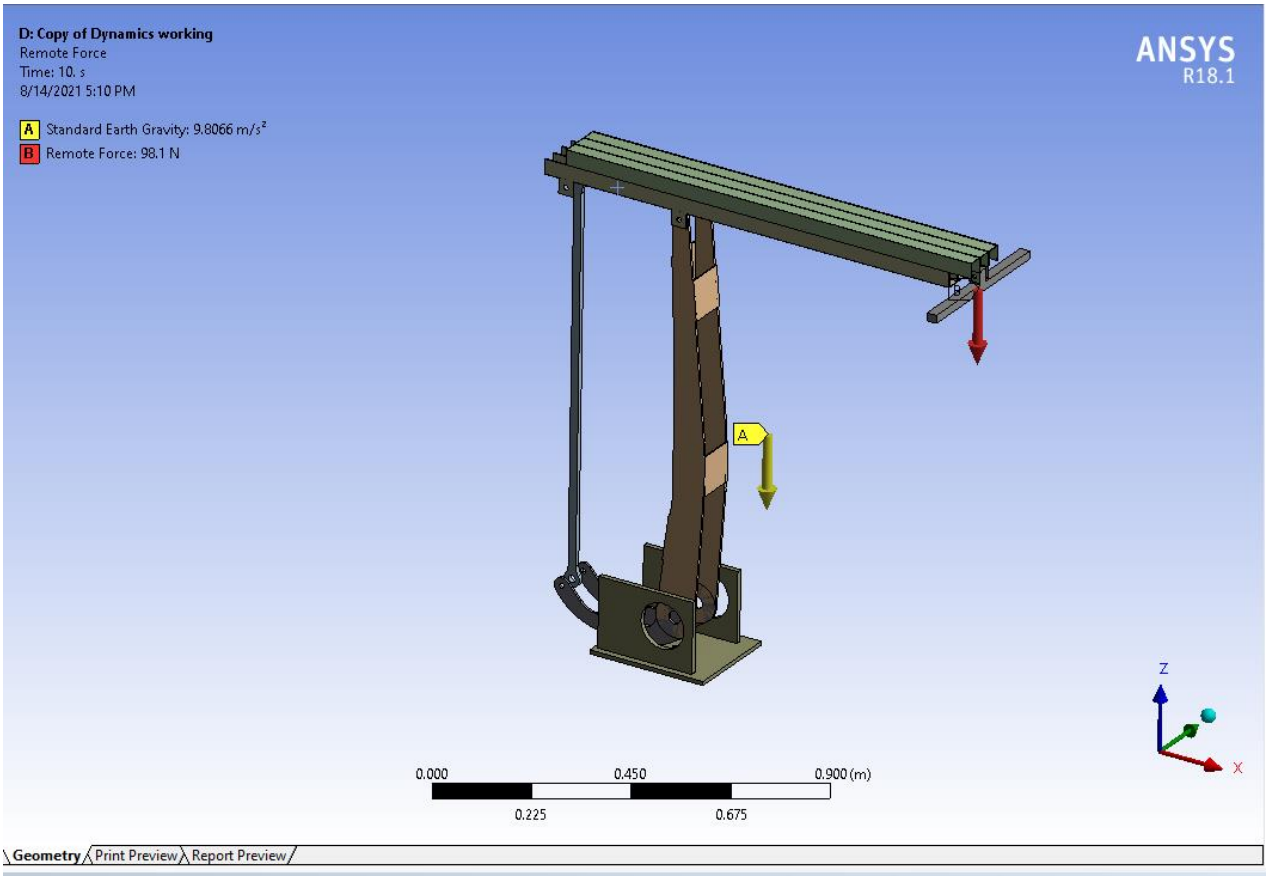



*Figure F.9: Boundary conditions applied*

Boundary conditions include the weight of the sensors at the end effector (10kg * 9.81 m/s$^2$), the acceleration due to gravity,  and fixing the position of the base plate, but allowing for rotation along the z axis.

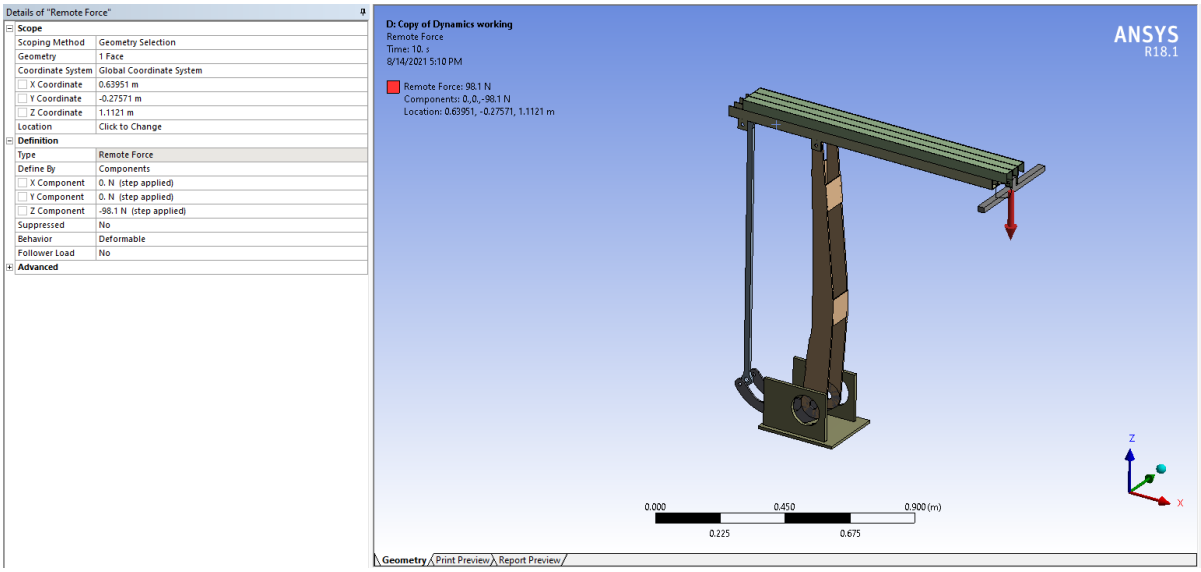



*Figure F.10: Specific details of end effector load*

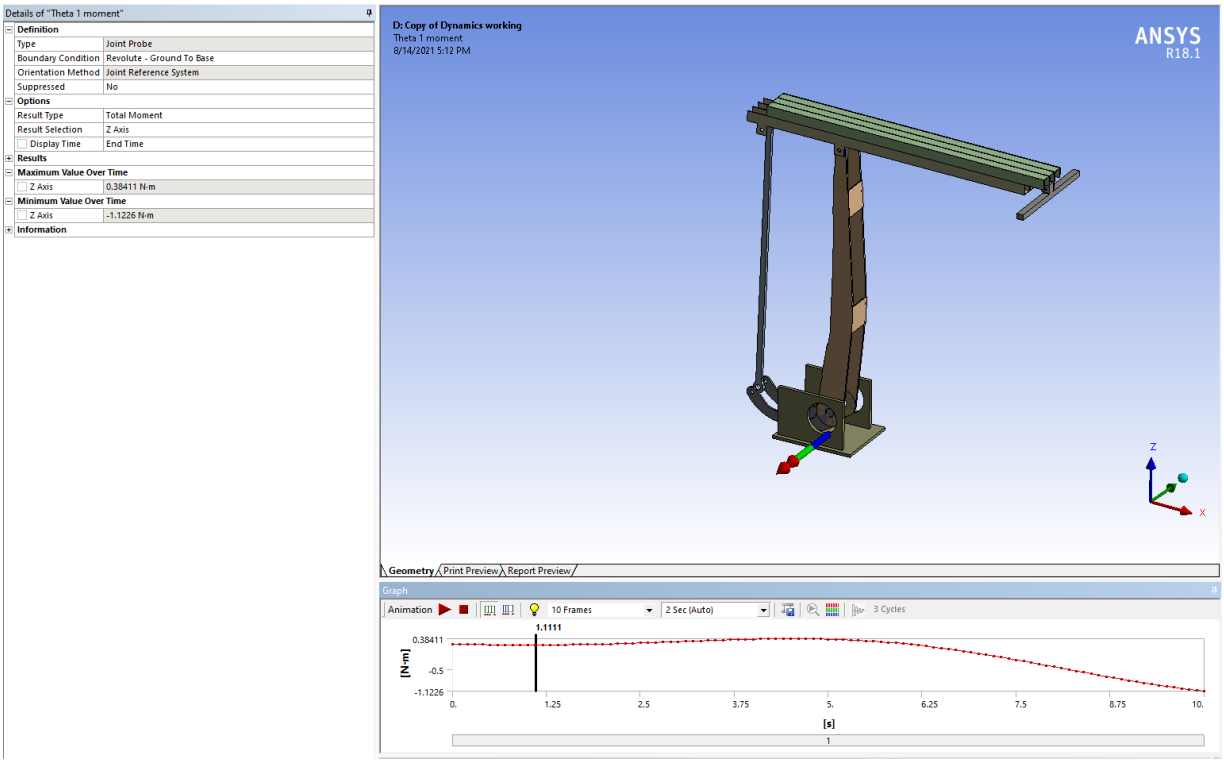



*Figure F.11: screenshot of the joint 1 moment measurement probe*

Once the model is complete, a joint probe may be placed at each of the joints to record the forces/moments that are generated during the motion of the robot.

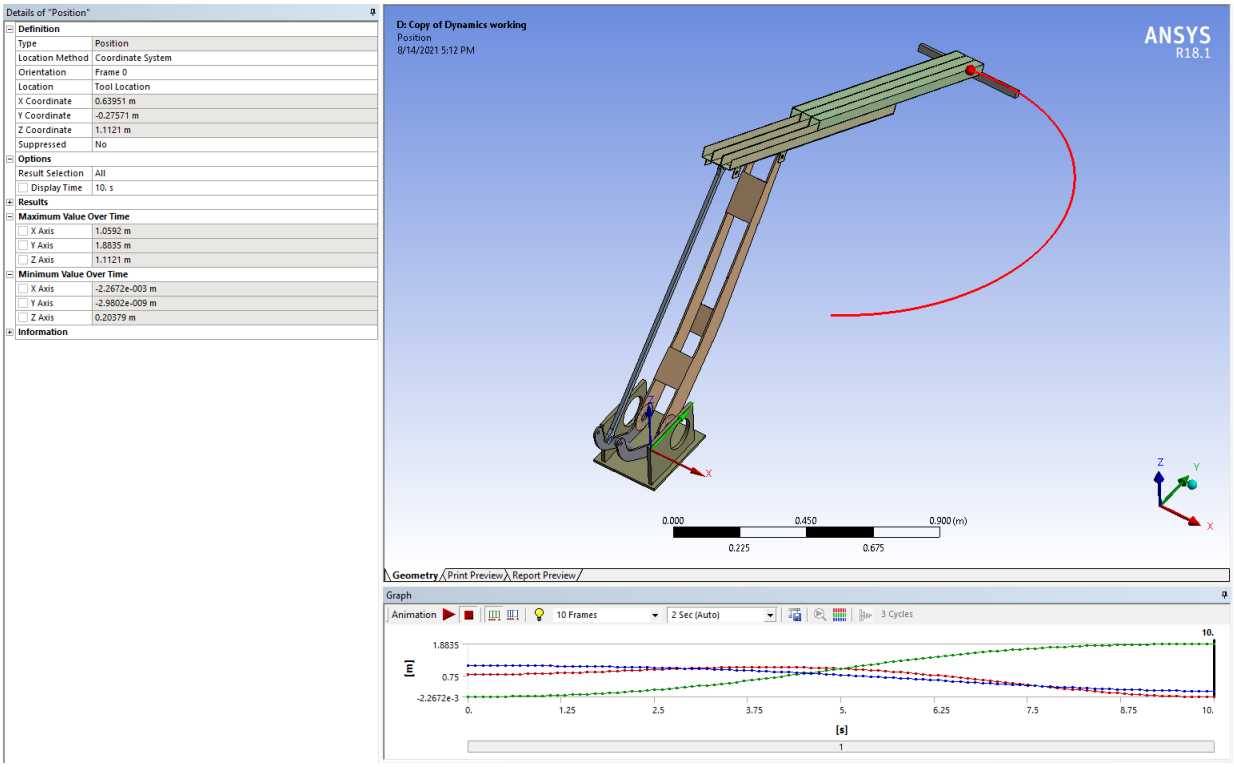



*Figure F.12: Details of the position probe placed at the end effector*

Additionally, the position probe can be place on the end effector to track the position of the end effector relative to the base of the robot.