

ABSTRACT

Title of dissertation: **TASK-BASED OPTIMIZATION
OF MULTI-ARM SPACE ROBOTICS**

Katherine McBryan, Doctor of Philosophy, 2018

Dissertation directed by: **Professor Dr. David L Akin
Department of Aerospace Engineering**

There are many benefits to using multi-arm systems over a single arm system including higher flexibility in planning, better payload handling capacity, and reduction of joint torques. However, multi-arm systems are inherently more complex. This complexity does not necessarily translate to “bigger” and “heavier”. This research seeks to answer the question of whether or not a multi-arm system can have lower mass than a single arm system.

Using a task-based methodology, Independent single-arm and cooperative dual-arm manipulator systems are designed. A task defines the payload’s motion and thus the manipulator’s trajectory. Utilizing linear programming, a new method is developed in order to optimize the distribution of forces among the multiple arms in order to guarantee a minimum system mass. The mass of the motors and gears are estimated based on the required torque and speed, obtained from the trajectory and force-distribution.

This study shows that a well-designed multi-arm system can in fact have a lower mass than a single-arm system. Further optimization demonstrates that a

multi-arm system, when designed as a complete system rather than individual parts, can significantly reduce the total system mass.

TASK-BASED OPTIMIZATION
OF MULTI-ARM SPACE ROBOTICS

by

Katherine McBryan

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor David L. Akin, Chair/Advisor
Professor Norman Wereley
Professor Raymond Sedwick
Professor Craig Carignan
Professor Linda Schmidt

© Copyright by
Katherine McBryan
2018

Acknowledgements

There are many people I would like to thank over the last couple of years. First and foremost I would like to thank my advisor, Dr Akin. Thank you for the years of support and guidance- words cannot express how much your guidance has meant to me. I would also like to thank the other members of my committee for their support throughout this process and over the years.

I would like to give a special thanks to the Satellite Servicing Projects Division at the NASA Goddard Space Flight Center, and to Brian Roberts, technical monitor. This work was sponsored under NASA SEAS contract NNG15CR66C. I would like to thank the Alcyon Technical Services Joint Ventures, who have provided the contractual arrangements to support the University of Maryland in this endeavor.

I want to thank all of the individuals at the Space Systems Laboratory for their continual support. My time at the Space Systems Lab has been an incredible experience and I wouldn't trade it for anything. There are so many people in the SSL that have helped me, I'm sorry I cannot mention you each by name. Nick Limparis, Chris Carlsen, Sharon Singer you all helped me so much.

In addition, I would like to thank my family and friends for their patience. In particular I want to thank everyone for not letting me quit to run away to join the circus or become a traveling bagpiper. Special thanks to my sister Sarah, truly the greatest of my all my siblings.

Finally, I would like to thank Wayne Yu, the most caring, patient, loving person ever.

Table of Contents

List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	3
1.3 Overview	4
1.3.1 Generation of Task	6
1.3.2 Estimating the Mass	7
1.3.3 Optimizing the Kinematic Structure	8
1.4 Organization of this Thesis	8
1.5 Literature Review	9
1.5.1 Background on Multi-Arm Robotics	9
1.5.2 Benefits of Multi-Arm Robotics	9
1.5.3 Multi-arm Space Robotics	12
1.5.4 Dynamic Models	16
1.5.5 Force Distribution Problem	17
1.6 Optimization of Robotic Systems	21
2 Kinematics and Dynamics	25
2.1 Task Description	25
2.2 Single Manipulator	28
2.3 Multiple Arms	30
2.4 Internal Grasp Forces	32
2.4.1 Dual Arm Planar Example	34
2.5 Conclusion	38
3 Mass Estimation	39
3.1 Estimating Motor Mass	39
3.1.1 Continuous Stall Torque	41
3.1.2 Closer look at Motors by a Single Manufacturer	44

3.1.3	Torque vs Speed	47
3.1.4	Estimating the Motor Mass as a Function of Torque and Speed	52
3.1.5	Sensitivity Analysis	59
3.1.6	Motor Mass Estimation Conclusion	63
3.2	Gearbox estimation	64
3.3	Link Mass Estimation	66
3.4	Conclusion	68
4	Mass Estimation of Single Manipulator	70
4.1	Examining a Docking Maneuver	75
4.2	Conclusion	80
5	Force Distribution Between Multiple Manipulators	82
5.1	Literature Review	83
5.2	Force Distribution: Even Force	84
5.3	Force Distribution: Minimizing Energy	91
5.4	Force Distribution: Minimize Mass	97
5.5	Conclusion	107
6	Dual Manipulators Moving a Large Spacecraft	109
6.1	Task Definition	110
6.1.1	Payload Properties	111
6.1.2	Trajectory	112
6.1.3	Manipulator Parameters	112
6.2	Force and Mass Estimation	113
6.3	Example Results	115
6.3.1	Study 1: Single Arm Systems	115
6.3.2	Study 2: Dual Arm System	118
7	Further Optimization: Preferred Link Lengths	122
7.1	Background	122
7.2	GA Implementation	124
7.2.1	Fitness Function	124
7.2.2	Variables	125
7.2.3	Initial Population	126
7.2.4	Bounds and Function Tolerance	126
7.3	Optimize Independent Systems	127
7.3.1	Independent Arm 1 Optimization	127
7.3.2	Independent Arm 2 Optimization	132
7.3.3	Conclusion for Independent Single Arm System Analysis . . .	134
7.4	Base Location Study: Independent Arm 1	135
7.5	Optimization of Multi-Arm Systems	139
7.5.1	System Optimization	139
7.5.2	Individual Optimize	143
7.5.3	Cooperative Dual Arm Study with Changing Base Location .	144

7.6	Conclusion	145
8	Contributions and Conclusions	147
8.1	Contributions	147
8.2	Design Insight for Multi-Arm Systems	149
8.2.1	When can moving to multi-arms reduce the total mass?	150
8.2.2	Shorter Arms are Not Always Lighter	152
8.2.3	When to Optimize a Multi-arm system	152
8.3	Future Work	153
8.3.1	Multiple Tasks	153
8.3.2	Trajectory Design	153
8.3.3	Internal Forces	154
8.3.4	Friction	154
8.3.5	Transferring Payloads	154
8.3.6	Further Optimization	155
8.4	Conclusion	155
	Kollmorgen Motor Curves	157
	Maxon Motor Curves	170
	Mag Motor Curves	174
	Woodward Motor Curves	176
	Moog Motor Curves	178
	Gems Motor Curves	180
	Anaheim Automation Motor Curves	182
	Code: Load Variables	186

List of Tables

1.1	Summary of DH-Parameters	24
3.1	Maxon Motor Family Descriptions	44
3.2	Coefficients for Linear and Power fit (Non-Robust)	54
3.3	Coefficients for Linear and Power fit (Robust)	54
3.4	Coefficients for Piecewise Non-Robust Linear and Power fit	55
3.5	Coefficients for Piecewise Robust Linear and Power fit	55
3.6	Coefficients for Linear and Power fit (Robust) for reduced datasets . .	59
3.7	Linear Coefficient for Mass/Torque Relationship for Harmonic Gearing	65
4.1	Simulated SRMS with link lengths	72
4.2	Material Properties for graphite epoxy	73
6.1	Task Definition: Payload Parameters	111
6.2	Task Definition: Trajectory Parameters	113
6.3	DH parameters for Docking Maneuver	114
6.4	Task Definition: Manipulator Parameters	115
7.1	GA Parameters	126
7.2	System mass with new link lengths from GA for Arm 1	130
7.3	System mass with new link lengths from GA for Arm 2	134
7.4	Original Single Arm System vs Optimized Single Arm System	135
7.5	Table of link Length Studies	137
7.6	System mass with new link lengths from GA for Arm 2	140
7.7	Summary of Optimization Results	143
7.8	Multi-arm Optimization with Base	145

List of Figures

1.1	Mass estimation flowchart for single-and multi-arm systems	5
2.1	Example 1-dimensional trapezoidal trajectory	26
2.2	Example 2DOF Planar Dual Manipulator System	35
3.1	Replication of Mass/Torque plot by Dermizanki et al	41
3.2	Maxon BLDC: Torque vs Mass, Limited Dataset	43
3.3	Maxon BLDC: Torque vs Mass, Full Dataset	43
3.4	Maxon Motor per torque	45
3.5	Probability-Residual plot for each Maxon BLDC Family	46
3.6	Speed/Torque Curve for Kollmorgen Motor	47
3.7	Mass/Torque Curve at 0RPM	48
3.8	Speed/Mass Curve at 5000RPM	49
3.9	Raw data for Motor Mass, Continuous Torque, Motor Speed	50
3.10	Regression analysis for all motor data	51
3.11	Regression analysis of full dataset	56
3.12	Regression analysis of small motor dataset	57
3.13	Regression analysis of large motor dataset	58
3.14	Sensitivity Analysis for Linear and Power Fit	61
3.15	Sensitivity Analysis for Linear and Power Fit, Variables c_3 and c_4	62
3.16	Harmonic Drive, LLC, Gearing Mass Relations	65
3.17	Stepped cantilever beam used for link mass estimation	66
4.1	SRMS mechanical parts. Image Source: IEEE.ca [1]	71
4.2	Research Overview: Single Arm Study	73
4.3	Mass vs Manipulability metric for each task. Dotted lines denote the +/- 10 % of the actual RMS Mass	74
4.4	Mass comparison between worse case task and actual mass for SRMS	75
4.5	Starting, Midpoint, and Final point of Docking Maneuver	76
4.6	Summary of Docking Trajectory	77
4.7	Joint Torques during Docking Maneuver	78
4.8	Link and Actuator Mass per Joint	79
4.9	Link and Actuator Mass per Joint	79

4.10	Percentage of Joint Mass to Total Mass	81
5.1	Simple Planar motion of SRMS type arms: symmetric case	86
5.2	Force Distribution for Evenly-Distributed Symmetric Case	87
5.3	Joint Torques for Evenly Distributed Symmetric Example	88
5.4	Simple Planar motion of SRMS type arms: Non-Symmetric Case	89
5.5	Force Distribution for Evenly Distributed Non-Symmetric Example	89
5.6	Max Joint Torque for Evenly Distributed Non-Symmetric Example	90
5.7	Minimum Energy Force Distribution, Symmetric Example	95
5.8	Maximum Joint Torque for Minimum Energy Symmetric Example	96
5.9	Total Energy for Symmetric Example	97
5.10	Force Distribution for Mass-Minimization Method	101
5.11	Maximum joint torque for a minimum-mass force distribution	102
5.12	Force from each manipulator on the payload at start of trajectory	103
5.13	Total joint mass for each force-distribution method	104
5.14	Maximum Joint Comparision for Force Distribution Methods	105
5.15	Total Joint Mass Comparison for all Force-Distribution Methods	106
5.16	Total Energy Comparison for all Force-Distribution Methods	106
6.1	Research Methodology Overview	110
6.2	Starting and Ending points for Arms 1 and 2	116
6.3	Starting and Ending points for Arms 1 and 2	117
6.4	Mass for links and joints for single arm docking example	117
6.5	Total Mass for Single Arm Docking Example	118
6.6	Starting and Ending points for Arms 1 and 2	118
6.7	Starting and Ending points for Arms 1 and 2	119
6.8	Starting and Ending points for Arms 1 and 2	120
6.9	Starting and Ending points for Arms 1 and 2	120
6.10	Percentage of total system mass due to Link masses	121
7.1	Mass Estimation for Arm 1 with a variety of link lengths	128
7.2	Position Error for Arm 1 over a variety of link lengths	128
7.3	Estimated mass of Single Arm1 as a function of link lengths	129
7.4	Joint Torque and Mass for Optimized Arm 1 Performing Docking Task	131
7.5	Joint Torques and Mass of Arm 1 Performing Docking Task	132
7.6	Arm 2 Optimization Results	133
7.7	Total Mass Comparison of Single Arm Optimization	134
7.8	Joint Torque and Mass of Arm 1 Design with Base Optimization	136
7.9	Comparison of Arm 1 Base Optimization Results	137
7.10	Mass Summary of Independent Arm 1 Base Location Optimization	138
7.11	Arm 1 with optimized link and new base location	138
7.12	Single Arm1 Torque: Original vs Optimized	141
7.13	Dual Arm Torque: Original vs Optimized	141
7.14	Dual Arm Joint Mass: Original vs Optimized	142
7.15	Dual Arm Link Mass: Original vs Optimized	142

7.16 Dual Arm Mass: Original vs Optimized 144

Chapter 1: Introduction

1.1 Motivation

Human expansion, both geographically and scientifically, drives the development of newer machines, computer systems, machine learning, and robots. These newer technologies help us to perform tasks we cannot do, and to work in environments we cannot currently work in. Space is one of the harshest environments in which humans live and work today. Serial manipulators on the International Space Station (ISS) help to accomplish spacecraft berthing, station maintenance, and were instrumental in the assembly of the station itself. In the scope of space operations, dexterous space manipulators are a great asset, expanding the frontier of in-space assembly and repair, assisting astronauts, and supporting experiments.

As robotic systems become more sophisticated, they can accomplish increasingly more complex and difficult tasks. Working with one dexterous manipulator at a time is limiting, like performing daily household chores with one hand tied behind your back. The addition of secondary manipulators is one method of expanding the task performance of current robotics systems. Multi-arm robotics can accomplish tasks which single arm systems are incapable of.

Multi-arm systems provide flexibility in planning and improve handling ca-

pability. Some tasks, such as the coupling of two objects, can only be done with multiple arms, e.g., dual manipulators. Other tasks, such as carrying heavy objects, are easier with two arms, requiring less force and effort. In terms of robotic systems, easier in a multi-arm system translates to lower joint torques required to carry out a given task. This reduction in joint torques allows a pair of weaker arms to move the same load as a single stronger arm. Furthermore, this reduction in joint torque can be turned into a reduction in motor mass.

Weight is an essential concern for any system that needs to get into space. Space is not only a harsh environment to work in; it is also exceedingly difficult to get there. Weight of robotic systems becomes an important factor when considering how to first get to, and then work in space. This leads to an important question: Is there a multi-arm dexterous manipulator system that is lighter in weight than a single arm system that can perform the same tasks?

Multi-arm systems are often created by taking an already designed manipulator (usually capable of performing the task individually) and simply adding an additional cloned arm. This results in many multi-arm systems being at least twice as heavy as the single-arm system. In robotics, the idea that a dual (or even three) arm system could be lighter than a single arm is therefore counter-intuitive. The individual manipulators in a multi-arm system do not need to be as strong as that of a single arm, since they do not need to produce as much torque as a single manipulator system. One key way of reducing the weight of the system is in reducing the mass of the motors.

As previously mentioned, many multi-arm systems are made by attaching pre-

designed and existing arms together. This leads to the erroneous conclusion of multi-arm systems not only being complex, but also heavy. While the first conclusion is true in many cases, especially facing challenges of efficiency and safety, the second doesn't have to be. If each arm of a multi-arm system is capable of producing 100% of the necessary forces to accomplish the task, then the resulting system will be over-designed and will weigh more than a single arm. These over-engineered systems can be avoided by designing for a whole system rather than designing individual manipulators. It is unnecessary for every arm to be capable of performing 100% of the task. By working together, forces can be distributed and optimized to reduce the overall mass.

By changing the focus of design to the whole system rather than individual elements (e.g., individual manipulators), each arm can be optimized and sized to work together to perform a desired task; resulting in a lower mass.

1.2 Contributions

A multi-arm robotic simulation was created to model how both a single and multi-arm system would perform, given a specific task. The simulation calculates the manipulator(s) dynamics and then estimates the mass of the manipulator structure and its motors using estimated joint torques. In the case of the multi-arm system, the forces are optimally distributed among the arms in order to minimize the overall mass. This optimization allows a multi-arm system design to be as light, or lighter, than a single arm system.

Determining how much force and torque each arm must contribute to the overall task is commonly known as the "force distribution problem". Many methods look at the minimization of power in order to reduce the mass; however, while lowering the power can lead to a lower mass, it is not guaranteed. While multiple methods of solving the force distribution problem have been studied, none focus on the minimization of the system mass.

This thesis looks at how to optimize a multi-arm system for space applications. The study compares the system mass of a single arm system to that of multi-arm systems. In addition to comparing the systems, a genetic algorithm optimizes the individual link lengths for both the single arm system and that of the multi-arm system. This shows how the optimization and design of the multi-arm system, as a whole, further reduces the total system mass.

Multi-arm systems can be a lighter solution for many tasks. However, the multi-arm system must be designed as such from the ground up, taking into consideration the system as a whole. This thesis will demonstrate that a well-designed multi-arm system can be lighter in weight than a single manipulator.

1.3 Overview

This section will provide an overview of the research methodology used. A simulation was written in MATLAB using the optimization toolbox as well as Peter Corke's robotics tool box. Using these tools, and custom code, a simulation was made to estimate the mass of manipulators as they perform a given task. The mass

of the manipulator(s) is based on the required joint torques as well as structural limits.

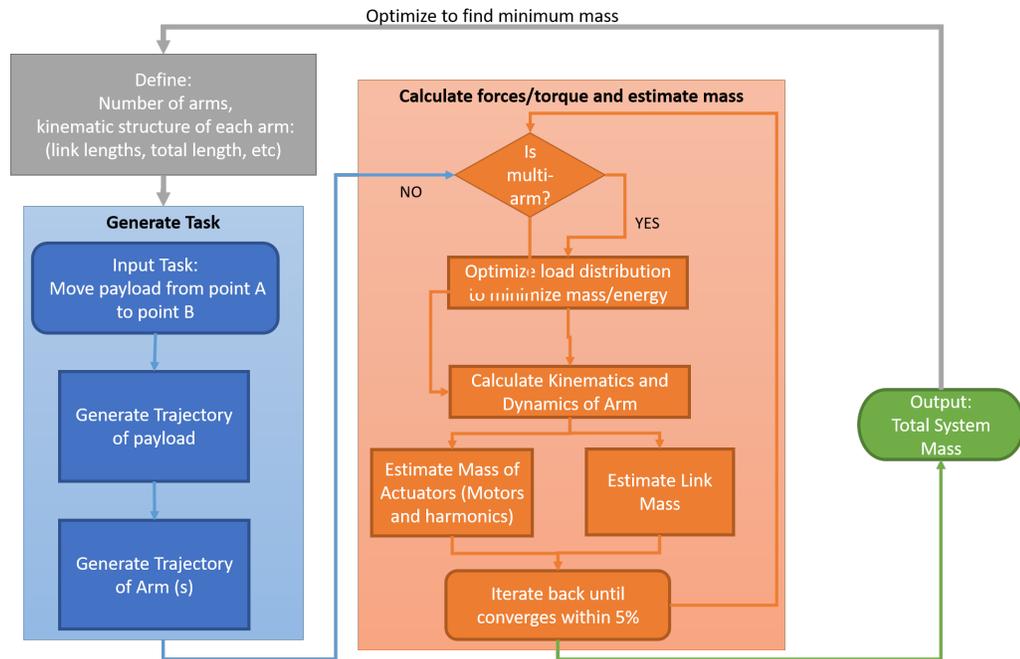


Figure 1.1: Mass estimation flowchart for single and multi-arm systems

This simulation reduces a complex manipulator system to a total system mass. For a single arm system, this is the mass of the single arm. For a dual-arm system, this is the summation of both arm masses. This allows direct comparison and optimization between the two systems. The various link lengths of the arm are then optimized to produce the system with the lowest system mass, not only allowing comparisons between single arm and multi-arm systems, but between the optimized systems as well.

1.3.1 Generation of Task

The task-based methodology starts with describing which task(s) must be performed. This can be seen in the first column of Figure 1.1. There are several constant variables embedded within the task definition, including key characteristics (mass, gripping locations, and general size) of the payload. For the purpose of this study, payload mass is set to be 14,000kg, which is near the maximum rated mass the Shuttle Remote Manipulator Systems (SRMS) can move at a rate of 0.6m/s [1].

This study focuses on moving a large inertial payload from point A to point B. The starting and ending points are both described as positions in 3D space (x, y, z) at a rotation about the x,y, and z payload axis, R_x, R_y, R_z . The rotation terms do not have to remain constant from start to finish. The task is described as moving the payload, not only translationally, but rotationally as well.

Once the end-points of a task are defined, a trajectory is generated for the payload. This is created using a trapezoidal trajectory function. Some additional requirements need to be imposed on the simulation, such as maximum speed of the payload/end-effector, maximum acceleration, and desired time to accomplish the trajectory. If the desired time to accomplish the trajectory is set too low, then the system will complete the trajectory as fast as possible, moving at the specified maximum acceleration and maximum velocity.

The manipulator(s) are assumed to be rigidly attached to the payload. As mentioned above, the location and angle of the manipulators relative to the payload are pre-defined. While the gripping locations must remain constant throughout

the task completion, the gripping angle does not. If the starting angle is defined differently from the resulting angle, then a change of angle will occur throughout the course of the trajectory. While the desired location and angle of each manipulator's end-effector are not explicitly given, they can be calculated based on the path of the payload. Using inverse kinematics, the pose of the each arm can be determined. For redundant manipulators, there is an infinite number of solutions, and the program will select the one closest to a defined 'default' pose.

1.3.2 Estimating the Mass

Once the trajectory for each arm has been calculated, it is possible to obtain the force and torque for each joint. For a single manipulator system, this is done using forward dynamics. For a multi-arm system, this becomes more complex, as the force from the payload can be divided among various arms. The amount of force each arm provides to the payload will directly affect the resulting joint torques and, thus, motor masses.

The mass of the motors and the structure will affect the dynamics of the system. Therefore an initial estimation of the dynamics is found without masses. The kinematics and dynamics are then recalculated with the estimated masses. This process is then reiterated until the change in system mass converges within a 5% change from the previous iteration. More information can be found regarding the heuristics for estimating the motor mass, gearbox mass, and structural mass in Chapter 3.

1.3.3 Optimizing the Kinematic Structure

In order to first calculate the arms' trajectories, the kinematics of each arm must first be known: specifically, the link lengths and joint configurations. Variable link lengths will result in different trajectories, dynamics, joint torques, and thus different manipulator masses.

The final step of this research will look at how to further optimize a multi-arm system for mass reduction. This study will focus on the differences between the optimization of individual arms and of the system as a whole. This optimization uses a genetic algorithm to calculate the optimal link lengths for single and dual-arm systems. These results are presented in chapter 7.

1.4 Organization of this Thesis

Chapter 1 focuses on background and previous work which has been done on multi-arm robotics. The forces and torques of robotic systems are described in Chapter 2. Chapter 3 describes the model used for estimating the mass of a manipulator. An example single arm is described in Chapter 4 utilizing the forces described in Chapter 2 and the mass model from Chapter 3. Chapter 5 describes how the force/torque can be distributed among the multiple robotic manipulators to better reduce the joint torques and, thus, the mass. Chapter 6 examines the results of moving from a single arm system to a multi-arm system. Chapter 7 describes the optimization process for each arm to further reduce the overall system mass. Conclusions and future works can be found in Chapter 8.

1.5 Literature Review

1.5.1 Background on Multi-Arm Robotics

There are a variety of ways in which multiple robotic systems work together, including performing individual sub-tasks to finish an overarching task. In this case, the robots themselves have no real interaction with each other, and are generally in different work volumes. This is common on the floor of automotive plants. There, multiple robots are used to handle tasks which do not require too much delicacy, such as heavy lifting, welding, painting, etc. Other tasks, such as wrapping tape around a pipe, means the manipulators must work closely together. However, the task must be safely transferred from one robot to the other[2].

This method of utilizing a multi-robot system is a coordinated effort means every robot knows its own task and performs them without aid from one another. However, this research focuses on cooperative multi-robot systems, in which the robotic systems actually cooperate with one another in order to better perform the task. Forces and torques pass from one robot to another; the robotic systems truly interact with each other and cooperate to accomplish a given task.

1.5.2 Benefits of Multi-Arm Robotics

Using multiple robots, to solve a problem or perform a task, is becoming more and more common as the capabilities of systems increase. The obvious time to use multi-arm systems is when the task requires it: a task which a single robot cannot

accomplish independently. Fujii and Kurono are some of the first researchers to study the control of dual manipulators, with a focus on tasks that required dual manipulators. They focus on the joint control of dual manipulators to perform a desired Cartesian motion, such as screwing a bolt into a nut. One arm holds the nut while the other turns the screw. However, it should be noted that this method does not account for the coupling effect between the two manipulators [3].

Multi-arm systems are beneficial even when a task does not necessarily require multiple arms. [4, 5, 6, 7, 8, 9, 10] discuss the uses and classification of multiple robot systems. Sirouspour talks about the increased dexterity, improved handling capacity and enhanced robustness, due to redundancy, of multi-arm systems. Their main focus is on the teleoperation of such a system using multiple "master/slaves" to control robots [11]. In their survey on dual arm manipulation, Smith et al. states that dual arm manipulators are far more familiar to the operator since it has the common human form factor. This familiarity aids in control and planning [12, 13].

Payload handling is one area that benefits from the use of multi-arm systems. Multi-arm systems handle larger objects better than a single arm [14]. In addition, they allow for multiple grasping points, which, in turn, allows any necessary moments/reactions applied to the payload to be countered. The various arms help balance out the moments and torques necessary to move a payload. This is important when handling large objects, where grasping an outer edge can result in large undesirable torques. This is especially true for space manipulators, as the center of mass, which is inside a frame or structure, is a highly coveted space for any and all propulsion tanks and sensors. [15]

Many researchers focus on how multi-arm systems aid in handling complex objects. These complex objects can be flexible, fragile, or mobile. Sugar and Kumar focus on performing tasks which require more than a single robot with flexible objects [16]. Bais and Erhart focus on payloads, in the case of constant and slow time-varying load constraints [17]. While it is highly preferable to have a constant load, it is possible for an uncooperative satellite to start thrusting or moving as a manipulator grabs it. Nakamura et al. studies how using multi-arms can minimize the strain put on a payload in order to reduce the possibility of damage [18].

Many multi-arm systems focus on holding objects using friction, similar to how the fingers on a hand work together to hold an object. This allows the movement and manipulation of payloads without requiring special fixtures for robots to rigidly attach to. However, relying on friction to hold a payload in space is often a poor choice [19]. The complex dynamics result in operators wanting a firm, predictable, and often dangerous grip on the payload, which are typically extremely expensive satellites or even modules of a space station. These objects are all man-made and generally adding grasping fixtures or finding an appropriate places to grab hold would not be too difficult. Generally, the operator will have a full, or close to full, model of the object they need to handle. This research focuses more on multi-arm systems which do not rely on a friction hold, but rather are rigidly attached at pre-defined grasping locations.

Often, the goal of aerospace designs is to reduce the system mass. In this study, this is accomplished by reducing the motor mass by distributing and reducing joint torques. Yoshida et al. designed a coordinated controller for a dual-arm space robot

and showed that the torque required to follow a determined path is smaller when using two arms [20]. Carignan and Akin show that the force each arm provides to the task is based on the current geometry of the system, which can be used to reduce the joint torques and to minimize the energy in order to accomplish a task [21]. Researchers have found that multi-arm systems can lead to a reduction in joint torques, improved payload handling, and overall more intuitive planning than single arm systems.

1.5.3 Multi-arm Space Robotics

Space robotics have proven themselves to be useful tools in assembling the International Space Station, EVA (extravehicular activity) assistance, and on-orbit servicing. There are a few key defining characteristics of a space manipulator as compared to their ground-based counterparts. As discussed earlier, weight is a major concern for any aerospace project.

The coupled dynamics between a manipulator and its base is another major concern for space manipulators. Ground-based manipulators are firmly mounted to either the ground or to a platform mounted to the ground. The movement of a manipulator is not going to change the movement of the base (i.e. Earth) that it is mounted to. This is not true for on-orbit manipulators. A space manipulator must react all forces through the spacecraft they are attached to. As the manipulator imparts a force on the base, the base will move. How much the base moves is dependent on the inertial mass of the base, the arm, and the acceleration. The

movement of the base must be considered, making planning and control of the system more difficult [22, 23, 24]. A reduction in the coupled dynamics translates to space manipulators behaving more like ground-based ones.

One method of reducing this coupled dynamics is to move slowly with a limited acceleration. Momentum wheels or control moment gyros can be utilized to stabilize a spacecraft, but the amount of torque they can counter is limited. Carignan and Akin describe how reaction stabilization systems can be used to counter the moment of a space manipulator [25].

Many industrial robots are performing the same task over and over again. The quicker the task time, the more productive the robot. Therefore, there is a high premium put on speed. In comparison, space manipulators move much slower. This reduction in speed also reduces the coupled dynamics. In addition, space manipulators generally do not perform the same repeating task. Instead, each individual task is carefully planned and practiced.

Researchers have been studying the control of space manipulators for decades. Many researchers focus on how to move the space manipulators, such that the satellite movement is minimal. Umetani and Yoshida developed an inverse kinematic solution by defining the Generalized Jacobian Matrix for a multi-arm system [20]. This creates a Jacobian which is a function of the joint angles and inertia parameter for multiple manipulators. They proposed a method for using two arms: one as the main focus and the other as a “balancing arm”. The balancing arm would counter the primary arm’s movements to maintain satellite attitude. Yale et. al. use the redundant degrees of freedom from a multi-arm system to minimize a weighted norm

of the actuator torques in order to reduce motion of the base [26].

Yoshida and Hashizume proposed the zero reaction maneuver. This uses the manipulator's degrees of freedom to carefully plan a maneuver which results in no effect to the base during movement of the manipulator [27]. There is a limited set of maneuvers for a 6 degrees of freedom (DOF) manipulator; however, as the DOFs increase, the number of possible zero reaction maneuvers also increases. A 9-DOF manipulator can use the extra DOF to perform zero reaction movements and keep the motion of the manipulator separate from that of the base. However, the additional degrees of freedom greatly increase the complexity of the system.

Other researchers have studied how to predict, and thus eventually control, the motion of the base as a manipulator moves. Dubowsky and Vafa developed a virtual manipulator method, which allows ground-based manipulator control methods to be used on a free-flyer [28]. Papadopoulos described the dynamics and the control of a manipulator on a free-floating spacecraft in terms of the barycenter [29]. This allows the the dynamics of the arm and the spacecraft base to be calculated relative to the non-moving center of mass. This method allows the base motion to be predicted for additional control of the system. Dubowsky and Papadopoulos continued studying free-flying manipulators and found there exists dynamic singularities as a result of being on a free-flying base [30]. These singularities are path dependent, and should therefore be taken into account when developing trajectories for space manipulators. Papadopoulos and Nanos focused on avoiding the singularities for a free-floating manipulator in Cartesian motion [31].

Longman studied how a robotic manipulator, mounted to a free-floater, can

move a large load. He found that the work space of the arm on a free-floater is larger than a fixed-based manipulator [32]. This is difficult, as the dynamics and the satellite altitude are a function of the path, and not just the current joint angles. His research shows that the work space of the manipulator is load dependent, and is inversely proportional to the load. He found that there is a “desired base orientation” and that should be taken into account with path-planning.

Many of the existing manipulators in space are multi-arm systems, such as Dextre. Dextre, also known as the Special Purpose Dexterous Manipulator (SPDM), was launched in 2008 and is currently aboard the ISS. It was built by MacDonald, Dettwiler, and Associates (MDA) and is maintained by the Canadian Space Agency (CSA). Dextre is a dual-arm manipulator: each arm has seven joints, a length of 3.35 m, and a total mass of 1560 kg [33][34].

Dextre and other multi-arm space systems are highly capable; however, operationally, the multiple arms do not typically interact with one another. Dextre, despite having dual dexterous manipulators, only operates one at a time [35]. There are very few instances of space manipulators working together. 2001 saw the first in space robot-to-robot transfer as a payload was transported between robot arms. A space lab pallet was passed from the SSRMS (Space Station Remote Manipulator System) to the SRMS (Shuttle Remote Manipulator System) [36]. This was done using single joint motions so as not to put major demands on the Station’s computers, [13] and was done in such a way that the robot arms had very minimal or no interaction. Though multi-arm systems are currently working in space, they are not being used to their fullest potential.

1.5.4 Dynamic Models

There are many benefits to the cooperative manipulation of multi-arm systems and their ability to move a payload in terms of capacity, dexterity, and even flexibility[11, 37, 38, 39, 40]. This allows for more complex tasks to be completed.

In general, there are two approaches when planning how robots manipulate a payload: model-based and reactionary-based. The first approach utilizes a model which generates a set of desired physical parameters to accomplish the task, such as desired path, joint torques, etc. The robots are then controlled to move to those desired parameters [41][42]. The second approach is reactionary based, where the robots interact with the payload and are controlled to deal with the resulting dynamics. This can be seen in the leader-follower method of dual arm transport: a leader moves and the follower, estimating the motion of the leader, performs the cooperative task estimation. All desired motion information is passed to the follower robots by moving the object. The follower's path is determined by the payload's actions[43]. This method requires a powerful leader. Alberts and Soloway state that the master/slave or leader/follower architecture for dual-arm control lacked the primary benefit insofar as the arms do not truly share the load bearing task. Space manipulation requires careful planning and practice; a model-based system is used to ensure predictability[37].

Many of the early studies focus on the coupled dynamics between the manipulator and the payloads. Hybrid position/force controllers are used to control dual manipulators as they are allow to exert forces and torques onto each other [20, 44].

Hemami used a master/slave scheme and studied the relative position of the slave's end-effector to the master end-effector [45]. He ensured the motion between the manipulators were symmetric either about a point or a plane. Pagilla and Tomizuka did not assume the payload's center of mass or inertial properties were certain [46]. They developed an adaptive motor controller and parameter adaptation laws for estimating the uncertainties. They show that the nullspace of the grasp matrix depends only on the location of the grasp points, and not the payload's center of mass.

1.5.5 Force Distribution Problem

Moving a large payload is a common task for studying multi-robot cooperation. Here, multiple systems work together to move an object which is beyond the capability of a single system. There exists a near infinite number of combinations to distribute the load across the multiple robots. As previously mentioned, this is commonly referred to as the load distribution problem, which has been studied for a number of years. Alberts and Soloway present a force compensated control method for multiple manipulators, with the load distribution being handled by minimizing a quadratic cost function in task-space force and torque [37]. Bais et al. present an allocation strategy to distribute a desired payload torque between two heterogeneous cooperative manipulators using weighting factors[17]. Experimentally, they demonstrate that different weighting factors give rise to additional torques which act on the payload. In an effort to minimize power, Nahon and Angeles minimize

joint torques; this is an important consideration for space-based systems[47]. Their proposed method show that minimizing internal forces was actually preferred to minimizing the power.

Nakamura proposed a hybrid position/force control system to provide a necessary internal force, concluding that both position and force control are required[41]. They divided the manipulator control into two phases: the first determines the resultant force to control object's trajectory and specifies the stiffness for the external forces, and the second phase considers uncertainty and variety of the static friction in order to determine the desired internal force. Wen and Kruez-Delgado also decompose the problem into two control problems: motion and force[48]. The internal force, also referred to as the squeeze force, is optimized about a desired set point.

Koga assumed all the arms had the same performance and, thus, the load sharing was equal among the manipulators[49]. He studies the motion of the payload in order to resolve the desired internal force. The results are twofold: it prevents the manipulators from damaging the object by applying too much internal force and also prevents them from dropping the payload by not applying enough force.

Kerr and Roth focused on a linearized friction model where they found the optimal internal force[50]. They did not assume the object was rigidly attached, but rather, that there was a pure rolling motion between the payload and each finger. This led to the point of contact between the payload moving in relation to the fingers. Hu and Goldenberg [51] proposed an optimal control approach to control the motion of the object and the internal force, this defining a constraint that the joint torques must satisfy in joint space. They developed two coordinated control schemes,

both of which uses torque optimization and internal force constraints. An adaptive law, which is used to estimate uncertain parameters, is based on error equations for position, contact force, and the internal force. Zheng and Luh studied a system of manipulators where one arm is under position control and the other arm(s) are subject to compliance force control to maintain a kinematic constraint[52]. They utilize a master/slave relationship. Gueaieb, Karray, Al-Sharhan focused on the real time control of a master/slave architect for joint trajectory generation for cooperative systems[53]. They focused on using neural networks and neurofuzzy systems to better calculate the inverse kinematics for the robotic system.

Many of the studies, based on internal forces, assume the load sharing is symmetric or known beforehand. Hayati developed a control architecture for position/force-torque control for cooperating arms and, using a simplifying artificial constraint, relates the load distribution between the multiple arms to the mass and the inertia tensor of the payload[44].

Carignan and Akin proposed an optimization of the total torque between multiple manipulators rigidly attached to a single payload[54]. They show that the system can be optimized to minimize the total torque during a motion, resulting in significantly lower energy consumption. They do not consider the internal forces.

Uchiyama introduces the concept of a ‘virtual stick’ which is fixed to the hand of each robot and extends through to the center of the payload. The absolute and relative positions/orientation of the payload and hands, respectively, are given as work-space coordinates allowing a simple coordinated control scheme to be used [40]. Laroussi, Hemami and Goddard focuses on dual manipulators lifting a load and

transporting it[45]. The two arms are rigidly attached to the payload. The contact forces were estimated using knowledge of the trajectory and the input torques and does not use force feedback.

Recently, there has been a paradigm shift in the focus of coordinated cooperative control. It is no longer assumed that specific characteristics of each payload are completely known. Uncertainties in the dynamics and kinematics of the payload and the manipulators themselves have resulted in the use of adaptive and fuzzy control schemes. Wang, Liu and Zhang [55] utilize the adaptive Jacobian control method to accomplish a trajectory tracking task. This method uses a position/force hybrid controller and optimization to both minimize the power consumption and ensure the squeeze force does not deviate too far from the set-point. The internal force is treated as an external disturbance, since it does not affect the motion.

While many researchers deemed the use of feedback sensors necessary, alternative and cost effective methods have also been explored. Liu and Malek use impedance control without external force feedback to control each robot using a sliding mode control algorithm. Experimentation is used to find a selection of the proper internal force vectors when necessary[56]. Rastegari and Moosavian [57] specifically focuses on space robotics. They propose the use of an multiple impedance control to enforce reference impedance, not only on both the manipulator's end-effector, but also the payload. An impedance controller enforces the relationship between the external forces/torques acting on the environment and the position, velocity and acceleration error of the end-effector. Mohajerpoor [58] presents an adaptive robust controller based on the position and the exerted force on the environment. They

break the vector of the external forces and moments exerted on the object into two parts: from the robot and from the environment. Tinos and Terra use a hybrid control of motion and the internal force (squeeze force). They include the use of passive joints, and relate the actuated and passive joint velocities. The Jacobian matrix is used to relate the velocities of the load and the velocity of the actuated joints [8].

Transporting or handling a single payload is a common task for cooperating manipulators [46, 52, 59, 60, 61]. Lui and Arimoto look at not only controlling the configuration of the payload, but also the internal forces [61]. Zhen and Luh assume an evenly distributed force among multiple industrial manipulators to ensure one manipulator does not experience large peak torques [52].

1.6 Optimization of Robotic Systems

Generally, there are multiple solutions for any given problem. This holds true in designing robots for space systems as well. Designers and engineers are left with a myriad of different variables they can tweak and change in order to optimize how the system to operates (i.e. soft requirements). For example, an industrial pick-and-place robot may have a task of picking of X types of parts and placing them at location B. The goal would be to do this repetitively and as fast as possible. When performing the task thousands of times a day, cutting the time down even by a fraction of a second can have a large impact. The design criteria may be the precision and accuracy of doing this task; however, it is also desirable to minimize

time. For a large manipulator in space, performing a maneuver very quickly is not necessarily a criteria that is considered; rather, power and mass are important factors[20].

In 1997, Chocron and Bidaud[62] showed that a genetic algorithm (GA) could be used to optimize robotic system parameters without using any design experience or database to start from. They demonstrated that optimization techniques could be applied to the robotic design process, and that experience and intuition were not the only ways to design a manipulator. However, in 2007, ten years after [62] published their results, Rout and Mittal state that, despite this point, many manipulator parameter tolerances are still selected by experience and intuition[63]. One of the reasons for this is that a manipulator can be optimized for a number of different criteria, e.g., minimizing mass [64][65], dexterity [66][67][68], task completion time [69][70], minimize end-effector velocity [71], maximizing the work-space [72], minimizing the dynamic stress [73], etc. Patel and Sobh survey a number of manipulator performance metrics and how they are applied [74].

The mass of the link lengths, the type of motor, the gearing on the motor, and the controller all affect the manipulator performance. Many authors focus on an integrated structure/control design of robotic systems. [75] focuses on reducing the settling time for a fast pick-and-place operation. [76] focuses on minimizing the higher harmonic portion of the torques that are required to perform a task/trajec-tory; allowing for high-speed maneuvers.

There are additional common parameters often modified to optimize a ma-nipulator, regardless of the goals. Link lengths, angle of the joints, number of

joints, etc., all contribute to the physical properties and robot kinematics of the manipulator. Adjusting any of these parameters can alter the performance of the manipulator. Many authors have used these parameters in order to optimize their designs [77][78][79].

Denavit-Hartenberg (DH) parameters are a common method used to describe the kinematic synthesis of a manipulator. These parameters include the number of joints, the types of joints, and the configuration, as well as the link lengths and offsets. It is a method of describing a manipulator using only four quantities per link. The DH parameters describe the frame transformation from one joint to another. Table 1.6 provides a brief description for each of these four parameters; for more information see table [80].

Park, Chang and Yang use the grid method optimization with a very fast simulated annealing to determine the kinematic properties to best reach a number of task points [81]. Sobh, Tarek, et al. optimize the DH parameters to optimize an average manipulability index [82]. Varalakshmi, Srinivas minimize the joint torques by optimizing the redundant prismatic joint locations (link lengths) for a planar parallel mechanism. The joint torques are calculated assuming a static force analysis without the use of the dynamics of the system [83].

Kumar et al uses a global condition to optimize the link lengths ratios given a series of task points [68]. Jarari et al. use a genetic algorithm to optimize the mass of the gearboxes and link lengths of a 3DOF serial manipulator in order to minimize manufacturing cost [84].

Dual arm and multi-arm optimization is not as common as single arm opti-

Name	Common Symbol	Description
Link length	a_i	Length between the origin of the frame placed $joint_i$ to $joint_{i+1}$ along the major X_{i+1} axis
Link twist	α_i	Rotation between the frame placed $joint_i$ to $joint_{i+1}$
Link offset	d	Length between the origin of the frame placed $joint_i$ to $joint_{i+1}$ along the minor axis (Z_{i+1})
Joint Angle	θ_i	Angle between links. For revolute joints this is variable

Table 1.1: Summary of DH-Parameters

mization. [85] focuses on optimizing the joint configuration of a dual arm system, creating a task-oriented dual-arm manipulability metric. Zhang et al develop a manipulability metric specific for dual-arm space manipulators: a metric which takes into account the coupled dynamics [86].

This research looks at the optimization of a multi-arm system in order to minimize the total system mass. The mass of the system is studied with the following changing variables: the number of arms, the link lengths of each arm, and the base locations.

Chapter 2: Kinematics and Dynamics

This chapter focuses on the kinematics and dynamics of a multi-arm system. The first section focuses on how the task description is used to calculate the desired kinematics and the resulting dynamics of the payload. Sections two and three pertain to the discussion of dynamics and kinematics of single- and multi-manipulator(s) respectively. Lastly, a brief discussion will be made on the internal force and how this affects the overall dynamics of the kinematics.

2.1 Task Description

The goal of this section is to translate a task definition into the dynamics of the payload; specifically, defining the equation of motion of the payload. While this section goes over how the model finds the trajectory of the payload, it is not the only method. Obstacle avoidance, which is not considered in this thesis, could also be used to generate more complex trajectories.

A task is defined as moving a payload of a given mass, $M_{payload}$, from point A to point B (i.e. the starting and end point). The task presented in this research assumes zero obstacles, and that the payload is moved in a straight line from the starting to the end point. If the task has a rotational difference from the start to

the end point, then the payload is rotated along the duration of the trajectory.

A trapezoidal trajectory is generated for the payload based on the maximum acceleration and velocity given in the task description. The maximum velocity of the trajectory is limited by the maximum velocity of the manipulator tip. Rather than define the maximum acceleration as a function of maximum force applied to the manipulator tip, it is based on how quickly the payload can safely go from zero velocity to maximum velocity. Generally, this is on the order of one second.

Trapezoidal Trajectory Example

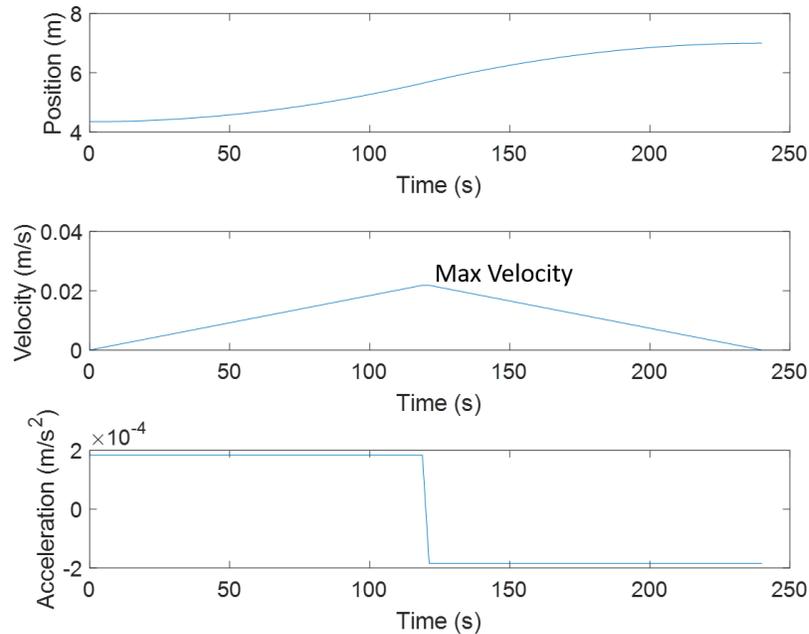


Figure 2.1: Example 1-dimensional trapezoidal trajectory

The shortest execution time for a given specified task is dependent on the maximum acceleration and velocity that can be achieved by the system. The maximum acceleration and velocity limits the minimum time in which a task can be completed.

This minimum time involves the payload accelerating and then decelerating at the maximum rate during the entirety of the trajectory. If this task time is less than the minimum time, the desired task time is ignored and the minimum time is used instead.

If the desired time is greater than the minimum time, than the manipulator will accelerate at the maximum rate, but will then proceed at a constant velocity in order to adjust the maneuver (the movement from point A to B) to the desired time.

Equation 2.1 describes the motion for the payload. x is the location of the payload, \dot{x} is the velocity of the payload, and \ddot{x} is the payload's acceleration. This provides the complete trajectory for the payload. Using the task defined velocities, accelerations, and time, the equation of motion for the payload can be calculated.

$$M_{payload}(x)\ddot{x} + cg(\dot{x}, x) + g(x) = F_{task} \quad (2.1)$$

$c(\dot{x}, x)$ is the Coriolis and $g(x)$ gravity terms on the payload. Since this thesis focuses on space manipulators moving large inertial masses in micro gravity, the gravity terms here are assumed to be zero.

F_{task} is the force required on the payload to perform the desired motion. This is the force that the manipulator(s) must apply in order to move the payload along its desired trajectory.

Equation 2.2 defines the task force as a function of the manipulators' grasping location on the payload and the force applied from the end-effector to the payload.

J_{EE} defines the force transformation from the end-effector to the payload's center of mass. F_{EE} is the amount of force applied by the manipulator at the end-effector.

$$F_{task} = -J_{EE}^T F_{EE} \quad (2.2)$$

One of the most challenging problems with using a multi-arm system is determining how to control each arm in order to move the object in order to accomplish the desired task.

2.2 Single Manipulator

This section briefly reviews the dynamics for a single manipulator as it performs a task. Inverse kinematics solve the joint angles, q_i , for a desired end-effector location which is based on the payload's location and grasping point, defined in the J_{EE} . The joint velocities \dot{q}_i and the accelerations \ddot{q}_i can be calculated as shown below.

There are multiple solutions to the inverse kinematic problems, even for a non-redundant manipulator. In order to ensure a smooth motion, the joint angles, which minimize the distance from the previous time step, are used. This provides a smooth manipulator motion from start to finish. The joint angles, q , for a N degree of freedom arm can be found in equation 2.3.

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \dots \\ q_N \end{bmatrix} \quad (2.3)$$

Let v denote the end-effector's linear and angular velocity. Assuming that the payload is rigid and the arm is firmly attached, this is defined by the payload's motion. The end-effector's velocities can be related to the joint velocities through the use of the Jacobian matrix, $J(q)$.

$$v = J(q)\dot{q} \quad (2.4)$$

The end-effector accelerations can be found by differentiating equation 2.4 with respect to time.

$$a = \dot{J}(q, \dot{q})\dot{q} + J(q)\ddot{q} \quad (2.5)$$

The equation of motion for a single manipulator is given in equation 2.6. The Coriolis and gravity forces are written in terms of the joint angles and joint velocities, q and \dot{q} respectively. The joint torques are represented by T . The Jacobian, J , is written as a function of the joint angles.

$$M(q)\ddot{q} + cg(\dot{q}, q) = T + J(q)^T F_{tip} \quad (2.6)$$

The equation of motion for a single manipulator can be combined with the equation of motion for that of the payload, see equation 2.7. This expresses the

required end-effector force in terms of the desired task force, or the amount of force required to move a payload in a desired trajectory.

$$M(q)\ddot{q} + cg(\dot{q}, q) = T - J(q)^T J_{EE}^{-T} F_{Task} \quad (2.7)$$

Equation 2.7 contains a term, $J(q)^T J_E^{-T}$, which is based on the geometry of the arm. This contains the Jacobian term for the manipulator, which is based on the joint angles and link lengths. J_{EE} is the force transformation from the grasping location to the payload's center of mass. This term is also based on the geometry of the situation. Carignan and Akin combined these terms into a single value, see 2.8

$$D = J_{EE}^{-1} J \quad (2.8)$$

Substituting this value for D yields:

$$M(q)\ddot{q} + cg(\dot{q}, q) = T - D^T F_{Task} \quad (2.9)$$

The above analysis describes the motion of the payload in terms of a single manipulator. A coordinated cooperating system consists of multiple manipulators handling a single object at the same time.

2.3 Multiple Arms

The equations of motion for fixed based arms rigidly attached to a payload or environment can be found below. The joint torques are a function of the joint angles, q_i , and are defined as $\tau_i^T = [\tau_{i1}, \tau_{i2}, \dots, \tau_{iN}]$ for a manipulator with N joints.

$$M_i(q_i)\ddot{q}_i + c_i(c_i, \dot{c}_i) + g_i(q_i) = \tau_i + J_i(q_i)^T f_i \quad (2.10)$$

M_i is the symmetric positive-definite inertia matrix for each manipulator. The gravity vector is represented by g_i and the Coriolis and centripetal force vectors are combined into c_i . The force exerted on the payload E by an arm is $f_i^T = [f_{ix}, f_{iy}, f_{iz}, N_{ix}, N_{iy}, N_{iz}]$. The first three terms are the forces in the x,y, and z directions at the grasp point, while the other terms represent torque.

The dynamics for K manipulators can be written in extended joint space:

$$M(q)\ddot{q} + c(q, \dot{q}) + g(q) = \tau + J(q)^T F \quad (2.11)$$

where

$$M = \text{diag}(M_1, M_2, \dots, M_k) \quad (2.12)$$

$$F = \begin{bmatrix} f_1 \\ \cdot \\ \cdot \\ \cdot \\ f_k \end{bmatrix} \quad \tau = \begin{bmatrix} \tau_1 \\ \cdot \\ \cdot \\ \cdot \\ \tau_k \end{bmatrix} \quad g = \begin{bmatrix} g_1 \\ \cdot \\ \cdot \\ \cdot \\ g_k \end{bmatrix} \quad c = \begin{bmatrix} c_1 \\ \cdot \\ \cdot \\ \cdot \\ c_k \end{bmatrix} \quad (2.13)$$

The equation of the payload's center of mass is written below as a function of the multiple manipulator, where W represents an $M \times M$ matrix defined by the grasping geometry.

$$M_e(X_e)\ddot{X}_e + c_e(X_e, \dot{X}_e) + g_e(X_e) = WF \quad (2.14)$$

When the payload is rigidly grasped, with no possibility of sliding, then the matrix W is constant[87]. The W_c matrix also contains the load sharing between the K different manipulators.

$$W_c = \begin{bmatrix} w_1 & w_2 & \dots & w_k \end{bmatrix} \quad (2.15)$$

2.4 Internal Grasp Forces

In general, the total number of forces and torques exerted by the manipulators are greater than the work-space. This results in an over constrained object. The internal grasp forces are the statically indeterminate components which are used when the manipulators have enough joints to control the components of the contact forces.

$$\text{Total Force on Payload} = H_o = W_c C$$

$$C = C_{internal} + C_{motion} \quad (2.16)$$

$$C_{motion} = W_c^{-+} H_o$$

The force vector acting on the payload F , an $n \times 1$ vector, can be divided into two components: the internal grasping force, which is in the null-space of the W_c matrix, and the forces which contribute to the motion of the payload. Here, C contains the contact forces, and W_c is the matrix relating the contact forces to the forces acting on the payload.

W_c^{-+} is the Moore-Penrose pseudo inverse of the grasping matrix, W_c . If W_c is an invertible square matrix, then the internal grasping force is zero, as the null-

space is empty. This is only true when the number of degrees of freedom of the manipulator match that of the work-space. For example, a 6DOF manipulator can exert forces and torques along x, y, z . A single 6DOF manipulator results in a square invertible W_c matrix.

If the manipulators rely entirely on friction to maintain a stable grasp, such as fingers, the friction at the contact points must be controlled. This is done through the W_c matrix. The internal grasp force should be also considered, if there is a limit on the forces that can be applied to the payload (i.e., a fragile payload).

$$\mathit{minAllowableForce} \leq C_{\mathit{internal}} \leq \mathit{maxAllowableForce} \quad (2.17)$$

While the internal grasping forces do not contribute to the motion of the payload, they do affect the joint torques. In the simplified static case, the joint torques are related directly to the end-effector forces. The internal grasping force is in the null space of the grasping matrix.

$$\begin{aligned} \tau &= J^T F_{\mathit{AppliedByArms}} \\ F_{\mathit{AppliedByArms}} &= C_{\mathit{internal}} + C_{\mathit{motion}} \\ \tau &= J^T (C_{\mathit{internal}} + C_{\mathit{motion}}) \end{aligned} \quad (2.18)$$

Constraints are assigned to ensure the internal grasping forces are within specified limits.

Depending on the number of constraints, there may be many different solutions and the system should be optimized. [50] approximated the frictional constraints

using linear constraints, and utilized a linear programming method to compute the optimal internal forces. Other researchers choose to minimize the energy consumption [88, 89].

Internal forces are important when the object is being held with friction, such as fingers pinching or grasping a payload [90]. When the manipulators are rigidly attached, there is no minimum internal force required. Ideally, all the force from the manipulator will go into the motion of the payload.

Using the assumption that there are no restrictions on the internal motion, the grasping matrix can be simplified. The case of rigidly attached manipulators allows the extended Jacobian to be separated into individual grasping matrices. These matrices are square and invertible, and therefore have an empty null-space.

$$M_e(X_e)\ddot{X}_e + c_e(X_e, \dot{X}_e) + g_e(X_e) = F_{task} - \sum J_{ei}(X_e)^T F_i \quad (2.19)$$

2.4.1 Dual Arm Planar Example

This section describes a dual arm planar example moving a payload. The payload is not exerting any force on the environment, therefore $F_{task} = 0$. The only force the manipulators must exert is to ensure the payload moves as required by the task. For this example, the payload will move along the X direction from point X_1 to X_2 . The two manipulators are designated as arms ‘a’ and ‘b’.

2DOF Planar Example

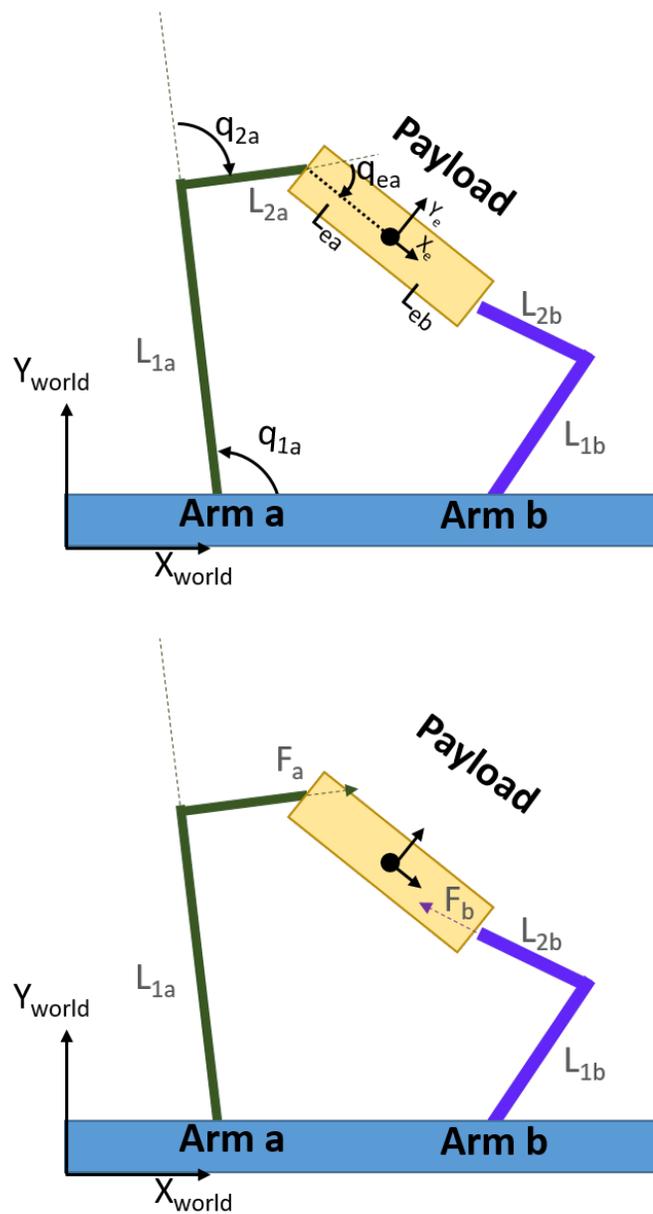


Figure 2.2: Diagram of arms 'a' and 'b' attached to a payload. (Top) Shows the joint angles between the manipulators (Bottom) Show the forces acting on the payload from the two manipulators

$$M_e(X_e)\ddot{X}_e + c_e(X_e, \dot{X}_e) + g_e(X_e) = -J_{ea}(X_e)^T F_a - J_{eb}(X_e)^T F_b \quad (2.20)$$

J_{ea} and J_{eb} are the force transformation matrices for arms ‘a’ and ‘b’ respectively, where L_{ea} and L_{eb} represent the length from the center of the payload to the grasping points for each arm. θ_{ea} is the angle of the payload relative to the end of arm ‘a’. s_{ea} denotes the $\sin(\theta_{ea})$ and c_{ea} denotes the $\cos(\theta_{ea})$. θ_{eb} is the angle of the payload relative to the end of arm ‘b’. The forces acting on the center of the payload from each arm is found using the force transformation matrices, J_{ea} and J_{eb}

$$J_{ea} = \begin{bmatrix} 1 & 0 & L_{ea}s_{ea} \\ 0 & 1 & -L_{ea}c_{ea} \\ 0 & 0 & 1 \end{bmatrix} \quad J_{eb} = \begin{bmatrix} 1 & 0 & -L_{eb}s_{eb} \\ 0 & 1 & L_{eb}c_{eb} \\ 0 & 0 & 1 \end{bmatrix} \quad (2.21)$$

The equation for the ‘a’ and ‘b’ manipulators are given in Equation 2.22

$$M_a(q_a)\ddot{q}_a + c_a(q_a, \dot{q}_a) + g_a(q_a) = \tau_a + J_a(q_a)^T F_a \quad (2.22)$$

$$M_b(q_b)\ddot{q}_b + c_b(q_b, \dot{q}_b) + g_b(q_b) = \tau_b + J_b(q_b)^T F_b$$

The Jacobian for arm ‘a’, a 2 link planar manipulator, is given in Equation 2.23. This transforms the external forces applied at the wrist to joint torques. S_i is the sign of joint angle q_i and C_i is the cosine of joint angle q_i .

$$J_a = \begin{bmatrix} -L_{1a}S_{1a} - L_{2a}S_{12a} & -L_{2a}S_{12a} \\ L_{1a}C_{1a} - L_{2a}C_{12a} & L_{2a}C_{12a} \end{bmatrix} \quad (2.23)$$

This example provides the equation of motion for a planar two degree of freedom dual-arm system. The motion of the payload and joint angles are a function of the task definition. It is clear to see that if the payload is not moving (static) and not exerting any force onto the environment, the force contributed by the two manipulators would either need to cancel each other out or be zero.

Static 2DOF Example

$$\begin{aligned}
 M_e(X_e)\ddot{X}_e + c_e(X_e, \dot{X}_e) + g_e(X_e) &= 0 \\
 0 &= -J_{ea}(X_e)^T F_a - J_{eb}(X_e)^T F_b
 \end{aligned}
 \tag{2.24}$$

The arms produce equal but opposite forces onto the static payload. One solution would be if both arms produced zero force; the payload would remain static (zero external forces) and there would be no internal forces as well. However, it is clear from Equation 2.25 that the forces from each arm are not required to be zero, just that they are equal but opposite between the dual manipulators.

$$-J_{ea}(X_e)^T F_a = J_{eb}(X_e)^T F_b
 \tag{2.25}$$

The static equation for each manipulator reduces to Equation 2.26.

$$\begin{aligned}
 \tau_a &= -J_a(q_a)^T F_a \\
 \tau_b &= -J_b(q_b)^T F_b
 \end{aligned}
 \tag{2.26}$$

2.5 Conclusion

This chapter provides an overview of the dynamics for single-and multi-arm systems. The equation of motions for a serial joint manipulator and a payload are given. These equations of motion are used together to determine the forces/torques necessary to accomplish the given task.

A multi-arm system can provide both internal and external forces. External forces contribute to the motion of the payload, whereas an internal force does not. However, the internal forces are reacted through the payload, and the payload must be strong enough to withstand these forces. Finger-type end-effectors, which rely on friction to maintain a grasp, must maintain a minimum internal force in order to maintain control of the payload. Space manipulators are generally rigidly attached to the payload through the end-effector, and do not have a minimum required internal force. The exact distribution of forces between the multiple manipulators is the focus of Chapter 5.

Chapter 3: Mass Estimation

Mass is a critical driving design requirement in getting space systems into orbit. Reducing the mass of the system can significantly reduce launch costs and increase launch opportunities.

This chapter provides methodology and equations for estimating the mass of a manipulator for a given task, and is broken into two different sections: joints and links. Each manipulator is made with various combinations of these two essential parts. Joints are any actuators, motors, and gearboxes that produce movement, and links provide the structural support of the manipulator.

3.1 Estimating Motor Mass

Sizing motors can be difficult, as there are many different parameters given to describe a motor. Throughout the last decade, several researchers have sought to provide a methodology to address this. Dermizanski focuses on sizing motors based on the peak torque, [91] while others, like Wilcox, use power to find the stall torque, which is related to the motor constant K_m [92]. Hollerbach et al. found that the motor constant has a linear relationship to the mass, also noting a dependence on the diameter[93].

It is highly desirable to obtain an estimation for a motor's mass based on a given performance metric. The datasheet for motors does not always provide the same information, although there are some parameters that are consistently provided, including peak torque, or the maximum amount of torque a motor can provide for a short time, and the continuous peak torque, the maximum amount of torque a motor can provide continuously.

Dermitzaki et al. looks at scaling motors based on peak (stall) torque. They compiled a survey of over 400 motors (see Figure 3.1), theorizing that the typical motor had a linear relationship between the mass of the motor and the peak torque. Most of the motors that were studied consisted of typical consumer type motors, such as low torque servo drives with a plastic gearbox. They also looked into one manufacturer of high-end motors: Maxon. The high-end motors did not fit with the linear trend. It was noted that unlike the lower cost motors, they did not have a plastic gearbox; in fact, the Maxon motors did not include any gearing. For these reasons, it is unsurprising that the high-end motors do not fit the same linear model, as they produce less torque per weight than the consumer grade motors with gearboxes.

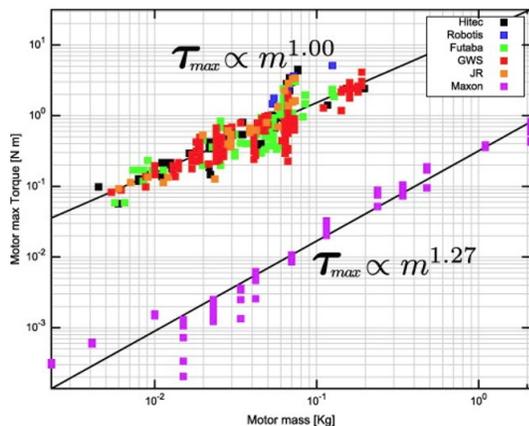


Figure 3.1: Replication of Mass/Torque plot by Dermizanki et al. [91]

Dermitzaki’s survey demonstrates that not all motor heuristics are created equal. This survey shows that the heuristics for high-end motors without gearboxes do not have the same trend as consumer grade motors with gearboxes, and that using the trend found for one type of motor does not mean that it will hold true for a second type.

While Dermitzaki uses consumer motors, this research focuses on purely high end DC brushless motors typically found in high-end robotics. This study does not include motors with pre-assembled gearing; gearbox sizing is considered in a later section. A database was compiled of over 100 motors from seven different motor manufacturers [94, 95, 96, 97, 98, 99, 100].

3.1.1 Continuous Stall Torque

Motor datasheets list the motors maximum torque as the peak torque. Generally, this peak torque is reached when first starting, or starting from ‘cold’. As the

motor continues to run, it will produce heat and warm up. They can only maintain the peak torque for a short time before they have to reduce their output for thermal reasons. Therefore, motors cannot continuously produce their peak torque. As a result, using this dimension as a means of selecting a motor is not the best choice unless the exact motion profile (trajectory) is known. The designer would need to know how long a motor could produce a specific torque for. If the time is on the order of a few seconds, using peak torque to size a motor is likely fine. However, if the motor must run at a desired torque for a longer amount of time (minutes to hours), then this is a poor metric.

The continuous stall torque (another common motor parameter) is the amount of torque that a motor can continuously output at stall (i.e. 0 RPM). Using this for motor selection and estimations puts fewer requirements/assumptions on the system's analysis; the designer does not need to know precisely how long the torque must be maintained. Figure 3.1.1 shows the difference in relationship between the *continuous* stall torque and the stall torque as a function of mass. It should be noted that this is for a limited dataset to better compare with Dermitzaki's dataset. Fig 3.1.1 shows the full dataset collected for Maxon BLDC motors. The relationship between the motor and continuous stall torque is closer to a linear relationship than Dermitzaki found.

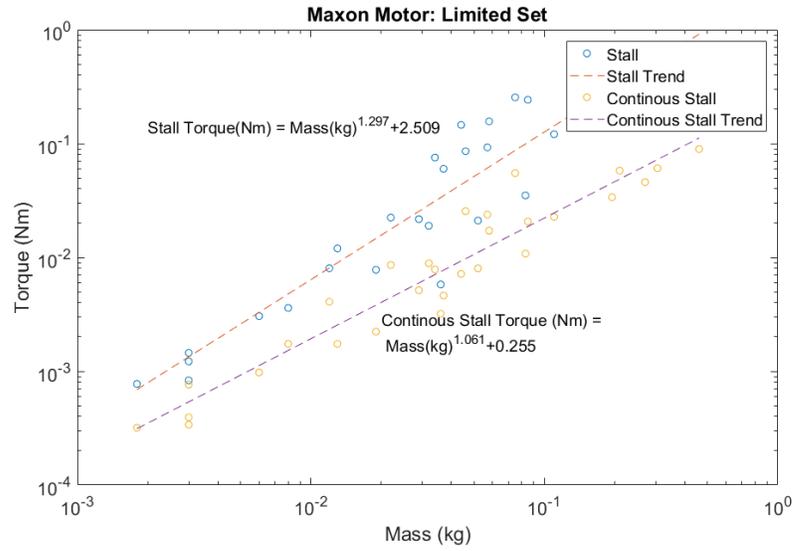


Figure 3.2: Max Stall Torque and Continuous Stall Torque vs Mass; limited dataset of Maxon BLDC motors to compare with Dermitzaki's results

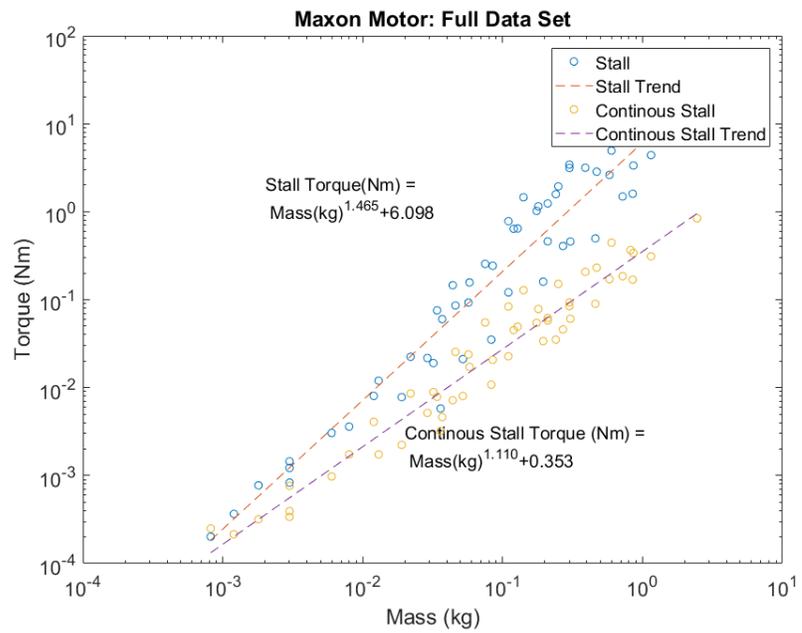


Figure 3.3: Max Stall Torque and Continuous Stall Torque vs Mass; full dataset of Maxon BLDC motors

3.1.2 Closer look at Motors by a Single Manufacturer

Before comparing motors from multiple manufacturers, it is important to first consider how motors vary across a single manufacturer. Normally, it is assumed that motors made by different manufacturers have more deviation than those created by the same manufacturer. This is true, to a certain extent. Motor manufacturers generally make several different families of motors. For example, Maxon has a five different BLDC motor families: the EC, EC-flat, EC-i, EC-max, and the 4pole motor. Each one of these families have a different design focus, with the overall goal of still maintaining high performance. (see Table 3.1)

Maxon Motor Family Description	
Motor Family	Design Focus
EC	Standard BLDC motor family
EC-Flat	Maintain a low profile
EC-i	Producing high torque
EC-max	Emphasize on modular design
4 pole	Focuses on delivering high performance per volume and weight

Table 3.1: Maxon Motor Family Descriptions

Figure 3.4 show the relationship between the continuous stall torque and the mass of framed BLDC motors for various series by Maxon. Looking at the individual

families, there is a strong linear relationship due to the fact that the manufacturing and design processes within a family are similar.

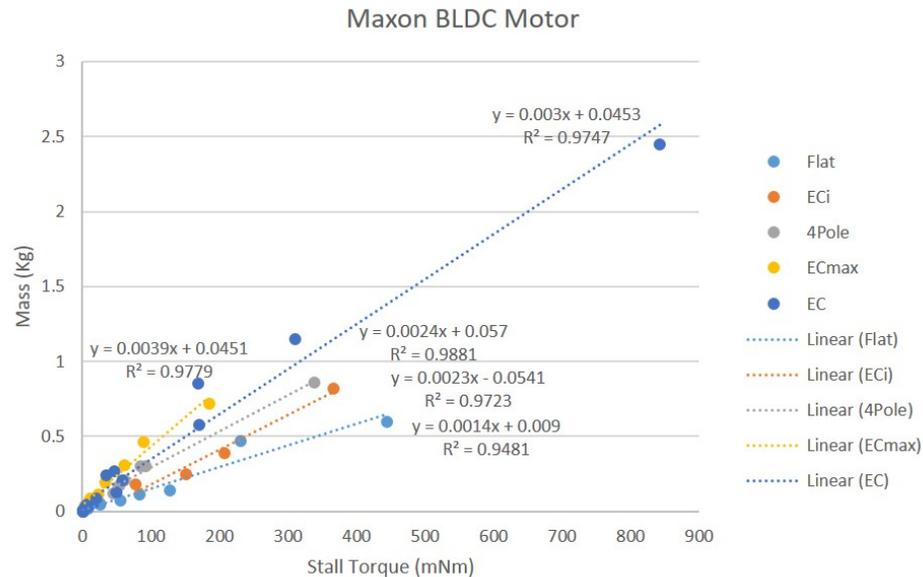


Figure 3.4: Maxon Motor per torque

Figure 3.5 shows the probability residual plot for each family. Each graph show very small tails and that most of the points lie along a line. This indicates that the data has a normal error distribution, and that using standard regression techniques to minimize the residual error are valid. Again, this is the type of relationship one would expect within a motor family or manufacturer.

However, taking a step back and looking at Maxon as a whole, it is clear that the same linear relationship does not hold true for every family for this single manufacturer. The probability-residual graph, looking at all Maxon motors, shows that the tails are becoming longer, see Figure 3.5(f). This indicates that the residuals (and hence the errors) are not normally distributed. Thus, using a least squares

regression model may not be the best approach to take, as it minimizes the square sum of the residuals. A robust regression scheme, such as weighted least squares, can be useful when working with a non-normal error distribution.

Statistical Analysis for Maxon Motor Families

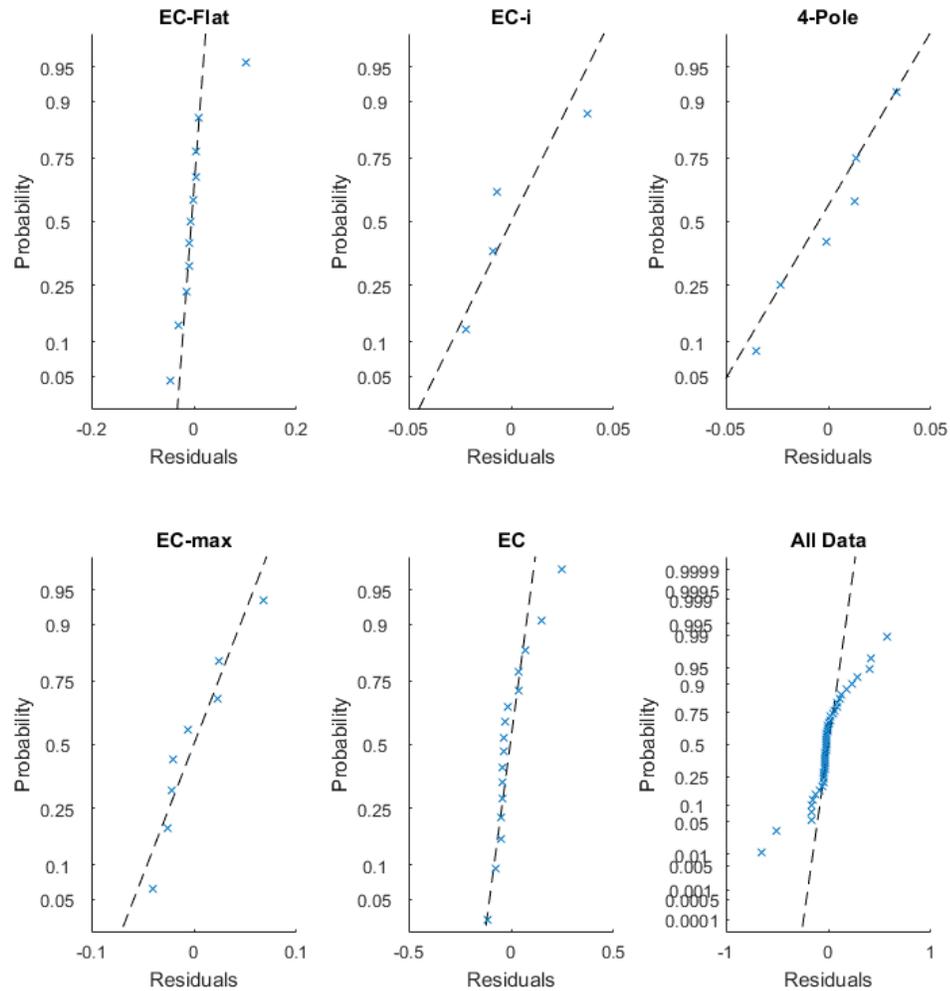


Figure 3.5: Probability-Residual plot for each Maxon BLDC Family

As the number of different families and manufacturers included in the database increase, the strong linear relationship between continuous stall torque and mass

becomes weaker. As more manufacturers are added with more motor families, the residuals will grow. Moreover, motors are rarely only operated at stall. The next step is to look at how the output torque varies as a function of the motor speed across multiple manufacturers.

3.1.3 Torque vs Speed

Motor torque output depends on the speed of the motor. A typical speed/torque curve is shown below in Figure 3.1.3. This is the amount of torque which can be continuously produced at the various motor speeds. Using data from the speed/torque curves for each of the motors, a relationship is found between the continuous torque and the desired speed.

Continuous Duty Capability for 130°C Rise — RBE - 00410 Serie

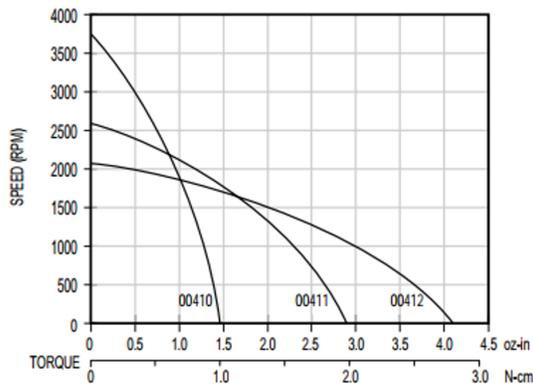


Figure 3.6: Replication of a speed/torque curve for a high performance motor [94]

The speed/torque curve data was collected for each of the 100+ motors across seven different manufacturers: Anaheim, Gems, Kollmorgen, Mag, Maxon, Moog, and Woodward. Each of these curves were digitized, allowing the torque at any

motor speed to be interpolated. This allows a relationship to be found between mass, speed, and torque. Figures 3.7 and 3.8 show that the continuous torque to mass relationship changes from 0 RPM to 5000 RPM. Both a linear fit and power fit describe the relationship; the change in coefficients between the two speeds demonstrates that the motor speed should not be neglected.

0 RPM (Stall) Mass/Torque Fit

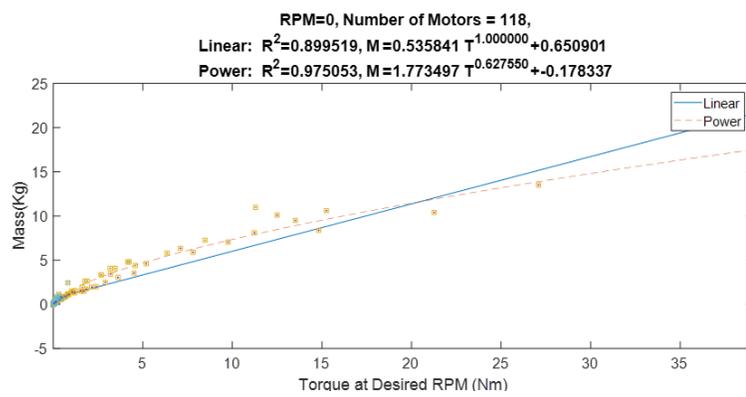


Figure 3.7: Example of a speed/torque curve for a high performance motor at 0 RPM

5,000 RPM Mass/Torque Fit

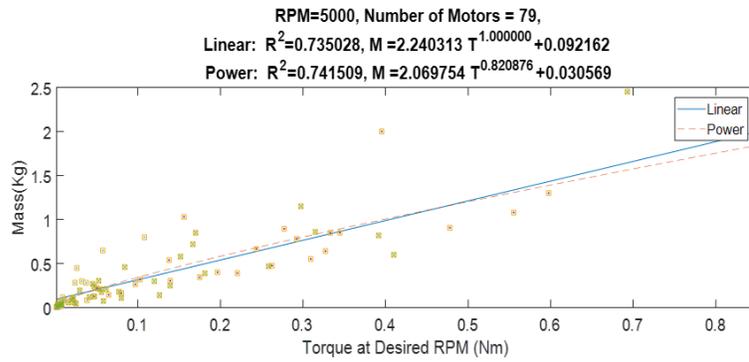


Figure 3.8: Example of a speed/torque curve for a high performance motor at 5000RPM

Figure 3.1.3 shows the motor mass plotted as a function of the motor speed (kRPM) and the continuous output torque (Nm). The continuous torque was calculated from the speed/torque curve for each motor at regular steps from 0 RPM to 20 kRPM.

Motor Mass, Continuous Torque, Motor Speed for all Manufacturers

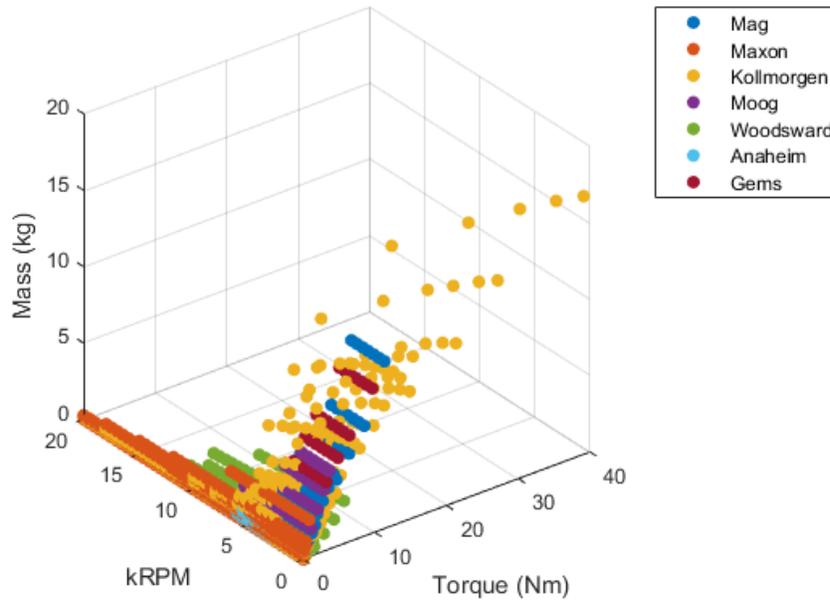


Figure 3.9: Raw data for Motor Mass, Continuous Torque, Motor Speed

Interestingly to note, the raw data is not normally distributed. The mass of each motor remains constant at each of the different points on the curve. This is the reason for the 'horizontal bands' of data: each one represents a single motor along the speed/torque curve. Knowing this, and taking an in-depth look at Maxon's motors, the following study does not consist of data with a normal distribution of errors.

Regression Analysis for All Motor Data

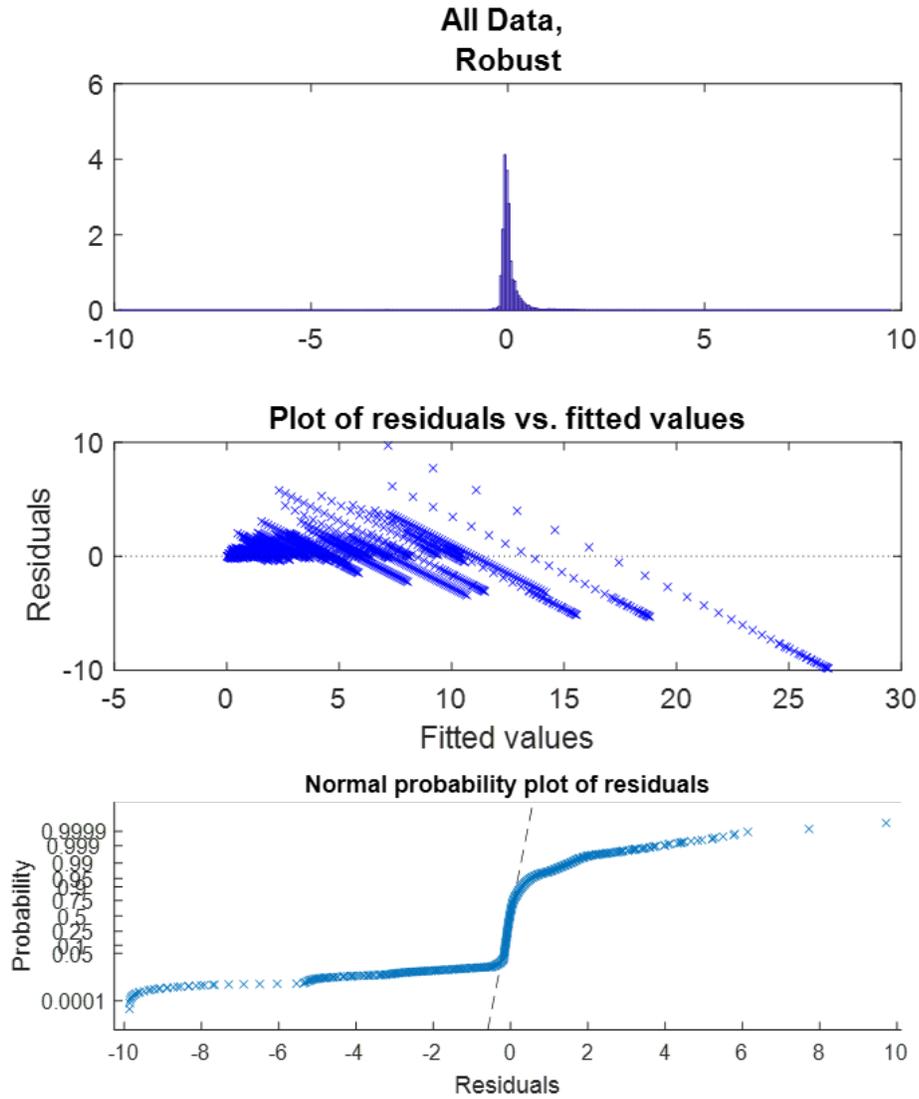


Figure 3.10: Regression analysis for all motor data

Figure 3.1.3(a) shows the results of the statistical analyses performed on the raw data, assuming a least-square linear model. From the probability-residual plot, the data shows long tails. From looking closely at Maxon motors, motors across

different families indicated a non-normal error. Different motor manufacturers have different design criteria and different manufacturer processes; it is expected that motors across different manufacturers would continue to show a non-normal error. The long tails on the probability-residual plot indicates that the data has a non-normal error variance.

The residual vs. fitted plot value shows a “megaphone”-type pattern made up of bands of data, see Figure 3.1.3(b). If the error variance was constant across the dataset, then the residual vs fitted graph would show randomly distributed points. The bands are a result of the nature of the data: the motor mass remains constant for each motor as the torque and speed change. The distinct shape indicates that the error variance is not constant.

These two graphs show the data to be heteroscedastic. As a result of this non-constant variance, the method of regression must be chosen carefully. The least squares methods minimizes the sum of the error and works best with homoscedastic data. A robust method, one that would minimize the absolute difference of the residuals, will be less susceptible to outliers and provide a better fit.

3.1.4 Estimating the Motor Mass as a Function of Torque and Speed

Using the speed/torque curves for over a hundred brushless DC high-end motors, a relationship can be found to estimate the motor mass as a function of the desired continuous torque at a desired motor speed. Both linear and power fits were utilized and compared as found in Equation 3.1.

$$MotorMass = c_1 + c_2Torque + c_3Speed \tag{3.1}$$

$$MotorMass = c_1 + c_2(Torque)^{c_4} + c_3Speed$$

The coefficient c_1 is the intersection value and can be thought of as a penalty for having a motor. The torque coefficient, c_2 , is the weighting of the continuous torque, and c_3 is the weighting for the motor speed. The power fit has an additional term: the exponent on the torque, c_4 .

The solved coefficients can be found in Tables 3.2 and 3.3. These were found using non-linear regression techniques. The results contain both robust and non-robust techniques to demonstrate that, even with the non-constant error variance and non-normal distribution, the least squares method still produces usable results with a reasonable goodness of fit (GOF). The non-normal distribution and non-constant error covariance does not mean that a least squares method cannot be used, but rather that the results are not guaranteed to be unbiased.

Ideally, c_1 should be positive or zero, as it is the intersection point. For the linear case, this coefficient is positive. Even when a motor is not being used, the linear relationship estimates a mass for the motor. The power fit calculates c_1 to be less than zero. A user must be careful to ensure this equation does not result in a negative motor mass, which would be the case of the motor not being used (produces zero torque at zero RPM).

Non-Robust Linear and Power Fit

Method	Goodness of Fit	c_1	c_2	c_3	c_4
Linear	0.8584	0.56996	0.7247	-0.0324	1
Power	0.9147	-0.0629	1.6355	0.0036	0.7016

Table 3.2: Coefficients for Linear and Power fit (Non-Robust)

Robust Linear and Power Fit

Method	Goodness of Fit	c_1	c_2	c_3	c_4
Linear	0.9969	0.1829	0.9559	-0.0062	1
Power	0.9981	-0.0132	1.5305	0.0017	0.7414

Table 3.3: Coefficients for Linear and Power fit (Robust)

Even though the linear fit and power fit provide a good GOF, it should be noted that there are a larger number of smaller motors with higher maximum speeds than the larger motors. Therefore, if the data is looked at as a whole, there is a bias towards small motors, which will lead to greater error when estimating the mass for larger motors. To look at this in more detail, the data was broken up into two different types: small motors and large. A small motor is defined as one producing less than 2 Nm of torque. This piece-wise function should allow a better motor mass estimate with less bias towards small motors.

Non-Robust Piecewise Linear and Power Fit

Method	Goodness of Fit	c_1	c_2	c_3	c_4
Linear Small Motors	0.7277	0.1691	1.1666	-0.0030	1
Linear Large Motors	0.7496	2.7778	0.5325	-0.2503	1
Power Small Motors	0.7501	-0.0174	1.3940	0.0021	0.6476
Power Large Motors	0.8136	-6.3557	6.9533	-0.1691	0.3431

Table 3.4: Coefficients for Piecewise Non-Robust Linear and Power fit

Robust Piecewise Linear and Power Fit

Method	Goodness of Fit	c_1	c_2	c_3	c_4
Linear Small Motors	0.9902	0.0676	1.3788	0.0010	1
Linear Large Motors	0.9961	2.5067	0.6071	-0.2796	1
Power Small Motors	0.9938	-0.0179	1.3741	0.0016	0.6802
Power Large Motors	0.9816	-5.3463	6.4673	-0.2323	0.3484

Table 3.5: Coefficients for Piecewise Robust Linear and Power fit

The GOF values, presented in both Tables 3.4 and 3.5, show that the GOF actually decreases when separated into large and small motor datasets. However, if we take a closer look at the residual, it is apparent that the maximum error residual is greatly reduced when the sets are broken apart. This is especially true of the larger motors.

Figures 3.11, 3.12, 3.13 show the entire dataset, the small motor dataset, and the large motor dataset, respectively, with the robust linear fit residuals.

Regression Analysis for Full Dataset

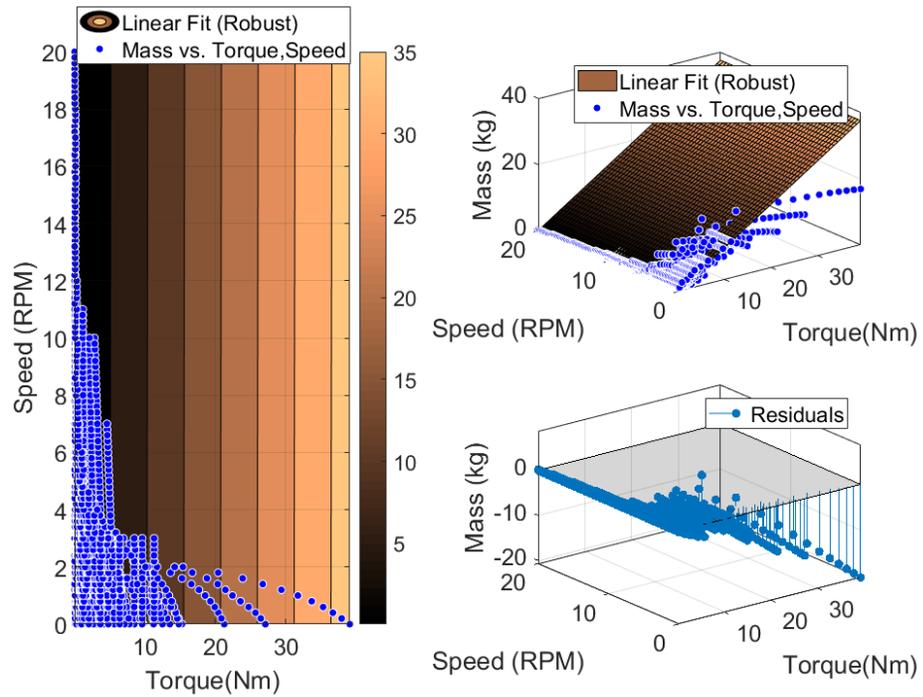


Figure 3.11: Regression analysis of full dataset

Regression Analysis for Small Motors ($T \leq 2\text{Nm}$) Dataset

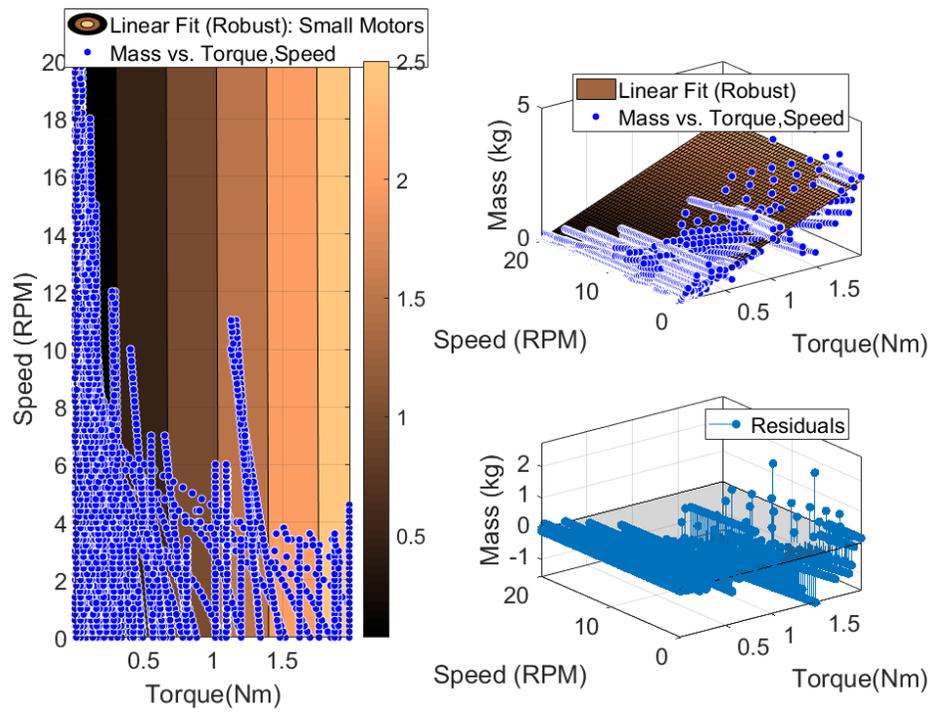


Figure 3.12: Regression analysis of small motor dataset

Regression Analysis for Large Motor ($T > 2\text{Nm}$) Dataset

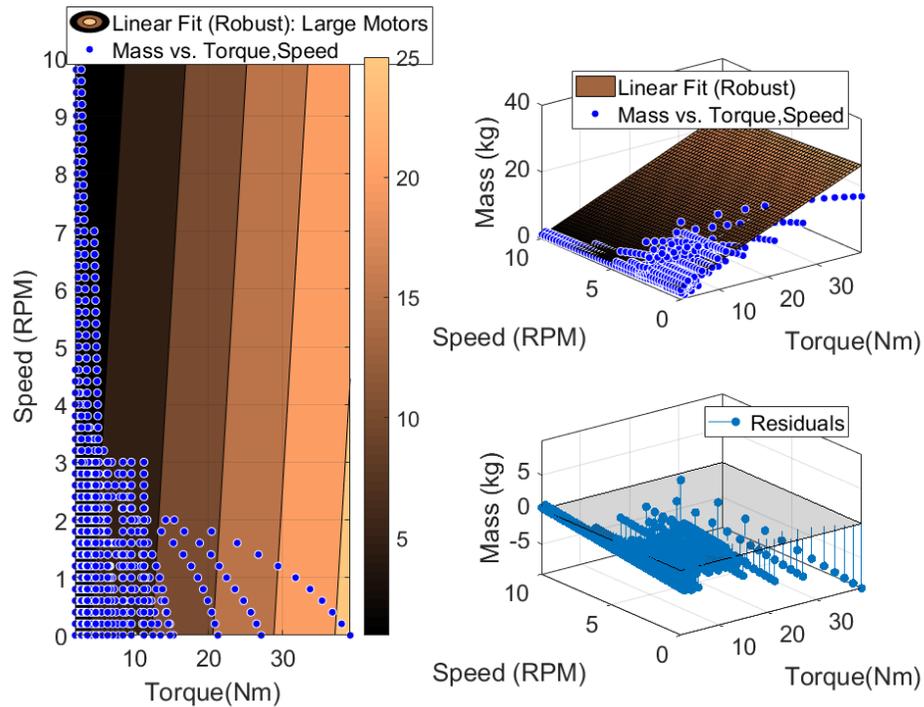


Figure 3.13: Regression analysis of large motor dataset

It is clear that breaking up the dataset into two different curves reduces the error residuals. There are a few design differences between small motors and large motors. Smaller motors have a lower inertia, and are designed to operate at higher motor speeds than larger motors. Larger motors often have a higher maximum power and more poles. Figure 3.13 has the maximum residual error of -10kg for the large motor dataset, whereas Figure 3.11 shows the maximum residual error of approximately -20kg . While the GOF values for the piece-wise datasets are lower than the entire dataset, the values themselves are still quite high and indicate a good model.

3.1.5 Sensitivity Analysis

Earlier, it was shown that adding more motors can have a large effect on the resulting analyses (see Figures 3.1.1 and 3.1.1). A sensitivity model determines how a relationship is altered by a change in data. In doing this, a new model was generated with the same process described above, but with half the motors in each dataset.

The sensitivity analysis shows the change in coefficients as the dataset changes. Ideally, adding (or subtracting) a few motors would not drastically change the results. A large change in the estimated coefficients suggests that the analysis is very sensitive and should be used with caution.

Coefficients for Linear and Power fit (Robust) for small motors (torque $\leq 2\text{Nm}$) and large motors (torque $> 2\text{Nm}$) with reduced dataset

Method	Goodness of Fit	c_1	c_2	c_3	c_4
Linear Fit All Motors	0.9987	0.1806	0.9711	-0.0056	1
Linear Small Motors	0.9966	0.0712	1.3551	0.0009	1
Linear Large Motors	0.9865	2.4345	0.5864	-0.2220	1
Power All Motors	0.9990	-0.0181	1.5650	0.0024	0.7530
Power Small Motors	0.9954	-0.0158	1.3945	0.0020	0.7028
Power Large Motors	0.9833	-4.5449	5.7450	-0.1927	0.3593

Table 3.6: Coefficients for Linear and Power fit (Robust) for reduced datasets

Table 3.6 shows the calculated coefficients using only 50% of the motor data. This simulation was completed for ten different reduced datasets, each randomly generated from 50% the datasets.

The coefficients from the sensitivity analysis are compared to those found with the full dataset in Tables 3.5 and 3.3. To better show how the models change with the reduced data, the following figures show the change in the coefficient values for each dataset over multiple runs.

Reduced Dataset Coefficients over 10 runs

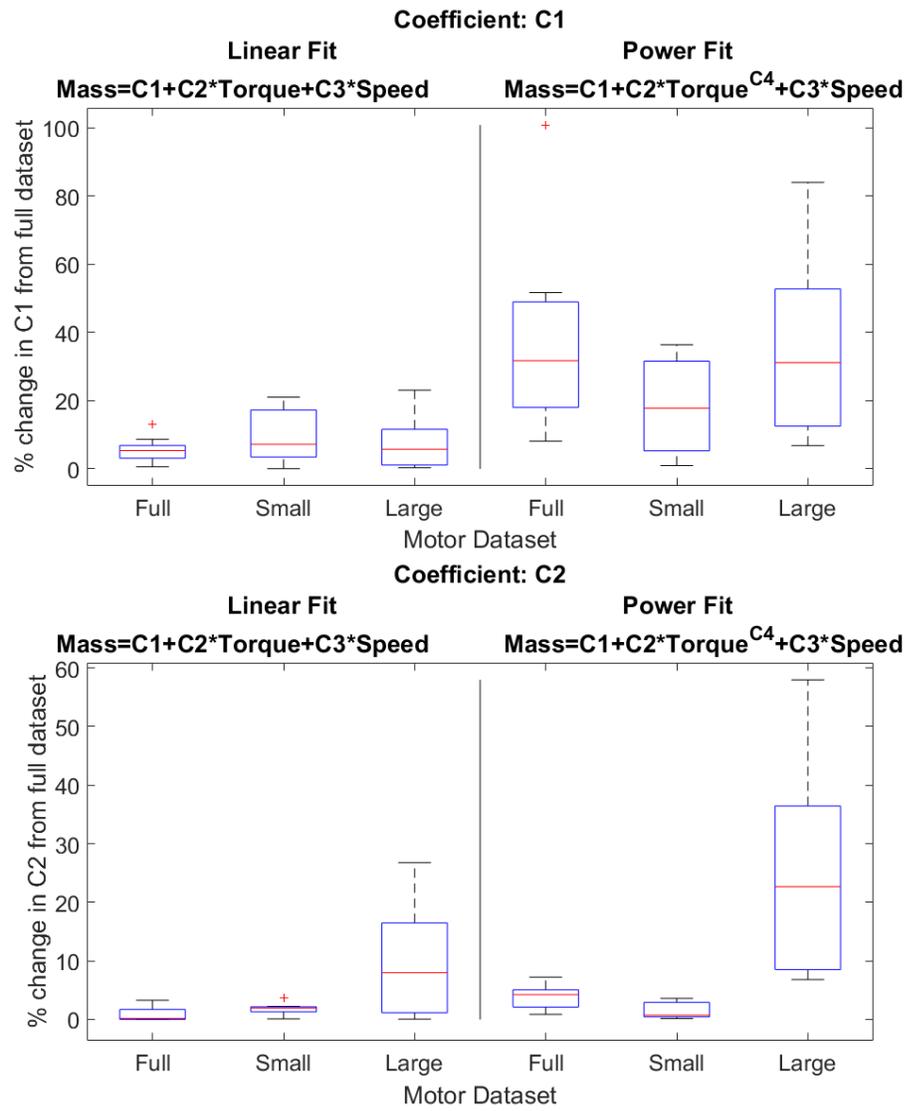


Figure 3.14: Sensitivity Analysis for Linear and Power Fit for Intercept Point (c_1) and Torque coefficient (c_2)

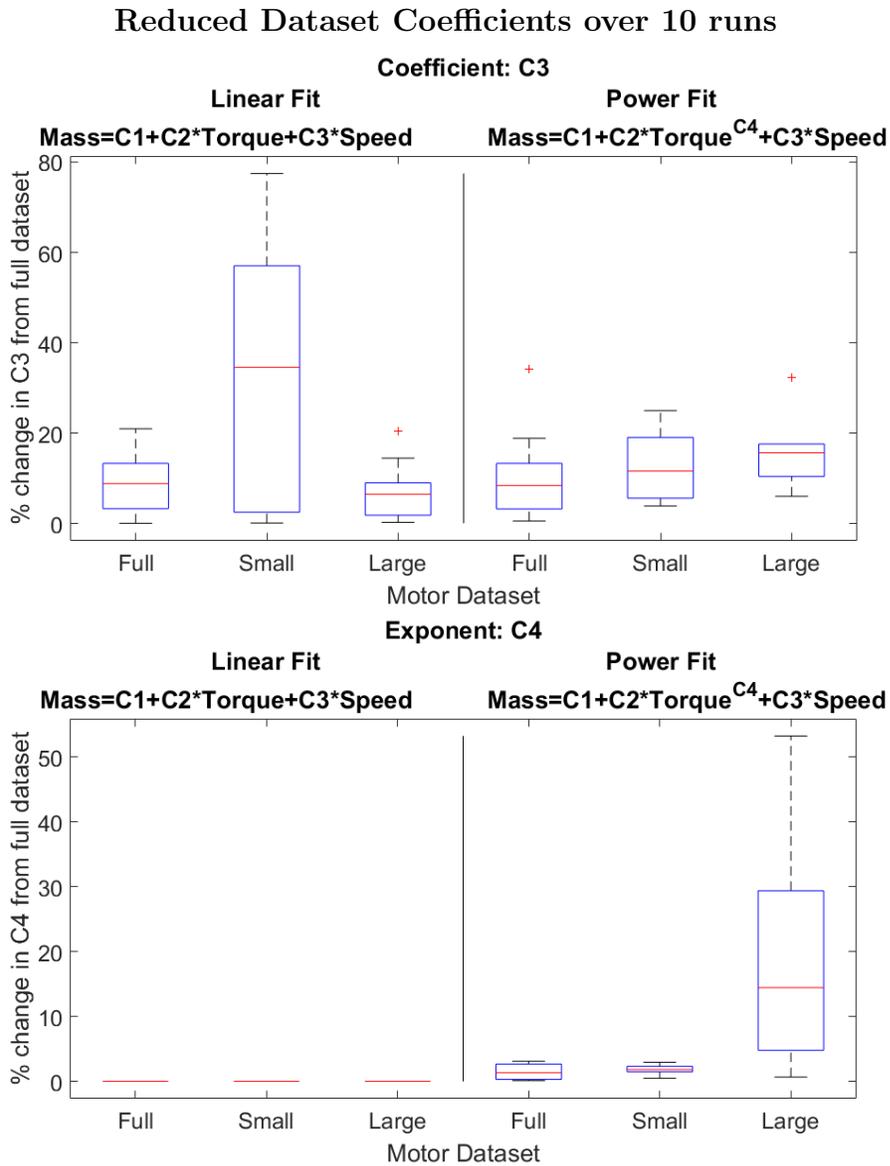


Figure 3.15: Sensitivity Analysis for Linear and Power Fit for Motor Speed coefficient (c_3) and Torque exponent (c_4)

Figures 3.14 and 3.15 show that the linear fit is not sensitive to a 50% data reduction. The greatest change occurs in c_3 , the scaling value on the motor speed. The average percent change for the small motor datasets is 35%. While this percentage may seem high, the actual c_3 value is very small, and its impact is minimal.

This study focuses on larger manipulators capable of handling large inertial payloads; therefore, despite the variation, the study is still improved by the use of the linear fit model for large and full datasets.

The sensitivity analysis for the power fit shows that separating the motors into the two piece-wise functions increases the sensitivity of the results, especially in the case of the exponent c_4 . This change in the exponent value greatly affects the results. The coefficients and the exponents of the power fit model are more sensitive to the initial motors than the linear fit model. Therefore, a linear fit model is used to more accurately estimate the motor mass.

3.1.6 Motor Mass Estimation Conclusion

Linear and power fit regressions have been used to model the motor mass as a function of continuous torque and motor speed. In order to reduce the residual error for larger motors, the datasets were broken into two categories: small and large motors. Breaking the motors into these two categories reduces the residuals for both models, as well as increases the sensitivity to the motors used in the initial data. The linear model is less sensitive than the power fit model and provides a high goodness of fit (GOF).

The robust linear model, with the two categories of motors, will be used for the rest of this study. This model has proven itself to be insensitive (i.e. robust), able to produce a high goodness of fit, and small residuals for both the small and large motor categories.

This model estimates the motor mass given the desired torque and speed. Equation 3.2 summarizes the motor mass estimation used in the rest of this study. Here T is the motor torque given in Nm; S is the speed given in kRPM. While the models were made with small motors, with a torque less than 2Nm, and with large motors, the intersection point of both models is at 3.17 Nm.

$$MotorMass(T, S) = \begin{cases} 0.0676 + 1.3788T + 0.0010S & |T| < 3.17Nm \\ 2.5067 + 0.6071T - 0.2796S & |T| > 3.17Nm \end{cases} \quad (3.2)$$

As George Box famously noted: “...the statistician knows that in nature there never was a normal distribution, there never was a straight line, yet with normal and linear assumptions, known to be false, he can often derive results which match, to a useful approximation, those found in the real world.” (1976)

3.2 Gearbox estimation

The motor mass estimation does not include any gearing. Instead, this estimation is performed in the gearbox estimation found below using the desired output torque and an assumption of harmonic gearing. This type of gearing eliminates backlash, provides high gear ratios (on the order of 100:1), and is very compact. Large space manipulators, like the SRMS, usually utilize multi-stage gearboxes which can result in a gear ratio on the order of 1000:1. Ranger, a highly dexterous manipulator designed for satellite servicing, makes use of harmonic gearing which ranges from 100:1 to 160:1, depending on the joint. [101] describe a lightweight space manipulator with harmonic drive gearings.

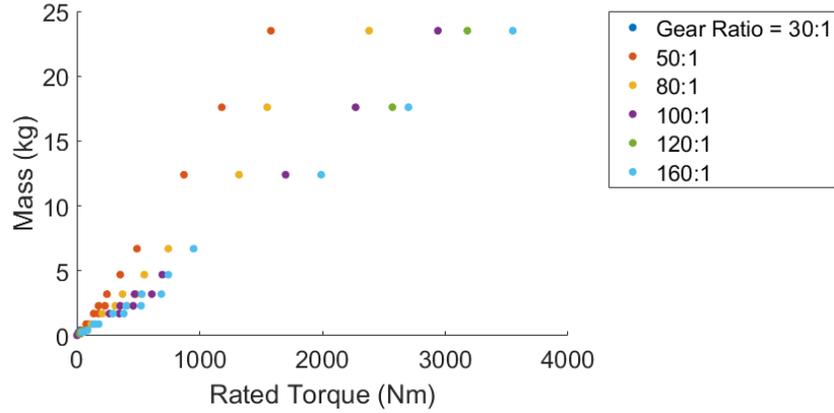


Figure 3.16: Harmonic Drive, LLC, gear box mass data, collected from harmonic net using two series of gearboxes[102].

The mass of a harmonic gearbox is dependent on the rated output torque and the gear ratio. As seen in Figure 3.16 and Table 3.7, a linear fit can be made for each gear ratio. Taken together, the GOF is lower at 0.9017, see Equation 3.3.

$$HarmonicMass(kg) = 0.0078(RatedTorque(Nm)) \quad (3.3)$$

$Mass = C_0 * RatedTorque$						
GearRatio	30:1	50:1	80:1	100:1	120:1	160:1
C_0	0.0101	0.0146	0.0098	0.0076	0.0069	0.0065
GOF	0.7936	0.9954	0.9802	0.9905	0.9859	0.9933

Table 3.7: Linear Coefficient for Mass/Torque Relationship for Harmonic Gearing

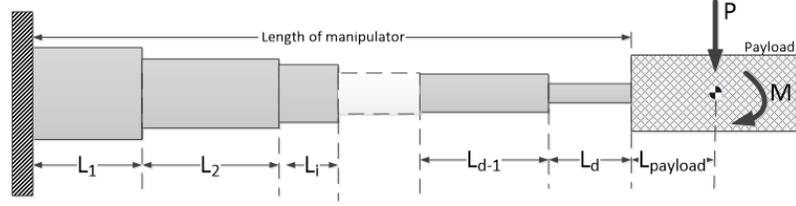


Figure 3.17: Stepped cantilever beam used for link mass estimation

3.3 Link Mass Estimation

The structural mass is calculated by sizing each link to ensure that the manipulator is not buckling and that a maximum deflection is not exceeded. For the purpose of this study, the link deflections are designed to be two orders of magnitude less than that of the total manipulator length.

$$TotalLinkMass = \sum_{i=1}^{EveryLink} \pi \rho (Length) (r_{outer}^2 - r_{inner}^2) \quad (3.4)$$

A non-linear constrained optimization is used to calculate the thickness and outer diameter of each link. It is assumed that the manipulator is fully extended with a maximum force and moment applied at the end (e.g. in a worse case pose). The maximum force and moment are based on the payload mass and the maximum acceleration of the payload as dictated by the task definition. It is assumed the manipulator is able to withstand this maximum force both tangentially and axially.

The deflection of each joint segment is modeled as a stepped-cantilever beam. Equation 3.5 shows the deflection due to both a force and a moment[103].

$$\begin{aligned}
\delta_{force} &= \frac{P}{3E} \sum_{i=1}^d \frac{1}{I_i} \left(\left(\sum_{j=i}^d L_j \right)^3 - \left(\sum_{j=i+1}^d L_j \right)^3 \right) \\
\delta_{moment} &= \frac{M}{2E} \sum_{i=1}^d \frac{1}{I_i} \left(\left(\sum_{j=i}^d L_j \right)^2 - \left(\sum_{j=i+1}^d L_j \right)^2 \right)
\end{aligned} \tag{3.5}$$

$$\delta_{total} = \delta_{force} + \delta_{moment}$$

A second constraint ensures the manipulator will not buckle. Equation 3.6 calculates the critical buckling load based on axial compression. γ is a correlation factor used to align the theoretical results to the experimental[104]. It is assumed that the link length is a thin-walled cylinder.

$$\gamma = 1 - 0.731(1 - e^{-\phi})$$

$$\phi = (1/16)(Outer\ Radius/thickness)^{.5} \tag{3.6}$$

$$\sigma_x = \gamma E / (3(1 - \mu^2)^{.5}) * (t/r)$$

Equation 3.7 is the critical buckling stress when under torsion. A correlation factor is used, and the value below is recommended for moderately long cylinders.

$$\gamma^{3/4} = 0.67 \tag{3.7}$$

$$\tau_{xy} = 0.747 * \gamma^{3/4} E / (r/t)^{5/4} (l/r)^{.5}$$

The total critical buckling stress is a combination of the buckling under torsion and an axial force.

$$Rc + Rb = 1$$

$$Rc = \sigma_{critical}/\sigma_x \tag{3.8}$$

$$Rb = \tau_{critical}/\tau_{xy}$$

$$\sigma_{critical} = ((1/\sigma_{axial}) + (/ \sigma_{torsion}))^{-1}$$

The optimization is bounded, and the minimum allowable thickness accounts for manufacturing limitations. This changes depending on the material used and is part of the task description: 0.8mm (1/32 inch) is used as a default minimum thickness. The maximum bounded value is used to ensure the thin wall cylinder assumption remains true.

3.4 Conclusion

This chapter describes the details for estimate the motor mass, the harmonic gear mass and the link mass for a manipulator. The motor and harmonic mass are both a linear function of the required torque. This allows the joint mass to be directly calculated once the manipulator's trajectory is determined from the task definition. The link mass is also derived from the task description. Once the task has been defined and the individual joint torques calculated, the total mass of the system can be estimated.

The link mass estimation is based on ensuring that the links can withstand thin wall buckling and are stiff enough to avoid excessive deflection. A nonlinear optimization is performed to find the thickness which will meet the constraints

while minimizing the mass. The frequency of the structure is another method of estimating the structural mass. This is based not only on the spring constant of the links, but also of the joints. In addition, this requires the desired control bandwidth to be known.

Chapter 4: Mass Estimation of Single Manipulator

This chapter demonstrates how to model a single manipulator using the described research methodology. An existing space manipulator provides a comparison point for the mass estimation. The Shuttle Remote Manipulator Systems (SRMS) was a large 6-DOF space manipulator. This particular manipulator was first launched in 1981 and was retired with the space shuttle program after 30 years and 90 flights. [105]

This manipulator is a prime example of a large space manipulator, designed to move large inertial payloads. The SRMS was capable of maneuvering payloads of up to 14,515 kg at a rate of 0.06 m/sec with a maximum contingency operation payload weight of 265,810 kg [1]. In this example, a large payload requirement will drive joint torques and thus the mass. Large systems like these, with a multi-jointed single manipulator, have masses dominated by joints. The most mass-savings occur when these systems are exchanged with multi-arm systems. Later, this study will be expanded to include an example using multi-arm systems rather than a single manipulator.

The trajectory planning for the SRMS is simple, as it is a non-redundant manipulator. The following methodology is valid for both redundant and non-

redundant manipulators, although it should be noted that the trajectory generation for redundant manipulators can be complex. Redundant degrees of freedom (DOF) would provide more freedom in designing trajectories, which could be used to further reduce the joint torques.

The total length of the SRMS is 15.2m. The upper and lower arm boom are approximately 5m and 5.8m respectively; see Figure 4.1 and Table 4.1 for the estimated values of other link lengths, which are significantly shorter than the upper and lower arm segments.

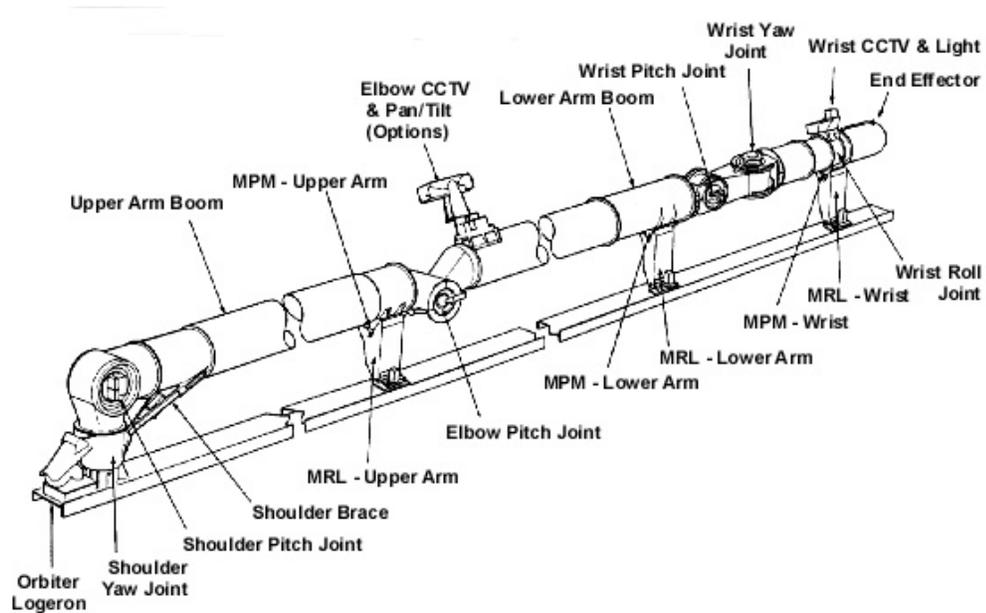


Figure 4.1: SRMS mechanical parts. Image Source: IEEE.ca [1]

The upper and lower arm links are a lightweight graphite epoxy composite; it is assumed that the material of every link remains constant. The material properties for the graphite epoxy can be found in Table 4.2. These values are necessary in

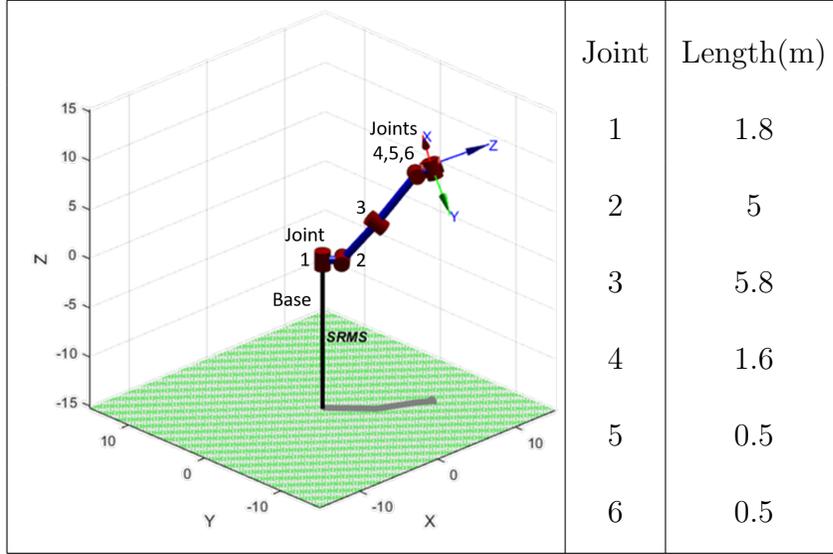


Table 4.1: Simulated SRMS with link lengths

estimating the structural mass for the system.

Figure 4.2 details the methodology for estimating the motor mass for a single manipulator. Once the desired task is generated, the dynamics and mass estimates can be calculated. This provides a mass estimate for a manipulator designed to perform a given task.

While the exact task(s) driving the design of the SRMS are unknown, this methodology works with all given tasks. Figure 4.3 shows the resulting mass of 100+ different tasks which are pseudo-randomly generated with different starting and ending positions/orientations. Many of the tasks actually result in a lower mass estimate than the actual SRMS.

The dotted and slashed lines in Figure 4.3 represent 10% and 20% of the actual mass, respectively. Most of the mass estimates calculated are significantly lower than the actual mass of the SRMS, although there are a few tasks that are higher. This

Material Properties for Graphite Epoxy

Property	Value
Poisson	0.2
Density(kg/m ³)	1700
Modulus of Elasticity (GPa)	160
Allowable Max Stress (Pa)	138x10 ⁶ (based on yield with a 2 factor of safety)
Minimum thickness	0.8mm (1/32inch)

Table 4.2: Material Properties for graphite epoxy

is to be expected, as robot designers must design for the worse allowable case.

Research Methodology for Single Arm System

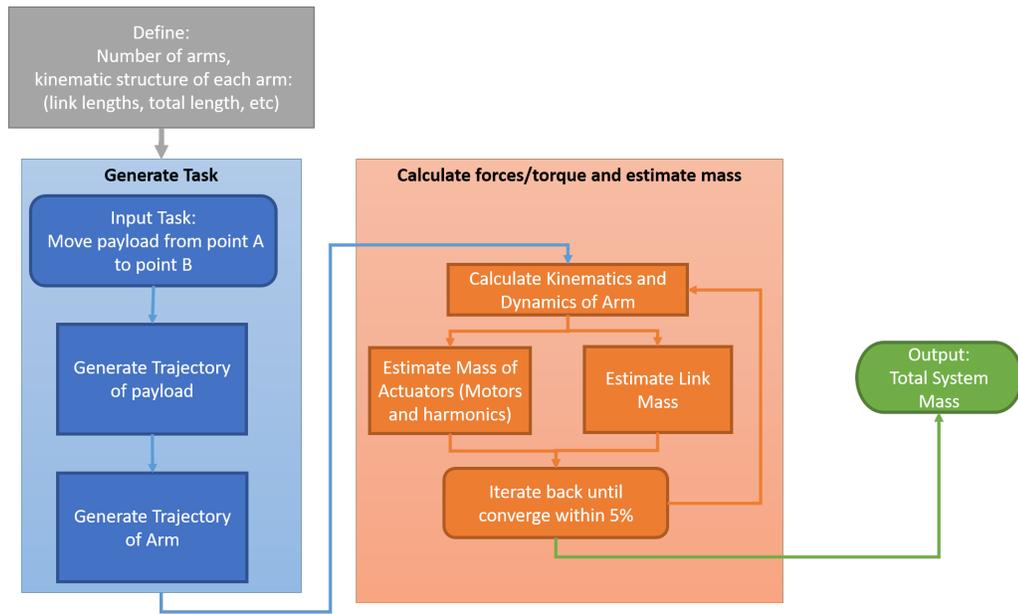


Figure 4.2: Research Overview: Single Arm Study

The maximum calculated mass is 504kg, which is within 20% of the actual SRMS mass. The total distance traveled by the arm is relatively small, only 5.2m.

During this task, the manipulator is tasked with operating the tip near the base of the arm, with the manipulator nearly folded in half. This is a difficult position for any manipulator, and is not recommended for safely utilizing the system. The manipulability of this task is very low, resulting in large joint torques and masses. Figure 4.3 shows the minimum translational manipulability measurement for each task.

Mass vs Manipulability for various tasks for Independent single arm

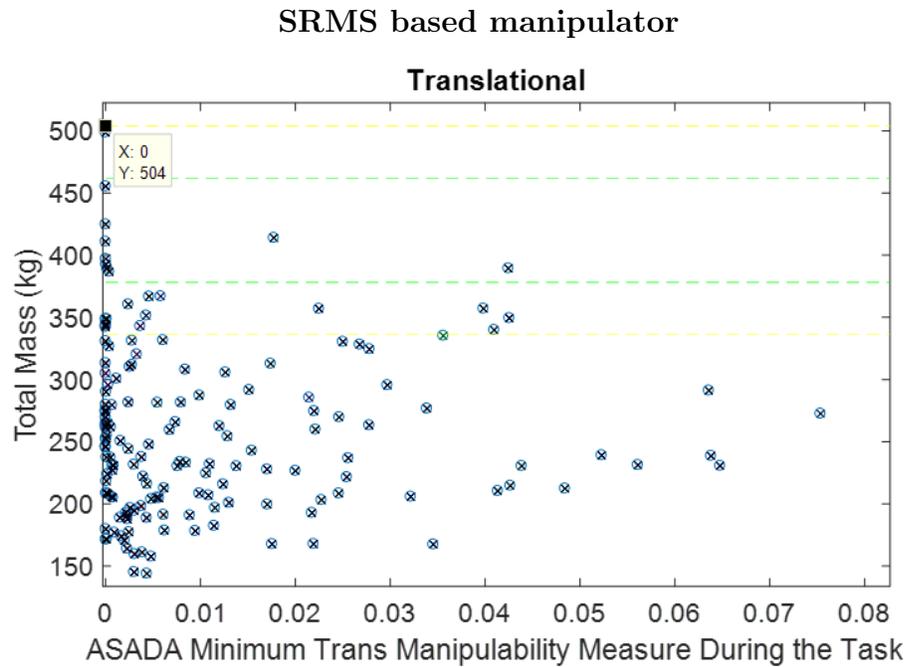


Figure 4.3: Mass vs Manipulability metric for each task. Dotted lines denote the +/- 10 % of the actual RMS Mass

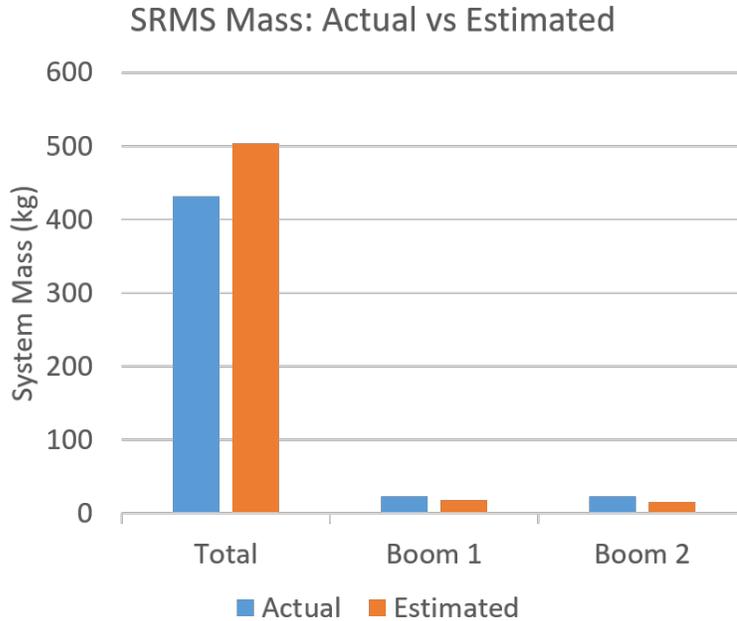


Figure 4.4: Mass comparison between worse case task and actual mass for SRMS

Figure 4.4 shows the comparison between the total mass estimation and actual mass of the SRMS. The structural mass of the each of the major links are also presented, showing the estimated mass to be lower than the actual mass. Again, the material of the links are assumed to be constant, without any interfacing hardware.

4.1 Examining a Docking Maneuver

This section details the different results for a task which fully utilizes the capabilities of the SRMS manipulator: the docking maneuver. This is a design -driving task which results in larger joint torques and, thus, mass. See Figure 4.5 for images of the manipulator performing this maneuver. This maneuver involves moving a payload from the edge of the manipulator’s work-space to a docking position.

Points Along Trajectory for Moving An Inertial Payload into a Docking Position

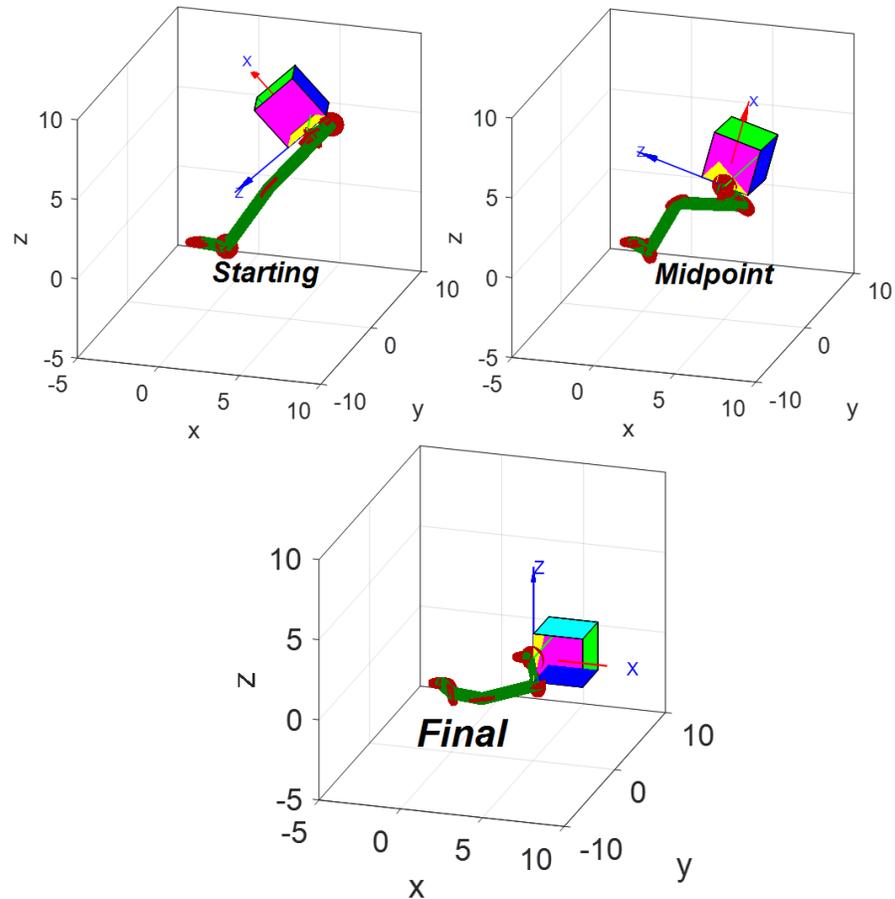


Figure 4.5: Docking Example: moving a large payload near the robot base

Figure 4.6 shows the joint accelerations, velocities, and angles over the course of the maneuver which utilizes all six joints. The joint movements are continuous and controlled, with no sudden jumps or flips through the joints. This creates a smooth movement across the trajectory.

The total position error is the squared error between the desired and actual payload location. If the error is large, then the arm is incapable of moving the

payload to the desired point. This is an important consideration, especially as optimization techniques are applied to adjust the link lengths.

Angle, Velocity and Acceleration of Payload During Docking Maneuver

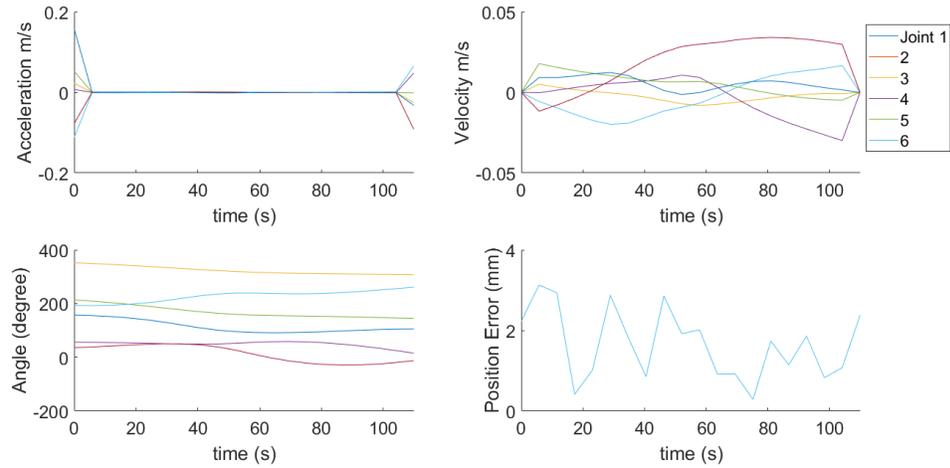


Figure 4.6: Summary of Docking Trajectory

Maximum joint torques occur during the periods of acceleration and deceleration, see Figure 4.7. During the constant velocity portion of the trajectory, the joint torques are insignificant. The maximum joint torques are used to perform motor and gearbox mass estimations.

Joint Torques during docking maneuver

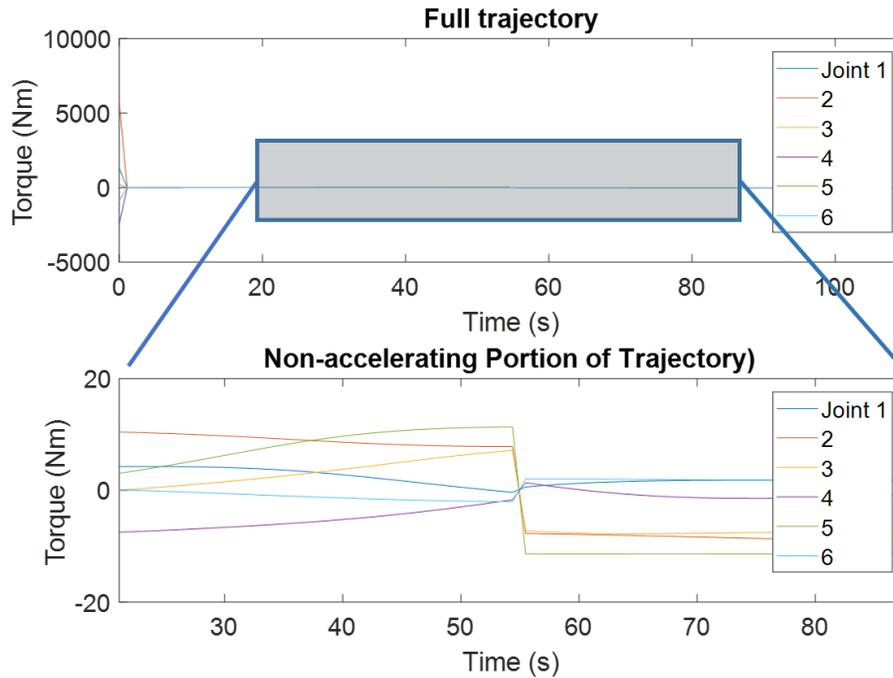


Figure 4.7: Joint Torques during docking maneuver. Lower Image contains the zoomed in portion to show that the joints are producing torque during the entire maneuver

The estimated and actual mass are 450 kg and 431 kg respectively, with a 4.4% error. The majority of the mass comes from the combination of the motor and gearboxes (i.e. the actuators or joints), see Figure 4.8 and 4.9.

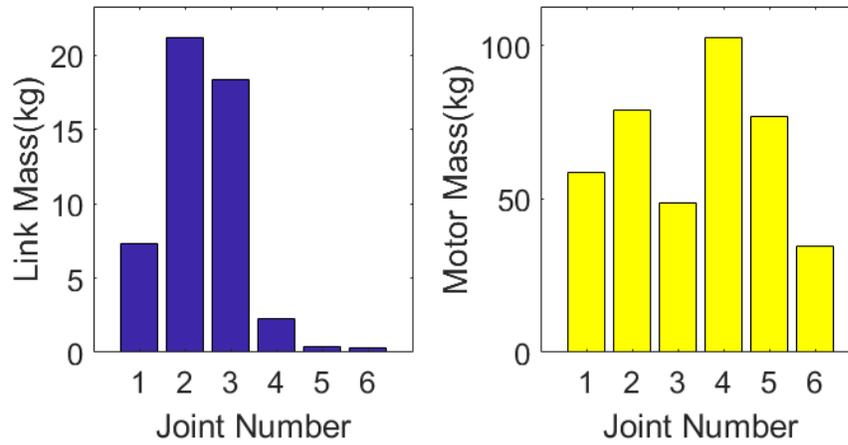


Figure 4.8: Link and Actuator Mass per Joint

Figure 4.9 shows which joints contribute the most mass.

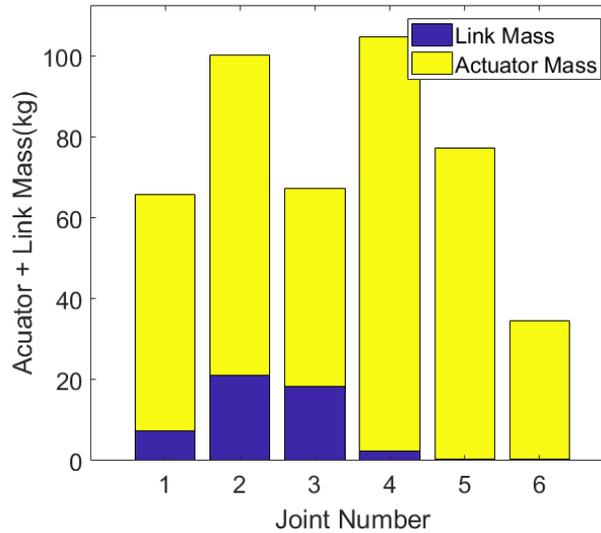


Figure 4.9: Link and Actuator Mass per Joint

The structural mass is heavily dependent on the required stiffness of the system. Part of the task definition is allowable tip deflection. This maximum deflection is set at 1/100 of the total length of the manipulator, or 15.2cm, when under maxi-

mum load and torque at full extension.

Decreasing the allowable deflection at the tip increases the structural mass. For example, reducing the allowable deflection to 1/200 of the total length results in a 6 kg increase in mass. The estimated link masses increase to 24kg and 21kg, for the lower and upper booms respectively. This allowable deflection is based on a assumed static force, and not on dynamic impacts.

4.2 Conclusion

The mass estimation of the task-based design simulator is within 20% of the actual mass of the SRMS space manipulator. Over 100 tasks were pseudo-randomly generated to validate the mass estimation model. Many tasks result in mass estimations within 10% of the actual mass of the SRMS, 431kg. The maximum estimated mass is 504 kg for a difficult task with low manipulability. A representative task, chosen to fully utilize the capabilities of the SRMS, shows an error within 5%.

This task highlights the conclusion that the mass estimator is heavily reliant on design bias when selecting “critical” task(s). Often in space robotics, this is difficult to define as the manipulators are expected to work for long periods of time and perform unpredictable tasks. Without knowledge of the “critical” task(s), a *series* of tasks can be used to provide insight and bound the situation, as seen in this chapter. Many of the outliers, primarily those with larger masses, are due to low manipulability or poor trajectory generation.

When a critical task is used, like the docking maneuver, the mass estimation

for the SRMS was within 5% of the actual mass. Therefore, this method of mass estimation is useful for a systems level analysis in comparing manipulator designs.

Finally, the example shows that the system mass is primarily driven by the joint masses rather than the link masses. Figure 4.2 shows the percentage of the total mass of each actuator.

The addition of a second manipulator reduces the required torques and allows for smaller and weaker motors to be used. However, if the structural mass (link mass) is responsible for over 50% of the total manipulator mass, then moving to dual, or multiple arms will not result in a lighter system. The mass saving from selecting smaller actuators can only be greater than the entire mass of the second arm, if the actuator mass is significantly driving the system mass.

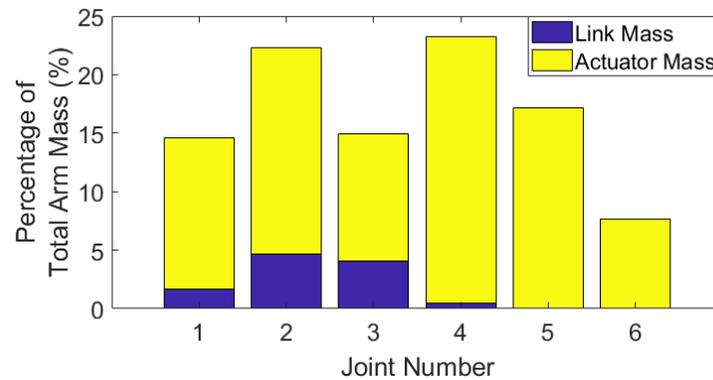


Figure 4.10: Percent the total arm mass that each joint contributes; broken down into actuator mass and link mass

Chapter 5: Force Distribution Between Multiple Manipulators

The task is defined as moving payload along a trajectory which require manipulators to provide forces along the desired directions. Typically, there are two forces that act on a payload: internal and external. While the external forces result from the motion of the payload itself, leading to the task, or trajectory completion, the internal force, or squeeze force, is the force applied to the payload by each of the multiple manipulators. The net internal force applied by the manipulators is zero; the individual manipulator forces cancel, and no movement is translated to the payload trajectory. However, these internal forces are reacted through the payload; high values can potentially cause damage and low values may result in slipping at the grasping points on the payload.

Due to multiple manipulators participating in the cooperative task, there exists an infinite number of individual end-effector forces and torques which result in the same desired force/torque applied to the object. The force distribution must be selected to yield the best results as dictated by the task definition and the design constraints. The next section reviews some of these constraints considered by previous researchers.

5.1 Literature Review

In order to avoid damaging the object, Nakamura (1988) focuses on minimizing the strain energy in dual arm systems. Koga (1992) assumes that every arm has the same performance and that load sharing is equal among them. Their aim is to resolve both the internal and external forces, yielding a parameter required in the task definition. This prevents the manipulators from breaking the object if they applied too much internal force and also prevents them from dropping the payload by not applying enough [49].

Orin and Oh (1981) and Cheng and Orin (1990) minimize the energy consumption. Orin suggests that the load distribution problem could be eliminated through an optimization approach: joint torques minimize energy consumption and balance forces among the arms. This approach has the disadvantage of excessive computational expense[88, 89].

Many researchers focus on the controller necessary to implement multi-arm motion, using a force/torque sensor to aid in measuring the internal forces. These sensors provide feedback used to ensure the internal force stays within acceptable bounds. Dai and Liu show that the internal and external loads can be decoupled; they can be designed to achieve different missions for mobile networked manipulators [106]. Liu et al. simulate a dual arm space manipulator and shows that their controls strategy, based on the knowing the task's trajectory, can realize the dual-arm coordinated operation. They also show that the internal force, which is not considered in the controller, is still within reasonable range[107].

Carignan and Akin focus on the minimization of total energy for dual cooperating manipulators, assuming the manipulators are rigidly attached to the payload[21]. Thus, there are no minimum required internal forces, and all the provided force can, and should, go into producing the desired motion of the payload.

Recently, Erhard and Hirche state that the complete characterization of internal forces and torques is still an open question[108]. They propose that characterization of the internal loading violates kinematics constraints imposed on the manipulator, and instead focus on an analytical expression for the external load distributions. This characterization of internal stress offers additional degrees of freedom (DOFs) for choosing a desired load distribution and allow incorporation of heterogeneous payload capacities. Like Carignan and Akin, they found that with all the end-effector forces contributing entirely to the desired motion, no internal wrenches are present.

5.2 Force Distribution: Even Force

There are multiple methods of determining how forces are distributed among multiple arms. Zheng and Luh focus on optimizing the load distribution in leader-follower robots to ensure the exerted forces are only external; there are no internal forces [52]. They asserted that, to reduce the computational complexity (a common issue with other optimization solutions [88]), the forces contributed by each manipulator should be equal.

Here F_{leader} and $F_{follower}$ are the exerted forces by each manipulator onto the

payload. F_{COM} is the force applied to the payload's center of mass necessary to accomplish the desired task.

$$F_{leader} = F_{follower} = 0.5F_{COM} \quad (5.1)$$

Equation 5.2 shows the relationship between the task force and the resulting joint torques. The joint torques are described for the i^{th} manipulator, as the force is evenly distributed across n manipulators. More detail on the equations of motion for manipulators can be found in Chapter 2.

$$T_i = M_i\ddot{q}_i + g_i - (1/n)J_i^T F_{COM} \quad (5.2)$$

Zheng and Luh were concerned with finding a computationally efficient method. They focus on using industrial manipulators, whose peak force is generally known. The simple solution of evenly distributing the forces allow Zheng and Luh to apply this method onto their systems[52].

A simple trajectory is used to describe the payload's motion as it is moved from point A to point B. The trapezoidal trajectory accelerates the payload to a constant velocity. The payload then travels at the constant velocity for the majority of the trajectory and finally decelerates to zero velocity ending at point B. This example has two manipulators moving a payload along the x-axis; this is a planar motion without any rotation. This example uses two simple 3DOF planar manipulators. Each manipulator is the same total length as the SRMS, 15.2m. Here, the payload is placed equidistantly between the two manipulators, resulting in symmetric manipulator motions throughout the trajectory. This can be seen in Figure 5.1.

In this task, the two manipulators move the payload from $(-2,8)$ to a $(2,8)$.

Symmetric Planar Example

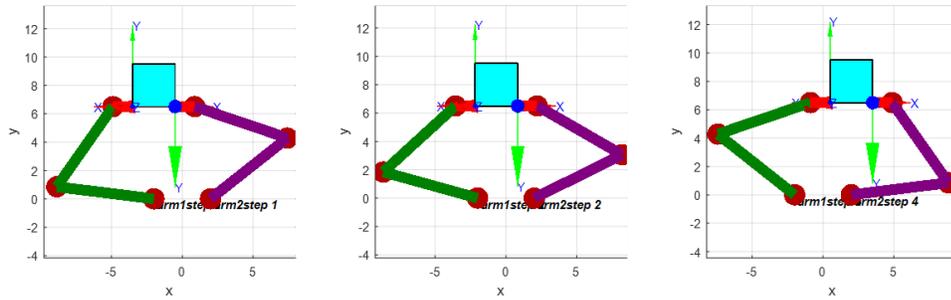


Figure 5.1: Simple Planar motion of SRMS type arms: symmetric case

Figure 5.2 shows the total force required to move the payload in the world frame, which is divided by half through the use of evenly distributed forces among the two manipulators. The majority of the imparted forces occurs at the beginning and end of the trajectory when the payload is accelerating/decelerating.

Symmetric Planar Example: Even Load Distribution

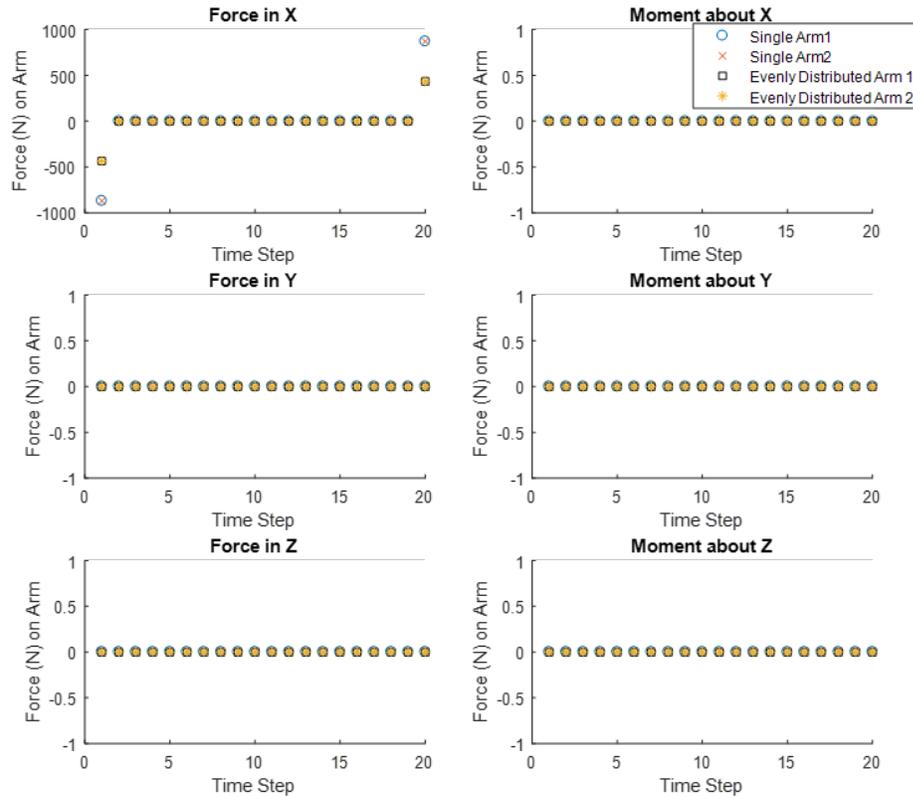


Figure 5.2: Symmetric Case: Force evenly distributed between dual SRMS type arms for simple planar motion

Once the forces on each arm are known, the joint torques can be calculated. The required joint torques are a function of forces on the manipulator and the joint angles. Figure 5.3 shows the maximum (absolute) joint torques for the simple motion described above. A single arm must produce joint torques which are twice as much as the evenly distributed dual arm case. These results may seem very simple: evenly distributing the forces between two arm reduces the joint torques by half.

Symmetric Planar Example: Even Load Distribution

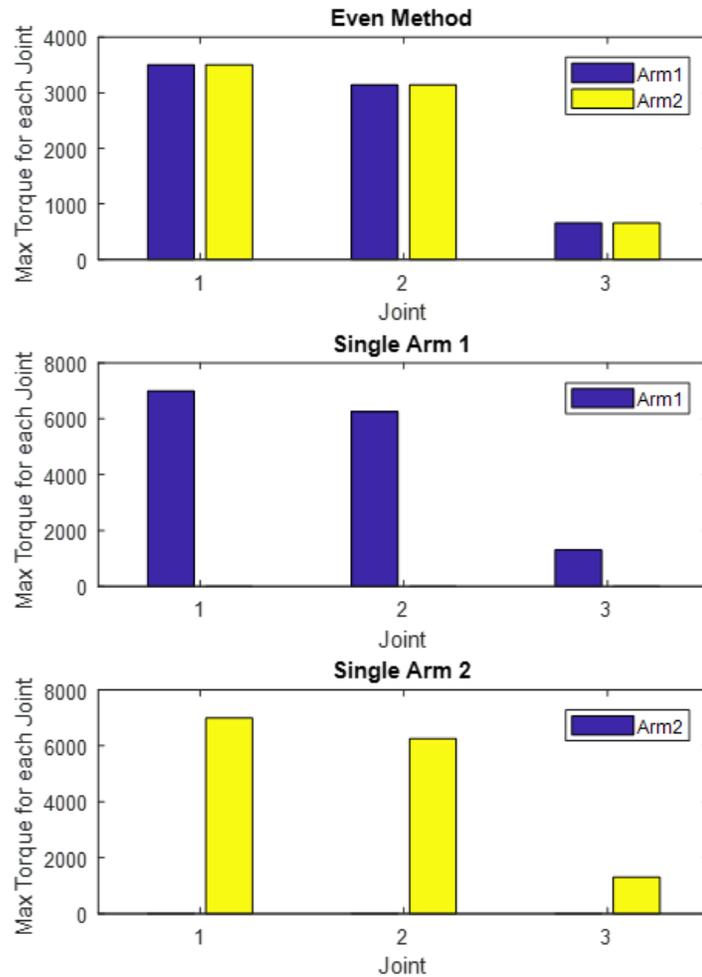


Figure 5.3: Symmetric case: Joint torques for forces evenly distributed between dual SRMS type arms for simple planar motion

Figure 5.4 takes a closer look at a planar case with non-symmetric motion. Here, the manipulators from previous example move the payload from $(-2, 8)$ to $(-2, 9)$. Note that the only change between this example and the previous is the additional motion along the y-axis.

Non-Symmetric Planar Example

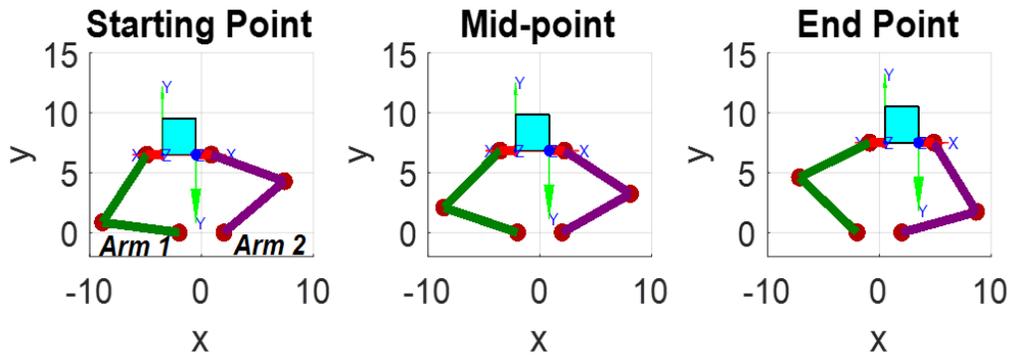


Figure 5.4: Simple Planar motion of SRMS type arms: Non-Symmetric Case

Non-Symmetric Planar Example: Even Load Distribution

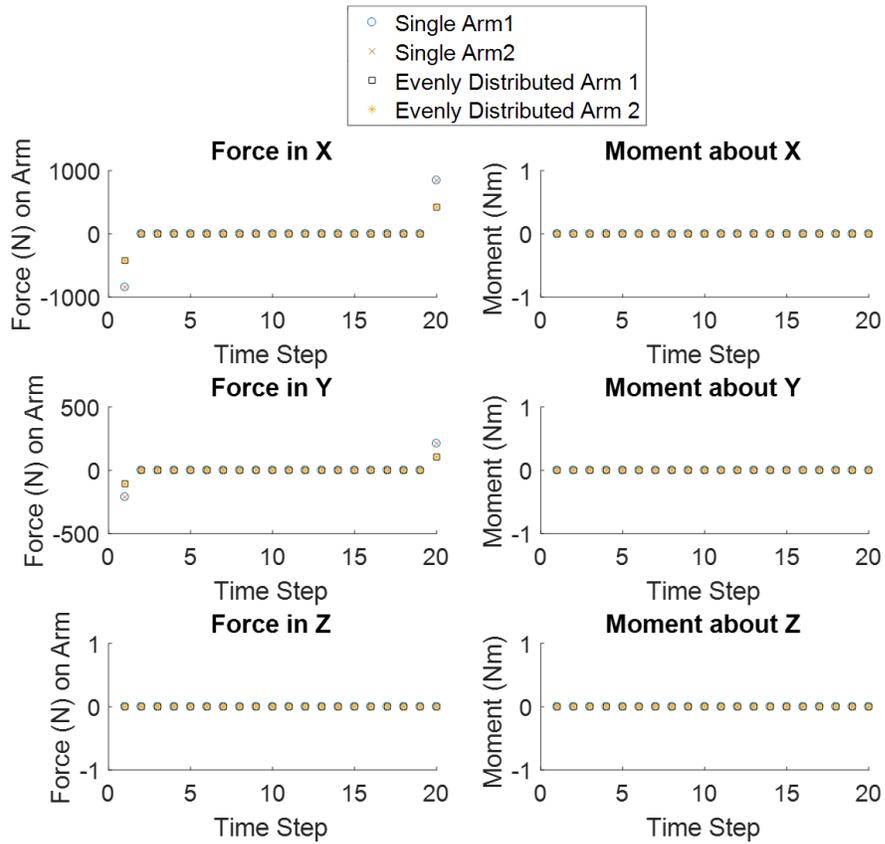


Figure 5.5: Force evenly distributed between dual SRMS type arms for simple planar motion

The required force to move the payload is divided evenly between the two manipulators, see Figure 5.5.

In this case, the motion between the two manipulators are no longer symmetric; some joints must now produce more torque than their counterparts (see Figure 5.6). The joint torques are not same between the non-symmetric Arms 1 and 2; Arm 2 requires more torque to provide the required force.

Non-Symmetric Planar Example: Even Load Distribution

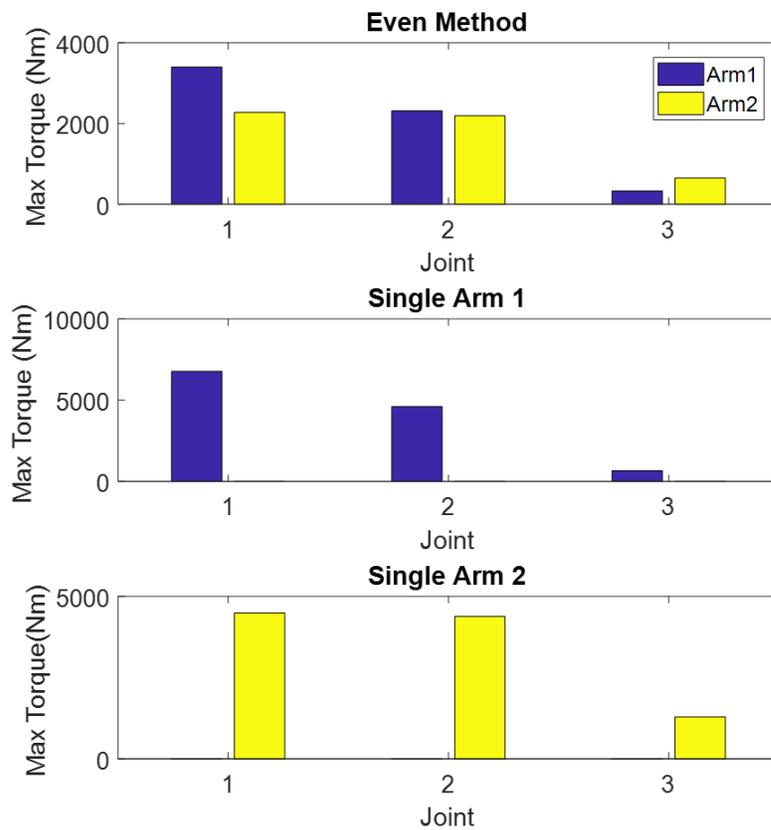


Figure 5.6: Max Torque per joint with load evenly distributed

These examples show that, even with a simple force distribution scheme, the required joint torques can be greatly reduced. The even distribution method assumes each arm will produce the same force. Other force distribution methods do not

assume the force is evenly distributed, and can better utilize each manipulator.

5.3 Force Distribution: Minimizing Energy

Carignan and Akin find a load distribution between arms which results in the lowest possible motor energy consumption, see Equation 5.3. A weighting matrix, W_i , defines the motor efficiency and converts the torque values to energy. When the weighting matrix is set to the identity matrix, then minimizing the energy results in the minimization of the total squared torque.

Using Carignan's methodology, the following section describes the optimization for an n-arm system. The goal, as defined in Equation 5.3, is to minimize the motor energy. W_i is assumed to be known and a positive definite matrix. In the following examples, the identity matrix is used for all motor efficiency matrices.

$$C = (1/n) \sum_{i=1}^n T_i^T W_i T_i \quad (5.3)$$

In a static system, all the forces would balance and cancel each other out; this results in the manipulators applying a net zero force and results in no motion. Carignan and Akin considers the dynamics of the system and the equation of motion for the payload as found in 5.4.

$$M_e \ddot{X}_e + cg_e(X_e, \dot{X}_e) = \sum_{i=1}^{n \text{ arms}} (F_{i,COM}) \quad (5.4)$$

M_e is the mass matrix of the payload, \ddot{X}_e is the acceleration, and cg_e is the combined Coriolis and gravitational terms. $F_{i,COM}$ is the force from each end-effector acting

on the payload's center of mass (COM).

$$M_i \ddot{q}_i + cg_i - J_i^T F_{i,EE} = T_i \quad (5.5)$$

The equation of motion for each manipulator is given in Equation 5.5. Here J_i is the Jacobian, which is a function of the joint angles, q_i . The joint torques for the i^{th} manipulator are defined as T_i . $F_{i,EE}$ is the force exerted by the i^{th} manipulator at the end-effector.

$$J_{ei}^T F_{i,EE} = F_{i,COM} \quad (5.6)$$

Equation 5.6 contains the relationship between the end-effector's force and the desired force on the payload's COM. The force transformation matrix, J_{ei} , is defined by the angle and location of the end-effector on the payload.

Carignan and Akin introduce a geometry term, D_i , based on the instantaneous geometry of each manipulator and the payload. J_{ei} is the geometry between the end-effector and the payload's COM. J_i is a function of the current joint angles.

$$D_i = J_{ei}^{-1} J_i \quad (5.7)$$

The goal is to write the equation of motion for the payload as a function of the joint torques.

$$M_e \ddot{X}_e + cg_e = \sum_{i=1}^{narms} (D_i^{-T} (M_i \ddot{q}_i + cg_i - T_i)) \quad (5.8)$$

Lagrange multipliers provide a method for finding the optimal torque distribution. The objective function, f , is defined by the minimization of the total motor energy, see equation 5.9. This is a quadratic measure and gives an estimate of the motor power.

$$f(T_1 \dots T_n) = (1/n) \sum_{i=1}^{narms} T_i^T W_i T_i \quad (5.9)$$

Without any constraints, the minimum energy force distribution would result in a static system: the manipulators would not move the payload. Constraints ensure that the manipulators produce the force necessary to move the payload along the desired trajectory. The constraint function, g can be found in 5.10.

$$g(T_1 \dots T_n) = \left(\sum_{i=1}^n D_i^{-T} (M_i \ddot{q}_i + c g_i) - D_i^{-T} T_i \right) - M_e \ddot{X}_e - c g_e \quad (5.10)$$

Equation 5.11 contains the full system for the Lagrangian multiplier optimization.

$$\begin{aligned} L(T_1 \dots T_n, \lambda) &= f(T_1 \dots T_n) - \lambda g(T_1 \dots T_n) \\ \delta L / \delta T_i &= 0 \end{aligned} \quad (5.11)$$

The results of optimization are found in 5.12.

$$\begin{aligned} \delta L / \delta T_i &= 2T_i^T W_i + \lambda D_i^{-T} = 0 \\ T_i^T &= -(1/2) \lambda D_i^{-T} W_i^{-1} \end{aligned} \quad (5.12)$$

This simplifies to equation 5.13. The relationship between the manipulator torques is a function of the geometry, D_i .

$$D_1 W_1^T T_1 = D_2 W_2^T T_2 = \dots = D_N W_N^T T_N \quad (5.13)$$

Equation 5.13 shows a clear relationship between the torques of each manipulator, the geometry, and the motor efficiency. The joint torques of each manipulator can be substituted and re-written in terms of a single manipulator, see equation 5.14. $F_{dynamic}$, in 5.14, combines the dynamic terms for the payload and the manipulators.

The task statement defines the trajectory and geometry of the payload and the manipulators. D_i can be calculated at every step. At this point, the only assumptions that have been made about the system are that the Jacobian and D_i are invertible or, at least, pseudo-invertible.

$$T_1 = W_1^{-T} D_1^{-1} F_{dynamic} \left(\sum_{i=1}^n D_i^{-T} W_i^{-T} D_i^{-1} \right)^{-1} \quad (5.14)$$

$$F_{dynamic} = \left(\sum_{i=1}^n D_i^{-T} (M_i \ddot{q}_i + c g_i) \right) - M_e \ddot{X}_e - c g_e$$

At every time-step in the trajectory, the new torque values are calculated based on equations 5.14 and 5.13. While this load distribution is significantly more complex than just evenly distributing the loads, it produces a maneuver requiring less motor energy without changing the payload's or manipulators' trajectory.

Using the planar symmetric example, figure 5.3 shows the force from the minimum-energy load distribution, and compares it to those of a single arm. It is assumed that each motor has the same efficiency, and W_i is assumed to be the identity matrix for every arm. The arms exhibit a symmetric motion. The maximum torque for the joints of Arm 1 and Arm 2 are equal but occur at different

points along the trajectory. Unlike in the single arm cases, the dual manipulators also apply a force in the Y direction and a moment about Z. These forces/torques are produced by both arms in opposite directions, resulting in net zero forces/torques. These forces/torques are internal and do not contribute to the payload's motion.

Symmetric Planar Example: Energy Minimization Method

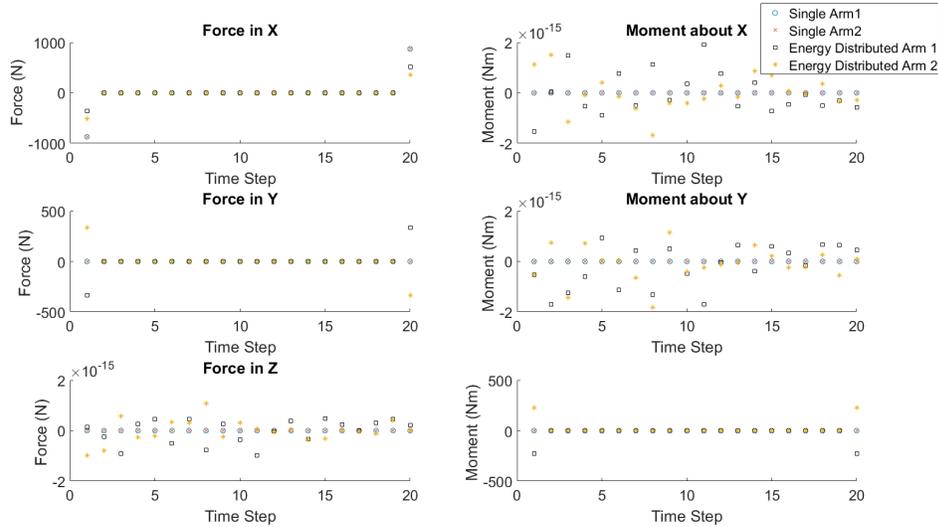


Figure 5.7: Force distribution for between dual SRMS type arms for simple planar motion

Figure 5.8 shows the maximum torque on each joint of the dual arm system. As expected, due to symmetry of the motion, the maximum torque is the same for each arm. The values are significantly less than the evenly distributed force method, especially in joint 2. There is a 15% and a 34% reduction in the peak torque for Joint 1 and Joint 2, respectively. The peak torques in Joint 3 are the same for both load distribution methods.

Symmetric Planar Example: Energy Minimization Method

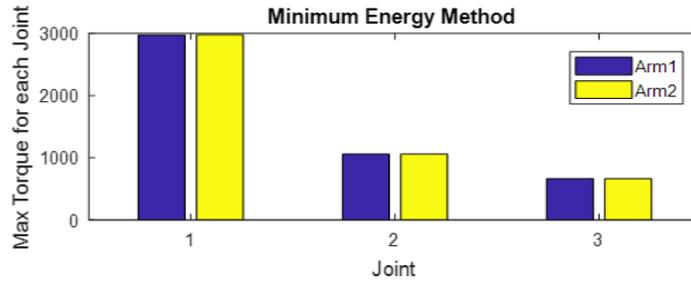


Figure 5.8: Force distributed between dual SRMS type arms for simple planar motion to minimize energy

The minimum-energy force distribution method takes advantage of the dual arm system. Forces and torques are reacted through the payload to the other manipulator, reducing the total joint torques and the required motor energy, T^TWT . Figure 5.9 compares the total motor energy for each load distribution method performing the same symmetric task. The even-load distribution method requires 54% more energy than the minimum-energy distribution method, with a single manipulator requiring 77% more. The main disadvantage to this method is that it is assumed that the payload is able to survive the internal forces imparted onto it by the manipulators. This method does not provide constraints to ensure the internal forces do not overload the payload.

Symmetric Planar Example: Energy Comparison

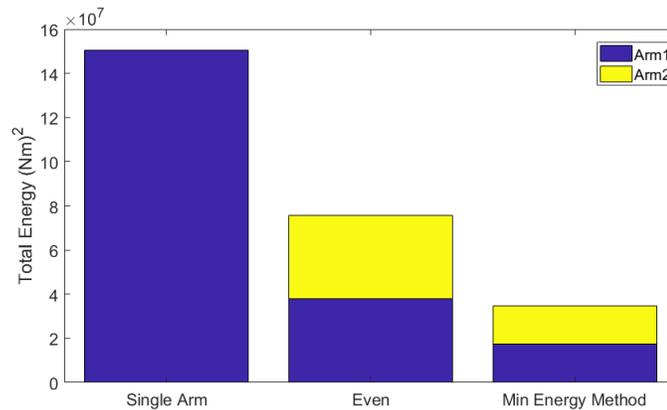


Figure 5.9: Energy requirement for Simple Symmetric Example: For a single arm, Evenly distributed force

Reducing the motor energy also reduces the required joint torque, providing a reduction in the motor mass. The minimum-energy distribution also results in a mass savings.

5.4 Force Distribution: Minimize Mass

Carignan focuses on minimizing the total motor energy, which reduces the joint torques and thus the motor mass. However, minimizing the energy does not necessarily result in a minimum mass. A new force distribution method minimizes the total joint mass and thus the system mass.

Chapter 3 concludes that the mass of a motor is a linear function of the torque and speed. The speed is derived from the known trajectory of the payload. The joint torques are a function of the force distribution. Once this is known, then the joint torques can be calculated for the entire trajectory. The joint mass is linearly

related to the joint torque, see equation 5.15.

$$M_{joint} = Motor(T, Speed) + HarmonicMass(T) \quad (5.15)$$

The total joint mass, see equation 5.16, is found using the previous equation and the results from Chapter 3. The joint mass is a function of the desired joint torque (Nm), joint speed (kRPM), and the gear ratio. The gear ratio, denoted as GR, converts the joint torques and joint speeds to *motor* torques and speeds. C_0 , C_T , C_S , and $C_{harmonic}$ are the coefficients for the intercept, the torque, the speed, and the harmonic mass terms, respectively.

$$M_{joint} = C_0 + C_T(|T|/GR) + C_S(S)GR + C_{harmonic}|T| \quad (5.16)$$

The objective function is a linear equation that minimizes the total mass of the system.

$$TotalJointMass = \sum_{n=1}^{Narms} \sum_{j=1}^{Njoints} M_{joint} \quad (5.17)$$

Combining the objective function with equation 5.16 leads to a linear equation for the total system joint mass.

$$TotalSystemJointMass = \sum_{i=1}^{Narms} \sum_{j=1}^{Njoints} C_0 + C_T(1/GR_{i,j})|T_{i,j}| + C_S(S_{i,j})(GR_{i,j}) + C_{harmonic}|T_{i,j}| \quad (5.18)$$

Constraints ensure the manipulators maintain the required forces, accelerations, and velocities in order to accomplish the given task. These are the same con-

straints used in the minimum-energy force distribution method. For convenience, it is restated in equation 5.19.

$$\sum_{i=1}^{Narms} (D_i^{-T}(M_i\ddot{q}_i + cg_i)) - M_e\ddot{X}_e - cg_e = \sum_{i=1}^{Narms} D_i^{-T}T_i \quad (5.19)$$

The left hand side of equation 5.19 is independent of the load distribution; all the terms are a function of the payload trajectory, manipulator trajectory, or the task definition. The constraint equation can be re-written as a linear constraint, see equations 5.20 and 5.21.

$$b_{eq} = A_{eq}T_{matrix} \quad (5.20)$$

With:

$$A_{eq}T_{matrix} = \sum_{i=1}^{Narms} D_i^{-T}T_i \quad (5.21)$$

$$b_{eq} = \sum_{i=1}^{Narms} (D_i^{-T}(M_i\ddot{q}_i + cg_i)) - M_e\ddot{X}_e - cg_e$$

The T_{matrix} contains the joint torques for every manipulator. A_{eq} is a coefficient matrix made up of the D_i^{-T} terms.

The objective function and the constraints are both linear functions. This allows the force distribution problem to be re-written as a Linear Programming problem.

The mass of each joint is a function of the torque, speed, and gear ratio. Once a gear ratio is chosen, the manipulator cannot change hardware mid-task. The force-distribution optimization can be performed for different gear ratios, but the selection

must occur before the optimization begins. The speed is task dependent and does not change depending on how the force is distributed among the manipulators.

The only variable in the force-distribution is torque, which allows the linearization to be further condensed, see equation 5.22. The gear ratio, speed term, and the intercept term are combined into a 'Motor Penalty' term. This is the mass penalty for having/using a motor.

$$TotalJointMass = MotorPenalty + \sum_{i=1}^{Narms} \sum_{j=1}^{Njoints} ((C_T(1/GR_{i,j}) + C_{harmonic})|T_{i,j}|) \quad (5.22)$$

It should be noted that the objective function is a function of the *absolute value* of the torque. Equation 5.22 is the same objective function modified to account for these absolute values. $P_{i,j}$ is a substitution variable for the absolute value of each joint torque, $T_{i,j}$.

$$TotaljointMass = MotorPenalty + \sum_{i=1}^{Narms} \sum_{j=1}^{Njoints} ((C_T(1/GR_{i,j}) + C_{harmonic})P_{i,j}) \quad (5.23)$$

Additional constraints are placed to ensure that $P_{i,j} = |T_{i,j}|$.

$$\begin{aligned} P_{i,j} &\geq -T_{i,j} \\ P_{i,j} &\geq T_{i,j} \end{aligned} \quad (5.24)$$

Equation 5.23, 5.20, and 5.24 allow the use of linear programming to solve the force-distribution problem in order to directly minimize the joint mass of a

multi-arm system.

Symmetric Planar Example: Mass Minimization

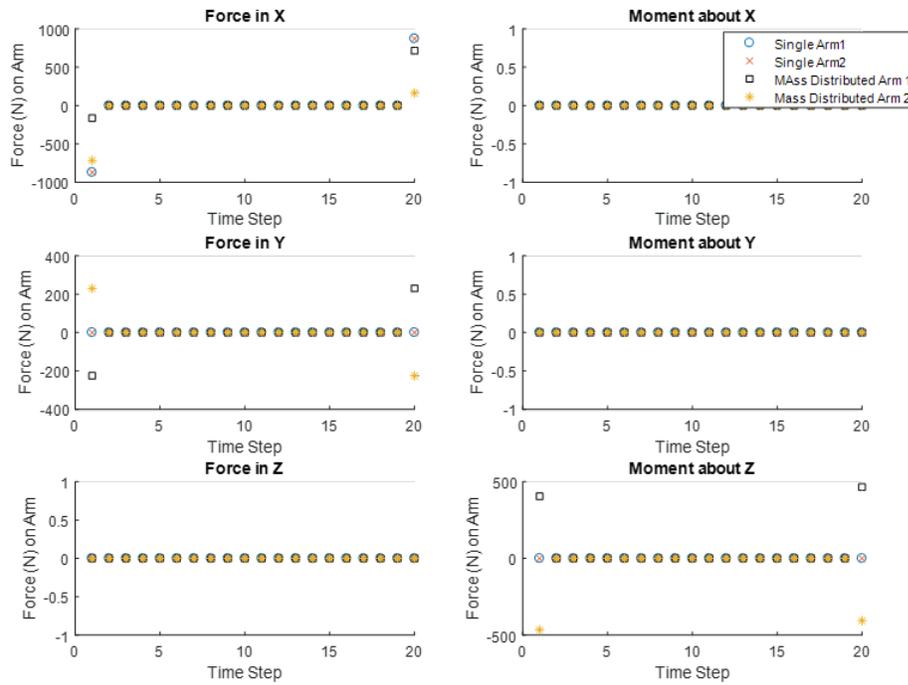


Figure 5.10: Force distribution for between dual SRMS type arms for simple planar motion

Figure 5.11 shows the maximum joint torques for the minimum-mass force distribution method. Joints two and three produce near-zero torque in both manipulators, which is only enough to overcome the motor friction and to move the manipulator. Joint one provides the torque necessary to complete the actual task.

Symmetric Planar Example: Mass Minimization

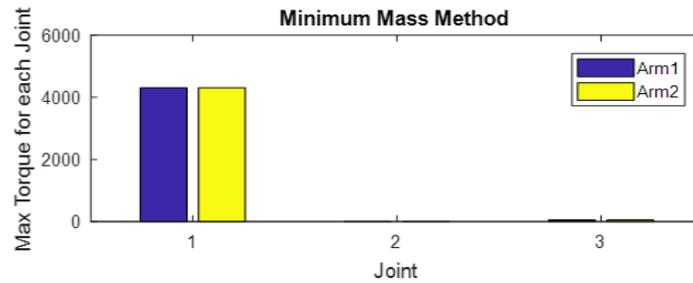


Figure 5.11: Maximum joint torque for a minimum-mass force distribution

The majority of the forces are applied at the beginning and end of the trajectory, during the acceleration and deceleration of the payload. Figure 5.12 show the forces/torques at the start of the trajectory for each of the distribution methods. This model assumes no friction, though as the number of manipulators increases, it is expected that the total losses due to friction will increase with the increased number of joints.

The single arm cases only provide force in the direction of motion, along the X-direction. Unlike the dual manipulator cases, these do not have a second arm with which to counter any other forces or moments; thus, the single arm must react all other torques out through the joints/motors. The even-force distribution method does not allow for any internal forces and thus, like the single arm, must react all forces through each manipulator independently. The minimum-energy and minimum-mass force distributions both allow forces to be reacted through the payload, resulting in significantly lower joint torques.

Symmetric Planar Example: Mass Minimization

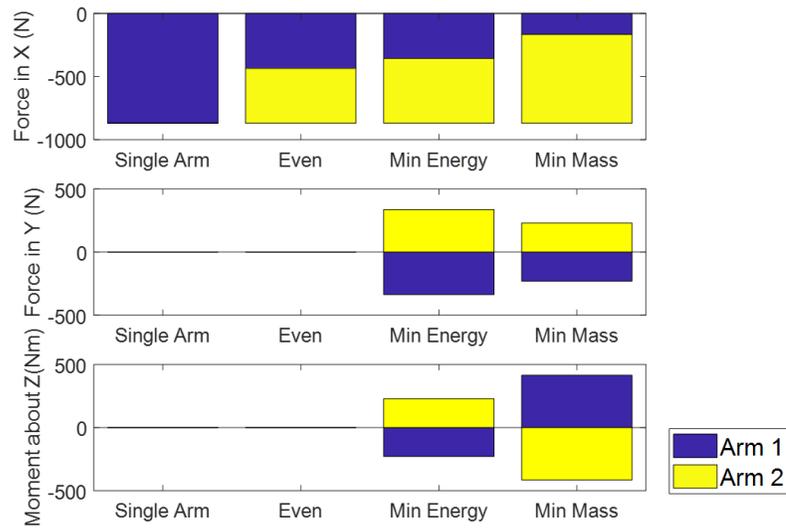


Figure 5.12: Force from each manipulator on the payload at start of trajectory

The total actuator mass for each force distribution method is in figure 5.13. Both single arm cases produce the same actuator mass, 197kg. The even-force distribution estimates a total joint mass of 102.5kg per arm, for a total joint mass of 204.4kg. While the joint torques are half that of a single arm case, the motor mass estimate has a non-zero intercept, resulting in the evenly-distributed dual arm case having a slightly higher joint mass than a single arm.

A minimum-energy and minimum-mass distribution yields a total joint mass of 137kg and 120kg respectively. The minimum-mass distribution reduces each arm to 60kg, which is a 70% reduction compared to the joint mass of a single arm.

Symmetric Planar Example

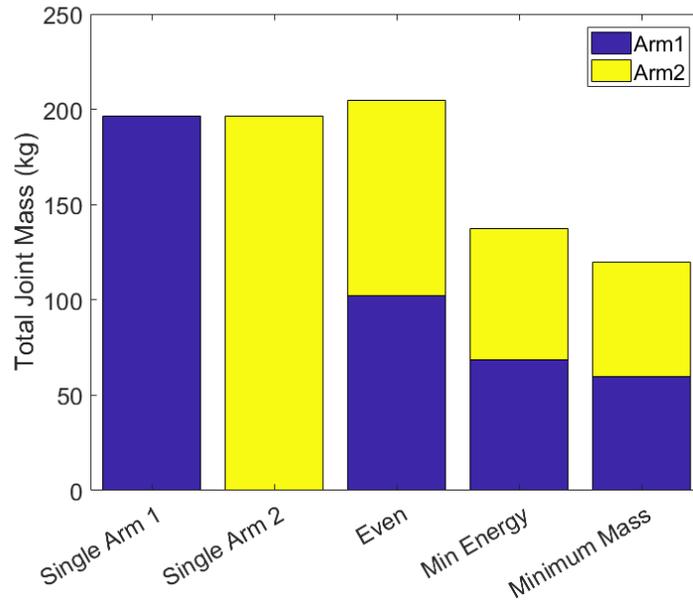


Figure 5.13: Total joint mass for each force-distribution method

This shows that, for a symmetric case, the joint mass is greatly reduced with the addition of a second manipulator when a proper force-distribution is chosen. This continues to hold true for non-symmetric cases as well. The maximum joint torques for each force-distribution methods is shown in figure 5.14 for the previously described non-symmetric example. Comparing the maximum torque for each of the single arm cases, it is clear that Arm 2 must produce more torque in order to accomplish the same task. The minimum-energy and minimum-mass force distributions utilize Arm 1 over Arm 2. This allows for a further reduction in the total joint torques.

Non-Symmetric Planar Example

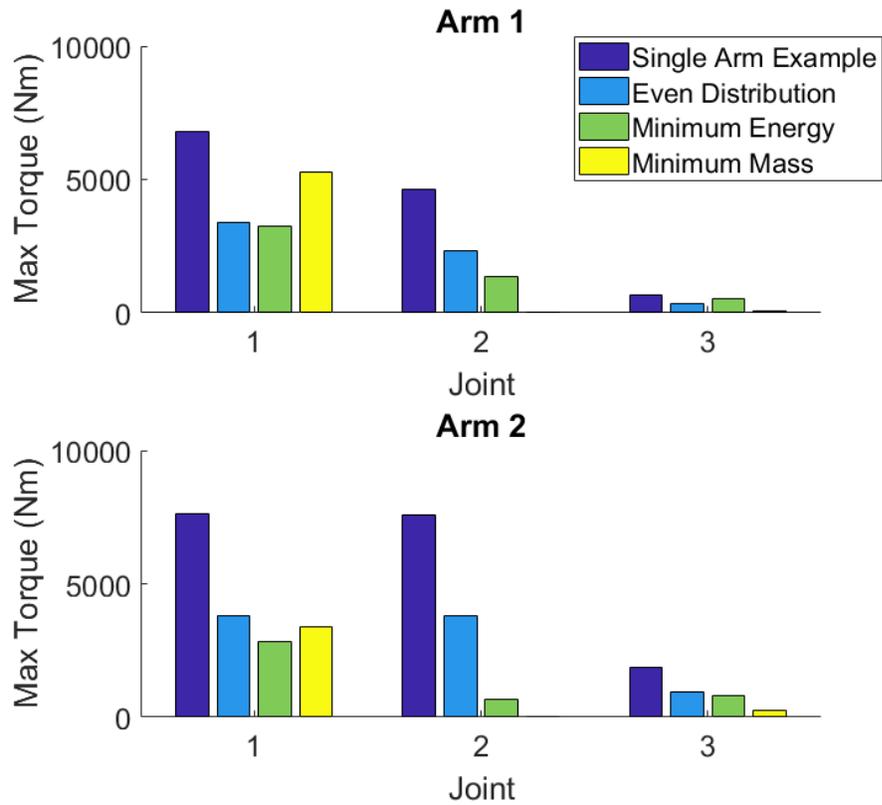


Figure 5.14: Maximum Joint Torque for Single Arm, Even force distribution, Minimum-energy force distribution and minimum-mass force distribution

Figure 5.4 shows the total energy for each force distribution method. Each of the multi-arm systems reduces the motor energy required for the maneuver, as well as the total joint mass. As expected, the minimum-energy method produces the lowest total energy; the minimum mass method produces the lowest joint mass. Both the minimum-mass and minimum-energy methods reduce the mass and energy as compared to the single arm and the even-force distribution method. Either of these force distribution methods would work well for space manipulators as reducing

the energy and mass are both driving criteria for designing space systems.

Non-Symmetric Planar Example

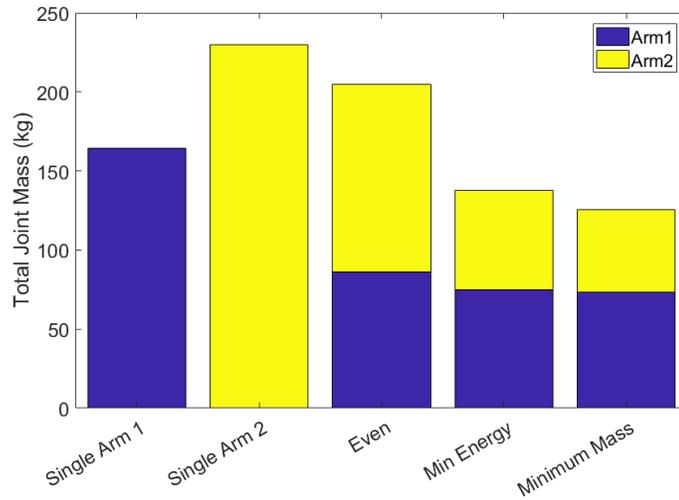


Figure 5.15: Total Joint Mass Comparison for all Force-Distribution Methods

Non-Symmetric Planar Example

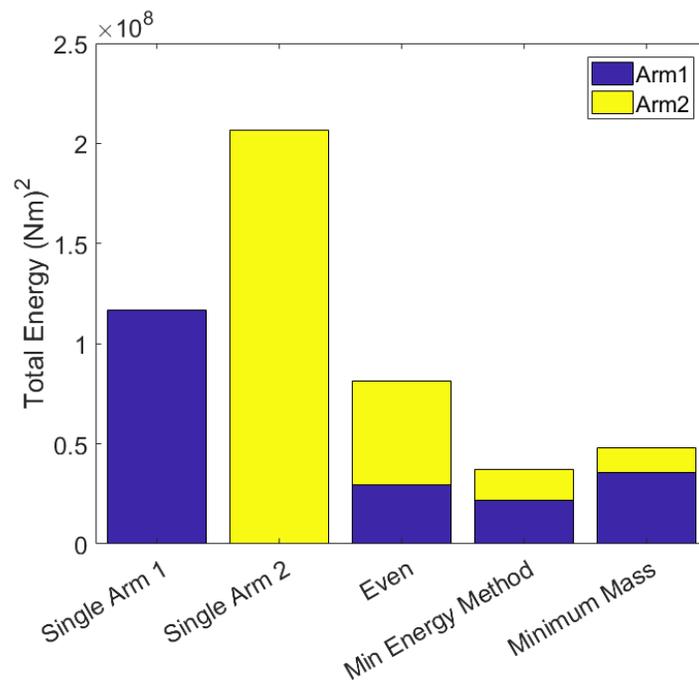


Figure 5.16: Total Energy Comparison for all Force-Distribution Methods

5.5 Conclusion

This chapter shows that multiple manipulators can work together to move a payload, allowing the forces to be divided between the different manipulators. This chapter presents three different methodologies: even-force, minimum-energy, and a minimum-mass distribution.

The even-force distribution method is the least computationally complex of these methods, and does not introduce any internal forces. When compared to using single manipulators, this method reduces the joint torques in half, but there are also twice as many joints. While this reduces the energy of the system and the individual joint torques, it does not reduce the total joint mass.

The minimum-mass and minimum-energy force distributions both result in internal forces or “squeeze forces” on the payload. These internal forces do not contribute to the motion of the payload. However, they should not be ignored if the payload is sensitive or fragile.

In the presented examples, both distributions reduce the total joint mass and the total energy. While the minimum-energy distribution *appears* to reduce the total joint mass, only the energy reduction is guaranteed. The minimum-mass method is formulated as a linear programming problem with linear constraints to find the minimum total joint mass. This results in a global solution and is guaranteed to minimize the total joint mass, and thus total system mass. The total joint mass was reduced by 40% in the symmetric case and 23% for the non-symmetric case by adding a second manipulator, as compared to the mass of Arm 1. Therefore, the

total system mass can be greatly reduced by adding a second manipulator and using the minimum-mass force distribution.

Chapter 6: Dual Manipulators Moving a Large Spacecraft

This chapter shows that it is possible for a dual, or multi-arm system to have a lower mass than a single arm system and still perform the same task. This analysis will show that, for some tasks, a dual manipulator system is not only lighter, but also provides all the bonuses inherent in a dual arm system and should therefore be considered as a viable option from the outset of the design process.

There are many tasks which can benefit from dual manipulators; this study focuses on transporting large payloads. This task is necessary for larger missions, such as satellite servicing, assembly, and berthing of a spacecraft.

The following example will go through the design methodology as shown in figure 6.1. Previous chapters describe the various components in detail (e.g. mass estimation and force-distribution). This chapter not only demonstrates how all these pieces come together, but also how a *well-designed* dual arm system compares to a single arm system.

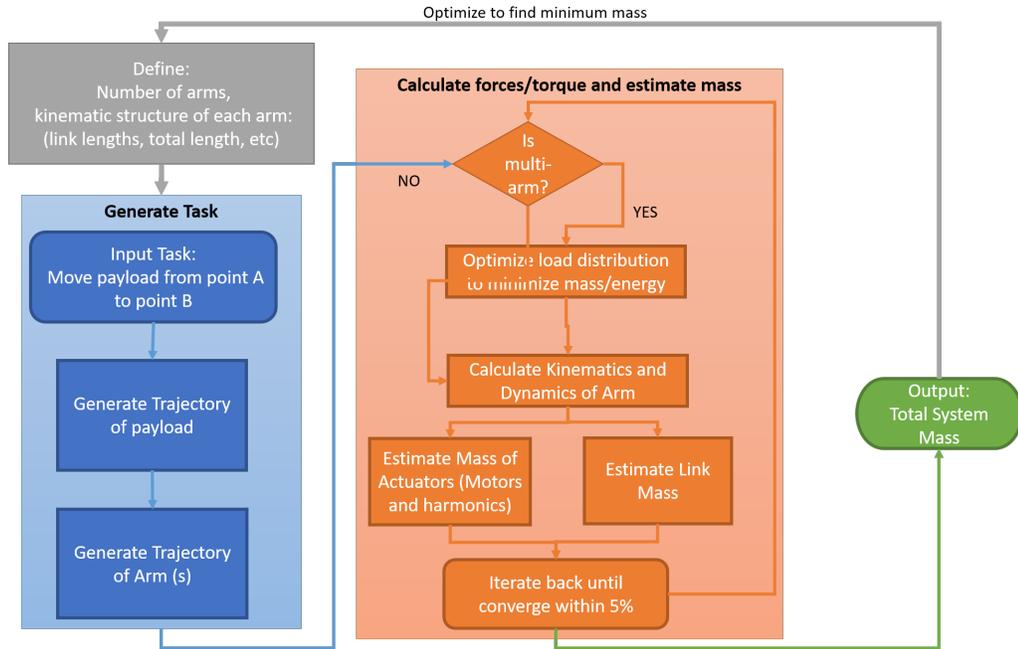


Figure 6.1: Research Methodology Overview

6.1 Task Definition

This is a task-based study; both single and dual arm systems are designed to perform the same task. All systems have the same design requirements, allowing a direct comparison. This methodology demonstrates how a multi-arm system can be more mass efficient than a single manipulator system.

This example uses a docking task; the goal is to dock a large inertial payload, a maneuver common in satellite servicing, assembly, and in berthing spacecraft. The payload ends in a 'docking pose' near the base of the manipulators after it is moved along a desired trajectory. Unlike previous examples, this maneuver requires the manipulators to rotate the payload while it moves along the trajectory.

This broad task definition leads to a number of design variables which can be

broken into three categories: the payload properties, the manipulator parameters, and the trajectory definition.

6.1.1 Payload Properties

In addition to the mass and size, the manipulator(s) must also know how and where to grasp the payload. Rather than requiring specific additional hardware, the manipulator(s) will attach to an existing band/structure on one side of the payload. This structure is common for many satellites and is used to attach to the payload fairing during launch.

The angle of attachment at the start and end of the payload movement along the desired trajectory must also be defined to allow the end-effector to attach/detach from the payload.

Table 6.1: Task Definition: Payload Parameters

Payload Parameters	Example
Payload Dimensions	3m x 3m
Mass	14,500kg
Grasping Location(relative to the payload COM)	Arm 1: Lower left corner, (-1.5m, -1.5m, 0)A
Starting Angle of Attachment	Arm 1: 0° relative to the payload
Ending Angle of Attachment	Arm 1: 0° relative to the payload

6.1.2 Trajectory

A trapezoidal trajectory is used to move the payload from the starting to the ending point. Even this simple trajectory requires acceleration and velocity limits. More complex trajectories, including obstacle avoidance and minimizing base movements, can also be applied here.

The trajectory for each manipulator is found using payload's trajectory and knowledge of the grasping points. Inverse kinematics are used to determine the joint angle along the trajectory.

The motion between joints is minimal between each time-step to ensure that the manipulator moves in a smooth continuous motion.

The parameters, found in table 6.2, are necessary for a simple trapezoidal trajectory.

6.1.3 Manipulator Parameters

Table 6.3 contains the DH parameters for each manipulator used in this study. These parameters are based on the SRMS manipulator; many are estimated from images of the manipulator. These used values can also be found in table 6.4, for those less familiar with DH-parameters. Table 6.4 also contains additional manipulator parameters, such as base location and information necessary to calculate the link masses (e.g. material type and allowable deflection).

Table 6.2: Task Definition: Trajectory Parameters

Trajectory Parameters	Example Values
Starting Location	(4.3, 2.5, 7.4)
End Location	(7, 0, 2)
Starting Angle (about X,Y,Z)	20°, 135°, 20°
End Angle(about X,Y,Z)	0°, 0°, 0°
Maximum Payload Velocity	0.06m/s
Maximum Joint Velocity	20 deg/second
Maximum Acceleration	0.06m/s ²
Joint Limits	None

Defining variables upfront not only allows a designer to compare different designs, but also provides the ability to see how changing these parameters can affect the overall design. This chapter focuses on how increasing the number of manipulators changes the overall design.

6.2 Force and Mass Estimation

In order to use dual manipulators, one must first determine how much force each arm is contributing to the task. The minimum-mass force distribution method introduces internal forces which are reacted through the spacecraft at the payload fairing interface, which is assumed to be able handle the additional strain.

Table 6.3: DH parameters for Docking Maneuver

Link	α_{i-1}	a_{i-1} (rad)	d_i (m)	θ_i
1	0	0	1.8	θ_1
2	$\pi/2$	0	5	θ_2
3	$-\pi/2$	0	5.8	θ_3
4	$\pi/2$	0	1.6	θ_4
5	$-\pi/2$	0	0.5	θ_5
6	$\pi/2$	0	0.5	θ_6

Using the minimal-mass force distribution method, the dynamics of each manipulator are calculated and the joint masses found. The joint masses, combined with the link masses, provide an initial mass estimate for the total system. The link mass is found using static requirements, and can be calculated based on the task definition.

The masses of the joints and links both affect the dynamics. An iterative loop first estimates the mass and then re-calculates the dynamics with these new masses. The new and estimated masses converge after a few iterations. There are a few cases where the mass estimations do not converge; these are near singularities when the estimated torque values are extremely large.

This methodology provides a total system mass for a manipulator and multi-arm system designed to perform a specific task. All systems are designed from the same starting point and use the same mass estimation, providing a true comparison

Table 6.4: Task Definition: Manipulator Parameters

Manipulator Parameters	Example Values
Link Lengths (m)	1.8, 5, 5.8, 1.6, .5, .5
Joint offsets(m)	0, 0, 0, 0, 0, 0
Joint Configurations	pitch,yaw,pitch, yaw pitch, yaw
Base location	Arm1 =(-2,0,0) Arm2 = (2,0,0)
Link Material	Graphite Epoxy
Allowable Deflection per Link Length	1/100
Minimum material thickness	1.587mm (1/16inch)

about how additional manipulators can affect the task.

6.3 Example Results

Two studies directly compare a single arm to a dual arm system. The first study considers how Arms 1 and 2 independently perform the task, while the second study focuses on a cooperative dual arm system. In the second system, the actuators are re-sized to meet the forces/torques required for the dual arm case.

6.3.1 Study 1: Single Arm Systems

This section presents the mass estimation for a single arm performing the given task. Arms 1 and 2 work independently and each have the same kinematic

Starting and Ending Points for Arms 1 and 2 During Docking Maneuver

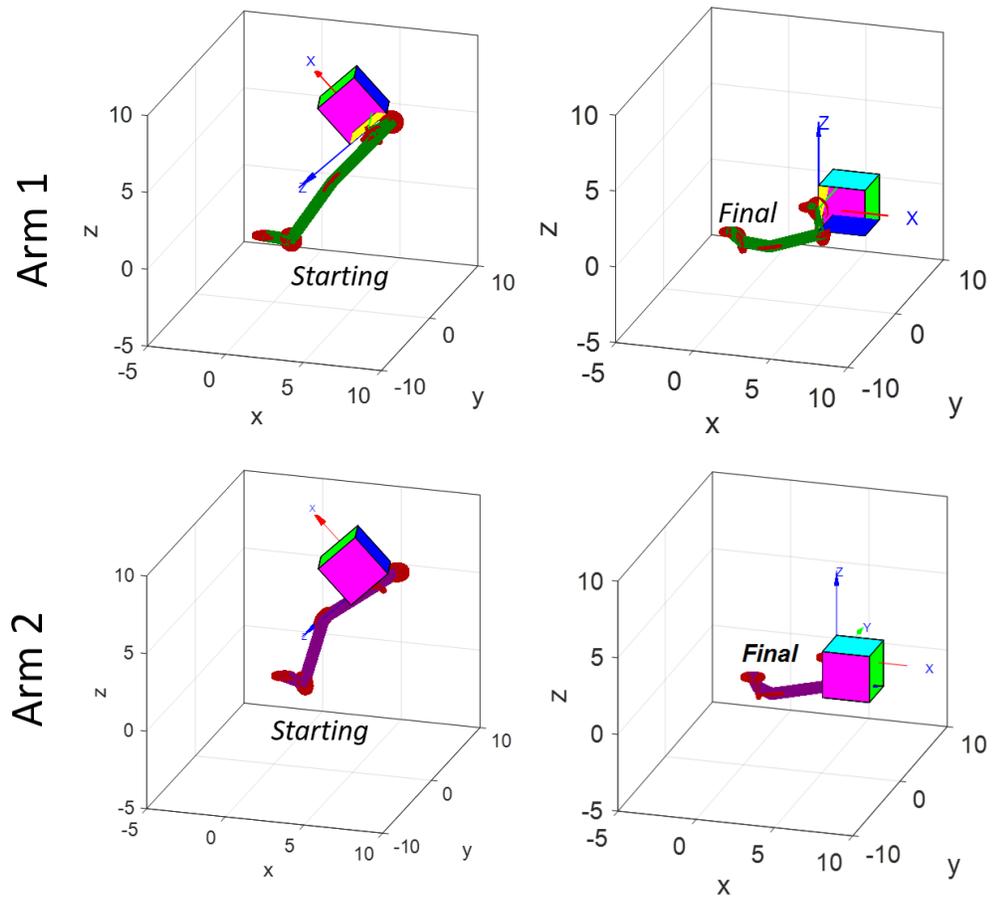


Figure 6.2: Starting and Ending points for Arms 1 and 2

properties; the only differences between them is their location with respect to both the bases and the grasping of the payload. Figure 6.2 shows the starting and ending points for a docking task.

Arm 1 must work harder to perform the same task as Arm 2, see Figure 6.3. Arm 1 has a higher maximum joint torque for joints 1, 2, 4, and 6; therefore it is expected to have a higher joint mass.

The link mass between the Arm 1 and 2 are the same, since both arms have the same kinematic properties and structural requirements. Despite these similarities,

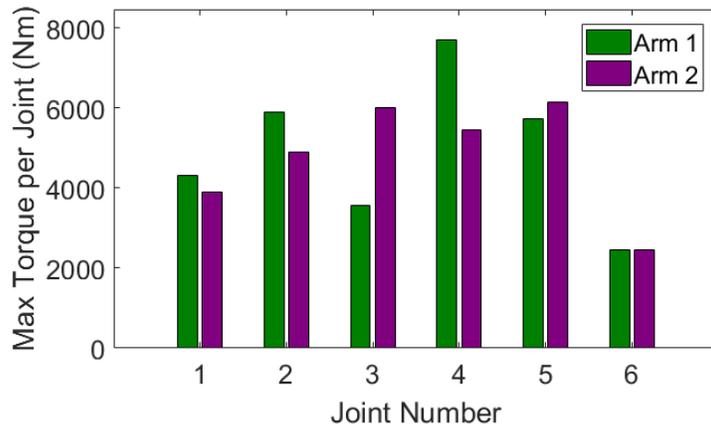


Figure 6.3: Starting and Ending points for Arms 1 and 2

however, the difference in joint torques significantly affects the total joint mass. Figure 6.4 shows that Arm 1 has a higher mass for most of the joints, and thus an overall higher total manipulator mass.

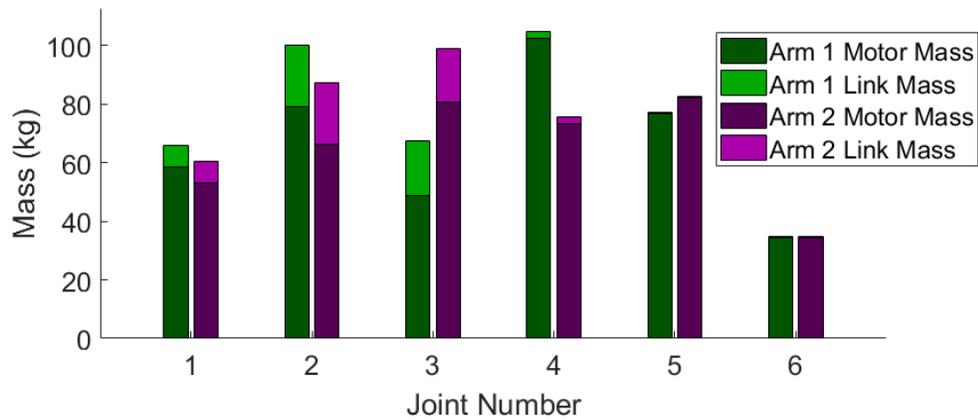


Figure 6.4: Mass for links and joints for single arm docking example

Figure 6.5 shows that Arm 1 has a slightly higher total mass than Arm 2, with a difference of only 10kg. The differences between Arms 1 and 2 are the base locations and where they grasp the payload. Arm 1 grasps the payload's lower left corner, while Arm 2 grasps the lower right corner.

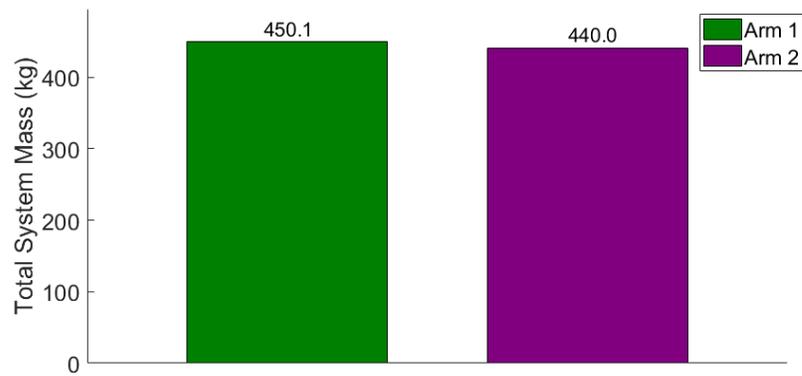


Figure 6.5: Total Mass for Single Arm Docking Example

6.3.2 Study 2: Dual Arm System

The dual arm system design is based on the same task definition as the single arm systems, including the same manipulator parameters for Arms 1 and 2. In the dual arm case, both manipulators cooperatively work together to move the payload.

Dual Arm Trajectory

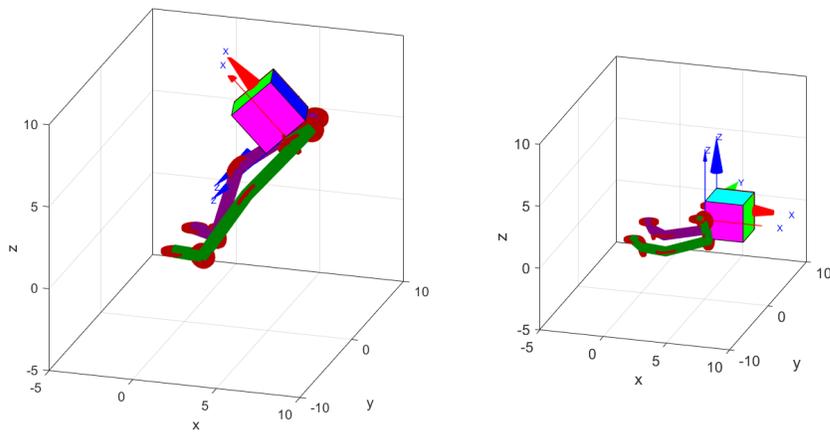


Figure 6.6: Starting and Ending points for Arms 1 and 2

Figure 6.7 shows the maximum joint torques for both the single arm and dual arm studies. In the dual arm case, less torque is required because the arms are working cooperatively. Interestingly, Joint 3 of Arm 1 contributes very little to the movement of the payload, only producing enough torque to overcome the motor friction.

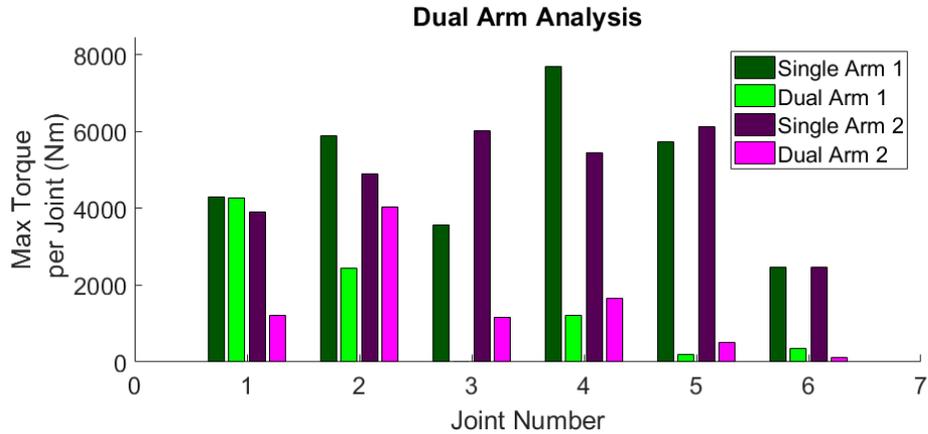


Figure 6.7: Starting and Ending points for Arms 1 and 2

The joint and link masses can be seen in Figure 6.8. The link masses for each arm remain constant, as their design is dictated by the static requirements and the task definition, rather than the dynamics.

Figure 6.9 shows the mass between the three systems: Independent Arm 1, Independent Arm 2, and the cooperative dual arm case. In the dual arm case, the mass of Arm 1 and 2 are reduced to 172kg and 176kg, respectively. The total mass for the dual arm case is 348kg, which is 102kg lower than that of Arm 1 in the independent arm case; a 23% mass savings.

While the introduction of additional manipulators reduces the total joint mass, it also increases the total link mass. Figure 6.10 shows the percentage of the system

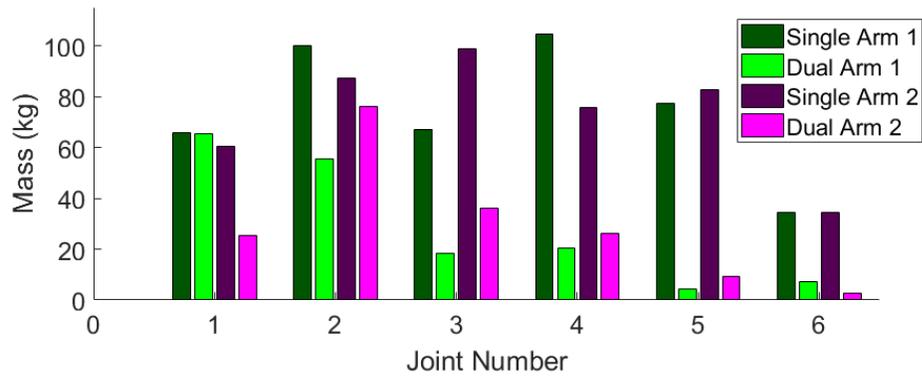


Figure 6.8: Starting and Ending points for Arms 1 and 2

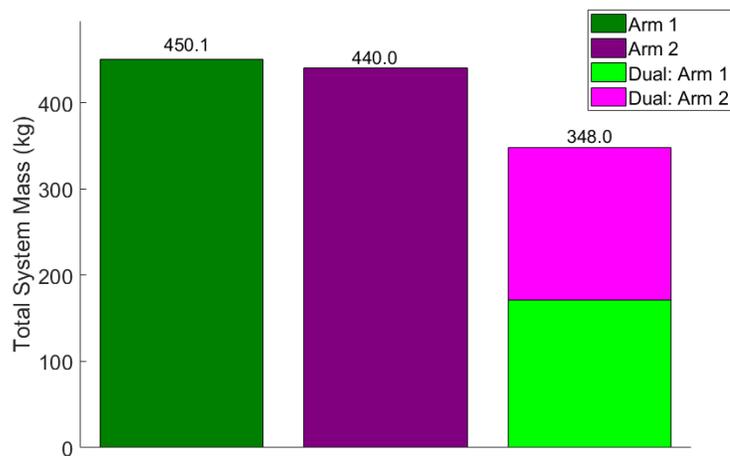


Figure 6.9: Starting and Ending points for Arms 1 and 2

mass made up of the link mass. The mass saving gained, from moving to a dual arm system, originates from the reduction in the joint mass.

This chapter demonstrates that, for a typical docking maneuver, a cooperative dual arm system can be significantly lighter in weight than independently working single arm systems. The total system mass for the dual arm system is calculated to be 348kg, a mass savings of 100kg and 90kg as compared to the independent single systems Arms 1 and 2, respectively.

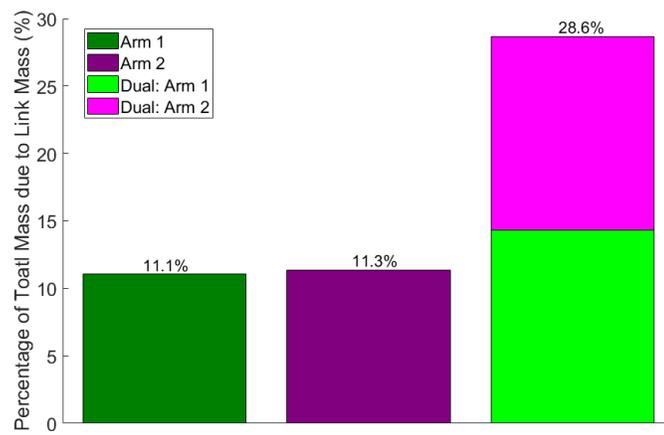


Figure 6.10: Percentage of total system mass due to Link masses

Chapter 7: Further Optimization: Preferred Link Lengths

Previous chapters demonstrate that the inclusion of a second manipulator can greatly reduce the overall system mass. This chapter focuses on methods to further reduce the mass of a dual arm system by optimizing the manipulator parameters. Many researchers have focused on optimizing a single manipulator [62, 68, 69, 70, 81, 109, 110, 111]. This research looks at the differences in optimizing for multi-arm systems compared to single manipulator systems.

Kivela divides the kinematic synthesis of a manipulator into two parts: topology and dimensions [112]. The former synthesis involves the structure of the manipulator (e.g. the number of joints, joint types, etc.). The latter synthesis is devoted to the dimensions associated with the manipulator (e.g. link lengths). This study focuses on the dimensional synthesis, specifically looking at the upper and lower link lengths (i.e. the links associated with joints 2 and 3). These two link lengths have the most variation.

7.1 Background

The genetic algorithm, GA, was first used by Holland in 1975 and is a type of evolutionary algorithm based on Darwin's theory of the Survival of the Fittest. A

GA evolves the variables, or chromosomes, over a number of generations until only the 'strongest' survives. The chromosomes which best solve the problem, i.e. the fittest, produce more "children", passing the best "genes" from one generation to the next[113, 114]. In this study, the "chromosomes" are comprised of the link lengths for the manipulators, and the "strongest" solution is the one which minimizes the total mass of the system.

Unlike other optimization schemes, the GA also includes mutations or random variations. These mutations ideally allow the algorithm to move out of local minima in order to find the global minimum, which results in a better solution.

One advantage of using a GA is that it does not require a smooth fitness function. The fitness function, presented here, is the total mass of the system. While one may expect the total system mass to increase smoothly as the link lengths increase, the joint masses are highly dependent on the joint angles and the trajectory, and will vary widely.

A number of researchers use a GA to optimize single serial manipulators; many focus on optimizing the link lengths and using different objective functions. The manipulability, or dexterity design parameter is a popular indicator of how well a manipulator can perform a given task [62, 68, 110, 115].

Multi-objective GAs (MOGAs) are studied in order to balance various competing performance requirements. Barissi focuses on combining different dexterity and manipulability metrics [109]. Recently, Zeinoun uses a multi-objective GA in order to weigh the manipulability while providing the desired force and velocity at the end-effector [116]. Kivela et al. optimize an existing commercial heavy-duty

hydraulic manipulator in order to reach and operate in multiple locations. Their results conclude that the industrial heavy-duty manipulator is longer than their optimized one (i.e. over-designed) [112].

MOGAs focus on how to combine different performance metrics to produce a manipulator better able to accomplish a given task(s). This paper uses a constrained GA to minimize the mass of the manipulator under the condition that the task(s) must be completed.

7.2 GA Implementation

A GA is implemented using MATLAB's Global Optimization Toolbox[114]. This section goes into details about the specific major variables used to define the GA.

7.2.1 Fitness Function

The fitness function includes a “penalty function” to ensure that the manipulator(s) move the payload along the specified trajectory. If the manipulator(s) are not able to reach desired the task points, then the result are heavily penalized. The penalty function, equation 7.1, uses the pose error and the mass of a non-optimized individual manipulator, see Chapter 6.

$$Mass_{Penalized} = 450kg + 10(PoseError) \quad (7.1)$$

This penalty function is only used when a manipulator is unable to successfully

accomplish the required task; the maximum pose error across the entire trajectory is over 5%. The pose error is the distance between the desired and the actual position divided by the arm length. Dividing by the arm length provides a generic error value, which allows long and short manipulators to be compared without a bias.

The mass of the manipulator(s) are estimated according to the methodology described in previous chapters. An optimization is performed on the independent Arms 1 and 2, as well as the cooperative dual arm system from Chapter 6. This shows how optimizing the individual arms compares to optimizing for a dual arm system.

7.2.2 Variables

The main link lengths, those associated with joint 2 and 3, make up the majority of the manipulator. This study focuses on optimizing those link lengths. The links close to the end-effector are kept small to provide the same precise motion the end-effector may need. For the single arm examples, there are two variables: the upper and lower link lengths. The dual arm example has twice the number of variables because it is changing the parameters of each arm.

The location of the manipulator(s) are also included in some of the studies. For those studies, it is assumed that the manipulator(s) are on a railing, allowing the Y-coordinate of the manipulator's base to be re-positioned before the task begins. For these studies an additional variable, the Y-coordinate of the base location, is added for each manipulator.

	Single Arm	Dual Arm	Dual arm with Base Position
Number of Variables	2	4	6
Population Size	100	200	300
Max Number of Generations	200	400	600
Function Tolerance	10kg	10kg	1kg

Table 7.1: GA Parameters

7.2.3 Initial Population

A population size of 100 is used to provide ample diversity without excessively long run-times for the single arm studies with two variables. A low population of 50 did not provide enough diversity, and the GA solution was highly dependent on the initial population, which is randomly selected. Larger populations, such as 200 or 300, did not provide any additional insight and took significantly longer to run. The initial population is increased as the number of variables are increased.

7.2.4 Bounds and Function Tolerance

The function tolerance is set to 10kg. A number of smaller tolerances were used, however, the results were not significantly different. In addition, the mass estimation is not guaranteed to be accurate down to the single gram. Therefore a function tolerance of 10kg is used to ensure shorter run times. The GA concluded when the function tolerance was met, rather than the maximum number of generations or any other stall conditions. This generally occurred after 21 generations.

A lower function tolerance of 1kg resulted in a run with 33 generations, with no significant difference in the results, before the GA met the function tolerance.

Upper and lower bounds, 0.1m and 35m respectively, are placed on the link lengths. The upper bound of 35m is chosen to ensure the GA explores the entire trade space. An adaptive function ensures that mutations fall within these boundaries.

7.3 Optimize Independent Systems

The upper and lower link lengths are optimized in order to design an independent manipulator with a lower mass, while still performing the task described in Chapter 6. This optimization is done for Independent Arms 1 and 2.

7.3.1 Independent Arm 1 Optimization

Figure 7.1 is the mass estimation for Arm 1 as the upper and lower links lengths are varied. The original design has a mass of 450kg, which is on the lower end of the mass spectrum. This is, in part, due to the severe mass penalty on any arm design that cannot perform the given task (i.e. the docking maneuver). Figure 7.2 shows the pose error for each of the link lengths. This graph shows where the penalty function is used and where mass estimation without a penalty is used. This method of analysis includes many manipulator designs which are unable to reach the task points; the task points lie outside the workspace of the manipulator. These cases are heavily penalized.

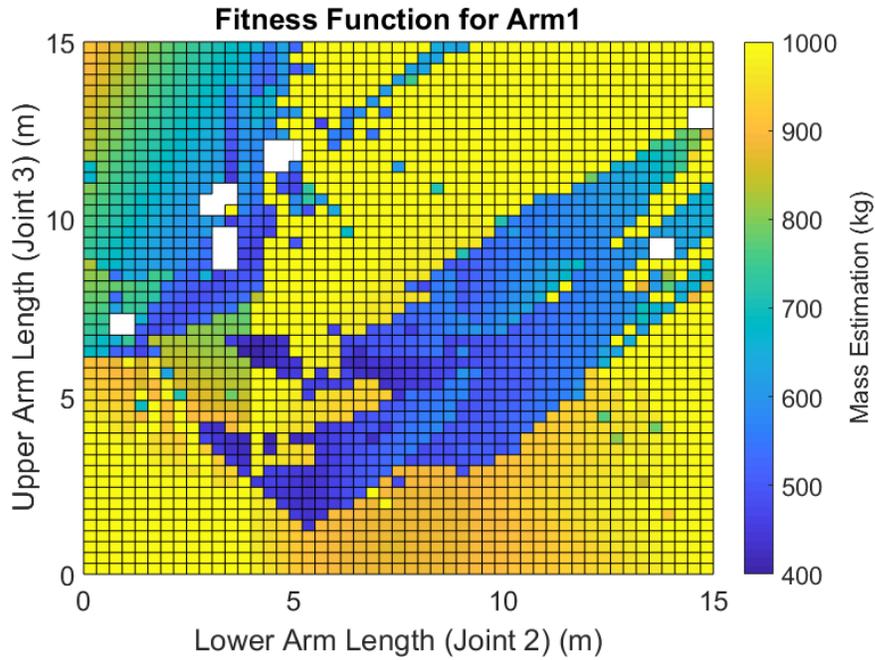


Figure 7.1: Mass Estimation for Arm 1 with a variety of link lengths

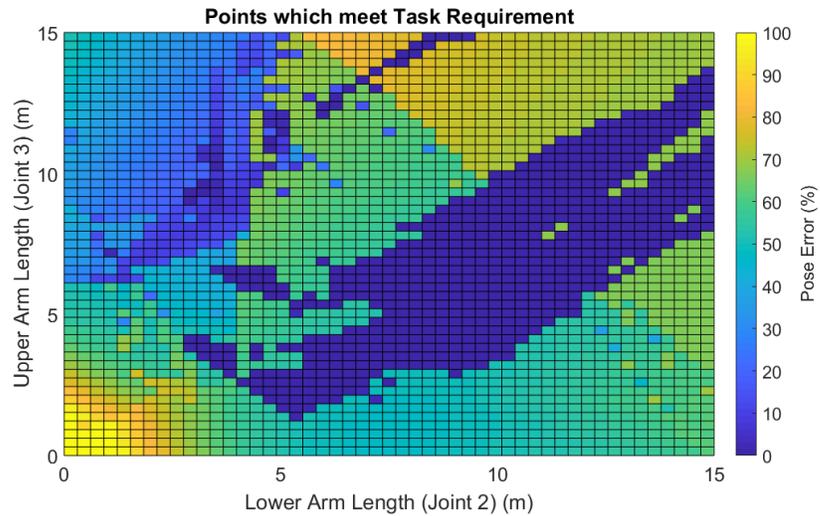


Figure 7.2: Position Error for Arm 1 over a variety of link lengths

Figure 7.1 shows the estimated mass over a large number of lower and upper link lengths. The minimum, 410kg, is found at link lengths of 3.6m and 6.2m. This minima is 40kg lighter than the original arm design. This shows that the total mass

for Arm 1 can be reduced by 6.8% when optimized for this particular task.

This minimum value is found by sampling points across the trade space and calculating the mass at each point. This exhaustive numerical approach is used to demonstrate the nature of the fitness function, which is discontinuous with many local minima. A GA can more efficiently search the trade space, especially when the number of variables increases beyond two.

A GA is applied to the same manipulator. Using an initial population of 100, and running for 21 generations, results in a final mass of 399.8kg. The optimal link lengths for the lower and upper links were found to be 3.6m and 6.1m, see figure 7.3

GA: Arm 1 Performing Docking Task

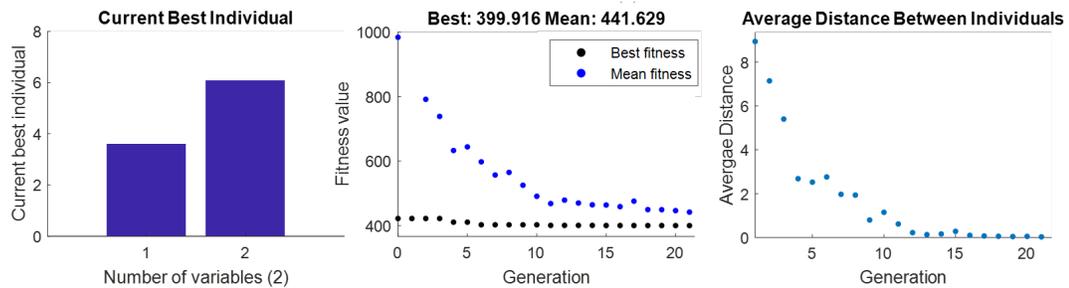


Figure 7.3: Estimated mass of Single Arm1 as a function of link lengths

A GA is dependent on the initial population, which is random. The GA is repeated multiple times in order to see the effect of the initial population on the results. One of the benefits of a GA is that the mutations will explore the trade space, and may result in “unexpected” optimized results which can give the designer a new direction in which to explore.

The GA settles on two primary designs. The first minimum has link lengths similar to those found in the exhaustive numerical approach, though more refined.

Table 7.2: System mass with new link lengths from GA for Arm 1

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6
Lower Link (m)	4.2939	3.6164	3.5963	3.5871	4.2977	3.5945
Upper Link (m)	3.7613	6.0793	6.0854	6.0970	3.7634	6.0922
Estimated Mass (kg)	405	400.2	399.9	399.8	405.2	399.9

Figure 7.4 shows how the optimized joint and link masses compares to the original design.

The second minimum has link lengths of 4.30m and 3.76m. While it is clear that this second design is not the global minima, it is still a significant reduction in mass from the original design. Though this design is heavier, the link lengths are shorter. So while the longer arm design produces the minimum mass, the second design may be the preferred design if there are additional volume restrictions. The GA can reveal different designs and allows the designers to choose from other options. The rest of this chapter focuses on the minimum mass solution.

GA: Arm 1 Performing Docking Task

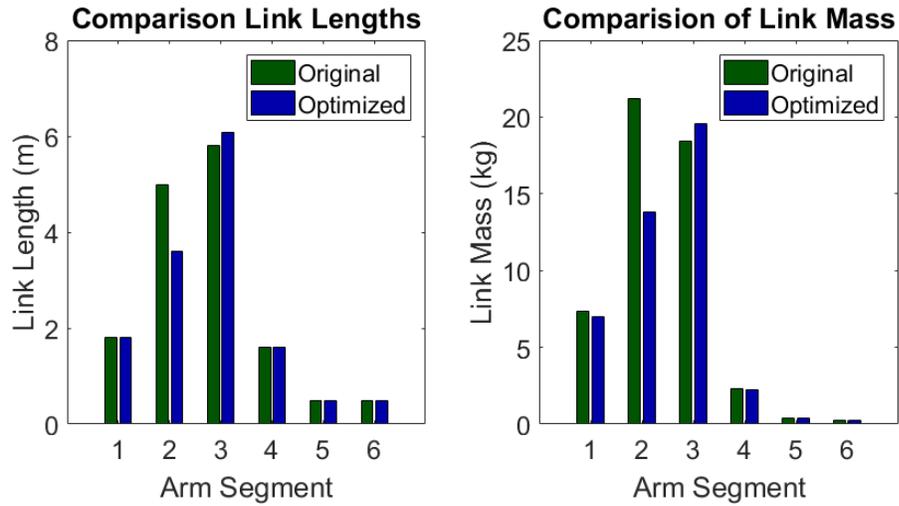


Figure 7.4: (Left) Estimated link mass of Single Arm 1, (Right) Total Mass per joint of optimized and original Arm 1 design

While the joints make up the majority of the total mass, changing the link lengths still affect the kinematics and the dynamics of the system. Figure 7.5 shows the comparison between the joint torques for the optimized Independent Arm 1 and the original design.

GA: Arm 1 Performing Docking Task

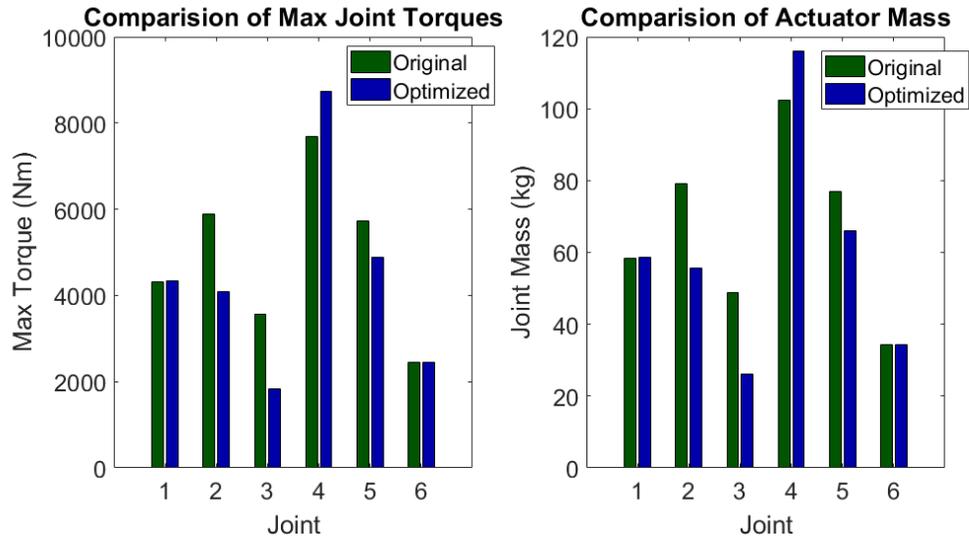


Figure 7.5: (Left)Maximum Torque per joint for Optimized and Original Arm 1 Design (Right)Estimated actuator mass of Single Arm1

7.3.2 Independent Arm 2 Optimization

The same optimization procedure is followed for Independent Arm 2. The mass estimation for Arm 2, originally estimated at 440kg, is reduced to 354kg. Figure 7.6 shows the optimized link lengths and how this affects the mass.

GA: Arm 2 Performing Docking Task

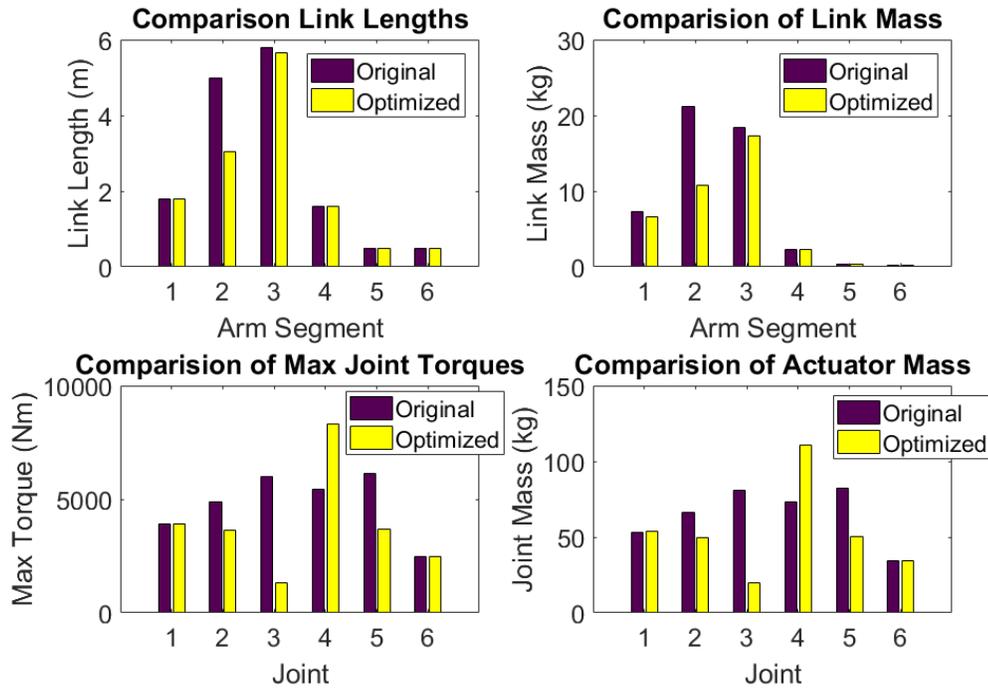


Figure 7.6: Arm 2 Optimization Results

A variety of initial populations are used to determine how they affect the GA. Unlike with Arm 1, the GA converges to the same point for all six trials. The average lower and upper link lengths are 2.96m and 5.63m, with standard deviations of 0.039 and 0.03, respectively. These trials produce an average mass of 354.6kg with a standard deviation of 0.64kg.

The mass of Arm 2 is reduced from an original mass of 440kg to 354kg, a 19% mass reduction. This shows that optimizing the link lengths can significantly reduce the total system mass.

Table 7.3: System mass with new link lengths from GA for Arm 2

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6
Lower Link (m)	3.0309	2.9203	2.9494	2.9576	2.9377	2.9822
Upper Link (m)	5.6812	5.6187	5.6356	5.6402	5.5918	5.6104
Estimated Mass (kg)	355.5670	353.8260	354.2887	354.4232	354.4565	355.2109

Single Manipulators Performing Docking Task

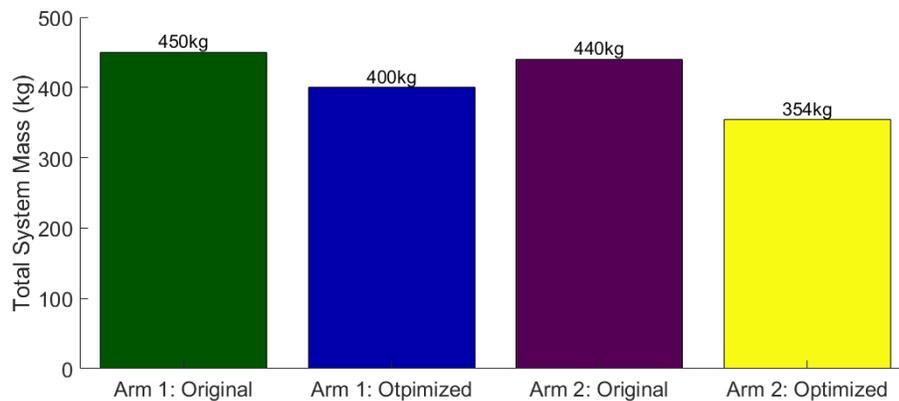


Figure 7.7: Comparison of Optimized and Original Total System Mass for an Independent Single Arm System

7.3.3 Conclusion for Independent Single Arm System Analysis

The link and joint masses change depending on the manipulator parameters, which can be optimized to reduce the overall mass. How these parameters affect the overall mass is highly dependent on the trajectory. One cannot say that reducing the arm lengths will *always* reduce the total mass; it is only guaranteed to reduce the link mass.

Arm 1 and Arm 2 were both optimized to better perform the docking task described in previous chapters. Adjusting the lower and upper link lengths can be used to significantly reduce the mass of each manipulator, see table 7.4.

Table 7.4: Original Single Arm System vs Optimized Single Arm System

Arm Name	Lower Link(m)	Upper Link (m)	Estimated Mass (kg)
Original Arm 1	5	5.8	450
Optimized Arm 1	3.60	6.09	399.8
Original Arm 2	5	5.8	440
Optimized Arm 2	3.03	5.65	354

7.4 Base Location Study: Independent Arm 1

There are only two differences between Arm 1 and Arm 2: the location of the base and where the end-effector grasps the payload. In the original design, there is only a 10kg difference between the mass estimate for Arm 1 and Arm 2. However, Arm 2 was able to see a significant improvement when optimized.

Originally, Arm 1's base is mounted at $(0, -2, 0)$ and Arm 2's base is mounted at $(0, 2, 0)$, both given in the world frame. Arm 2 has a lower mass than Arm 1, which leads to the question: is there a base location that significantly improves the optimization of Arm 1?

The Space Station Remote Manipulator System (SSRMS) is attached to a mobile base which travels along a rail [117]. This system allows the SSRMS to be

relocated in order to better perform a task(s). This adds an additional variable which can be further optimized [66, 118].

A single variable optimization is done on the Y-coordinate of Independent Arm 1's base, with the original link lengths of 5m and 5.8m. The optimized base location is found to be at (0, 0.078, 0), resulting in a new mas of 415kg. This mass saving comes from the large reduction in Joint 4's required maximum torque.

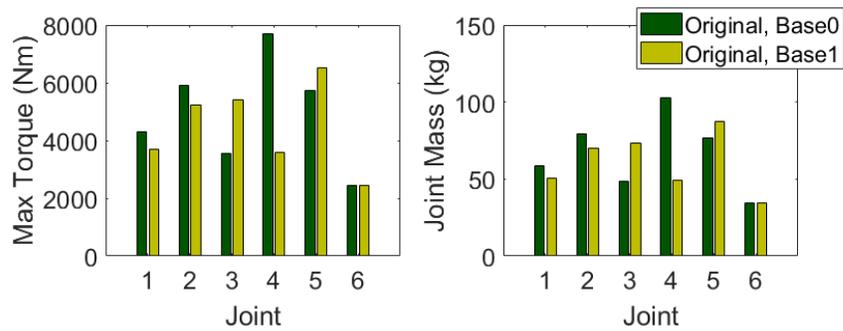


Figure 7.8: Joint Torque and Mass of Arm 1 Design with Base Optimization

A second study optimizes the Y-coordinate of Independent Arm 1's base, in addition to the link lengths. This three variable optimization is done using a GA with an initial population of 150.

Table 7.5: Table of link Length Studies

Study Name	link Length	Base Location	Description
Original, Base0	5m , 5.8m	(0, -2, 0)	Original Design, with original base
Original,Base1	5m , 5.8m	(0, 0.01, 0)	Original link lengths, base found as result of single variable (Y-coordinate of the base location) optimization
Optimize3,Base3	3.13m, 5.89m	(0, 0.38682, 0)	Optimized lengths and location (result of 3 variable optimization)

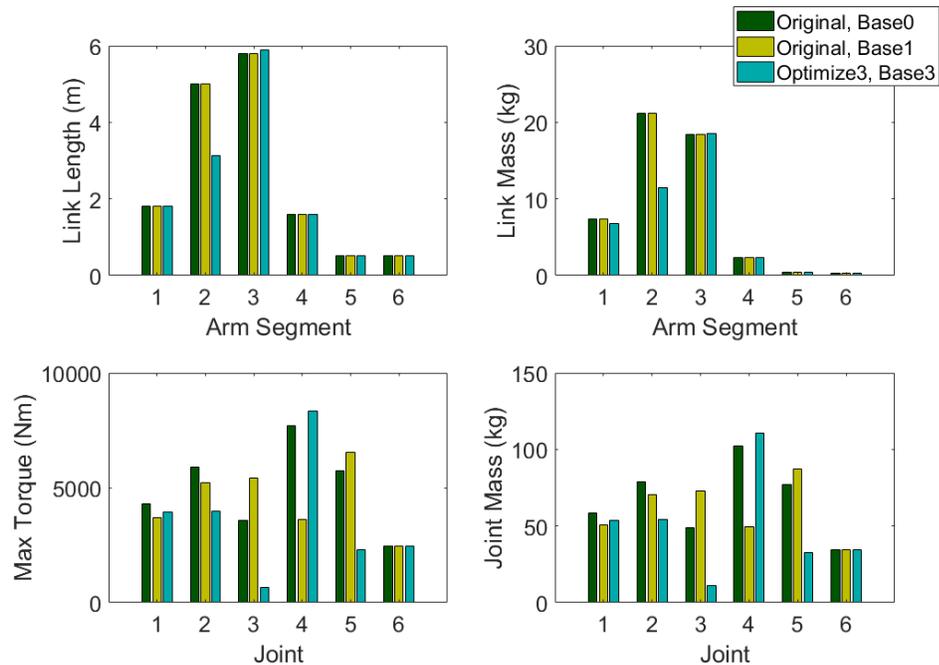


Figure 7.9: Three Variable Optimization Results: Link lengths and Y-Coordinate of Base location for Independent Arm 1

Changing the base location and optimizing the link lengths reduces Arm 1 to 336kg. The optimized link lengths, with the new base location, are 3.13m and 5.89m, for the lower and upper links respectively. The relocation of the manipulator allows for lower joint torques, resulting in a higher mass reduction.

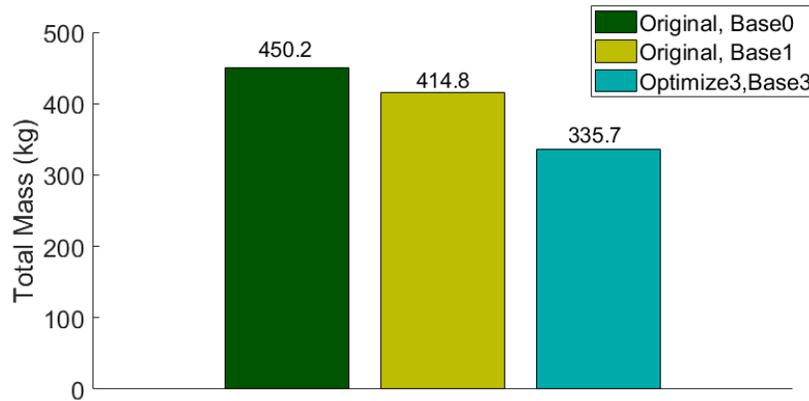


Figure 7.10: Mass Summary of Independent Arm 1 Base Location Optimization

Figure 7.11 shows the optimized Arm 1 at the new base location as compared to the original Arm 1 at the starting and ending points of the trajectory.

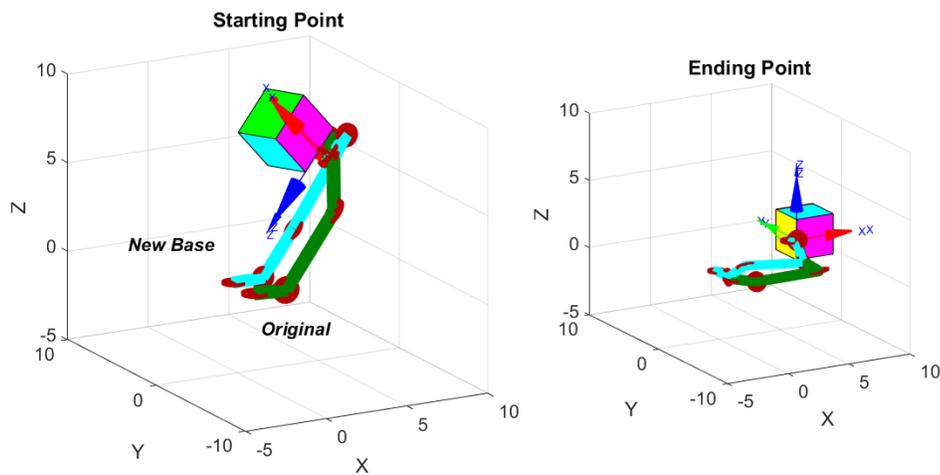


Figure 7.11: Arm 1 with optimized link and new base location

Many times, the base and grasping locations are dictated by available hardware or other restrictions. Optimizing the location, as well as the link lengths, results in significant mass savings (an estimated mass of 336kg). However, changing the base location *alone* will still result in mass savings; the mass of Arm 1 was reduced from 450kg to 415kg.

The more freedom one has in changing task parameters, the more a system can be optimized to better perform the task(s). The dual arm study assumes the original base locations are fixed and unable to be changed.

7.5 Optimization of Multi-Arm Systems

Often, a dual-arm system is created by using pre-existing manipulator designs and affixing them together, as oppose to designing the dual arm system from the ground-up (e.g. utilizing the optimized Independent Arm1 and Independent Arm2 designs and combining those into a dual arm system). When Arm1 and Arm2 are optimized together as part of the dual arm system, the results will be different. This section looks at the difference between a cooperative dual arm system comprised of individually optimized arms and those that have been optimized together.

7.5.1 System Optimization

The procedure for optimizing a dual-arm system is similar to that of the independent arm study. The number of variables double from two to four, as the upper and lower link lengths for each arm are optimized together. The force distribution

used between the dual arms is the minimization-mass method. Six optimization trials were done; Table 7.6 shows the results.

Table 7.6: System mass with new link lengths from GA for Arm 2

		Run 1	Run 2	Run 3	Run 4	Run 5	Run 6
Lower	Arm1	5.121	5.029	4.7936	5.161	5.053	4.7221
Link(m)	Arm2	5.221	3.756	3.0548	3.659	3.708	3.7372
Upper	Arm1	1.518	1.629	1.6087	1.513	1.554	1.6881
Link(m)	Arm2	3.732	3.241	3.3718	3.568	3.59	2.1238
Mass(kg)	Arm1	78.7	88.9	95.6	90.3	87.6	89.8
	Arm2	99.9	77.4	65.8	78.5	79.1	86.7
Total System Mass(kg)		178.67	166.3	161.4	168.9	166.7	176.4

The original dual-arm designs produced a system mass of 330kg, see Chapter 6. The optimized dual arm system reduces this mass to 160kg, a reduction greater than 50%, see Figure 7.12.

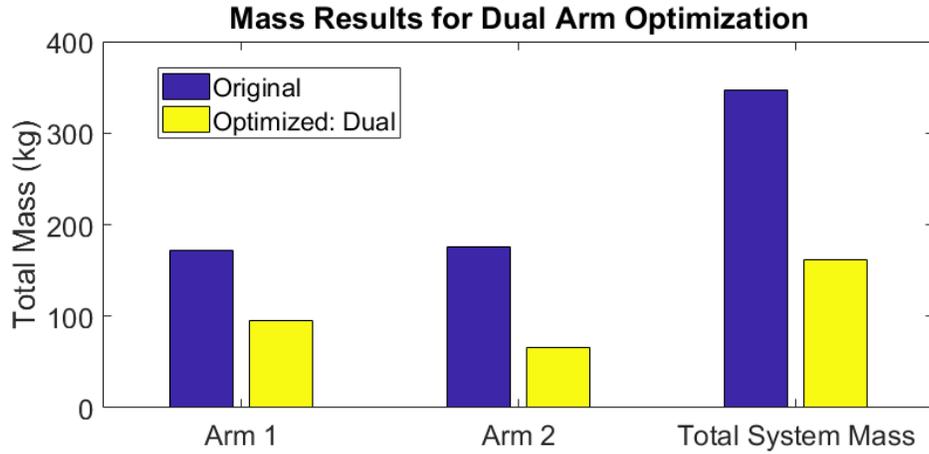


Figure 7.12: Single Arm1 Torque: Original vs Optimized

Figure 7.13 shows the maximum joint torques in the original dual arm system and the newly optimized system. The decrease in joint torques leads to a decrease in joint mass, see Figure 7.14.

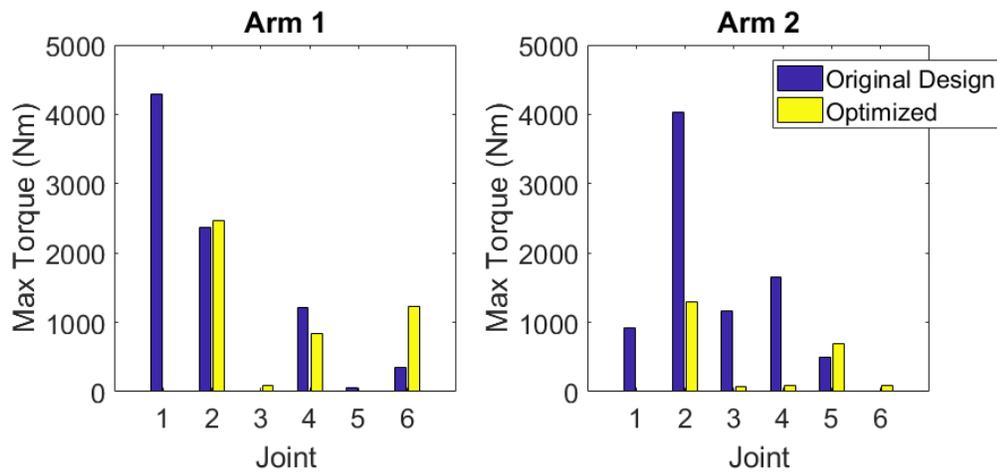


Figure 7.13: Dual Arm Torque: Original vs Optimized

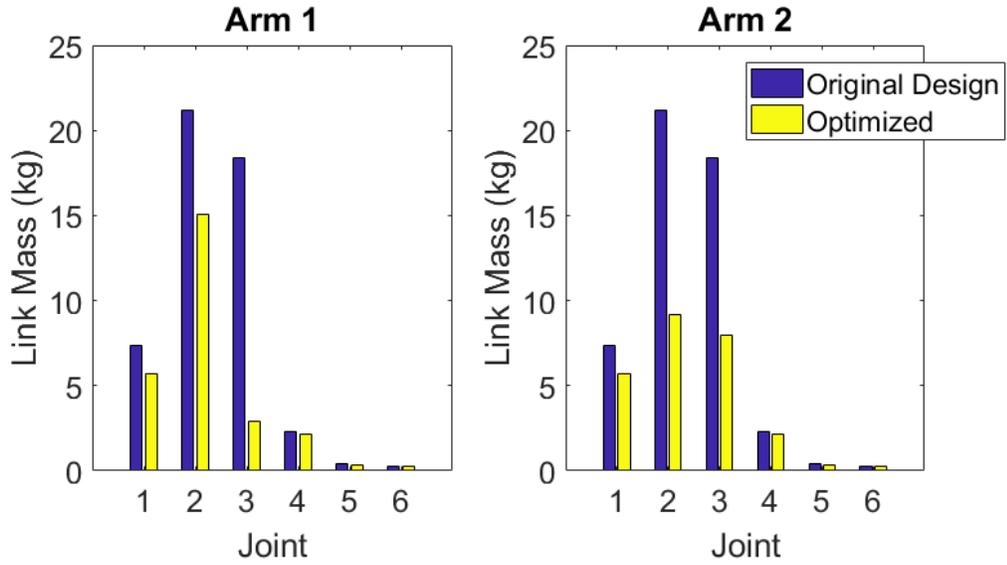


Figure 7.14: Dual Arm Joint Mass: Original vs Optimized

While the lower link length for Arm 1 is approximately the same as the original design, all other link lengths have significantly decreased. Figure 7.15 shows the link mass for each link of each arm.

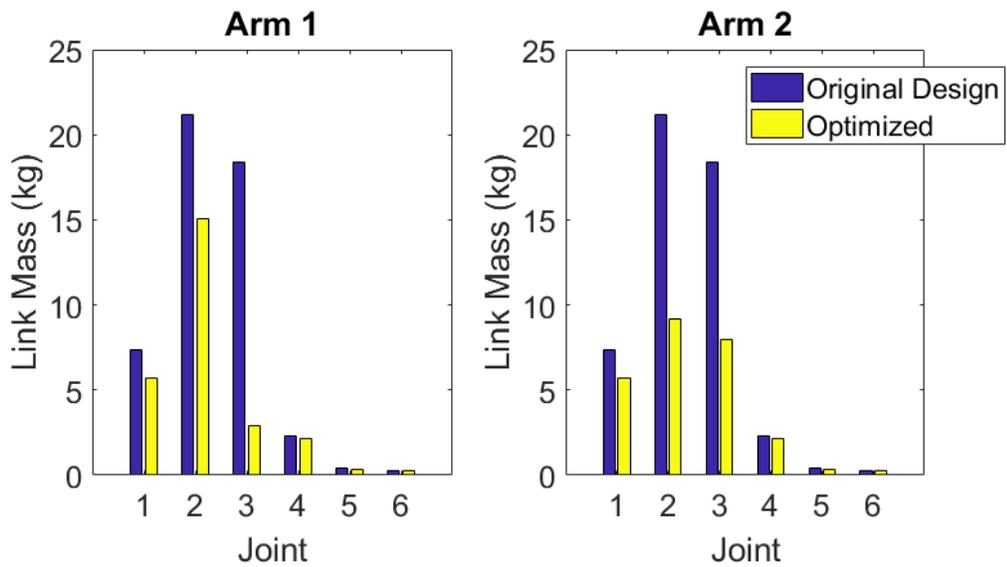


Figure 7.15: Dual Arm Link Mass: Original vs Optimized

7.5.2 Individual Optimize

Table 7.7 contains the optimized link lengths for the Independent and Cooperative dual arm systems. It is clear that optimizing the latter system as a whole results in different link lengths than those found when optimizing the Independent single arm systems.

This study looks at a dual arm system when it is comprised of Independently optimized manipulators. Figure 7.16 shows the mass for the individually optimized dual arm system as compared to the optimized Cooperative dual arm system.

Table 7.7: Summary of Optimization Results

		Independent Arm1	Independent Arm 2	Dual Arm
Lower Link(m)	Arm1	3.6m	-	4.8
	Arm2	-	2.9	1.6
Upper Link(m)	Arm1	6.1m	-	3.1
	Arm2	-	5.6	3.4
Mass(kg)	Arm1	399.9	-	95.6
	Arm2	-	354.6	65.8

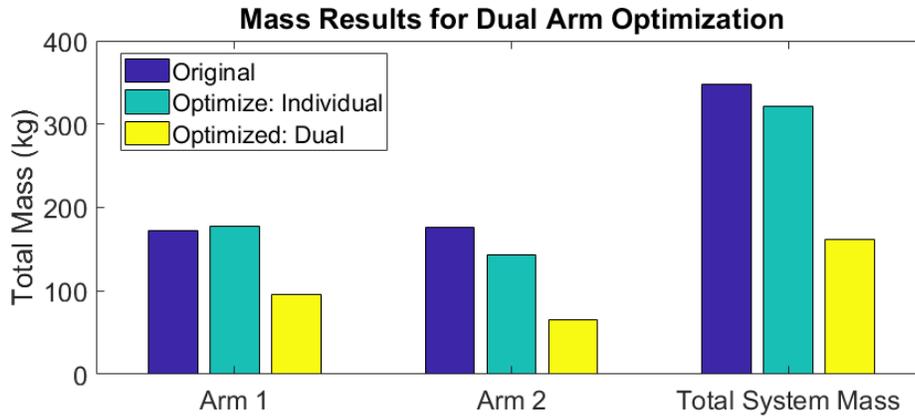


Figure 7.16: Dual Arm Mass: Original vs Optimized

The independently optimized system has a total system mass of 321kg, which is only 27kg lighter than the *original* cooperative dual arm design. The total mass for the optimized cooperative dual arm system is 160kg, significantly lower than the independently optimized system. In this example, the individually optimized results still produce a lighter design than the original; however, this is not guaranteed. Taking existing manipulators and using them together in an cooperative system, although they may each be optimized independently, still results in a sub-optimal Cooperative dual arm system.

7.5.3 Cooperative Dual Arm Study with Changing Base Location

This study optimizes the link lengths and the Y-coordinate of each arm in the dual arm system, for a total of six variables. The tolerance of the GA is reduced in order to ensure that the system mass converges to a minimum value.

The optimal design has a total mass of 160kg, which is the same mass as the previous optimization. This shows that adding more variables to optimize the

system may not significantly decrease the overall mass. Careful selection of variables can keep the GA run-time low.

Table 7.8: Multi-arm Optimization with Base

Arm Number	Lower Link (m)	Upper Link(m)	Base Y-coordinate
1	5.163	1.59	-2.90
2	2.78	4.13	3.761

7.6 Conclusion

The upper and lower links, associated with joints 2 and 3, make up the majority of the total manipulator lengths. Changing the lengths of these links can greatly affect the total mass of the system. These links are varied using a genetic algorithm (GA) to optimize an Independent single and Cooperative dual manipulators for mass reduction. The links further out (links 4-6) are kept small to ensure fine precision motion at the end-effector.

When both manipulators in a dual arm system are optimized together, the system as a whole is optimized, and the total system mass may be significantly reduced. If both manipulators are optimized individually, the system mass will be higher than that of manipulators that are optimized together.

Many multi-arm systems are built by combining two or more pre-existing manipulators together. While this method can create a capable multi-arm system, it can lead to the false impression that multi-arm systems are bigger and heavier than a single arm. Creating dual arm systems from the onset of the task definition allows

for comprehensive optimization of the total system. The arms are optimized together towards the completion of the task(s), resulting in a well-designed cooperative dual-arm system with a significantly lower mass.

Chapter 8: Contributions and Conclusions

8.1 Contributions

This thesis answers the question: can the mass of a multi-arm system be less than that of a single arm system? This paper lays out a task-based methodology that estimates the mass of single and multi-arm systems using the assumption that both systems are designed for the same task. In doing so, it shows that when the joint mass is a significant portion of the total mass, moving to a multi-arm system can result in a lighter total system mass. This research was able to:

- Develop a relationship to estimate the mass of a brushless-DC motor based on the desired torque at a desired speed.
- Develop a force distribution method for a multi-arm system which ensures that the mass is minimized
- Optimize a multi-arm system for space applications
- Minimize the system mass by changing the number of manipulators, the link lengths of each manipulator, and the base positions
- Determine that a multi-arm system can be more mass efficient than a single

manipulator in performing a task

- Optimize a multi-arm system for space applications

Over 100 brushless DC motors, from seven different manufacturers, are analyzed in order to determine motor mass. High-end motors, like the ones found in robotic systems, are used to find a linear heuristic which relates the motor mass to the torque for a given trajectory.

Shifting from an Independent single arm system to a Cooperative multi-arm system requires the force-distribution problem to be solved before the total mass can be estimated. Using this motor mass relationship allows linear programming techniques to be used in order to guarantee a resulting minimum mass system. The mass for a multi-arm system can now be estimated, allowing a direct comparison between the single-and multi-arm systems.

A multi-arm system *can* be lighter than a single arm system. The forces/torques between the multiple manipulators balance and cancel out opposing forces/torques imparted by the additional arms; the manipulators work together, allowing them to have significantly lower joint torques. Thus, multiple lighter manipulators, with lower joint torques, are just as powerful as one heavier arm.

In addition to looking at how the number of manipulators can lower the total system mass, the link lengths and base location are also analyzed. A genetic algorithm is used to optimize these variables for both the single and the dual arm system. Optimizing the link lengths showed a significant reduction in the total system mass. The single arm case showed a mass reduction of 11% in Arm 1 and 19%

in Arm 2.

The dual arm system can be optimized using two different methods. Method 1 optimizes each arm individually. The link lengths of each manipulator are optimized to perform the task as if they were working independently. The second method is to optimize both manipulators together. This method optimizes the *system* rather than the individual components. Optimizing the system as a whole results in a significantly lighter dual arm system, which is almost 50% lighter than the first method and 55% lighter than Arm 2 working independently (even when Arm 2 was optimized to perform the same task).

Multi-arm systems provide many benefits over a single arm system including increased flexibility in mission planning, lower joint torques, and improved handling capabilities. In addition, they can also have a larger workspace than a single arm. There are many benefits that are inherent when working with multi-arm systems. This paper shows that a multi-arm system can also be a lighter-weight option as well. Though they are more complex, multi-arm systems can be incredibly powerful tools.

8.2 Design Insight for Multi-Arm Systems

Mass estimations are done for single and dual arm systems using a task driven study. A docking task, with a large inertial payload, was used to directly compare the two systems. While the performances of each system are not guaranteed for every additional task, this example provides insights into the design of multi-arm

systems.

8.2.1 When can moving to multi-arms reduce the total mass?

Moving from an Independent single arm system to a Cooperative multi-arm system can reduce the total system mass when the system mass is driven by the joint masses. The mass reduction from moving to a multi-arm system results from reducing the joint torques and thus the joint masses. A multi-arm system can only be lighter than a single arm system when the joint masses account for the majority of the total system mass. A multi-arm system will have a higher link mass than a single-arm, assuming the manipulators have the same kinematic structure and thus comparable link masses.

Equation 8.1 is a simplified estimation for the total system mass for a single Independent manipulator. Equation 8.2 is the simplified system mass for an n-arm Cooperative system.

$$SystemMass_{Independent} = LinkMass + JointMass_{Independent} \quad (8.1)$$

$$SystemMass_{Cooperative} = n * LinkMass + \sum_{i=1}^{n \text{ arms}} (JointMasses_{Cooperative}) \quad (8.2)$$

The key is to find when the mass of a cooperative multi-arm system is less than (or equal to) the independent single arm system. This would allow a cooperative system to be used with all the benefits inherent to a multi-arm system, without a mass penalty.

Looking at the single arm design for a given task can give an indication of whether or not moving to a multi-arm system will provide a lower mass option.

$$SystemMass_{Cooperative} \leq SystemMass_{Independent} \quad (8.3)$$

Substituting equation 8.1 and 8.2 into 8.3, the inequality can be written in terms of the joint and link masses. The sum of the joint masses in the cooperative multi-arm system must be greater than zero, otherwise the resulting system would have no motors and joints.

$$0 < \sum_{i=1}^{n \text{ arms}} (JointMasses_{Cooperative}) \leq JointMass_{Independent} - (n - 1)LinkMass$$

$$0 < JointMass_{Independent} - (n - 1)LinkMass$$

$$(n - 1)LinkMass < JointMass_{Independent}$$

$$(8.4)$$

If the inequality in equation 8.5 is found to be true for an Independent single arm, then it is possible that a multi-arm system is a lower mass option. This equation does not *guarantee* that a multi-arm system will be lighter, but it is an indication that it is worth expanding the design study.

$$(n - 1) < JointMass_{Independent}/LinkMass \quad (8.5)$$

For a dual arm system, $n = 2$, the link mass should be less than the joint mass in order to justify moving to a dual arm system. Using the same equation (Equation

8.5 for a triple arm system, $n = 3$, the joint mass of the single arm must be at least twice that of the link mass in order to justify moving to a three arm system. This is a very simplified equation, and it assumes the link mass for all arms are the same. This does not guarantee that the multi-arm system will be lighter, but it does give an indication when moving to multi-arm design may provide some mass saving.

8.2.2 Shorter Arms are Not Always Lighter

Shorter links do not always lead to a lower system mass; shorter links lead to a lower *link* mass. Changing the link lengths will alter the trajectory and thus the required joint torques and the joint masses. It is difficult to say whether or not the new link lengths will increase or decrease the system mass.

An exhaustive numerical study estimates the total mass for Arm 1 over a number of link lengths. While the original Arm 1 design has a total mass of 450kg, with main link lengths of 5m and 5.8m, there are a number of manipulator designs with longer link lengths that can reduce the mass. For example, a design with link lengths of 6.4m and 6.1m results in a total system mass of 430.5kg.

8.2.3 When to Optimize a Multi-arm system

Despite being given the same task and same mounting points, the optimized single arm and multi-arm systems have different final link lengths. Optimizing each arm independently and then affixing them together to form a cooperating multi-arm system may not result in an optimal system, but rather an over-engineered

design. Optimization should be done on the *entire* system after the task has been well-defined.

8.3 Future Work

8.3.1 Multiple Tasks

While this work focused on a single task-based design methodology, it is important to consider how changing the task may impact the systems. For example, it is very important that the same design works well for many tasks, but the system cannot not be optimized for every task. Future design considerations should be expanded to include multiple tasks.

This paper also focuses on tasks that took place in the combined workspace. Tasks which occur outside the combined workspace, which are done by only one arm, should also be considered when optimizing the manipulators.

8.3.2 Trajectory Design

All payload trajectories are calculated assuming a simple trapezoidal trajectory between two points. More complex trajectory designs can be incorporated to minimize spacecraft disturbances or to reduce the joint torques.

Although this study focused on non-redundant manipulators, the methodology still holds true for redundant manipulators. The additional degrees of freedom in redundant systems allows a designer to incorporate more complex trajectories in order to further minimize joint torques. Future studies should include the incorporation

of more complex trajectory designs.

8.3.3 Internal Forces

This research assumes that the payload is capable of handling internal forces. If the payload is more fragile, the internal forces can damage the payload; additional constraints should be applied to the minimum-mass force distribution method to ensure the internal forces do not exceed a specified limit. Future work includes studying how constraining the internal forces affects the mass-minimizing and minimizing-energy force-distribution methods.

8.3.4 Friction

This thesis did not include motor friction in torque estimates. A multi-arm system with more motors will be more affected by friction. Including friction will not only affect the total weight of the system, but can also change the force distribution between the manipulators.

8.3.5 Transferring Payloads

One of the inherent benefits of multi-arm systems is the flexibility they provide in planning. Here, it was assumed that every manipulator in the multi-arm system is handling the payload at all times. However, a multi-arm system could also be used to pass a payload between themselves. Considering how many and when manipulators should be used, as well as where/when they should be grasping and transferring

payloads is part of the flexibility multi-arm systems provide.

8.3.6 Further Optimization

Many of the task parameters can be defined as a constraint or range of values rather than a single value. For example, the grasping locations on the payload may not be a single point, but may instead lie around a ring, or on a hand rail, or an entire side of the payload. In addition, the angle of attachment may be a range of angles rather than a single pre-specified one. There are many different variables which can be optimized to design a better multi-arm system.

The Y-coordinate of the base location was added to the optimization, in addition to looking at the major link lengths. Adding this additional variable was able to reduce the mass for the single arm system. However, adding this variable to the multi-arm system did not change the total optimized system mass.

The addition of different variables to the optimization should be considered carefully, as some variables will have more of an effect than others, and the variables will interact with each other.

8.4 Conclusion

This thesis shows that switching to a multi-arm system can reduce the total system mass when designed properly. While multi-arm systems are inherently more complex than single arm systems, that complexity does not necessarily translate to mass. In mass restricted systems, where the total mass is driven by the

joints, switching to a multi-arm system should be considered as a method of further reducing the mass.

Appendix : Kollmorgen Motor Curves

Kollmorgen Motor RBE Family: 410 Series

	Torque (Ncm)		
RPM	RBE_0410	RBE_0411	RBE_0412
-----	-----	-----	-----
0	0.01	0.01	0.01
500	0.0095	0.0095	0.0095
1000	0.009	0.009	0.009
1500	0.008	0.008	0.008
2000	0.0065	0.0065	0.0065
2500	0.0051	0.0051	0.0051
3000	0.004	0.004	0.004
3500	0.0015	0.0015	0.0015
4000	1e-06	1e-06	1e-06

Kollmorgen Motor RBE Family: 510 Series

RPM	Torque (Ncm)			
	RBE_0510	RBE_0511	RBE_0512	RBE_0513
1	0.025	0.05	0.072	0.088
2500	0.023	0.048	0.068	0.0855
5000	0.022	0.047	0.065	0.08
10000	0.02	0.041	0.055	0.065
15000	0.017	0.034	0.042	0.042
17500	0.016	0.03	0.03	0.015
20000	0.014	0.021	0.025	1e-06
25000	0.01	1e-06	1e-06	1e-06
30000	0.004	1e-06	1e-06	1e-06
32500	1e-06	1e-06	1e-06	1e-06

Kollmorgen Motor RBE Family: 710 Series

	Torque (Ncm)				
RPM	RBE_0710	RBE_0711	RBE_0712	RBE_0713	RBE_0714
-----	-----	-----	-----	-----	-----
1	0.055	0.105	0.15	0.195	0.25
2500	0.053	0.1	0.145	0.185	0.235
5000	0.05	0.0975	0.14	0.175	0.22
10000	0.0475	0.085	0.115	0.14	0.165
15000	0.04	0.0675	0.07	0.07	0.03
17500	0.038	0.055	0.05	1e-06	1e-06
20000	0.031	0.031	1e-06	1e-06	1e-06
22500	0.0275	1e-06	1e-06	1e-06	1e-06
25000	0.02	1e-06	1e-06	1e-06	1e-06
28000	1e-06	1e-06	1e-06	1e-06	1e-06

Kollmorgen Motor RBE Family: 1210 Series

RPM	Torque (Ncm)					
RBE_1210	RBE_1211	RBE_1212	RBE_1213	RBE_1214	RBE_1215	
0	0.12	0.22	0.32	0.4	0.49	0.65
2000	0.11	0.215	0.3	0.37	0.45	0.59
4000	0.105	0.2	0.27	0.34	0.38	0.45
6000	0.1	0.19	0.25	0.27	0.27	0.23
8000	0.095	0.17	0.2	0.2	0.1	1e-06
10000	0.08	0.13	0.11	0.04	1e-06	1e-06
12000	0.075	0.08	1e-06	1e-06	1e-06	1e-06
14000	0.06	1e-06	1e-06	1e-06	1e-06	1e-06
16000	0.05	1e-06	1e-06	1e-06	1e-06	1e-06
18000	0.02	1e-06	1e-06	1e-06	1e-06	1e-06
20000	1e-06	1e-06	1e-06	1e-06	1e-06	1e-06

Kollmorgen Motor RBE Family: 1510 Series

RPM	Torque (Ncm)						
	RBE_1510	RBE_1511	RBE_1512	RBE_1513	RBE_1514	RBE_1515	RBE_1516
0	0.19	0.34	0.49	0.62	0.78	0.88	1.06
1000	0.18	0.33	0.46	0.59	0.72	0.82	0.96
2000	0.17	0.32	0.44	0.57	0.66	0.76	0.86
3000	0.16	0.3	0.4	0.48	0.56	0.62	0.66
4000	0.15	0.28	0.35	0.4	0.44	0.45	0.36
5000	0.14	0.24	0.3	0.3	0.14	1e-05	1e-05
6000	0.12	0.2	0.2	0.05	1e-05	1e-05	1e-05
8000	0.1	0.02	1e-06	1e-06	1e-06	1e-06	1e-06
10000	0.06	1e-06	1e-06	1e-06	1e-06	1e-06	0
12000	1e-06	1e-06	1e-06	1e-06	1e-06	1e-06	0

Kollmorgen Motor RBE Family: 1810 Series

RPM	Torque (Ncm)					
	RBE_1810	RBE_1811	RBE_1812	RBE_1813	RBE_1814	RBE_1815
0	0.43	0.83	1.2	1.52	1.85	2.15
1000	0.42	0.8	1.15	1.44	1.72	2
2000	0.4	0.76	1.08	1.34	1.64	1.8
3000	0.38	0.72	1	1.18	1.32	1.48
4000	0.36	0.64	0.85	0.88	0.88	0.88
6000	0.3	0.44	0.3	1e-06	1e-06	1e-06
7000	0.28	0.24	1e-06	1e-06	1e-06	1e-06
8000	0.22	1e-06	1e-06	1e-06	1e-06	1e-06
10000	0.05	1e-06	1e-06	1e-06	1e-06	1e-06
12000	1e-06	1e-06	1e-06	1e-06	1e-06	1e-06

Kollmorgen Motor RBE Family: 2110 Series

RPM	Torque (Nm)					
	RBE_2110	RBE_2111	RBE_2112	RBE_2113	RBE_2114	RBE_2115
0	0.7	1.7	2.4	2.9	3.6	4.5
1000	0.68	1.6	2.3	2.8	3.3	4
2000	0.65	1.5	2.1	2.5	2.8	3.5
3000	0.6	1.35	1.75	2	2.05	2.5

4000	0.58	1.2	1.25	1.25	0.75	0.0001
4500	0.55	1	0.75	0.05	0.0001	0.0001
5000	0.45	0.85	0.4	0.0001	0.0001	0.0001
6000	0.4	0.5	0.0001	0.0001	0.0001	0.0001
7000	0.25	0.0001	0.0001	0.0001	0.0001	0.0001
8000	0.05	0.0001	0.0001	0.0001	0.0001	0.0001
9000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001

Kollmorgen Motor RBE Family: 3010 Series

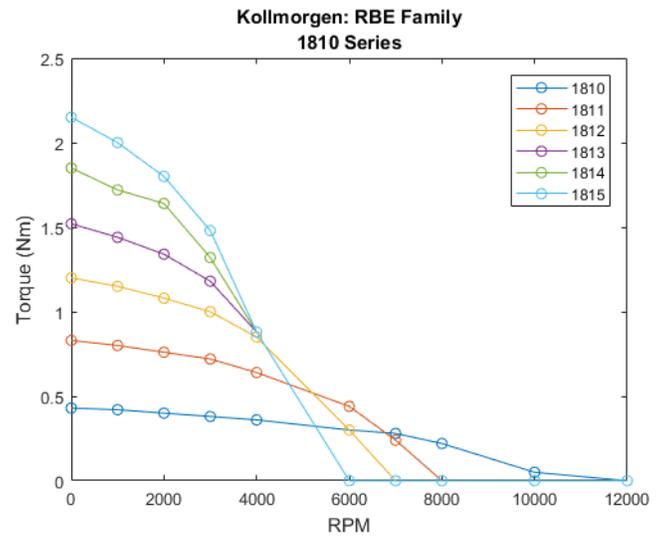
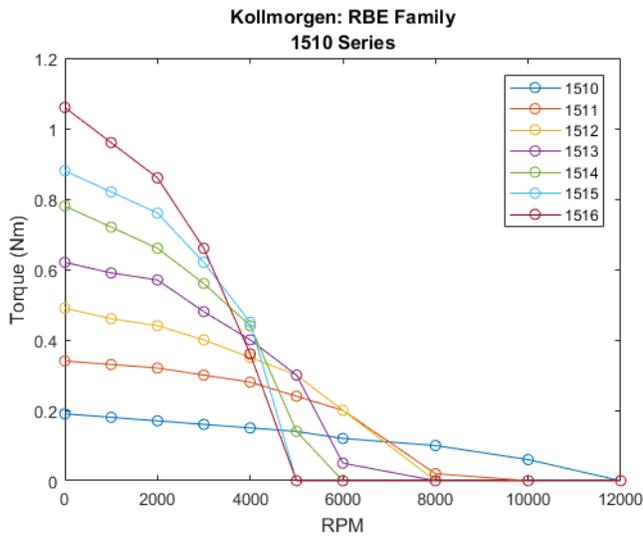
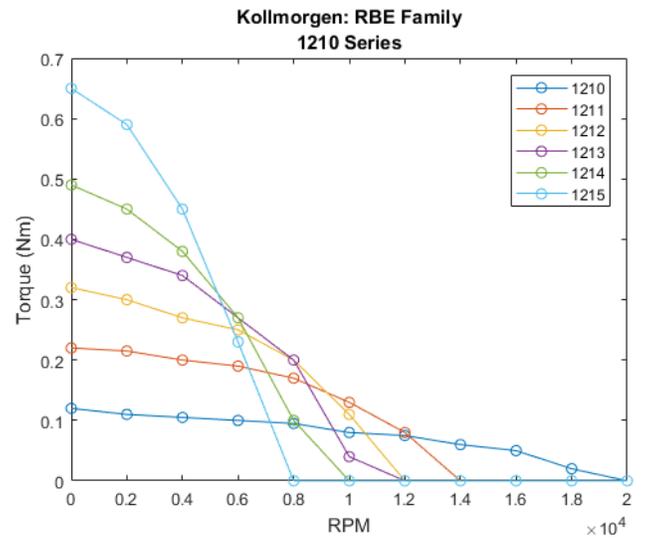
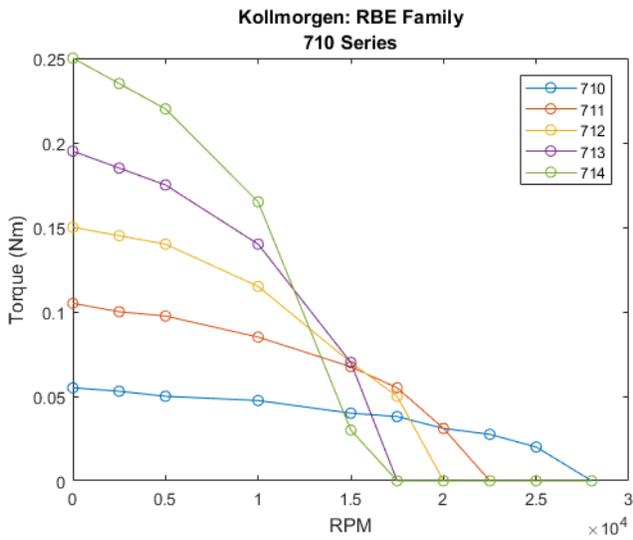
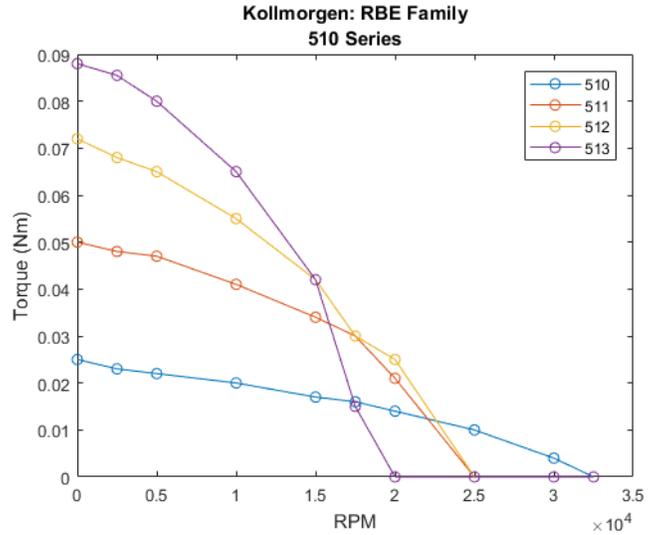
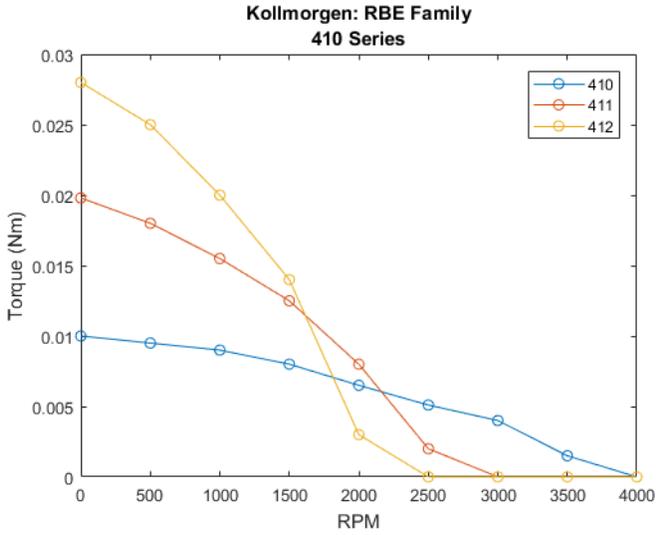
RPM	Torque (Nm)						
	RBE_3010	RBE_3011	RBE_3012	RBE_3013	RBE_3014	RBE_3015	RBE_3016
0	3.2	5.2	7.8	9.8	11.2	13.5	15.2
500	3.1	5	7.5	8.6	10.8	12.5	14
1000	3	4.9	7	8.5	9.5	11	12
1500	2.9	4.5	6.2	7.3	8	8.3	9.2
1750	2.85	4.25	5.6	6.25	6.25	6.25	6.2
2000	2.6	4	5.3	5.5	5.1	3.8	0.0001
2250	2.5	3.5	3.7	4.2	3	0.0001	0.0001
2500	2.4	2.4	3.1	2	2	0.0001	0.0001
3000	2.1	2.2	0.0001	0.0001	0.0001	0.0001	0.0001
3500	1.8	1	0.0001	0.0001	0.0001	0.0001	0.0001
4000	1.3	0.0001	0.0001	0.0001	0.0001	1e-05	0.0001
4500	0.8	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
5000	0.0001	0.0001	0.0001	0.0001	0.0001	0	0.0001

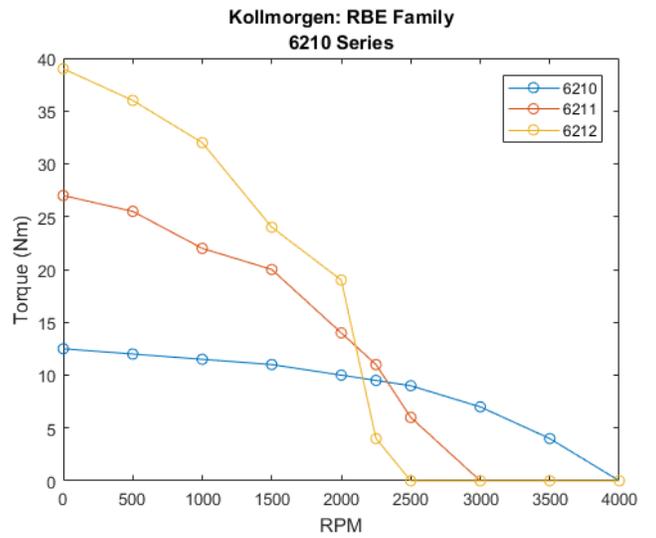
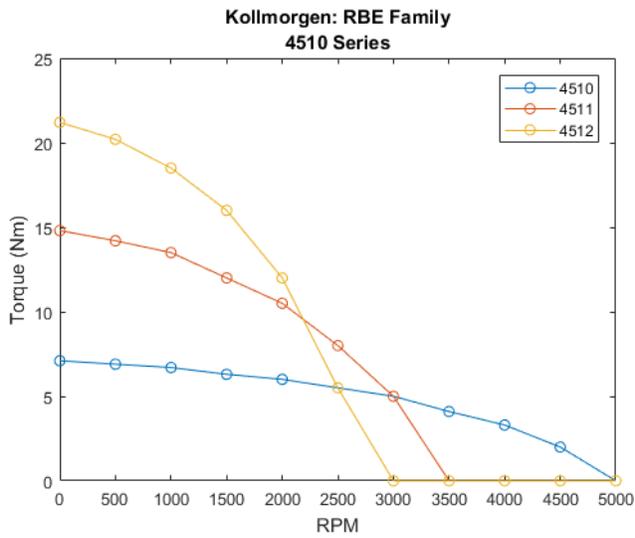
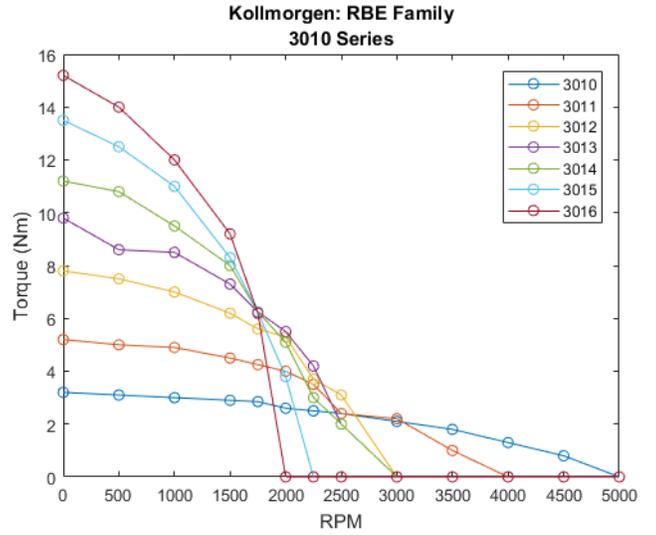
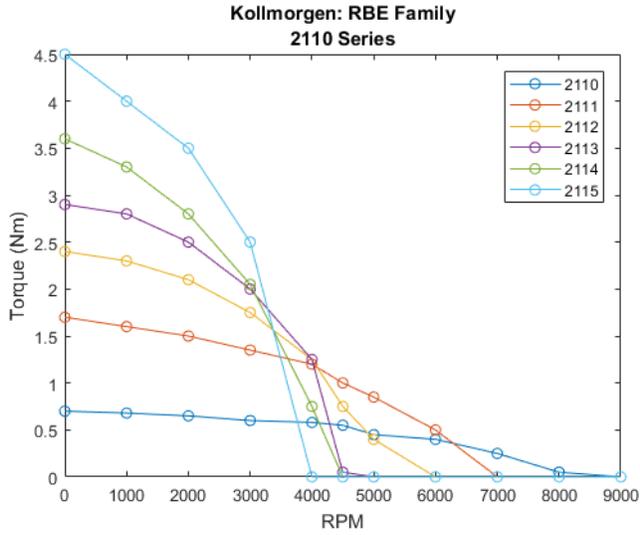
Kollmorgen Motor RBE Family: 4510 Series

Torque (Nm)			
RPM	RBE_4510	RBE_4511	RBE_4512
0	7.1	14.8	21.2
500	6.9	14.2	20.2
1000	6.7	13.5	18.5
1500	6.3	12	16
2000	6	10.5	12
2500	5.5	8	5.5
3000	5	5	0.0001
3500	4.1	0.0001	0.0001
4000	3.3	0.0001	0.0001
4500	2	0.0001	0.0001
5000	0.0001	0.0001	0.0001

Kollmorgen Motor RBE Family: 6210 Series

	Torque (Nm)		
RPM	RBE_6210	RBE_6211	RBE_6212
----	-----	-----	-----
0	12.5	27	39
500	12	25.5	36
1000	11.5	22	32
1500	11	20	24
2000	10	14	19
2250	9.5	11	4
2500	9	6	0.0001
3000	7	0.0001	0
3500	4	0.001	0
4000	0.0001	0.0001	0



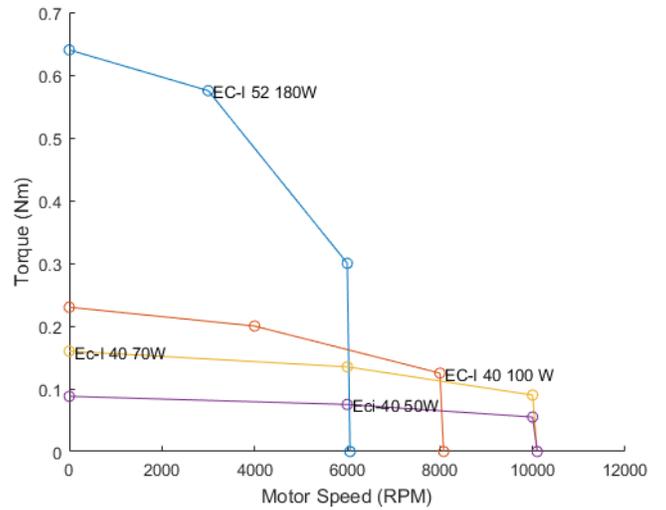
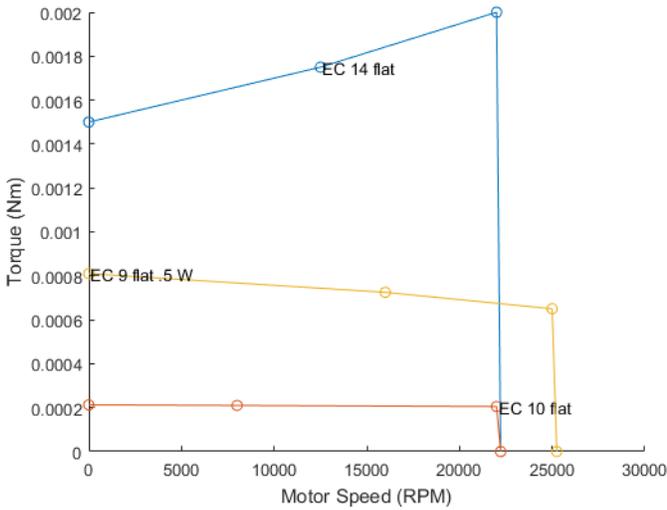
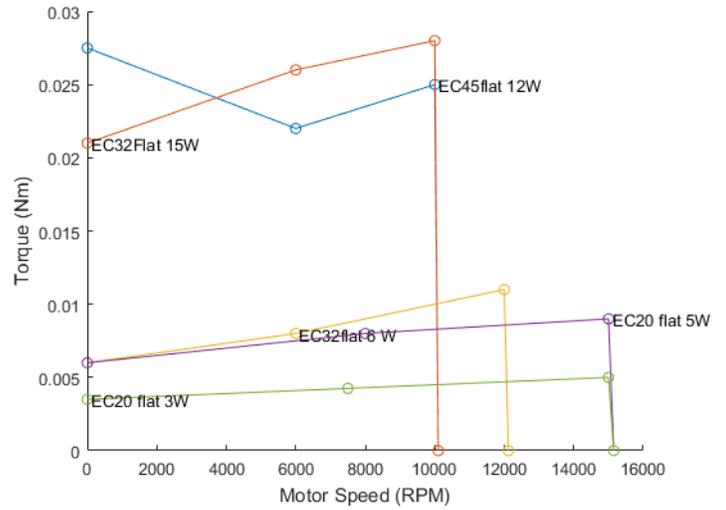
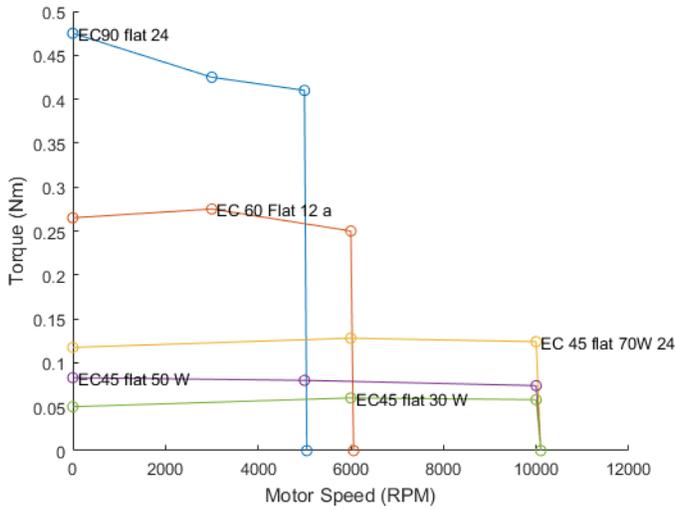


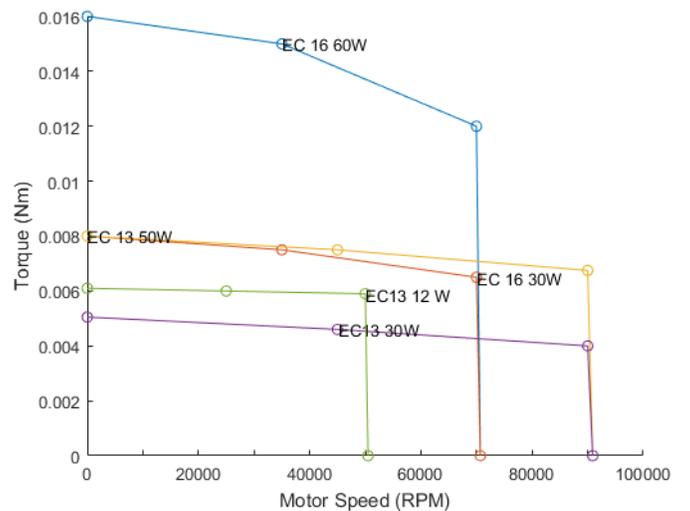
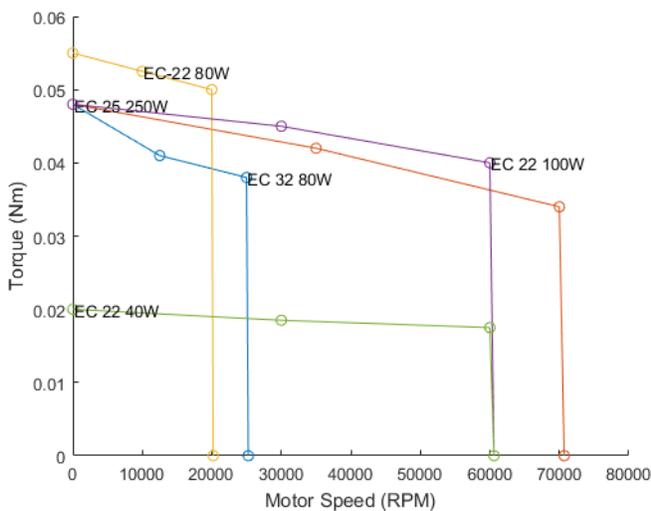
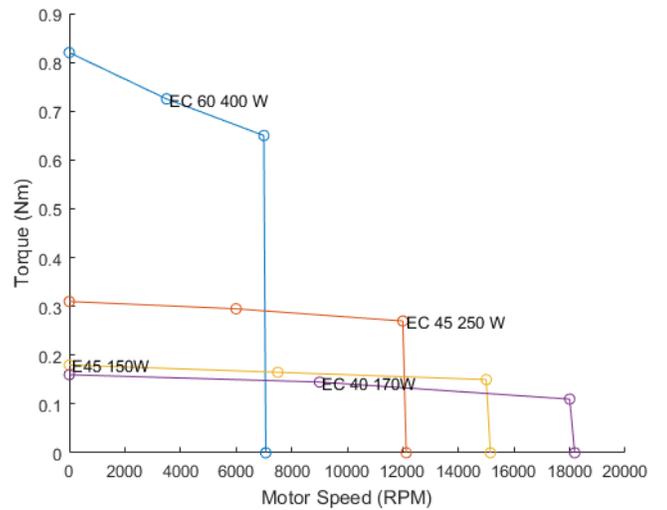
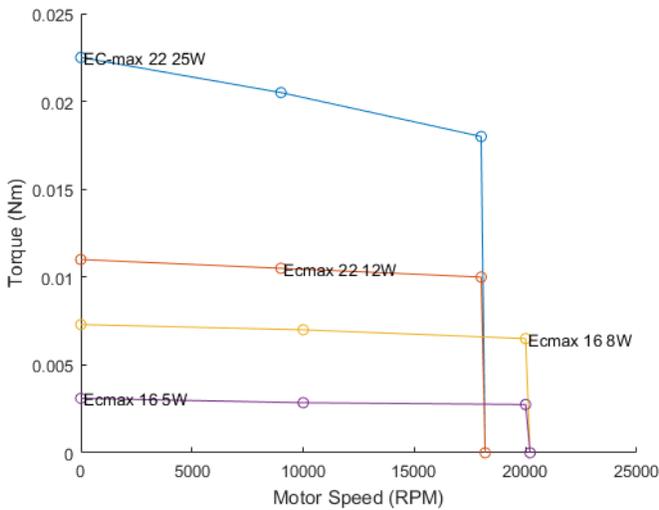
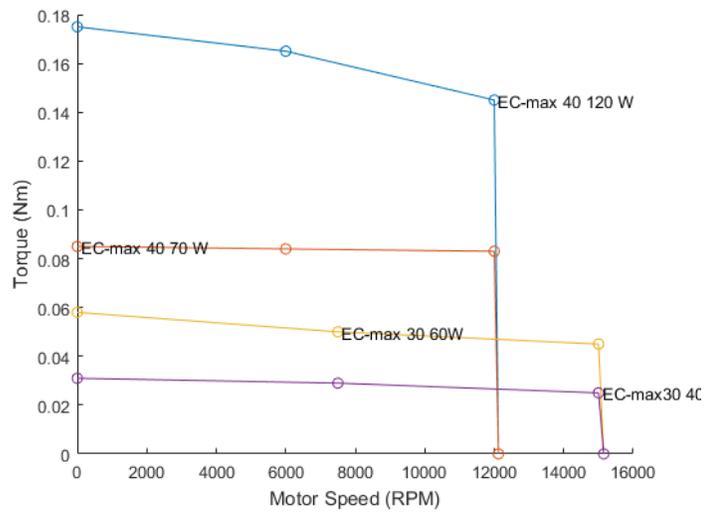
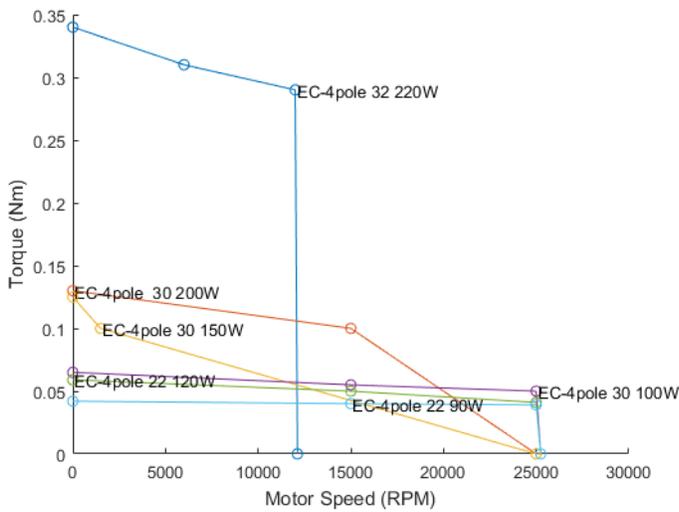
Series	Mass (kg)						
'0410'	0.0864	'0510'	0.108	'0710'	0.221	'1210'	0.32
'0411'	0.1116	'0511'	0.126	'0711'	0.265	'1211'	0.402
'0412'	0.1386	'0512'	0.145	'0712'	0.304	'1212'	0.477
		'0513'	0.165	'0713'	0.344	'1213'	0.552
				'0714'	0.391	'1214'	0.641
						'1215'	0.85

Series	Mass (kg)						
'1510'	0.539	'1810'	0.85	'2110'	0.907	'3010'	3.43
'1511'	0.665	'1811'	1.08	'2111'	1.46	'3011'	4.6
'1512'	0.78	'1812'	1.3	'2112'	2	'3012'	5.9
'1513'	0.895	'1813'	1.52	'2113'	2.5	'3013'	7.05
'1514'	1.03	'1814'	1.73	'2114'	3.04	'3014'	8.1
'1515'	1.13	'1815'	1.96	'2115'	3.54	'3015'	9.5
'1516'	1.38					'3016'	10.6

Series	Mass (kg)	Series	Mass (kg)
'4510'	6.35	'6210'	10.1
'4511'	8.39	'6211'	13.5
'4512'	10.4	'6212'	16.9

Appendix : Maxon Motor Curves

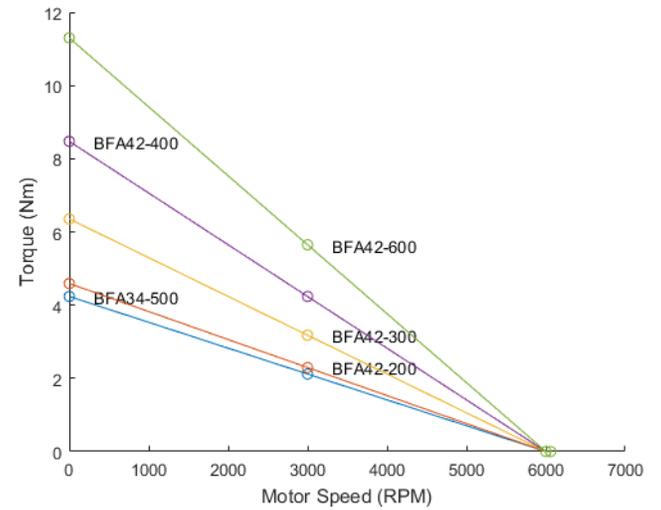
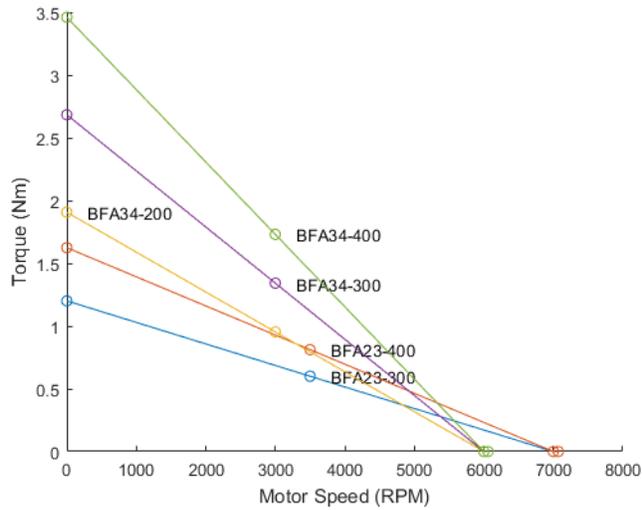
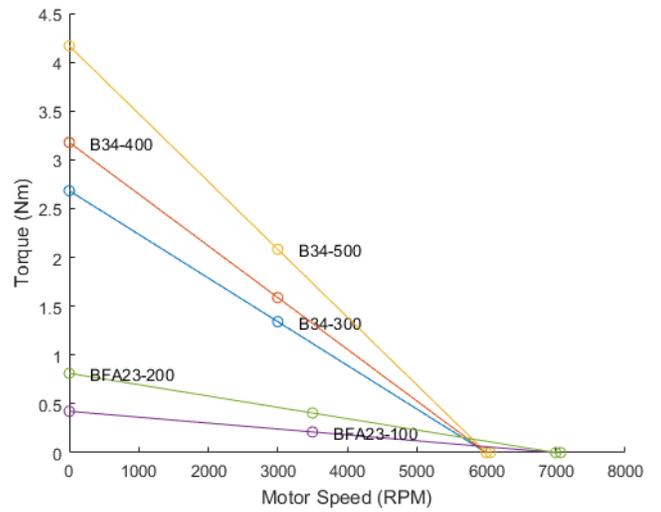
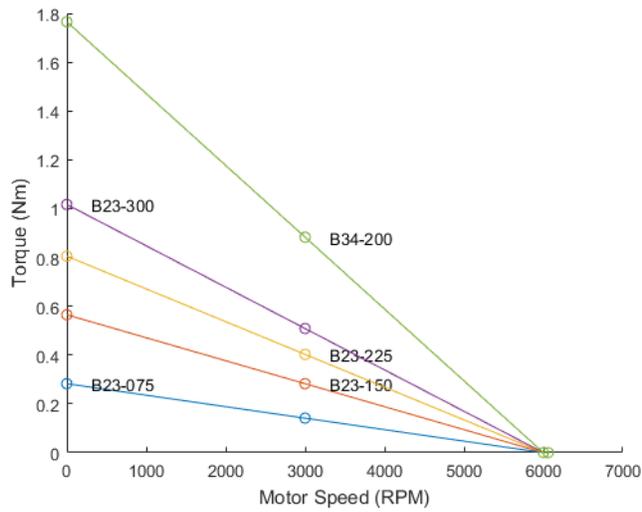




Series	Motor Weight (g)	Lower RPM	mid RPM	max RPM	Torque (Nm) at low RPM	Torque (Nm) at mid RPM	Torque (Nm) at max RPM
EC90 flat 24	0.6	0	3000	5000	0.475	0.425	0.41
EC 60 Flat 12 a	0.47	0	3000	6000	0.265	0.275	0.25
EC 45 flat 70W 24	0.141	0	6000	10000	0.1175	0.128	0.124
EC45 flat 50 W	0.11	0	5000	10000	0.083	0.08	0.074
EC45 flat 30 W	0.075	0	6000	10000	0.05	0.06	0.058
EC45flat 12W	0.057	0	6000	10000	0.0275	0.022	0.025
EC32Flat 15W	0.046	0	6000	10000	0.021	0.026	0.028
EC32flat 6 W	0.032	0	6000	12000	0.006	0.008	0.011
EC20 flat 5W	0.022	0	8000	15000	0.006	0.008	0.009
EC20 flat 3W	0.012	0	7500	15000	0.0035	0.00425	0.005
EC 14 flat	0.008	0	12500	22000	0.0015	0.00175	0.002
EC 10 flat	0.00082	0	8000	22000	0.0002125	0.00021	0.000205
EC 9 flat .5 W	0.003	0	16000	25000	0.00081	0.000725	0.00065
EC-I 52 180W	0.82	0	3000	6000	0.64	0.575	0.3
EC-I 40 100 W	0.39	0	4000	8000	0.23	0.2	0.125
Ec-I 40 70W	0.25	0	6000	10000	0.16	0.135	0.09
Eci-40 50W	0.18	0	6000	10000	0.088	0.075	0.055
EC-4pole 32 220W	0.86	0	6000	12000	0.34	0.31	0.29
EC-4pole 30 200W	0.3	0	15000	25000	0.13	0.1	0
EC-4pole 30 150W	0.3	0	1500	25000	0.125	0.1	0
EC-4pole 30 100W	0.21	0	15000	25000	0.065	0.055	0.05
EC-4pole 22 120W	0.175	0	15000	25000	0.059	0.05	0.041
EC-4pole 22 90W	0.12	0	15000	25000	0.042	0.04	0.039
EC-max 40 120 W	0.72	0	6000	12000	0.175	0.165	0.145
EC-max 40 70 W	0.46	0	6000	12000	0.085	0.084	0.083

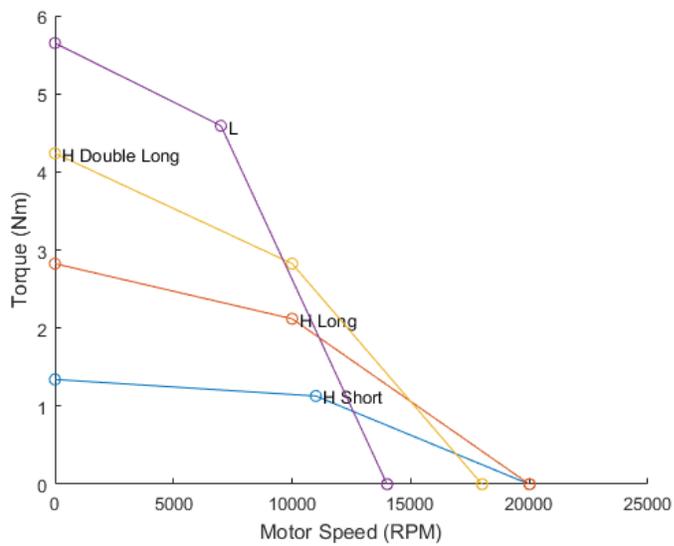
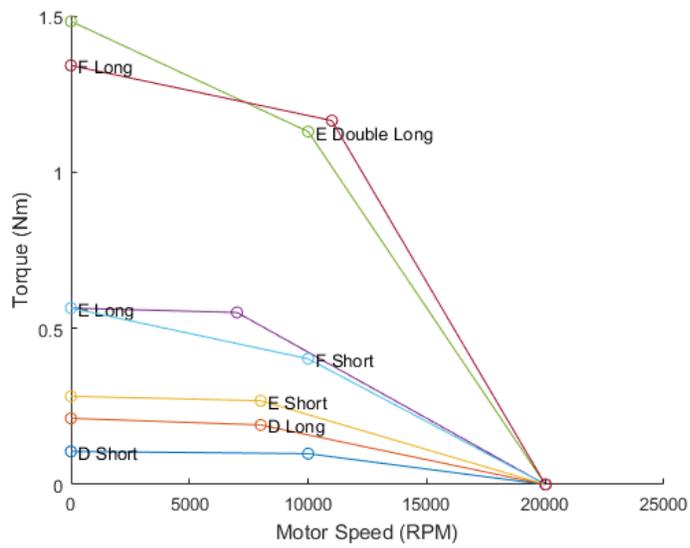
Series	Motor Weight (g)	Lower RPM	mid RPM	max RPM	Torque (Nm) at low RPM	Torque (Nm) at mid RPM	Torque (Nm) at max RPM
EC-max 30 60W	0.305	0	7500	15000	0.058	0.05	0.045
EC-max30 40w	0.195	0	7500	15000	0.031	0.029	0.025
EC-max 22 25W	0.11	0	9000	18000	0.0225	0.0205	0.018
Ecmax 22 12W	0.083	0	9000	18000	0.011	0.0105	0.01
Ecmax 16 8W	0.052	0	10000	20000	0.0073	0.007	0.0065
Ecmax 16 5W	0.036	0	10000	20000	0.0031	0.00285	0.00275
EC 60 400 W	2.45	0	3500	7000	0.82	0.725	0.65
EC 45 250 W	1.15	0	6000	12000	0.31	0.295	0.27
E45 150W	0.85	0	7500	15000	0.18	0.165	0.15
EC 40 170W	0.58	0	9000	18000	0.16	0.145	0.11
EC 32 80W	0.27	0	12500	25000	0.048	0.041	0.038
EC 25 250W	0.24	0	35000	70000	0.048	0.042	0.034
EC-22 80W	0.21	0	10000	20000	0.055	0.0525	0.05
EC 22 100W	0.128	0	30000	60000	0.048	0.045	0.04
EC 22 40W	0.085	0	30000	60000	0.02	0.0185	0.0175
EC 16 60W	0.058	0	35000	70000	0.016	0.015	0.012
EC 16 30W	0.034	0	35000	70000	0.008	0.0075	0.0065
EC 13 50W	0.044	0	45000	90000	0.008	0.0075	0.00675
EC13 30W	0.037	0	45000	90000	0.00505	0.0046	0.004
EC13 12 W	0.029	0	25000	50000	0.0061	0.006	0.0059
EC 13 6W	0.019	0	25000	50000	0.0025	0.00245	0.0024
EC10 8 W	0.013	0	32500	65000	0.00182	0.00175	0.0015
EC 8 2 W	0.006	0	40000	80000	0.000835	0.000825	0.000775
EC 6 2W	0.003	0	60000	100000	0.00048	0.00047	0.00046
EC6 1.5W	0.003	0	60000	100000	0.000325	0.0003175	0.00031
EC4 1 W	0.0018	0	30000	50000	0.00032	0.0003	0.00028
EC4 .5W	0.0012	0	30000	50000	0.0002	0.00019	0.00018

Appendix : Mag Motor Curves



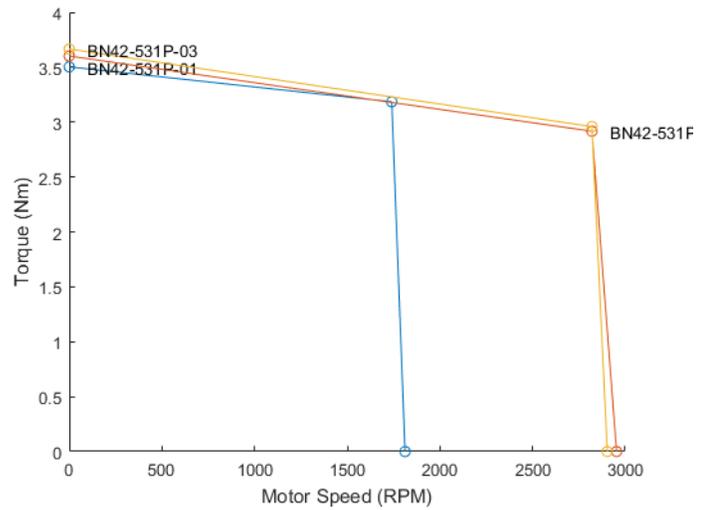
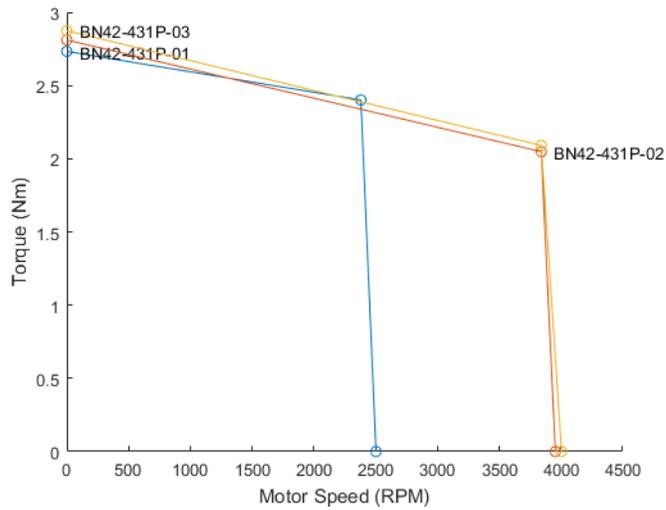
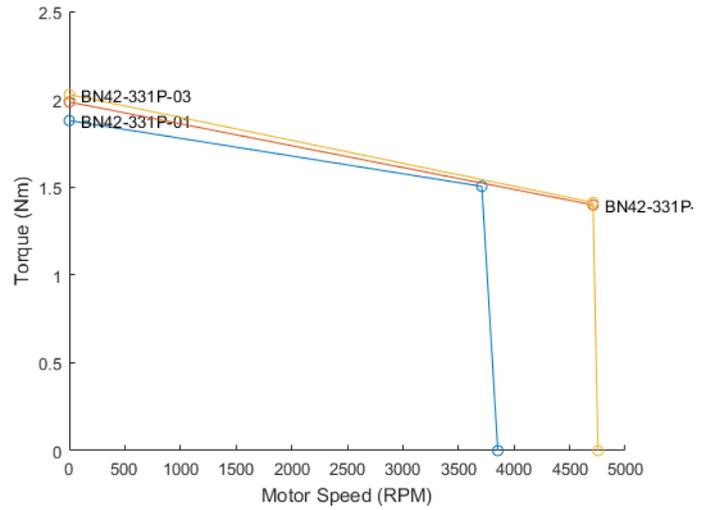
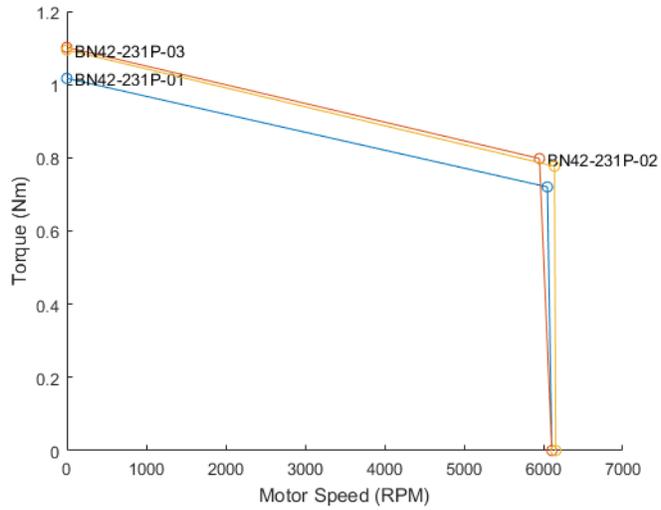
Series	Motor Weight (kg)	Lower RPM	mid RPM	max RPM	Torque (Nm) at low RPM	Torque (Nm) at mid RPM	Torque (Nm) at max RPM
B23-075	0.50	0	3000	6000	0.28	0.14	0
B23-150	0.82	0	3000	6000	0.56	0.28	0
B23-225	1.18	0	3000	6000	0.81	0.40	0
B23-300	1.50	0	3000	6000	1.02	0.51	0
B34-200	2.63	0	3000	6000	1.77	0.88	0
B34-300	3.36	0	3000	6000	2.68	1.34	0
B34-400	4.08	0	3000	6000	3.18	1.59	0
B34-500	4.81	0	3000	6000	4.17	2.08	0
BFA23-100	0.64	0	3500	7000	0.42	0.21	0
BFA23-200	1.09	0	3500	7000	0.81	0.41	0
BFA23-300	1.54	0	3500	7000	1.20	0.60	0
BFA23-400	1.95	0	3500	7000	1.62	0.81	0
BFA34-200	2.63	0	3000	6000	1.91	0.95	0
BFA34-300	3.36	0	3000	6000	2.68	1.34	0
BFA34-400	4.08	0	3000	6000	3.46	1.73	0
BFA34-500	4.81	0	3000	6000	4.24	2.12	0
BFA42-200	4.40	0	3000	6000	4.59	2.30	0
BFA42-300	5.81	0	3000	6000	6.36	3.18	0
BFA42-400	7.26	0	3000	6000	8.47	4.24	0
BFA42-600	10.98	0	3000	6000	11.30	5.65	0

Appendix : Woodward Motor Curves



Series	Motor Weight (kg)	Lower RPM	mid RPM	max RPM	Torque (Nm) at low RPM	Torque (Nm) at mid RPM	Torque (Nm) at max RPM
'D Short'	0.128	0	10000	20000	0.11	0.10	0
'D Long'	0.156	0	8000	20000	0.21	0.19	0
'E Short'	0.255	0	8000	20000	0.28	0.27	0
'E Long'	0.340	0	7000	20000	0.56	0.55	0
'E Double Long'	0.567	0	10000	20000	1.48	1.13	0
'F Short'	0.369	0	10000	20000	0.56	0.40	0
'F Long'	0.680	0	11000	20000	1.34	1.17	0
'H Short'	0.907	0	11000	20000	1.34	1.13	0
'H Long'	1.191	0	10000	20000	2.82	2.12	0
'H Double Long'	1.871	0	10000	18000	4.24	2.82	0
'L'	2.835	0	7000	14000	5.65	4.59	0

Appendix : Moog Motor Curves

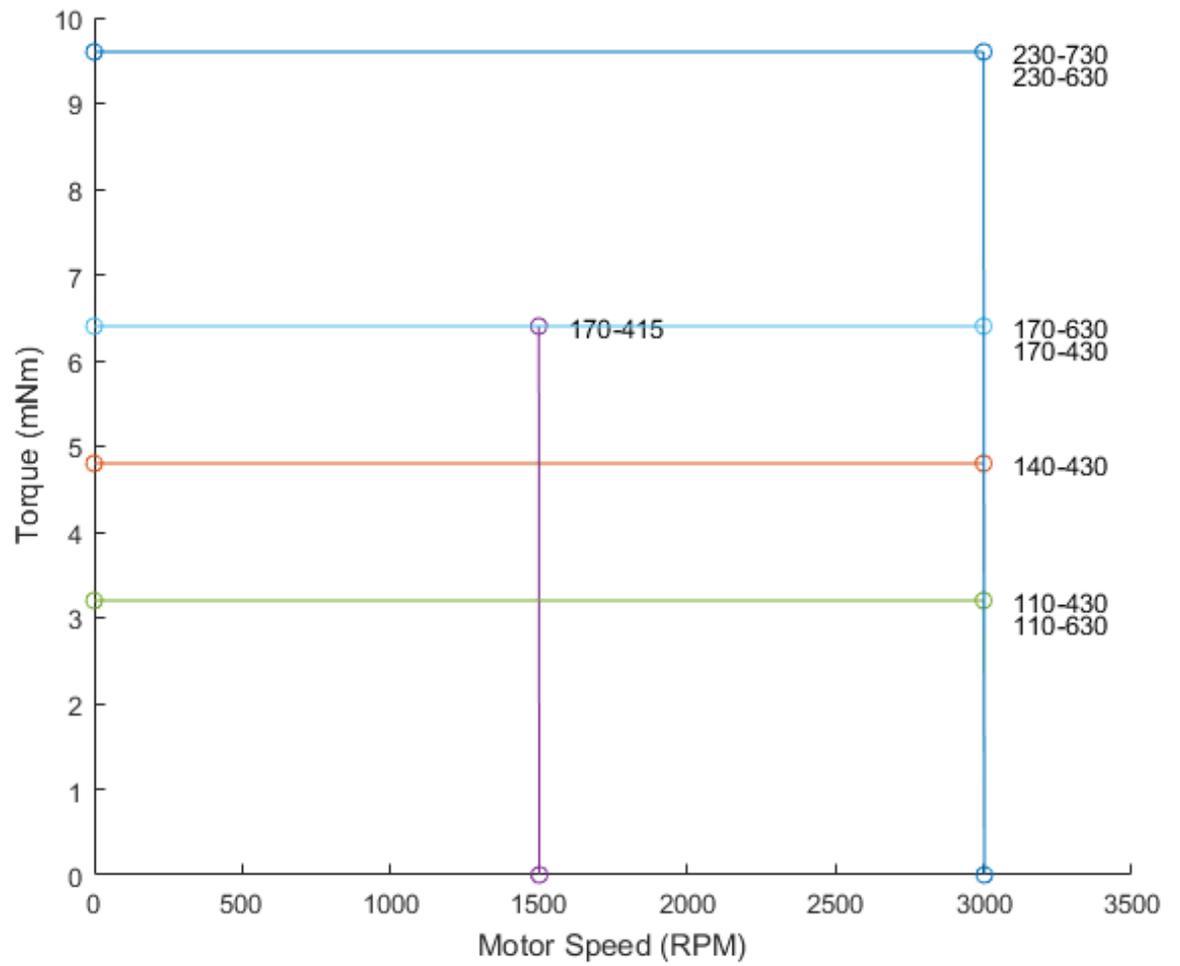


Series	Motor Weight (kg)	Lower RPM	mid RPM	max RPM	Torque (Nm) at low RPM	Torque (Nm) at mid RPM	Torque (Nm) at max RPM
BN42-231P-01	1.85	0	6050	6100	0	0.72	1.02
BN42-231P-02	1.85	0	5950	6100	0	0.80	1.10
BN42-231P-03	1.85	0	6140	6150	0	0.78	1.09
BN42-331P-01	2.95	0	3710	3850	0	1.50	1.88
BN42-331P-02	2.95	0	4710	4750	0	1.40	1.98
BN42-331P-03	2.95	0	4710	4750	0	1.41	2.03
BN42-431P-01	4.06	0	2380	2500	0	2.40	2.73
BN42-431P-02	4.06	0	3840	3950	0	2.05	2.81
BN42-431P-03	4.06	0	3840	4000	0	2.09	2.87
BN42-531P-01	5.17	0	1740	1810	0	3.18	3.50
BN42-531P-02	5.17	0	2820	2950	0	2.92	3.60
BN42-531P-03	5.17	0	2820	2900	0	2.96	3.66

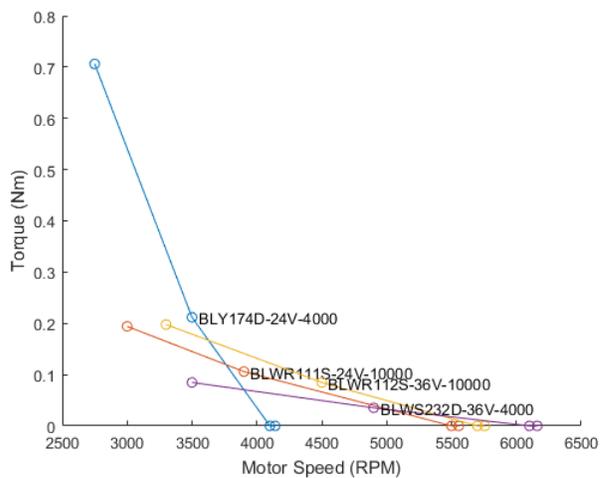
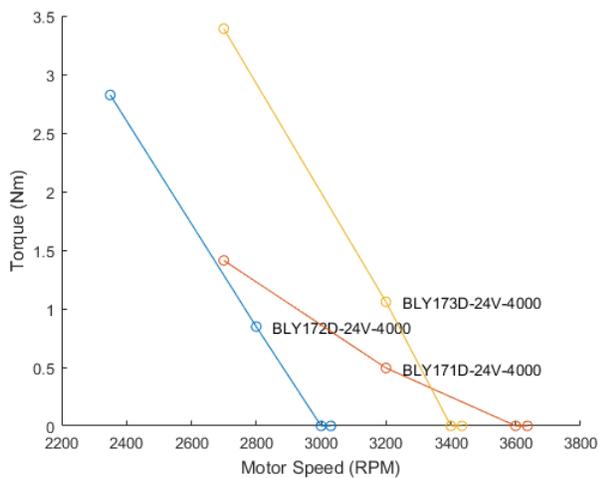
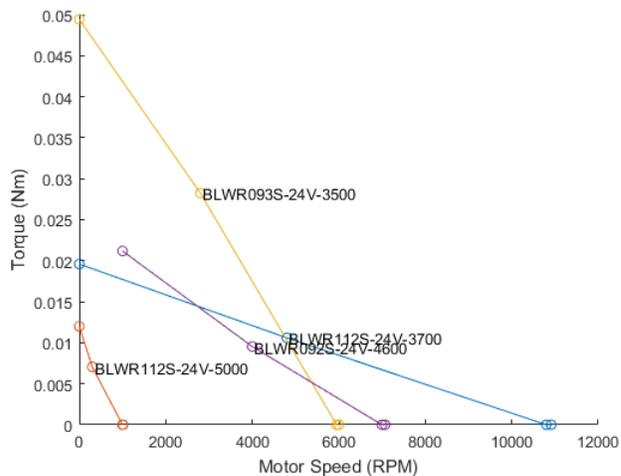
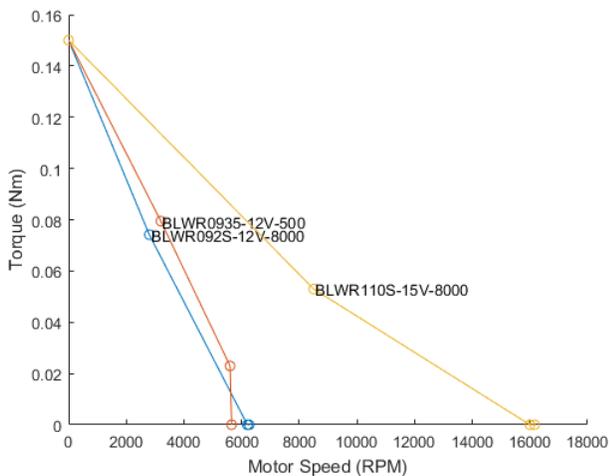
Appendix : Gems Motor Curves

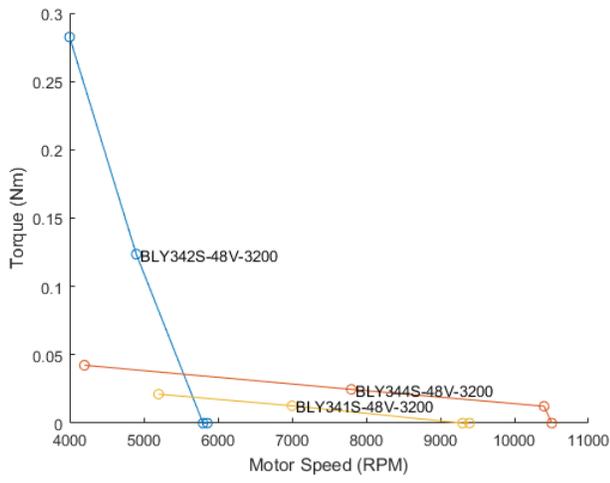
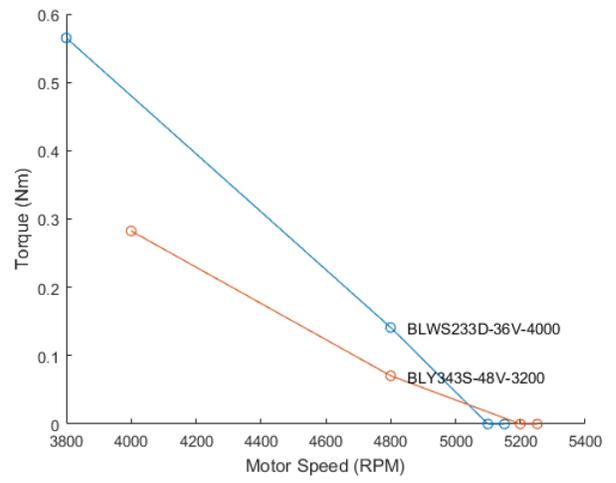
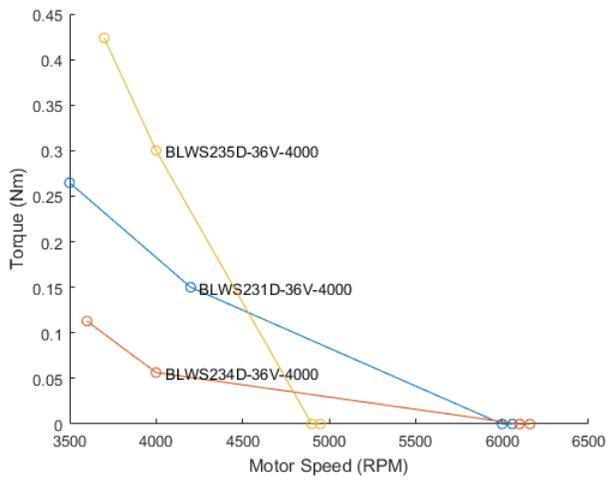
Series	Mass (kg)	Rated RPM	Torque (mNm)
110-430	4.5	3000	3.2
140-430	5.8	3000	4.8
170-430	7	3000	6.4
170-415	7	1500	6.4
110-630	4.5	3000	3.2
170-630	7	3000	6.4
230-630	9.5	3000	9.6
230-730	9.5	3000	9.6

Motor Series: GM110BLF



Appendix : Anaheim Automation Motor Curves





Series	Motor Weight (kg)	Lower RPM	mid RPM	max RPM	Torque (Nm) at low RPM	Torque (Nm) at mid RPM	Torque (Nm) at max RPM
BLWR092S-12V-8000	0.28	0	2800	6200	0.15	0.07	0.00
BLWR0935-12V-500	0.28	0	3200	5600	0.15	0.08	0.02
BLWR110S-15V-8000	0.28	0	8500	16000	0.15	0.05	0.00
BLWR112S-24V-3700	0.07	0	4800	10800	0.02	0.01	0.00
BLWR112S-24V-5000	0.13	0	300	1000	0.01	0.01	0.00
BLWR093S-24V-3500	0.12	0	2800	5950	0.05	0.03	0.00
BLWR092S-24V-4600	0.07	1000	4000	7000	0.02	0.01	0.00
BLY172D-24V-4000	2.60	2350	2800	3000	2.82	0.85	0.00
BLY171D-24V-4000	1.85	2700	3200	3600	1.41	0.49	0.00
BLY173D-24V-4000	4.00	2700	3200	3400	3.39	1.06	0.00
BLY174D-24V-4000	1.50	2750	3500	4100	0.71	0.21	0.00
BLWR111S-24V-10000	0.45	3000	3900	5500	0.19	0.11	0.00

Series	Motor Weight (kg)	Lower RPM	mid RPM	max RPM	Torque (Nm) at low RPM	Torque (Nm) at mid RPM	Torque (Nm) at max RPM
BLWR112S-36V-10000	0.50	3300	4500	5700	0.20	0.08	0.00
BLWS232D-36V-4000	0.30	3500	4900	6100	0.08	0.04	0.00
BLWS231D-36V-4000	0.65	3500	4200	6000	0.26	0.15	0.00
BLWS234D-36V-4000	0.38	3600	4000	6100	0.11	0.06	0.00
BLWS235D-36V-4000	1.00	3700	4000	4900	0.42	0.30	0.00
BLWS233D-36V-4000	1.25	3800	4800	5100	0.56	0.14	0.00
BLY343S-48V-3200	0.75	4000	4800	5200	0.28	0.07	0.00
BLY342S-48V-3200	0.80	4000	4900	5800	0.28	0.12	0.00
BLY344S-48V-3200	0.08	4200	7800	10400	0.04	0.02	0.01
BLY341S-48V-3200	0.06	5200	7000	9300	0.02	0.01	0.00

Appendix : Code: Load Variables

This code is used to load the various variables necessary to perform the task-based design. These variables are organized in a structure which allows variables to be easily passed between functions. These variables are NOT locked and are meant to be changed as part of optimization.

In addition to loading variables, this also generates the robotic arms using Peter Corke's Robotic Toolbox.

All the code for this dissertation can also be found on github under katzchen/Dissertation_OptimizedTaskBasedSpaceRobotics. The code was developed using MatLab 2017a and makes use of the Optimization toolbox, parallel computing toolbox, and Peter Corke's Robotics toolbox.

```
1 function AllVariables = LoadAllVariables_SRMS_simple()
2
3 'Loading SRMS'
4 path(path , 'C:/Users/.../MATLAB/RobotToolBox/rvctools')
5 path(path , 'C:/Users/.../MATLAB/RobotToolBox/rvctools')
6 startup_rvc
7 format compact
8
9 %-----
10 %Task Description
11 totaldistancetotravel =0;
```

```

12 x_task_start = -totaldistancetotravel/2;
13 x_task_end = -x_task_start;
14
15 %Create an array of all other variables htat will be passed in
16 %This is to just make it easier to keep track/pass all ...
    information through
17 %the GA/whatever easier
18
19 %-----
20 %All Task related variables
21 %-----
22
23 AllVariables.Task.MinThickness= 0.0254*(1/16);
24 %m, min thickness between outer and inner diaemters, 1/16" = 1.58mm;
25 AllVariables.Task.maxVelocity_xyz= .06;%m/s %SRMS
26 AllVariables.Task.maxVelocity_rpy= .06;%m/s %SRMS
27 AllVariables.Task.Mpayload=14515; %kg
28 AllVariables.Task.Lpayload=[3 3 3]; %length in m
29 AllVariables.Depth.currentLevel = 0; %NOT USED, was for ...
    workspace calculation
30 AllVariables.Depth.maxLevel = 3;%NOT USED, was for workspace ...
    calculation
31 AllVariables.Task.DesiredTime= 1; %seconds, if complex this is ...
    per segment
32
33 %-----
34 %Trajectory Variables

```

```

35  %-----
36
37  AllVariables.Task.ComplexMotion.Code = 0;
38  %1 for circle, 2 for square, 3= rotating square
39  if AllVariables.Task.ComplexMotion.Code == 3
40      AllVariables.Task.ComplexMotion.Nsegments = 8;
41      %number of segments
42  else
43      AllVariables.Task.ComplexMotion.Nsegments = 4;
44      %number of segments
45  end
46  AllVariables.Task.ComplexMotion.XDistance = 4;
47  %either the radii of the circle, or the length of the planar square
48  AllVariables.Task.ComplexMotion.YDistance = 4;
49  %either the radii of the circle, or the length of the planar square
50  AllVariables.Task.task_start=[4.3508    2.4570    7.4396];
51  % of the main payload relative to global;
52  AllVariables.Task.task_end= [7 0 2];
53  %theta is about [z Y X]
54  AllVariables.Task.theta_task_start=[20 135 20]*pi/180;
55  %of the main payload relative to global;
56  AllVariables.Task.theta_task_end=[0 0 0];
57  AllVariables.Task.npoints=10;
58  %HALF the number of points used in trajecotries
59
60  AllVariables.Task.DecelerationTime_xyz = 1;% must stop in 1 second
61  AllVariables.Task.DecelerationTime_rpy = 1;% must stop in 1 second

```

```

62 AllVariables.Task.maxAcceleration_xyz =
63 AllVariables.Task.maxVelocity_xyz/AllVariables.Task.DecelerationTime_xyz;
64 AllVariables.Task.maxAcceleration_rpy =
65 AllVariables.Task.maxVelocity_rpy/AllVariables.Task.DecelerationTime_rpy;
66 AllVariables.Task.ForceAtStart = [0 0 0 0 0 0]'; %these are ...
        glbal coordinates
67 AllVariables.Task.ForceAtEnd = [0 0 0 0 0 0]';
68 AllVariables.Task.ForceContinuous = [0 0 0 0 0 0]';
69 %continous torque while moving in addition to the moving in GLOBAL
70 AllVariables.Task.Velocity_angularlimit = 2*pi/10;
71 %1 full circle in 10 seconds
72 %-----
73 %Variables for plotting
74 %-----
75
76 AllVariables.Task.plot=0; %0 = no plot
77 AllVariables.Arm(1).plot=0;%0 = no plot
78 AllVariables.Arm(1).plotmovie=0;%0 = no plot
79 AllVariables.Arm(2).plot=0;%0 = no plot
80 AllVariables.Arm(2).plotmovie=0;%0 = no plot
81
82 %-----
83 %All Arm related variables
84 %-----
85
86 AllVariables.Arm(1).xyz_base = [0 -2 0]; %base of arm 1
87 AllVariables.Arm(1).ArmLength= [1.8 5 5.8 1.6 .5 .5];% .1 .6 .4];0

```

```

88 AllVariables.Arm(1).Armoffset= [0 0 0 0 0 0];
89 AllVariables.Arm(1).armtype= 2;
90 %armtype 1 == normal, %armtype 2= SRMS
91 AllVariables.Arm(1).PayloadGrip=[-AllVariables.Task.Lpayload(1)/2 ...
...
92 -AllVariables.Task.Lpayload(1)/2 0*AllVariables.Task.Lpayload(3)];
93 AllVariables.Arm(1).theta_grasp_start=[-0*pi/2 0 0];
94 %[-0*pi/180 0*pi/180 0*pi/180]; %relative to the payload
95 AllVariables.Arm(1).theta_grasp_end=AllVariables.Arm(1).theta_grasp_start;
96 AllVariables.Arm(1).IsPlanar = 0; %1 = yes
97 AllVariables.Arm(1).MotorTorqueSensitivity =
98 .07*ones(length(AllVariables.Arm(1).ArmLength),1);
99 AllVariables.Arm(1).gravity = [0 0 0]; %gravity vector
100 AllVariables.Arm(1).gearratio = 160;
101 AllVariables.Arm(1).MaxDeflectionPerLength = 1/200;
102 AllVariables.Arm(1).material.poisson = .2;
103 AllVariables.Arm(1).material.E = 160*10^9;%Pa
104 AllVariables.Arm(1).material.BendingStressMax =
105 (276*10^6)/2; %FOS = 2, based on yield (tensile)
106 AllVariables.Arm(1).material.Density = 1700; %kg/m^3
107 AllVariables.Arm(1).TotalArmLength = 15.2;
108 %is assigned in ChangeArmLength
109 AllVariables.Arm(1).material.MinThickness= 0.0254*(1/16)/2;
110 %m, min thickness between outer and inner diaemters, 1/16" = 1.58mm;
111 AllVariables.Arm(1).JointLimits = [-2*pi 2*pi; -2*pi 2*pi;...
112 -2*pi 2*pi;-2*pi 2*pi; -2*pi 2*pi;-2*pi 2*pi];
113 AllVariables.Arm(1).name = 'arm1';

```

```

114 AllVariables.Arm(1).linkcolor =[0 .5 0];
115 AllVariables.Arm(1).EEMaxMovementBy1Degree = .1; %m
116 AllVariables.Arm(1).EEJoints = 3;
117 %how many joints (from the end down) will have this limit
118 AllVariables.Arm(1).NumberOfStillViews= 2;
119 %used for plotting purposes, how many still views per row
120 AllVariables.Arm(1).TrajGenAngle=[32.4 50.5 263 50.4 151 ...
      88.2]*pi/180;
121 %gives an initial guess for the trajectory generation
122 %mostly used for more complex segments
123
124 AllVariables.Arm(2).xyz_base = [0 2 0];%base of arm 2
125 AllVariables.Arm(2).ArmLength= AllVariables.Arm(1).ArmLength;
126 AllVariables.Arm(2).Armoffset= AllVariables.Arm(1).Armoffset;
127 AllVariables.Arm(2).PayloadGrip=[-AllVariables.Task.Lpayload(1)/2 ...
      ...
128 AllVariables.Task.Lpayload(2)/2 0*AllVariables.Task.Lpayload(3)/2];
129 AllVariables.Arm(2).theta_grasp_start=[-0*90 0 0]*pi/180;
130 AllVariables.Arm(2).theta_grasp_end=AllVariables.Arm(2).theta_grasp_start;
131 AllVariables.Arm(2).IsPlanar = 0; %1 = yes
132 AllVariables.Arm(2).MotorTorqueSensitivity =
133 .07*ones(length(AllVariables.Arm(2).ArmLength),1);
134 AllVariables.Arm(2).gravity = [0 0 0]; %gravity vector
135 AllVariables.Arm(2).gearratio = 160;
136 AllVariables.Arm(2).MaxDeflectionPerLength =
137     AllVariables.Arm(1).MaxDeflectionPerLength;
138 AllVariables.Arm(2).material = AllVariables.Arm(1).material;

```

```

139 AllVariables.Arm(2).TotalArmLength = 15.2; %is assigned in ...
    ChangeArmLength
140 AllVariables.Arm(2).JointLimits = [-2*pi 2*pi; -2*pi 2*pi;...
141 -2*pi 2*pi;-2*pi 2*pi; -2*pi 2*pi;-2*pi 2*pi];
142 AllVariables.Arm(2).name = 'arm2';
143 AllVariables.Arm(2).linkcolor = [.5 0 .5];
144 AllVariables.Arm(2).EEMaxMovementBy1Degree = .1; %m
145 AllVariables.Arm(2).EEJoints = 3;
146 %how many joints (from the end down) will have this limit
147 AllVariables.Arm(2).NumberOfStillViews= ...
    AllVariables.Arm(1).NumberOfStillViews;
148 AllVariables.Arm(2).TrajGenAngle=[1.6780    0.1969 ...
149 -0.5575    1.6761    2.9749    -0.6735]*pi/180;
150 gives an initial guess for the trajectory generation - mostly ...
    used for more complex segments
151 AllVariables.Arm(2).armtype= 2;
152 armtype 1 == normal, %armtype 2= SRMS
153
154 %-----
155 %Generate Arms
156 %-----
157
158 for i = 1:length(AllVariables.Arm)
159     teach = 0;
160     AllVariables.bot(i)= ...
        GenerateBot_singleArm(AllVariables.Arm(i).ArmLength, ...
161     AllVariables.Arm(i).IsPlanar,AllVariables.Arm(i).armtype,....

```

```

162     AllVariables.Arm(i).Armoffset,teach);
163     AllVariables.bot(i).base = [eye(3,3) ...
        AllVariables.Arm(i).xyz_base';...
164     0 0 0 1];
165     AllVariables.bot(i).qlim =AllVariables.Arm(i).JointLimits;
166 end
167 \end{verbatim}
168
169 \end{landscape}
170
171 \hline
172 \small
173     \begin{verbatim}
174 function [bot]= ...
        GenerateBot_singleArm(Lengths,IsPlanar,armtype,offsets,teach)
175 %use robotics toolbox to find inverse kinematics and Jacobian
176 %step 1: put into DH parameters
177 %step 2: calculate joint angles
178 %Now with taking into account the velocity and acceleration of the
179 %end-effector. Note: V_ee and vdot_ee are given at the ...
        END-EFFECTOR not the
180 %payload center
181
182 %armtype 1 == normal
183 %armtype 2= SRMS
184
185 %If it is planar (isPlan

```

```

186 if (IsPlanar ==1)
187     PlanarFactor = 0;
188 else
189     PlanarFactor = 1;
190 end
191
192 m = zeros(length(Lengths),1);
193 if length(Lengths) < 6;
194     m = zeros(6,1);
195 end
196
197 %for initial on
198 density = 2700; %kg/m^3
199 outerradii = 0.1524/2; %3in
200 thickness = 0.003175; %1/8in
201 area = pi*outerradii^2-pi*(outerradii-thickness)^2;
202
203 if armtype == 1
204     %alpha = 0 for all planar
205     for i = 1:length(Lengths)
206         if i == 1
207             L(i) = Link('d',offsets(i), 'a',Lengths(i), 'alpha',0);
208         else
209             if mod(i,2) == 0
210                 L(i) = Link('d',offsets(i), 'a',Lengths(i),...
211                     'alpha',0*PlanarFactor*pi/2);
212             else

```

```

213         L(i) = Link('d',offsets(i), 'a',Lengths(i),...
214                 'alpha',PlanarFactor*pi/2);
215     end
216 end
217 L(i).qlim = [0 2*pi];
218 mass(i) = density*area*Lengths(i);
219 L(i).m = mass(i);
220 L(i).Jm = .1; %set high .02 for motormass masses of 18kg
221 L(i).B = 0*6.51/1000;
222 L(i).G= 0;
223 end
224 elseif armtype == 2
225     %alpha = 0 for all planar
226     for i = 1:length(Lengths)
227         if i == 1
228             L(i) = Link('d',offsets(i), 'a',Lengths(i), ...
229                     'alpha',pi/2);
230         elseif i == 2
231             L(i) = Link('revolute','d',offsets(i), ...
232                     'a',Lengths(i),...
233                     'alpha',pi/2,'standard');
234         elseif i==3
235             L(i) = Link('revolute','d',offsets(i), ...
236                     'a',Lengths(i),...
237                     'alpha',pi/2,'offset',pi/2);
238         elseif i==4
239             L(i) = Link('revolute','d',offsets(i), ...

```

```

                'a',Lengths(i),...
237                'alpha',pi/2,'standard');
238     elseif i==5
239         L(i) = Link('revolute','d',offsets(i), ...
                'a',Lengths(i),...
240                'alpha',pi/2,'offset',pi/2);
241     elseif i==6
242         L(i) = Link('revolute','d',offsets(i), ...
                'a',Lengths(i),...
243                'alpha',0,'standard','offset',pi/2);
244     end
245     %L(i).qlim = [-2*pi 2*pi];
246     mass(i) = density*area*Lengths(i);
247     L(i).m = mass(i);
248     L(i).Jm = .1; %set high .02 for motormass masses of 18kg
249     L(i).B = 6.51/1000;
250     L(i).G= 1;
251     %massoflink = density*area*Lengths(i);
252 end
253 elseif armtype == 3 %ranger
254     'Is Rnager'
255     %alpha = 0 for all planar
256     for i = 1:length(Lengths)
257         if i == 1
258             L(i) = Link('revolute','d',offsets(i), ...
                'a',Lengths(i),...
259                'alpha',pi/2,'standard','offset',0); %pitch

```

```

260     elseif i==2
261         L(i) = Link('revolute','d',offsets(i), ...
                    'a',Lengths(i),...
                    'alpha',pi/2,'standard','offset',pi/2); %roll
262
263     elseif i==3
264         L(i) = Link('revolute','d',offsets(i), ...
                    'a',Lengths(i),...
                    'alpha',pi/2,'standard','offset',0);
265
266     elseif i==4
267         L(i) = Link('revolute','d',offsets(i), ...
                    'a',Lengths(i),...
                    'alpha',0,'standard','offset',0);
268
269     elseif i==5
270         L(i) = Link('revolute','d',offsets(i), ...
                    'a',Lengths(i),...
                    'alpha',pi/2,'standard','offset',0);
271
272     elseif i==6
273         L(i) = Link('revolute','d',offsets(i), ...
                    'a',Lengths(i),...
                    'alpha',pi/2,'standard','offset',0);
274
275     end
276
277     L(i).qlim = [0 2*pi];
278
279     mass(i) = density*area*Lengths(i);
280
281     L(i).m = mass(i);
282
283     L(i).G = 0; %setting gear ratio to 0
284
285     L(i).Jm = .1; %set high .02 for motormass masses of 18kg
286
287     L(i).I = eye(3); %moment o inertia of link

```

```
282         %massoflink = density*area*Lengths(i);
283     end
284 end
285
286
287 gravityvector = [0 0 0];
288 bot = SerialLink(L);
289 if teach == 1
290     figure
291     if armtype ==3
292
293         bot.plot([0 200 0 320 30 30]*pi/180)
294     else
295
296         bot.plot([0 0 0 0 0 45]*pi/180)
297     end
298     bot.teach()
299 end
300 bot.gravity = gravityvector;
```

Bibliography

- [1] The shuttle remote manipulator system – the canadarm. URL https://www.ieee.ca/millennium/canadarm/canadarm_technical.html.
- [2] A Hernansanz, J Amat, and A Casals. Optimization criterion for safety task transfer in cooperative robotics. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–6. IEEE, 2009.
- [3] S Fujii and S Kuroono. Coordinated computer control of a pair of manipulators. In *Proceeding of 4th IFToMM World Congress*, pages 411–417, 1975.
- [4] Frederik W Heger, Laura M Hiatt, Brennan Sellner, Reid Simmons, and Sanjiv Singh. Results in sliding autonomy for multi-robot spatial assembly. In *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, volume 603, pages 5–8, 2005.
- [5] Gregory Dudek, Michael RM Jenkin, Evangelos Milios, and David Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4):375–397, 1996.

- [6] Okawa Takehiro and Satoshi Murata. Towards dynamic and robust robot division of tasks via local communication among robots-application to space solar power system construction. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2008.
- [7] Y. Uny Cao, Andrew B. Kahng, and Alex S. Fukunaga. Cooperative mobile robotics: Antecedents and directions. *Robot Colonies*, page 727, 1997. doi: 10.1007/978-1-4757-6451-2_1.
- [8] Renato Tinós, Marco H Terra, and João Yoshiyuki Ishihara. Motion and force control of cooperative robotic manipulators with passive joints. *IEEE Transactions on Control Systems Technology*, 14(4):725–734, 2006.
- [9] Yu Zhang, Wenfu Xu, Zhiying Wang, and Yu She. Dynamic modeling of self-reconfigurable multi-arm space robotic system with variable topology. *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013. doi: 10.1109/robio.2013.6739525.
- [10] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, and C. Curatella. Task-oriented motion planning for multi-arm robotic systems. *Robotics and Computer-Integrated Manufacturing*, 28(5):569582, 2012. doi: 10.1016/j.rcim.2012.02.007.
- [11] S. Sirouspour. Modeling and control of cooperative teleoperation systems. *IEEE Transactions on Robotics*, 21(6):1220–1225, Dec 2005. ISSN 1552-3098. doi: 10.1109/TRO.2005.852254.
- [12] Christian Smith, Yiannis Karayiannidis, Lazaros Nalpantidis, Xavi Gratal, Peng Qi, Dimos V. Dimarogonas, and Danica Kragic. Dual arm manipulationa survey.

- Robotics and Autonomous Systems*, 60(10):13401353, 2012. doi: 10.1016/j.robot.2012.07.005.
- [13] J Krüger, G Schreck, and D Surdilovic. Dual arm robot for flexible and cooperative assembly. *CIRP Annals-Manufacturing Technology*, 60(1):5–8, 2011.
- [14] Yuan-Chih Peng, David S. Carabis, and John T. Wen. Collaborative manipulation with multiple dual-arm robots under human guidance. *International Journal of Intelligent Robotics and Applications*, 2(2):252266, 2018. doi: 10.1007/s41315-018-0053-y.
- [15] Leonard Felicetti, Paolo Gasbarri, Andrea Pisculli, Marco Sabatini, and Giovanni B. Palmerini. Design of robotic manipulators for orbit removal of spent launchers stages. *Acta Astronautica*, 119:118130, 2016. doi: 10.1016/j.actaastro.2015.11.012.
- [16] T. Sugar and V. Kumar. Decentralized control of cooperating mobile manipulators. *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, 1998. doi: 10.1109/robot.1998.680672.
- [17] Andrea Zambelli Bais, Sebastian Erhart, Luca Zaccarian, and Sandra Hirche. Dynamic load distribution in cooperative manipulation tasks. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015. doi: 10.1109/iros.2015.7353699.
- [18] Yoshihiko Nakamura. Minimizing object strain energy for coordination of multiple robotic mechanisms. In *American Control Conference, 1988*, pages 499–509. IEEE, 1988.

- [19] Yoshihiko Nakamura. Minimizing object strain energy for coordination of multiple robotic mechanisms. *1988 American Control Conference*, 1988. doi: 10.23919/acc.1988.4789771.
- [20] Kazuya Yoshida, Ryo Kurazume, and Yoji Umetani. Dual arm coordination in space free-flying robot. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2516–2521. IEEE, 1991.
- [21] C. R. Carignan and D. L. Akin. Optimal force distribution for payload positioning using a planar dual-arm robot. *Journal of Dynamic Systems, Measurement, and Control*, 111(2):205, 1989. doi: 10.1115/1.3153038.
- [22] Suril V. Shah, Inna Sharf, and Arun Misra. Reactionless path planning strategies for capture of tumbling objects in space using a dual-arm robotic system. *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013. doi: 10.2514/6.2013-4521.
- [23] E. Papadopoulos and A. Abu-Abed. On the design of zero reaction manipulators. *Journal of Mechanical Design*, 118(3):372, 1996. doi: 10.1115/1.2826895.
- [24] Wenfu Xu, Yu Liu, and Yangsheng Xu. The coordinated motion planning of a dual-arm space robot for target capturing. *Robotica*, 30(5):755–771, 2012.
- [25] Craig R Carignan and David L Akin. The reaction stabilization of on-orbit robots. *IEEE control systems*, 20(6):19–33, 2000.
- [26] Gary Yale and Brij Agrawal. Cooperative control of multiple space manipulators. *Guidance, Navigation, and Control Conference*, 1994. doi: 10.2514/6.1994-3653.
- [27] Kazuya Yoshida, Kenichi Hashizume, and Satoko Abiko. Zero reaction maneuver: Flight validation with ets-vii space robot and extension to kinematically redundant

- arm. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 441–446. IEEE, 2001.
- [28] Z Vafa and Steven Dubowsky. On the dynamics of manipulators in space using the virtual manipulator approach. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, volume 4, pages 579–585. IEEE, 1987.
- [29] E. Papadopoulos and S. Dubowsky. Dynamic singularities in free-floating space manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 115(1):44, 1993. doi: 10.1115/1.2897406.
- [30] S. Dubowsky and E. Papadopoulos. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *IEEE Transactions on Robotics and Automation*, 9(5):531543, 1993. doi: 10.1109/70.258046.
- [31] Kostas Nanos and Evangelos Papadopoulos. Avoiding dynamic singularities in cartesian motions of free-floating manipulators. *IEEE Transactions on Aerospace and Electronic Systems*, 51(3):23052318, 2015. doi: 10.1109/taes.2015.140343.
- [32] Richard W. Longman. The kinetics and workspace of a satellite-mounted robot. *The Kluwer International Series in Engineering and Computer Science Space Robotics: Dynamics and Control*, page 2744, 1993. doi: 10.1007/978-1-4615-3588-1_2.
- [33] Dextre and rrm complete successful satellite refuelling demo, 2013. URL <https://www.nasaspaceflight.com/2013/01/dextre-rrm-satellite-refuelling-demo/>.
- [34] Glenn D. Cook. The canadian designed and built spar/mda canadarms,

2015. URL <https://documents.techno-science.ca/documents/CASM-Aircrafthistories-SparMDACanadarms.pdf>.

- [35] Angel Flores-Abad, Ou Ma, Khanh Pham, and Steve Ulrich. A review of space robotics technologies for on-orbit servicing. *Progress in Aerospace Sciences*, 68:126, 2014. doi: 10.1016/j.paerosci.2014.03.002.
- [36] Graham Gibbs. Canada and the international space station program: overview and status. In *54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, pages T-1, 2002.
- [37] T. E. Alberts and D. I. Soloway. Force control of a multi-arm robot system. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 1490–1496 vol.3, Apr 1988. doi: 10.1109/ROBOT.1988.12278.
- [38] A. Zambelli Bais, S. Erhart, L. Zaccarian, and S. Hirche. Dynamic load distribution in cooperative manipulation tasks. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2380–2385, Sept 2015. doi: 10.1109/IROS.2015.7353699.
- [39] S. Lee. Dual redundant arm configuration optimization with task-oriented dual arm manipulability. *IEEE Transactions on Robotics and Automation*, 5(1):7897, 1989. doi: 10.1109/70.88020.
- [40] Masaru Uchiyama and Pierre Dauchez. A symmetric hybrid position/force control scheme for the coordination of two robots. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 350–356. IEEE, 1988.

- [41] Y. Nakamura, K. Nagai, and T. Yoshikawa. Mechanics of coordinative manipulation by multiple robotic mechanisms. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 991–998, Mar 1987. doi: 10.1109/ROBOT.1987.1087941.
- [42] Yong Wang, Zhi Liu, and Yun Zhang. Adaptive jacobian coordinated manipulation for multiple robot manipulators. In *Control and Decision Conference (CCDC), 2016 Chinese*, pages 1342–1347. IEEE, 2016.
- [43] Tatsuzo Ishida. Force control in coordination of two arms. In *IJCAI*, pages 717–722, 1977.
- [44] Samad Hayati. Hybrid position/force control of multi-arm cooperating robots. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 82–89. IEEE, 1986.
- [45] R. E. Goddard, K. Laroussi, and H. Hemami. Lifting of an object by coordination of two planar robots. *CAD/CAM Robotics and Factories of the Future*, page 289290, 1989. doi: 10.1007/978-3-642-52326-7_47.
- [46] P. R. Pagilla and M. Tomizuka. Adaptive control of two robot arms carrying an unknown object. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 1, pages 597–602 vol.1, May 1995. doi: 10.1109/ROBOT.1995.525349.
- [47] Meyer Nahon and Jorge Angeles. Minimization of power losses in cooperating manipulators. *1991 American Control Conference*, 1991. doi: 10.23919/acc.1991.4791964.

- [48] John T Wen and Kenneth Kreutz-Delgado. Motion and force control of multiple robotic manipulators. *Automatica*, 28(4):729–743, 1992.
- [49] Masanobu Koga, Kazuhiro Kosuge, Katsuhisa Furuta, and Kageharu Nosaki. Coordinated motion control of robot arms based on the virtual internal model. *IEEE Transactions on Robotics and Automation*, 8(1):77–85, 1992.
- [50] Jeffrey Kerr and Bernard Roth. Analysis of multifingered hands. *The International Journal of Robotics Research*, 4(4):3–17, 1986.
- [51] Y-R Hu and AA Goldenberg. Dynamic control of multiple coordinated redundant manipulators with torque optimization. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 1000–1005. IEEE, 1990.
- [52] Yuan F Zheng and JYS Luh. Optimal load distribution for two industrial robots handling a single object. *Journal of Dynamic Systems, Measurement, and Control*, 111(2):232–237, 1989.
- [53] W. Gueaieb, F. Karray, and S. Al-Sharhan. A robust adaptive fuzzy position/force control scheme for cooperative manipulators. *IEEE Transactions on Control Systems Technology*, 11(4):516528, 2003. doi: 10.1109/tcst.2003.813378.
- [54] Craig R Carignan and David L Akin. Cooperative control of two arms in the transport of an inertial load in zero gravity. *IEEE Journal on Robotics and Automation*, 4(4):414–419, 1988.
- [55] Tianliang Liu, Yan Lei, Liang Han, Wenfu Xu, and Huaiwu Zou. Coordinated resolved motion control of dual-arm manipulators with closed chain. *International Journal of Advanced Robotic Systems*, 13(3):80, 2016. doi: 10.5772/63430.

- [56] Jin-Fan Liu and Karim Abdel-Malek. Robust control of planar dual-arm cooperative manipulators. *Robotics and Computer-Integrated Manufacturing*, 16(2-3):109–119, 2000.
- [57] R. Rastegari and S. A. A. Moosavian. Multiple impedance control of space free-flying robots using virtual object grasp. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3125–3130, Oct 2006. doi: 10.1109/IROS.2006.282332.
- [58] Reza Mohajerpoor, Seyed Mehdi Rezaei, Heidar Ali Talebi, and Reza Monfaredi. A robust adaptive control scheme for two planar manipulators handling an unknown object in an assembly process. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 156–161. IEEE, 2011.
- [59] K. Laroussi, H. Hemami, and R. E. Goddard. Coordination of two planar robots in lifting. *IEEE Journal on Robotics and Automation*, 4(1):77–85, Feb 1988. ISSN 0882-4967. doi: 10.1109/56.774.
- [60] Bin Yao and Masayoshi Tomizuka. Adaptive coordinated control of multiple manipulators handling a constrained object. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 624–629. IEEE, 1993.
- [61] Yun-Hui Liu and S. Arimoto. Distributively controlling two robots handling an object in the task space without any communication. *IEEE Transactions on Automatic Control*, 41(8):1193–1198, 1996. doi: 10.1109/9.533683.
- [62] Olivier Chocron and Philippe Bidaud. Evolutionary algorithms in kinematic design of robotic systems. In *Intelligent Robots and Systems, 1997. IROS'97., Proceedings*

- of the 1997 IEEE/RSJ International Conference on, volume 2, pages 1111–1117. IEEE, 1997.
- [63] BK Rout and RK Mittal. Optimal manipulator tolerance design using hybrid evolutionary optimization technique. *International Journal of Robotics and Automation*, 22(4):263–271, 2007.
- [64] Robert O Ambrose. Interactive robot joint design, analysis and prototyping. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 2, pages 2119–2124. IEEE, 1995.
- [65] Katherine M McBryan and David L Akin. Systems analysis of the benefits of multiple model-based cooperating serial-link manipulators. In *AIAA SPACE and Astronautics Forum and Exposition*, page 5121, 2017.
- [66] Karim Abdel-Malek, Wei Yu, and Jingzhou Yang. Placement of robot manipulators to maximize dexterity. *International Journal of Robotics and Automation*, 19(1): 6–14, 2004.
- [67] Sarosh Patel and Tarek Sobh. Goal directed design of serial robotic manipulators. In *American Society for Engineering Education (ASEE Zone 1), 2014 Zone 1 Conference of the*, pages 1–6. IEEE, 2014.
- [68] Virendra Kumar, Soumen Sen, Shibendu S Roy, Chandan Har, and SN Shome. Design optimization of serial link redundant manipulator: an approach using global performance metric. *Procedia Technology*, 14:43–50, 2014.
- [69] Lounell B Gueta, Jia Cheng, Ryosuke Chiba, Tamio Arai, Tsuyoshi Ueyama, and Jun Ota. Multiple-goal task realization utilizing redundant degrees of freedom of

- task and tool attachment optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1714–1719. IEEE, 2011.
- [70] Hatem Al-Dois, A. K. Jha, and R. B. Mishra. Task-based design optimization of serial robot manipulators. *Engineering Optimization*, 45(6):647658, 2013. doi: 10.1080/0305215x.2012.704027.
- [71] Athanasios Nektarios and Nikos A Aspragathos. Optimal location of a general position and orientation end-effector’s path relative to manipulator’s base, considering velocity performance. *Robotics and Computer-Integrated Manufacturing*, 26(2):162–173, 2010.
- [72] Daniel Di Marco, Oliver Zweigle, and Paul Levi. Base pose estimation using shared reachability maps for manipulation tasks. In *Autonomous Robot Systems (Robotica), 2013 13th International Conference on*, pages 1–6. IEEE, 2013.
- [73] A Khoukhi and A Ghoul. On the maximum dynamic stress search problem for robot manipulators. *Robotica*, 22(5):513–522, 2004.
- [74] Sarosh Patel and Tarek Sobh. Manipulator performance measures-a comprehensive literature survey. *Journal of Intelligent & Robotic Systems*, 77(3-4):547–570, 2015.
- [75] Jahng-Hyon Park and Haruhiko Asada. Concurrent design optimization of mechanical structure and control for high speed robots. *Journal of dynamic systems, measurement, and control*, 116(3):344–356, 1994.
- [76] Jahangir S Rastegar, Lidong Liu, and Dan Yin. Task-specific optimal simultaneous kinematic, dynamic, and control design of high-performance robotic systems. *IEEE/ASME transactions on mechatronics*, 4(4):387–395, 1999.

- [77] J-O Kim and Pradeep K Khosla. A formulation for task based design of robot manipulators. In *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 3, pages 2310–2317. IEEE, 1993.
- [78] Wan Kyun Chung, Jeongheon Han, Youngil Youm, and SH Kim. Task based design of modular robot manipulator using efficient genetic algorithm. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 507–512. IEEE, 1997.
- [79] T.m. Sobh and D.y. Toundykov. Optimizing the tasks at hand. *IEEE Robotics & Automation Magazine*, 11(2):7885, 2004. doi: 10.1109/mra.2004.1310944.
- [80] John J Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson-/Prentice Hall Upper Saddle River, NJ, USA:, 2005.
- [81] Joon-Young Park, Pyung-Hun Chang, and Jeong-Yean Yang. Task-oriented design of robot kinematics using the grid method. *Advanced robotics*, 17(9):879–907, 2003.
- [82] Tarek M Sobh and Daniel Y Toundykov. Optimizing the tasks at hand [robotic manipulators]. *IEEE robotics & automation magazine*, 11(2):78–85, 2004.
- [83] KV Varalakshmi and J Srinivas. Optimized configurations of kinematically redundant planar parallel manipulator following a desired trajectory. *Procedia Technology*, 14:133–140, 2014.
- [84] A Jafari, M Safavi, and A Fadaei. A genetic algorithm to optimum dynamic performance of industrial robots in the conceptual design phase. In *Rehabilitation*

- Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*, pages 1129–1135. IEEE, 2007.
- [85] Sukhan Lee and Jang M Lee. Task-oriented dual-arm manipulability and its application to configuration optimization. In *Decision and Control, 1988., Proceedings of the 27th IEEE Conference on*, pages 2253–2260. IEEE, 1988.
- [86] Bo Zhang, Bin Liang, Xueqian Wang, Gang Li, Zhang Chen, and Xiaojun Zhu. Manipulability measure of dual-arm space robot and its application to design an optimal configuration. *Acta Astronautica*, 128:322329, 2016. doi: 10.1016/j.actaastro.2016.07.040.
- [87] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani. Six-dof impedance control of dual-arm cooperative manipulators. *IEEE/ASME Transactions on Mechatronics*, 13(5):576586, 2008. doi: 10.1109/tmech.2008.2002816.
- [88] DE Orin and SY Oh. Control of force distribution in robotic mechanisms containing closed kinematic chains. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):134–141, 1981.
- [89] F-T Cheng and David E Orin. Efficient algorithm for optimal force distribution-the compact-dual lp method. *IEEE Transactions on Robotics and Automation*, 6(2):178–187, 1990.
- [90] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 2000. doi: 10.1109/robot.2000.844081.

- [91] Konstantinos Dermitzakis, Juan Pablo Carbajal, and James H Marden. Scaling laws in robotics. *Procedia Computer Science*, 7:250–252, 2011.
- [92] Brian H Wilcox. Athlete: Trading complexity for mass in roving vehicles. In *AIAA SPACE 2013 Conference and Exposition*, page 5382, 2013.
- [93] John M Hollerbach, Ian W Hunter, and John Ballantyne. A comparative analysis of actuator technologies for robotics. *Robotics Review*, 2:299–342, 1991.
- [94] Kollmorgen LLC. Kollmorgen, 2013. URL <http://www.kollmorgen.com/en-us/home/>.
- [95] LLC Maxon Motor ag. High precision drives and systems, 2017. URL <https://www.maxonmotorusa.com/>.
- [96] LLC Global Electric Motor Solution. 12v 24v 36v and 48v brushless (blde) motor manufacturers, 2015. URL <https://www.gemsmotor.com/>.
- [97] Moog Inc. Direct drive brushless dc torque motors, 2018. URL <http://www.moog.com/1>.
- [98] Woodward Inc. Woodward motors and accessories, 2018. URL <http://www.woodward.com/motorsandservomotors.aspx>.
- [99] LLC Magmotor Technologies Inc. Brushless motors, 2012. URL <https://www.magmotor.com/brushless/brushless.html>.
- [100] Inc. Anaheim Automation. Brushless dc, 2018. URL <http://www.anaheimautomation.com/>.

- [101] Gerd Hirzinger, Norbert Sporer, A Albu-Schaffer, M Hahnle, Rainer Krenn, Antonio Pascucci, and Markus Schedl. Dlr's torque-controlled light weight robot iii-are we reaching the technological limits now? In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 1710–1716. IEEE, 2002.
- [102] HarmonicDrive LLC. Harmonicdrive, 2017. URL <http://www.harmonicdrive.net/>.
- [103] George H Cheng, Adel Younis, Kambiz Haji Hajikolaie, and G Gary Wang. Trust region based mode pursuing sampling method for global optimization of high dimensional design problems. *Journal of Mechanical Design*, 137(2):021407, 2015.
- [104] JP Peterson, P Seide, and VI Weingarten. Buckling of thin-walled circular cylinders. 1968.
- [105] Canadarm, 2018. URL <http://www.asc-csa.gc.ca/eng/canadarm/default.asp>.
- [106] Gong-Bo Dai and Yen-Chen Liu. Distributed coordination and cooperation control for networked mobile manipulators. *IEEE Transactions on Industrial Electronics*, 64(6):5065–5074, 2017.
- [107] Tianliang Liu, Yan Lei, Liang Han, Wenfu Xu, and Huaiwu Zou. Coordinated resolved motion control of dual-arm manipulators with closed chain. *International Journal of Advanced Robotic Systems*, 13(3):80, 2016.
- [108] Sebastian Erhart and Sandra Hirche. Internal force analysis and load distribution for cooperative multi-robot manipulation. *IEEE Transactions on Robotics*, 31(5):12381243, 2015. doi: 10.1109/tro.2015.2459412.

- [109] Sasan Barissi and Hamid D Taghirad. Task based optimal geometric design and positioning of serial robotic manipulators. In *Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on*, pages 158–163. IEEE, 2008.
- [110] J. Kim and P.k. Khosla. A multi-population genetic algorithm and its application to design of manipulators. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1992. doi: 10.1109/iros.1992.587341.
- [111] Sarosh H. Patel and Tarek M. Sobh. Using task descriptions for designing optimal task specific manipulators. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015. doi: 10.1109/iros.2015.7353872.
- [112] Tuomo Kivelä, Jouni Mattila, and Jussi Puura. A generic method to optimize a redundant serial robotic manipulator’s structure. *Automation in Construction*, 81: 172–179, 2017.
- [113] Todd Rowland and Eric W. Weisstein. Genetic algorithm. *From MathWorld—A Wolfram Web Resource*. URL <http://mathworld.wolfram.com/GeneticAlgorithm.html>.
- [114] MATLAB Optimization Toolbox. Matlab optimization toolbox, 2017b. The MathWorks, Natick, MA, USA.
- [115] Sarosh Patel, Tarek Sobh, and Ausif Mahmood. Goal directed synthesis of serial manipulators based on task descriptions. *Chaos Modeling and Control Systems Design Studies in Computational Intelligence*, page 319350, 2015. doi: 10.1007/978-3-319-13132-0_13.

- [116] Gabriel Zeinoun. *Optimization of a Reconfigurable Manipulator with Lockable Cylindrical Joints*. PhD thesis, Concordia University, 2013.
- [117] Amiko Kauderer. Canadarm2 and the mobile servicing system, Jan 2013. URL https://www.nasa.gov/mission_pages/station/structure/elements/mss.html.
- [118] Wayne F Carriker, Pradeep K Khosla, and Bruce H Krogh. The use of simulated annealing to solve the mobile manipulator path planning problem. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 204–209. IEEE, 1990.