1-1-2007

Module-based kinetostatic analysis and optimization of a modular reconfigurable robot

Richard P. Mohamed
Ryerson University

MODULE-BASED KINETOSTATIC ANALYSIS AND OPTIMIZATION OF A MODULAR RECONFIGURABLE ROBOT

by

Richard P. Mohamed
Bachelor of Engineering
Ryerson University
Toronto, Ontario, Canada

A thesis
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in the Program of
Mechanical Engineering

Toronto, Ontario, Canada, 2007

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Borrower's Page

Ryerson university requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Name	Signature	Address	Date

Abstract

Kinetostatic Analysis and Module-Based Optimization of a Modular Reconfigurable Robot

Richard P. Mohamed

A thesis for the degree of
Master of Applied Science, 2007

Department of Mechanical Engineering, Ryerson University

In this thesis, a newly developed kinetostatic model for modular reconfigurable robots (MRRs) is presented. First, a kinematic computational method was created to allow for simple connectivity between modules which included the possibilities of angular offsets. Then, a flexibility analysis was performed to determine the static and dynamic flexibility of link and joint modules and the regions of flexibility were plotted to determine exactly which of the components can be considered flexible or rigid, depending on their sizes. Afterwards, the kinetostatic model was developed and compared to a finite element model and results give essentially the same tip deflections between the two models. This kinetostatic model was then used to determine the maximum allowable payload and maximum deflection position for a given MRR. Additionally, a direct method was created to determine the cross section properties of all modules in a given MRR for a given payload and maximum desirable tip deflection.

Acknowledgements

To begin, I would like to sincerely thank my supervisor Dr. Jeff Xi, who has taken me under his wing since my first year in undergraduate studies. Without his guidance and persistence, I honestly feel I would not be at the level where I am now. With his strong theoretical background and understanding of many engineering concepts, he has helped me solve many difficulties whenever I needed help. I was always intrigued by his innovative ideas and solutions to many problems, in engineering and in life. He is very devoted to his work, and is one of the few professors I know who would go out of his way to help his students in extreme situations. It has been my pleasure to work under his supervision. He has been my mentor over the past five years and I hope to somehow continue working with him in the future.

I would also like to thank my co-supervisor Dr. Guangjun Liu, for allowing me to work on this project and for allowing me to aid in the mechanical design of a modular reconfigurable joint in the summer of 2005. I also would like to thank the members of the MRR design team, Dr. Yingchun Zhang, Mr. Sajan Abdul and Mr. Mathew Adamson. It has been a pleasure to work with them and it was very satisfying to see the MRR joint when it was operational. I would like to thank Dr. Liu for his support, encouragement and kind words of advice throughout this project. In addition, I would like to thank Engineering Services Inc. and NSERC for their adequate support.

I also wish to thank my lab mates Mr. Yu Lin and Mr. Daniel Finistauri, who were also working on reconfigurable robots, for their advice when I was in need of an opinion. Furthermore, I would like to thank Mr. Liang Liao and Mr. Haibin Jia for their moral support over the past two years. I am grateful for working alongside them.

Finally, I would like to sincerely thank my family and close friends for their encouragement, support, and understanding over the course of my university career.

Table of Contents

List of Figures

List of Tables

Nomenclature

Symbol	Definition	Units
x, y, z	x, y or z coordinate position	m
L	length	m
E	modulus of elasticity	N/m^2
A	area	m^2
I_z	area moment of inertia about the z-axis	m^4
I_y	area moment of inertia about the y-axis	m^4
G	shear modulus	N/m^2
J	torsion constant	m^4
u	axial displacement	m
v	transverse bending displacement about y-axis	m
w	transverse bending displacement about z-axis	m
f_x	local force in x-direction	N
f_y	local force in y-direction	N
f_z	local force in z-direction	N
m_x	local moment about x-axis	N-m
m_x	local moment about y-axis	N-m
m_x	local moment about z-axis	N-m
P	applied force at tip	N
M	applied moment at tip	N-m
W	weight	N
I_p	polar mass moment of inertia	$kg\text{-}m^2$
σ	axial or bending stress	N/m^2
τ	shear stress	N/m^2
t	thickness	m
r	radius	m
b	rectangular cross section base length	m
h	rectangular cross section height	m
m	mass	kg

Greek Symbol	Definition	Units
θ_x	angle about x axis	rad (or deg if specified)
θ_y	angle about y axis	rad (or deg if specified)
θ_z	angle about z axis	rad (or deg if specified)
η	Boolean operator	-
ζ	transformed revolute joint input angle	rad (or deg if specified)
θ	torsional deformation about x-axis	rad
φ	bending angle about y-axis	rad
ψ	bending angle about z-axis	rad
ρ	material density	kg/m^3
ω_n	natural frequency	rad/s
α	torsional stress constant	-
κ_k	kinetostatic dexterity index	-
σ_y	yield stress	N/m^2

Subscript	Definition
config	configuration
i	i^{th} module
n	total number of modules
ICSU	initial configuration setup
input	joint motion input
s	static
m	motion
e	element (in the local coordinate frame)
G	global frame
offset	angular offset
axial	for axial deformation
bend,z	for bending about z-axis
bend,y	for bending about y-axis
torsion	for torsion about x-axis
shear,y	for transverse shear along the y-axis
j	joint
l	link
tip	variables at the tip of the module
inner	inner part of cross section
outer	outer part of cross section
max	maximum

1. Introduction

This thesis describes the kinetostatic modeling, analysis and design optimization of a serial (open-chain) modular reconfigurable robot. This chapter first introduces basic concepts of modular reconfigurable robots, and then describes the objectives and methodology of the thesis. Afterwards, a review of existing modular reconfigurable robot designs is presented. Then, a literature review is presented for the topics covered in this thesis. The final section in this chapter provides a brief outline of each of the following chapters presented in this thesis.

1.1 Background and Objectives

A modular reconfigurable robot (MRR) consists of several interconnected modules that can be assembled into different configurations in order to perform a desired task and meet the requirements for a specified range of motion. These modules can then be reconfigured to perform other tasks and obtain different ranges of motion. On the other hand, traditional non-reconfigurable robots are designed and configured for only one type of motion, thus their capabilities are limited. A typical MRR consists of separate revolute or prismatic joint modules connected to link modules [1-3]. Each joint module is connected to a link module usually by means of connection interfaces located at the ends of the link and the joint outer casing. Furthermore, typical MRRs have larger joint module sizes in comparison to conventional industrial robots because the joint casing must be reasonably sized to act as a connection interface to other joint or link modules. To simplify the reconfigurability of a MRR and reduce costs, separate joint modules should have similar shapes and contain similar components. It is also important to include a common connection interface to reduce the costs of designing and manufacturing separate adapter plates to connect modules with different interfaces. In addition, MRRs are easier to maintain than non-reconfigurable robots because each module can be easily removed and inspected, and can be repaired and replaced if necessary. The operation of the MRR is then limited to the number of working modules available until the malfunctioned module is replaced. If vital component failure occurs in a typical non-reconfigurable robot, the entire robot may become non-functional until the defective

component is repaired and replaced. This is especially true if the defective component is the base joint of a serial non-reconfigurable robot.

In manufacturing, MRRs can be used to perform a number of operations and the end-effector can be interchangeable, as well as the reconfigurable joint-link modules, in order to obtain desirable ranges of motion. The use of MRRs in multi-task manufacturing can also reduce the costs of purchasing several machines to perform specialized operations such as riveting and polishing. Multiple MRRs can be used and their individual modules can be changed to suit the type of assembly required in a production line. As with all robots, positional accuracy is a main concern with MRRs, and accuracy can be lost in the joints due to friction, backlash, wear and contamination. One of the main factors affecting the positional accuracy is the flexibility (ability to deform under an applied load) of the links and joints. The flexibility of reconfigurable link and joint modules depend mainly on four factors: the type of material used, module shape, size and the orientation of the module.

When considering the flexibility of MRR modules, there is a lot of work to be done. Quite often, the links and joints are considered as rigid bodies, and a simple kinematic analysis can be used to determine the position of the end-effector. However, when there is a large payload applied at the tip of the robot, large elastic deformations may occur, and the rigid body assumption may no longer be valid. Instead, the modules can be modeled as flexible bodies and the effects of deformations on the flexible bodies undergoing low-speed motion (quasi-static motion) can be studied using kinetostatic analysis. This involves creating the appropriate kinematic model and incorporating the stiffness or compliance parameters in order to solve for the tip deflections and ultimately, the flexible tip position in the robot's workspace.

With all MRRs, the number of different configurations can increase exponentially with an increase in the number of modules. Thus, the generation of the necessary kinematic equations can become increasingly difficult. Also, different configurations and positions can yield different magnitudes of the tip deformation. Therefore, when

performing a kinetostatic optimization for MRRs, whether optimizing the workspace volume, the shape of the modules, or the stiffness, it is highly favorable to account for all possible configurations. However, only feasible configurations are considered in order to reduce the total number of configurations, hence reducing the number of required computations.

This thesis focuses on the kinetostatic analysis and optimization of an open-chain MRR with any possible number of link and joint modules. To address the issues mentioned above, the objectives of this thesis are summarized as follows:

- Develop an easy to use method for module connectivity for a given number of links and joints. Afterwards, create a kinematic model that incorporates the connectivity method to determine the position, velocity and acceleration of an MRR in any given configuration.
- Investigate the effects of flexibility for reconfigurable joint-link modules under static loading and dynamic conditions. Define structural parameters for each component, and determine the type of flexibility based on their respective sizes.
- Create a kinetostatic model to obtain the tip deflections for a given configuration and orientation of an MRR. Then use post-processing to obtain additional information such as the local deflections, local forces and stresses of each module.
- Use the kinetostatic model to determine the maximum workspace boundary and volume. Develop criteria for obtaining the workspace boundary. Afterwards, for a given configuration, determine the position and orientation where the MRR has the highest tip deformation. Then, determine the maximum payload that the MRR can take.
- Optimize the cross section shape of the link and joint modules by using a genetic algorithm, or by explicitly solving for the unknown shape variables for a given payload and allowable tip deformation.

1.2 Existing Modular Reconfigurable Robot Designs

Reconfigurable robots can be classified depending whether or not their modules are identical. According to Castano and Will [4], a homogeneous robot has identical modules and the position of the module defines its function, whereas a heterogeneous robot has different modules and the function of the module defines the position of the robot. Also, lattice-based MRRs are usually homogeneous and require the modules to reconfigure in order to move, while non-lattice MRRs can reconfigure while moving [4]. Furthermore, reconfigurable robots can be classified according to the types of module reconfiguration they can undergo, namely: manual-reconfiguring, self-reconfiguring and self-assembly [5].

Manual-reconfiguring robots are perhaps the simplest type of MRRs since their modules can only be reconfigured and interchanged with manual assistance from a person. Common types of manual-reconfiguring MRRs include separate joint and link modules connected to each other to form an open-chain robot [2]. The joint modules are usually revolute or prismatic joints and the link modules are usually square beams with a constant square cross section. Li et al. [6] developed a self-reconfigurable robot consisting of interconnected link-type modules with pitch and yaw revolute joints at the ends of each link. Hamlin and Sanderson [7] developed the Tetrobot, a parallel reconfigurable robot that utilizes configurations that resemble tetrahedral truss structures consisting of passive and active joints and connector pins, where the passive joints do not have any motors or motion input in the robot, and the active joints contain motors used to actuate the robot. They tested manipulator, platform and walking configurations using only those three mechanical parts. The joints used were spherical joints which enabled the truss structures to move with a high number of degrees of freedom (DOF). Chen and Yang [8,9] used revolute joint modules connected to link modules to assemble their MRR in a SCARA configuration, as well as a "branched-tree" configuration, where the open chain robot has more than one end link, as shown in Fig. 1-1.

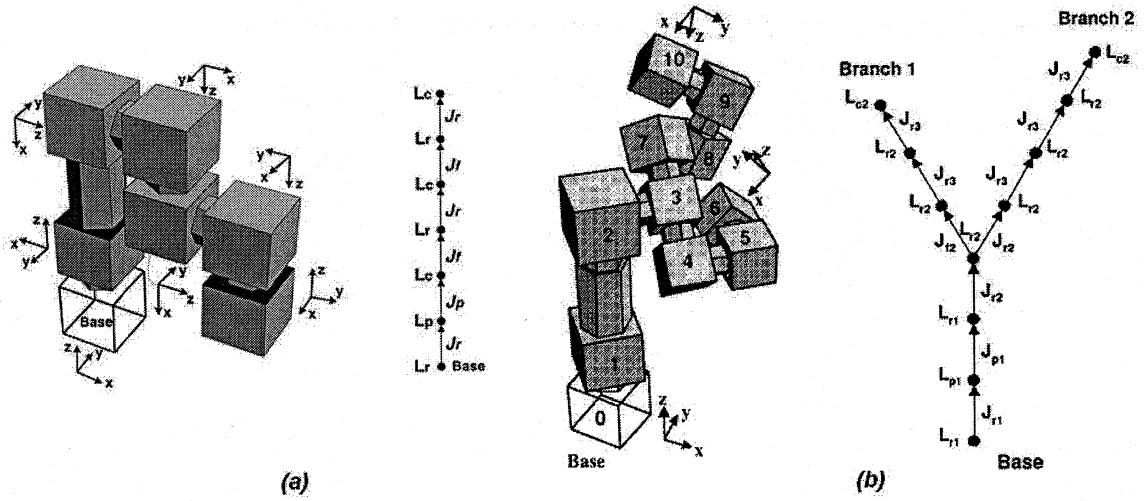


Figure 1-1: (a) SCARA configuration (from [8]), (b) Branched configuration (from [9])

Self-reconfiguring robots are initially assembled manually in order to perform a desired task. These robots can then automatically change their configuration without manual assistance to obtain a different shape and range of motion. Hafez et al. [10] developed modules that consist of a 3-DOF parallel stage connected to a rigid plate that attaches to another module forming a serial connection, as shown in Fig.1-2. Each module was driven using electromagnetic actuators and the modules connect to each other using magnetic plates. The use of parallel platforms as modules enables the robot to perform self-reconfiguration because the robot is hyper-redundant. In other words, there are a large number of unnecessary extra joints available to perform a desired range of motion. When the range of motion requirements change, the joints that were previously static may become active and the previous active joints may become static.

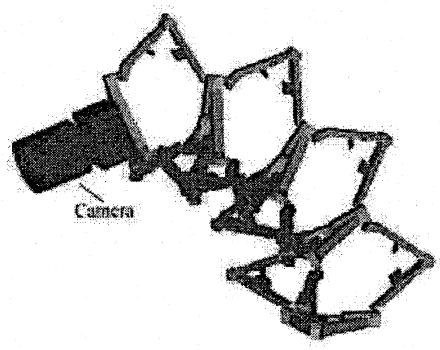


Figure 1-2: Self-reconfigurable robot (from Hafez et al. [10])

Li et al. [11] developed a self-reconfigurable robot consisting of interconnected link-type modules with pitch and yaw revolute joints at the ends of the link to form a mobile MRR with stair climbing abilities for search and rescue operations. The robot consists of three tracked modules, and reconfiguration occurs by the movement of the pitch and yaw revolute joints located at the ends of the link as shown in Fig. 1-3.

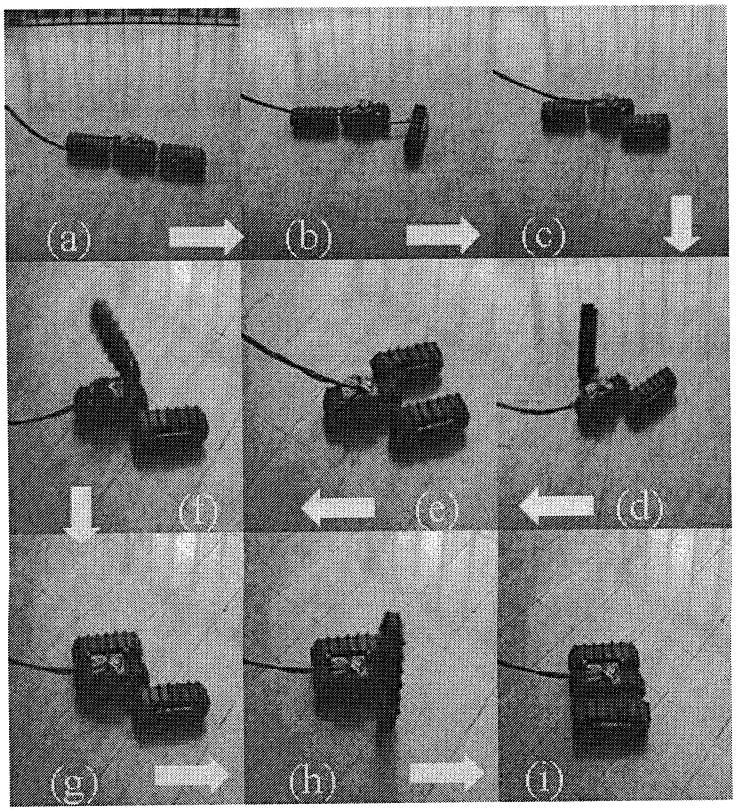


Figure 1-3: Mobile robot in the process of self-reconfiguration (from Li et. al [11])

Yim et al. [12] developed Polybot, which was the first self-reconfiguring MRR that demonstrated two distinct topologies for locomotion. The first configuration was a closed loop of interconnected modules, and the second configuration simply involved disconnecting one of the modules to form a snake-link chain. Wu et al. [13] designed a homogeneous lattice self-reconfigurable MRR which consists of cube-shaped modules that contain self-locking mechanisms. Each module can move its neighboring module by rotating one of its joints. To provide locomotion, one module acts as a carrier to its neighboring module (or passenger) and both the carrier and passenger modules must release their connections at some point to prevent locking before the next motion occurs.

Perhaps the most versatile MRRs are those that can perform self-assembly. These types of MRRs do not require manual assistance to assemble or reconfigure since their modules are able to attach and detach from one another automatically. For instance, Ünsal and Khosla [14] developed self-reconfiguring modules that consist of links with three revolute joints forming a 3-DOF manipulator that can connect to one of six faces on a separate passive cube shaped module. The active link modules can detach from one of the faces of a cube module and move to a different cube module, or can attach itself to a separate cube module and move the cube module attached at the other end, as shown in Fig. 1-4.

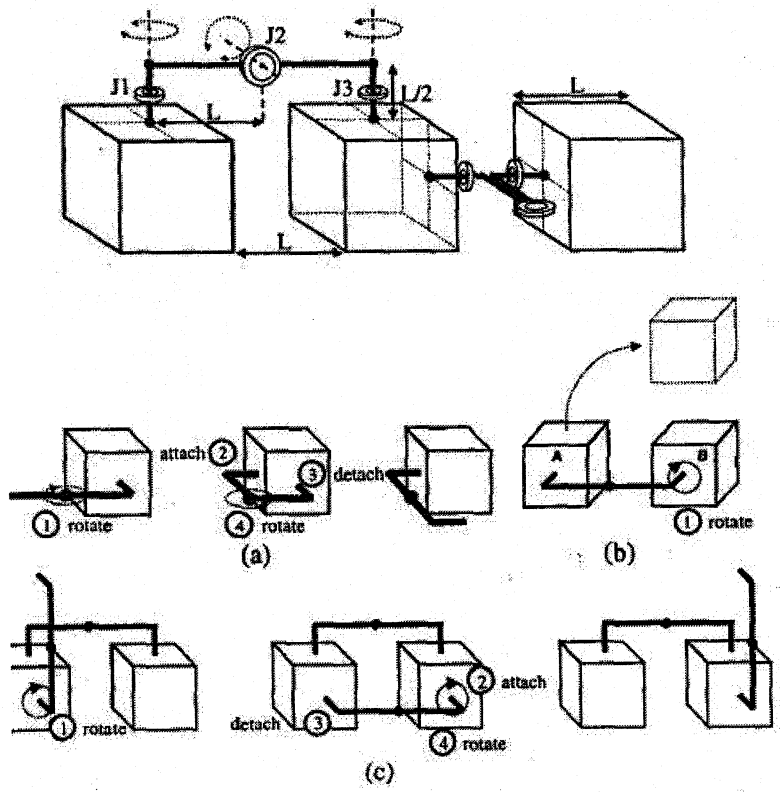


Figure 1-4: I-Cubes robot developed by Ünsal and Khosla [14]

Another example of a self-assembling MRR is the Conro robot developed by Castano and Will [4]. The Conro modules consist of three interconnected cubes with one yaw joint in between the first two modules and one pitch joint in between the second and third module, forming a snake-like structure. Connection interfaces exist at the extremities of the two end cubes, and each untethered module is entirely self-sufficient

and does not require another module to operate. Furthermore, the connection interfaces of each Conro module contain self-locking spring mechanisms which provide an entirely mechanical locking process. Tomita et al. [15] developed a self-assembling robotic system with modules that used connecting surfaces with permanent magnets that can attract and repel other modules by controlling the magnetic forces between modules. In addition, Suh et al. [16] designed a 6-DOF reconfigurable cube-shaped module consisting of six prismatic joints that expand each of the module's connecting faces outwards in order to provide motion. Utilizing the prismatic DOFs along with magnetic faces for connection, each module is able to attach and detach to another module by expanding or contracting one of its faces to form (or separate) a reconfigurable lattice structure.

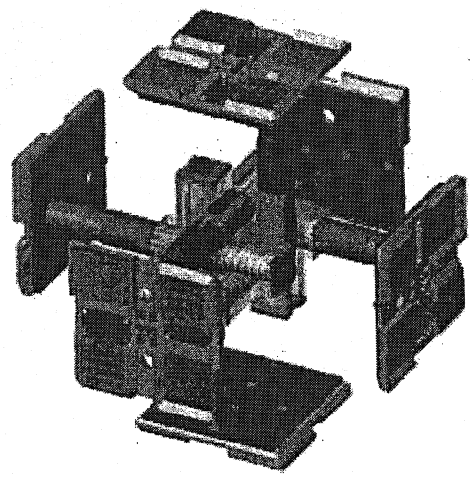


Figure 1-5: Telecube module, designed by Suh et al. [16]

In this thesis, the modules are manually reconfigured by the user and the joints can be either homogeneous or heterogeneous, depending on the size requirements and functionality of the MRR. There are separate revolute joint modules, link modules and possible prismatic joint modules. The revolute joint modules can be configured to rotate about one of six possible axes, and the link modules can connect to one of three faces of a joint module. After the modules are assembled, the joint-link connections can form a serial robot. The system description is discussed in more detail in the following chapter.

1.3 Overview of Relevant Research

To aid in the objectives of this thesis, considerable research was conducted in three categories. The first topic researched was about the different kinematic modeling methods available for MRRs; the second topic was about the flexibility of links and joints in typical robots; and the third topic researched was on the kinetostatic modeling and optimization of robots.

1.3.1 Modular Reconfigurable Robot Kinematics

The kinematic equations of an MRR can become quite complicated for an increasing number of modules. Also, with an increasing number of modules, the coordinates and connectivity of each module using a relative body-fixed coordinate system can become increasingly difficult to define. Therefore, computational methods are needed to solve the kinematic equations. Kelmar and Khosla [2] derived a method for the automatic generation of Denavit-Hartenberg kinematic parameters to obtain the forward kinematics for a serial MRR consisting of revolute joint and link modules. Forward kinematics solves for the position and orientation of a desired point along the robot given the joint motion angles. Inverse kinematics solves for the joint angles given the tip position and orientation in global Cartesian coordinates. Their model assumed that the rotation axis of the joint module was aligned with the link's z-axis, and the forward kinematics were solved by multiplying successive transformation matrices that contained the Denavit-Hartenberg parameters. However, MRR joint modules should be able to align with other available axes of the link as well. Therefore, Denavit-Hartenberg kinematic parameters are not sufficient for MRRs.

In addition, Kelmar and Khosla [2] connected each module using body-fixed relative coordinates as shown in Fig. 1-6. The orientation of each module was predefined using a rotation matrix. However, the D-H-parameters could not solve for all configurations of the MRR because of different reference frames, and they needed to create a transformation for the D-H parameters to coincide with the module coordinate frame. Also, with a high number of modules, it can be increasingly difficult to configure and connect each module using relative body-fixed coordinates. Fei and Zhao [17]

created the forward kinematics for link and joint modules in a similar fashion, but used recursive methods instead of D-H parameters. Again, the method of connectivity for each module was based on using body-fixed relative coordinate systems. The orientations of the modules were predefined using transformation matrices and group theory to classify the types of modules.

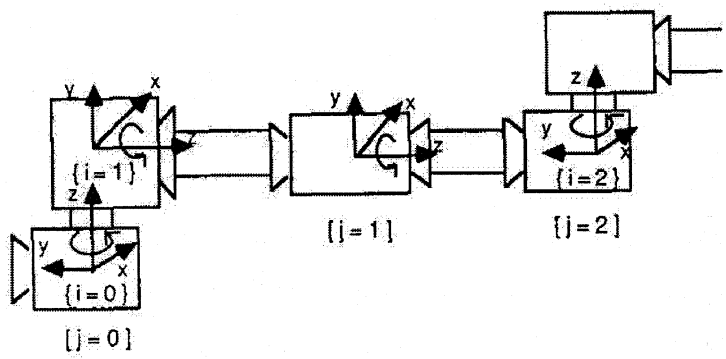


Figure 1-6: Module connectivity used by Kelmar and Khosla [2].

Chen and Yang [1] developed the recursive forward kinematics of prismatic and revolute joint and link modules using the Product-of-Exponentials formula. This forward kinematics model was configuration dependent and the Product-of-Exponentials formulation was uniform in representing revolute and prismatic joints for an open-chain robot. Also, this type of formulation allows for configurations when the joint axes are parallel and require no further modifications to the kinematic equations, such as those needed using D-H parameters. Also, they used a zero reference position method where each module is configured in terms of the joint axes directions and locations with respect to the zero reference position. Afterwards, Chen et al. [9,18] used the Product-of-Exponentials formula, along with a Newton-Raphson iterative method to solve for the inverse kinematics of serial and tree-branch configured MRRs, along with a three-leg parallel MRR.

In this thesis, the kinematic equations will be developed for an open-chain MRR with prismatic and revolute joints. The kinematic equations will also take the revolute joint length into account, and connectivity methods are presented to consider angular offsets and include relative coordinate module connectivity as well as two global

connectivity methods. Also, instead of defining each module's initial position and orientation by defining body vectors and rotation angles first, a simple method was developed to allow the modules to be configured according to coordinate system direction, in other words, configured according to the x, y or z direction.

1.3.2 The Flexibility of Joints and Links

Previously, flexible joint and link models were represented as torsional springs and cantilever beams, respectively. A number of researchers used models with a flexible link and rigid joint or flexible joint and rigid link, or both flexible links and joints. Anderson [19] studied the stability of a 2D manipulator with rigid links and flexible revolute joints by performing a free vibration analysis. The equations of motion were derived using Lagrange's equation. Tang and Wang [20] assessed the effects of link deflections and joint compliance on the overall positioning of manipulators by calculating the link displacements from classical beam theory, and approximating the joint compliance with a torsional spring. Martins et al. [21] created a dynamic model for a manipulator with flexible links and rigid joints by representing the links as Euler-Bernoulli beams. Howell and Midha [22] modeled flexible links by using a pseudo-rigid body model. This model assumes that large-deflection links follow a near-circular path at the tip and is represented by two rigid links connected by a non-linear torsional spring joint. Another pseudo-rigid body model that represents a flexible link contains up to six pseudo-joints to allow the required link deflections found in MRRs [23].

The finite element method has been used to model flexible links and joints and has proven to be accurate. Flexible revolute joints were not the only joint types to be considered for flexibility. Torby and Kimura [24] derived the equations of motion for a flexible link manipulator with prismatic links by representing each link as two 3-D finite elements and applying moving boundary conditions to represent the motion of a prismatic joint. Fallahi et al [25] developed a finite element to represent the flexible links of a crank-slider mechanism at high speeds; however, the joint flexibility was ignored. Z.C. Lin and D.T. Lin [26] studied the dynamic deflection of a two-link planar manipulator using a beam developed from Timoshenko beam theory, and again, the joint flexibility

was ignored. Shigang et al. [27] developed a flexible rotor beam element coupled with a flexible link element to account for joint flexibility. Their flexible joint element accounted for torsion in the joints only.

Yang and Sadler [28] studied the effect of joint flexibility on the dynamic performance of a completely flexible manipulator by using a joint-link finite element that consisted of two Euler beams connected to a shaft that allowed only torsional bending and neglected joint and axial deformation. They noted that the ratio of joint to link stiffness determines the overall deflection of the system and that there exists ranges where the either the torsional, bending, or combined torsion and bending responses are dominant. Also, the coupling effect of a flexible link and flexible joint was investigated and the link to joint stiffness and mass moment of inertia ratios play a role in determining the overall system flexibility, and there exists three regions of system softening, stiffening, and mixed flexibilities [29].

The above papers did not mention when the joint or link is flexible for different link and joint configurations. Also, the effects of possible joint bending and axial stretching were ignored. This thesis addresses the issue of joint and link flexibility by developing flexible link and joint models and comparing these models to rigid body models over a range of joint and link sizes and different joint configurations to determine when exactly the joint or link can be considered flexible. Details of the flexibility analysis are given in Chapter 4 of this thesis.

1.3.3 Kinetostatic Modeling and Optimization

The kinetostatic modeling for an MRR is necessary to predict any deflections if the link or joint modules elastically deform under an applied load for given positions and orientations of each module. Also, the kinetostatic modeling can be used to assess the kinetostatic performance of a manipulator by testing different positions in a given workspace. Kim and Ryu [30] developed a kinetostatic Jacobian matrix to account for dimensional inconsistencies found in velocity Jacobians, and then used the condition number of the modified Jacobian to determine the optimal position for a 6-DOF Stewart

platform. Similar work was done by Chablat et al. [31] for a 3-DOF parallel kinematic machine to find the largest dextrous workspace (workspace enclosed in a cube, cylinder or any other known 3-D shape). Chablat and Angeles [32] determined an index for the kinetostatic performance of a revolute-coupled planar manipulator by comparing the velocity Jacobian to a given isotropic matrix. Also, Khan and Angeles [33] optimized the D-H parameters of a serial robotic manipulator by directly minimizing the condition number of a dimensionally homogeneous Jacobian. It is important to note that the methods used in [30-33] involved rigid-body kinematics and did not involve the inclusion of any forces, stiffness terms, or displacements.

To account for link elastic deformations while a robot is undergoing quasi-static motion, Xi and Fenton [34] created a kinetostatic model which was based on the algebra of rotations. The links were modeled as Euler beams and the computational method involved using three Jacobians (one for rigid body motion, one for initial link deflections, and one for link bending variation). The proposed algorithm used two loops, one for solving the joint motions, and the other for solving link deflections. The algorithm used to solve the kinetostatic equations were then reduced to a single-loop problem and utilized the Newton-Raphson method [35]. Kim and Choi [36] assessed the kinetostatic capability of parallel and serial mechanisms to find the extreme magnitudes of the forces (or velocities) acting on the joints. They derived the forward and inverse kinetostatic capability equations, and solved the eigen-problem which yielded the force bounds. The forward analysis finds the forces (or velocities) at the tip of the robot, and the inverse analysis finds the forces (or velocities) at the robot joints. However, they did not include any elastic link or joint deformations in their analysis.

Kinetostatic methods that included the stiffness of the links and joints were created also. Roy et al. [37] used the finite element method to design a two-link robotic arm to have high static strength, high vibration frequencies and low weight. They tested a few joint angles of the robot and obtained the natural frequencies, then compared them to experimental results. Mroz and Notash [38] used finite element software to design a wire-actuated robot, and tested various directions for the end-effector force to obtain the end-

effector displacements. Xi et al. [39] used a compliance kinetostatic model instead of the stiffness for a parallel kinematic machine. They then created two global compliance indices to measure the compliance over the robot's workspace. Afterwards, a genetic algorithm was used to optimize the stiffness and the workspace of the same robot [40].

In this research, two kinetostatic models were presented to determine the tip deformation with an applied payload at the tip for a given configuration, position and orientation of a MRR. The first model uses the compliance and Jacobian for each module, and this was compared to the second model, which used finite element method (FEM). Post processing of both methods were also defined in order to solve for the local displacements, forces, and stresses. Afterwards, the workspace of an MRR with a given number of modules was determined for a number of configurations, along with the maximum displacement position and configuration, and the maximum allowable payload. After those design points were determined, an optimization was performed using a genetic algorithm to determine the required thickness of the module casings. The design problem was also solved using a direct method.

1.4 Outline of Thesis

Along with the introduction, this thesis is organized into eight chapters.

Chapter 2 contains a system description of the type of MRR used, and a description of the revolute joint module design.

Chapter 3 presents the kinematic equations of the MRR and introduces a new connectivity method for link and joint modules. In the initial configuration setup, the user simply defines the directions of each module using one three frames of references. The kinematic equations were then created to include revolute and prismatic joints as separate modules, along with links.

Chapter 4 discusses the flexibility of joint and link modules. Three static models were created to assess the flexibilities of each component. The first model is a hybrid FEM

model where flexibilities in both the joint and link are accounted for. This model was then compared to a rigid joint-flexible link model, and a rigid link-flexible joint model using size ratios to determine the flexibility in each component. Afterwards, a dynamic analysis was performed using the three flexibility models to determine the dynamic flexibility of the link and joint. A set of graphs were then created for the static and dynamic test cases to show the flexibility of the link or joint shaft over a range of sizes.

Chapter 5 discusses two kinetostatic models that can be used in performing a kinetostatic analysis. The two models are developed to solve for the tip deformation for a given configuration, position, orientation and applied tip load and compared to each other using a test case.

Chapter 6 discusses the applications of the kinetostatic model. First, the maximum deflection is determined using two methods. The first method involves searching the workspace to obtain a workspace visualization for each configuration and storing the maximum resultant tip deflection for an applied load. The second method involves optimizing a performance index to obtain the maximum deflection position. Afterwards, the maximum allowable payload can be determined at the maximum deflection position using the bisection method. To avoid testing all possible configurations, an enumeration scheme is presented to eliminate any infeasible configurations.

Chapter 7 involves using a genetic algorithm to optimize the thickness of the joint and link module casings using different size constraints. The results were compared to a direct method to solve for the thicknesses explicitly. The latter method involves using a least-squares pseudoinverse to approximate the local module deflections.

Chapter 8 provides the conclusions and discusses the future work.

2. System Description

This chapter describes the mechanical system description of the MRR. The type of MRR used in this thesis consists of separate revolute joint, prismatic joint and link modules. When the MRR is assembled, the system is driven by the joint modules. Each revolute joint module can be manually reconfigured to provide rotation about one of three possible axes (±x, ±y, and ±z). Successive revolute joint and link modules are connected to each other by attaching one end of a link module to one of the flat faces of the previous joint module. One of the connection interfaces of the subsequent joint module is then connected to the free end of the previous link module, and so on, forming a serial robot. Also, this type of MRR allows joint-joint serial connections instead of assembling joint-link serial connections since the joints are longer than typical robot joints. The revolute joint and link modules are discussed in further detail in the following sections.

2.1 Revolute Joint Module

The revolute joint module consists of two parts, the gear section and the motor section. The main parts of the motor section consist of the motor, brake, encoder, motor driver, and a square tube outer casing, as shown in Fig. 2.1. The brushless DC motor is the component that drives the revolute joint and allows any subsequent modules to rotate. This part was attached to the top of the motor casing. At the bottom end of the motor is a brake that was attached to the lower shaft extrusion of the motor. Located below the brake is an incremental encoder, which was used as a feedback device to measure the angular position of the motor shaft. Also, a motor driver was attached to one of the flat faces on the outside motor casing and was used to control the DC motor with feedback from the motor and encoder.

The main parts in the gear section consist of the harmonic drive, bearing, shaft extension and a cylindrical outer casing. The harmonic drive consists of a flexspline, a circular spline and a wave generator. In Fig. 2.1, the flexspline was fixed to the bottom of the gear casing, so the motor shaft drives the wave generator and output comes from the circular spline. This harmonic drive configuration acts as a gear reducer to slow the speed

of the motor shaft. The circular spline component of the harmonic drive was then attached to a shaft extension, which was surrounded by a bearing that can take axial and transverse forces. A top plate was then attached to the shaft extension which was used as a connection interface to attach other modules. The top connector plate also protected the bearing from possible interference from any large objects entering the gear casing.

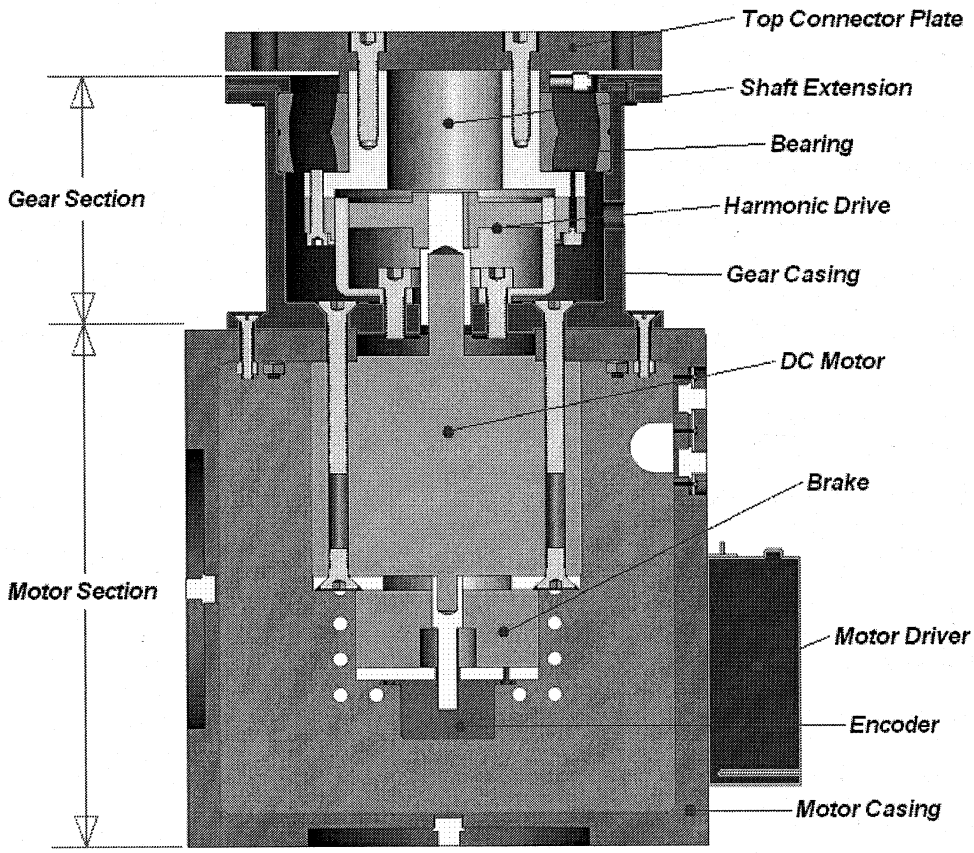


Figure 2-1: Major components of revolute joint module

Along with the top connector plate, the revolute joint module has two other connection interfaces located on two sides of the motor casing as shown in Fig. 2-2. This allows the joint to provide rotation output to the next module about one of three possible axes (±x, ±y, and ±z). In other words, the axis of rotation of the joint can be either parallel or perpendicular to the axis of rotation of the previous revolute joint module in the open chain. Also, on the two connection interfaces located on the motor casing, there are two identical input ports, which can be connected to the central controller and the

motor drivers. Furthermore, there were two removable doors which covered the open space on opposite ends of the square tube shaped motor casing. The doors allow for easy inspection, maintenance, and possible removal of the motor components. Also, a DC fan was installed on the front door, while the other door contained distributed holes for sufficient cooling and ventilation of the motor components.

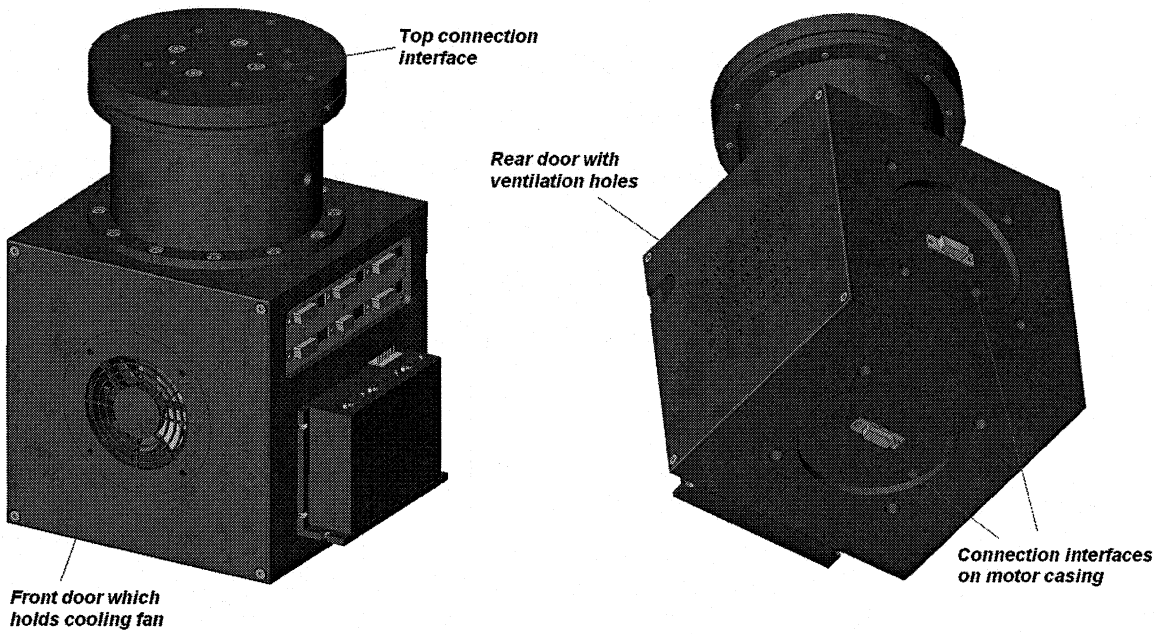


Figure 2-2: Revolute joint module showing the three connection interfaces

2.2 Link Module

The link modules are straight beams and have a hollow cross section area and the revolute joint modules can be attached to either end of the link. At one end of the link there exists an integrated connection interface, however, at the opposite end of the link, a separate plate is required to act as a connection interface to connect other link or joint modules. There is also a base link which can be connected to the ground or an experiment table. A connection interface is attached to the top of the base link in order to allow connectivity to other modules. In addition, the base link does not move and is assumed to be a rigid body with no deformation. The link components are shown in Fig. 2-3.

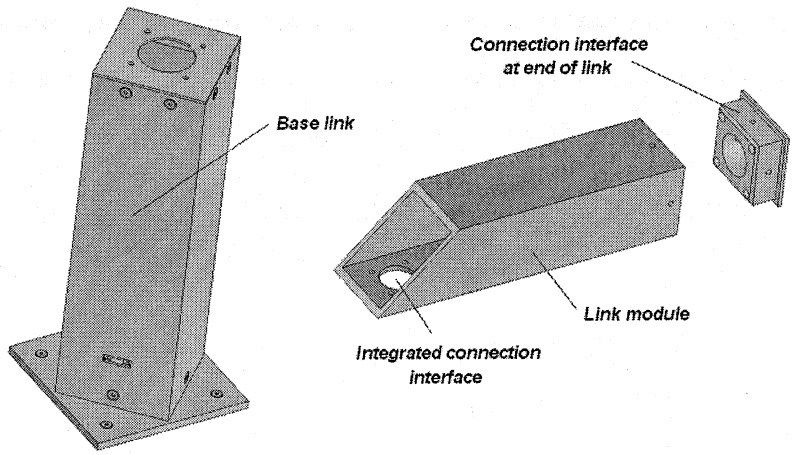


Figure 2-3: Base link and link module

2.3 MRR Assembly

To assemble the MRR, the base link is first attached to the ground, and the corresponding revolute joint model is attached to the base using one of the connection interfaces located on the sides of the motor casing, and is secured using bolts. Afterwards, the tapered end of a corresponding link module can be secured to the top plate connection interface of the joint module with four bolts as shown in Fig. 2-4. On the opposite end of the link, a separate plate is first attached to the corresponding revolute joint module in the open chain and is then attached to the end of the link, as shown in Fig. 2-5. Also, the wires needed to supply power to and connect other revolute joint modules can pass through the hollow link module. With the revolute joint modules, it is also possible to connect two or more joint modules together without the need of a link in between since the joints are longer than conventional joints. In addition, the size of different modules can vary and can also be scaled relative to each other to allow for higher or lower module weights and different payload capacities. Fig. 2-6 shows an assembled MRR in a spatial motion configuration with three link and revolute joint modules. It is important to note that this MRR design did not include prismatic joints at the time this thesis was written. Furthermore, for this design, the links cannot be connected to the revolute joint to form an axial joint-link configuration where the length of the link is aligned with the preceding joint's axis of rotation. However this type of configuration, along with prismatic joints,

and solid rectangular links (instead of links with hollow cross sections) are accounted for in the kinematic, flexibility and kinetostatic analyses in the following chapters.

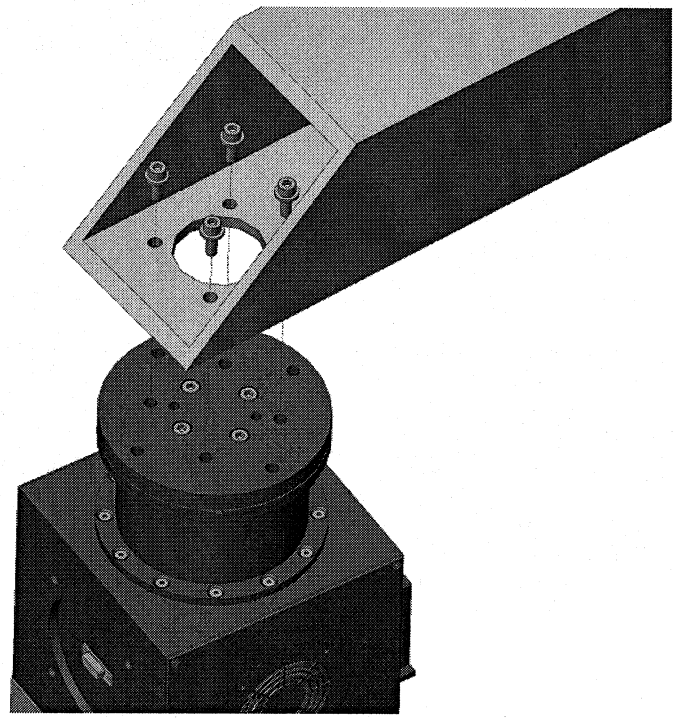


Figure 2-4: Tapered end of link attaching to revolute joint module

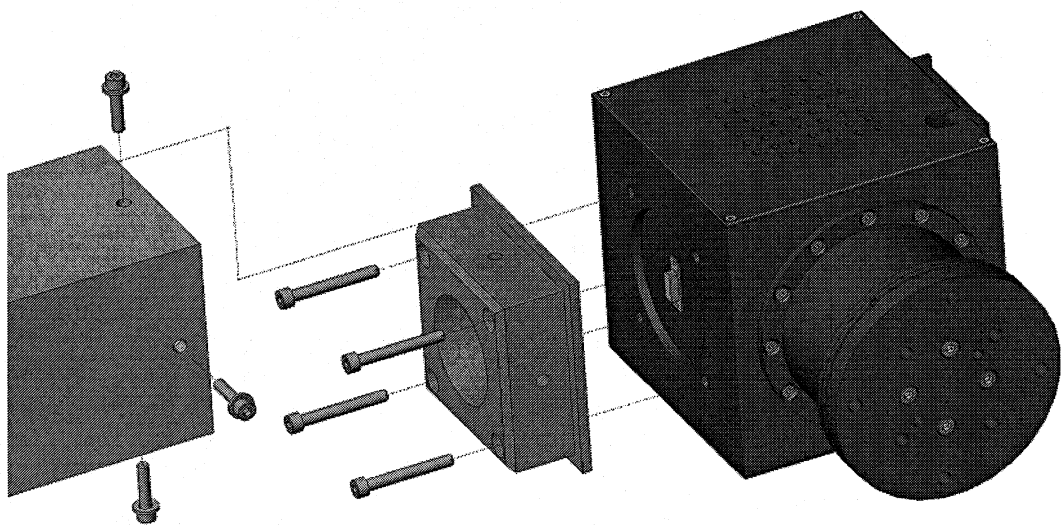


Figure 2-5: Connector plate attaching to joint and then attaching to square end of link

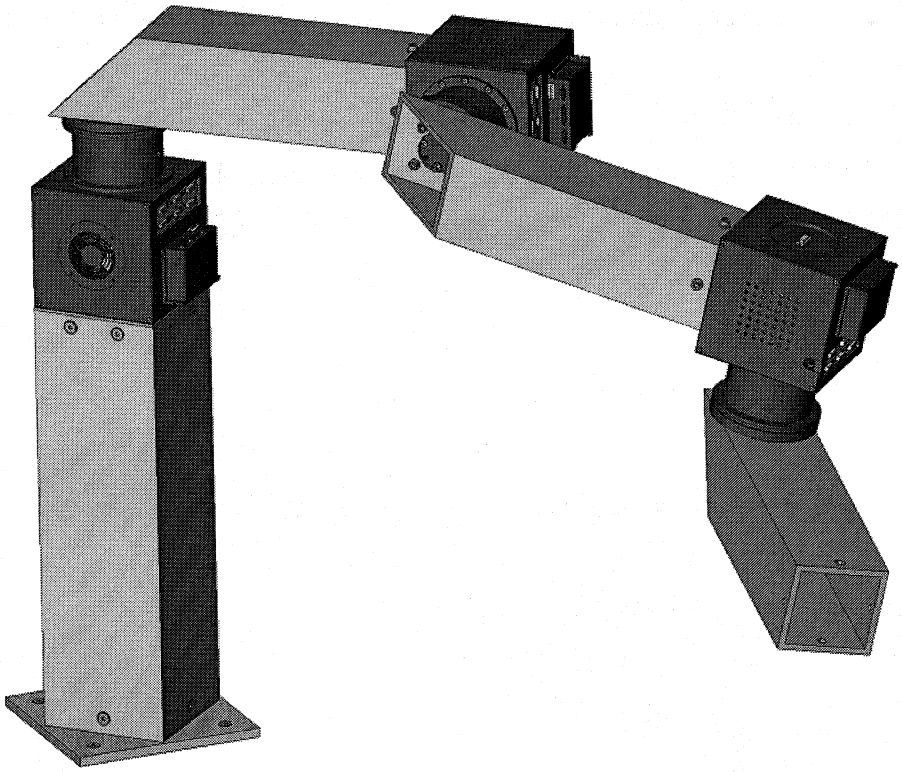


Figure 2-6: Assembled MRR with three revolute joints and link modules

3. MRR Kinematic Computational Method

This chapter describes the forward kinematics of an MRR with rigid bodies. The kinematic analysis is concerned with the geometric motion of multibody systems, and not the forces which cause that motion. The first step is the position analysis, where all the positions and orientations of each body are determined for an initial and final position. The next step is the velocity analysis, where the linear and angular velocities of each body are determined. This can be done by differentiating the position equations with respect to time. The last step is the acceleration analysis and is done by differentiating the velocity equations with respect to time. However, since the kinetostatic analysis methods presented in Chapter 5 assume that the MRR undergoes quasi-static motion, the velocity, as well as the acceleration, can be neglected.

In the following sections, recursive methods (instead of the constraint equation based method from Shabana [41]) are used to determine the position and orientation of a MRR since it is easier to define module connectivity and computationally less expensive. Also, the equations used in the following sections are left parameterized to allow for an infinite number of joint and link module configurations. In addition, the lengths of the joints are also taken into account since some MRRs have longer revolute joint module sizes than typical robots. The longer joint sizes can create a noticeable difference in the kinematics when the joint lengths are ignored and modeled as simple nodes that connect links.

3.1 Rotation and Translation of a Single Body

This section first shows the equations for the position and orientation, often referred to as the pose, of a single body in order to aid in the definition of the rotation matrix, then the connectivity of MRR modules is explained and the forward position and orientation equations are developed in the latter sections. For all robots, the main types of motion are rotation and translation. The rotations are defined with respect to a coordinate system, either global (space-fixed) or relative (body-fixed) coordinates. Thus, in robotics, a rotation matrix is used to transform the coordinates of a body to a more convenient

coordinate frame. An arbitrary body in 3-D space undergoing rotation and translation is shown in Fig. 3-1.

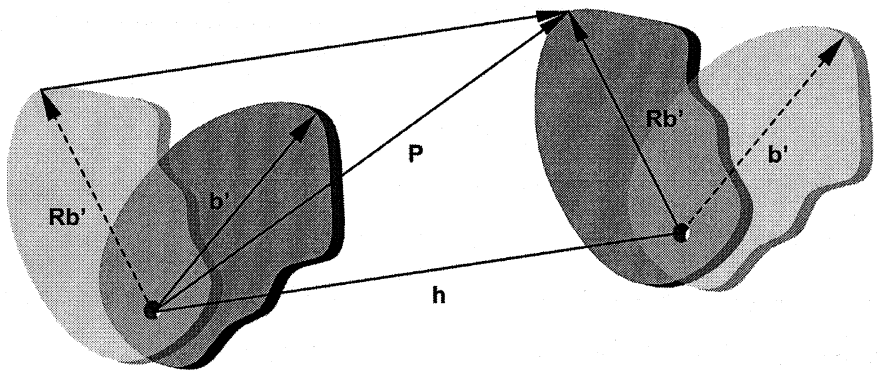


Figure 3-1: Rotation and translation of a single body

The tip position vector of the body, **P**, shown in figure 3-1 is due to rotation and translation of the body and is given by:

$$\mathbf{P} = \mathbf{Rb}' + \mathbf{h} = \mathbf{h} + \mathbf{Rb}' \tag{3.1}$$

where **b'** is the local body vector, **h** is the translation vector and **R** is the rotation matrix which transforms the body vector from the body-fixed frame to the desired global coordinate reference frame. When **h** is equal to zero, there is no translation and the body undergoes pure rotation. The position vector, **P,** can also be written as:

$$\mathbf{P} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}} = \mathbf{b} \tag{3.2}$$

where x, y and z are the Cartesian coordinates of the tip position of a single body after movement and **b** is the body vector in terms of the desired global coordinate frame. To transform a local body vector to global coordinates in terms of pitch, roll and yaw angles the following equations can be used [42]:

$$\mathbf{R}(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \tag{3.3}$$

$$\mathbf{R}(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{3.4}$$

$$\mathbf{R}(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

where θ_x, θ_y, and θ_z are the rotation angles about the x, y and z axis, respectively. Combining equations (3.3) to (3.5), the rotation matrix to convert a body in 3-D space from local coordinates to global coordinates is given by:

$$\mathbf{R} = \mathbf{R}(\theta_x)\mathbf{R}(\theta_y)\mathbf{R}(\theta_z) \tag{3.6}$$

Equation 3.6 can be written explicitly as:

$$\mathbf{R} = \begin{bmatrix} \cos\theta_y\cos\theta_z & -\cos\theta_y\sin\theta_z & \sin\theta_y \\ \sin\theta_x\sin\theta_y\cos\theta_z + \cos\theta_x\sin\theta_z & -\sin\theta_x\sin\theta_y\sin\theta_z + \cos\theta_x\cos\theta_z & -\sin\theta_x\cos\theta_y \\ -\cos\theta_x\sin\theta_y\cos\theta_z + \sin\theta_x\sin\theta_z & \cos\theta_x\sin\theta_x\sin\theta_y + \sin\theta_x\cos\theta_z & \cos\theta_x\cos\theta_y \end{bmatrix} \tag{3.7}$$

To convert global coordinates to local coordinates (body-fixed frame), the following equation can be used:

$$\mathbf{R}^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}}(\theta_z)\mathbf{R}^{\mathrm{T}}(\theta_y)\mathbf{R}^{\mathrm{T}}(\theta_x) \tag{3.8}$$

Furthermore, the pitch, roll and yaw rotation angles can be obtained using the following equations:

$$\begin{aligned} \theta_x &= \cos^{-1}\left(\frac{r_{33}}{\cos\theta_y}\right) \\ \theta_y &= \cos^{-1}\left(\sqrt{(r_{23})^2 + (r_{33})^2}\right) \\ \theta_z &= \cos^{-1}\left(\frac{r_{11}}{\cos\theta_y}\right) \end{aligned} \tag{3.9}$$

where r_{33} is the third row, third column entry of the rotation matrix in equation (3.7).

Similarly, r_{23} is the second row, third column entry of the rotation matrix in equation (3.7). In the following sections, equation (3.6) is used to setup the initial configuration of the modules, define angular offsets, and determine the final orientations of each module.

3.2 Initial Configuration Setup for Each Module

The first step in solving the kinematic equations for an MRR requires assembling each module in its initial desired position and orientation called the initial configuration setup. In this thesis, the concept of *directional configurability* is introduced. Here, instead of entering the position and orientation of each module by manually defining the body vectors for position and angles for orientation, each module can be initially setup simply by entering the module length and the required direction of the length about a selected reference frame. For example, in a kinematics computer program, if the length of the module is to be aligned with the x-axis of a given reference Cartesian coordinate system, the user simply enters the length of the module, then enters "x" for the direction of the module. The length of the module will then be aligned with the x-axis of the reference coordinate system in the same direction. Fig. 3-2 shows the directional configurability of a link module with respect to a reference Cartesian coordinate system.

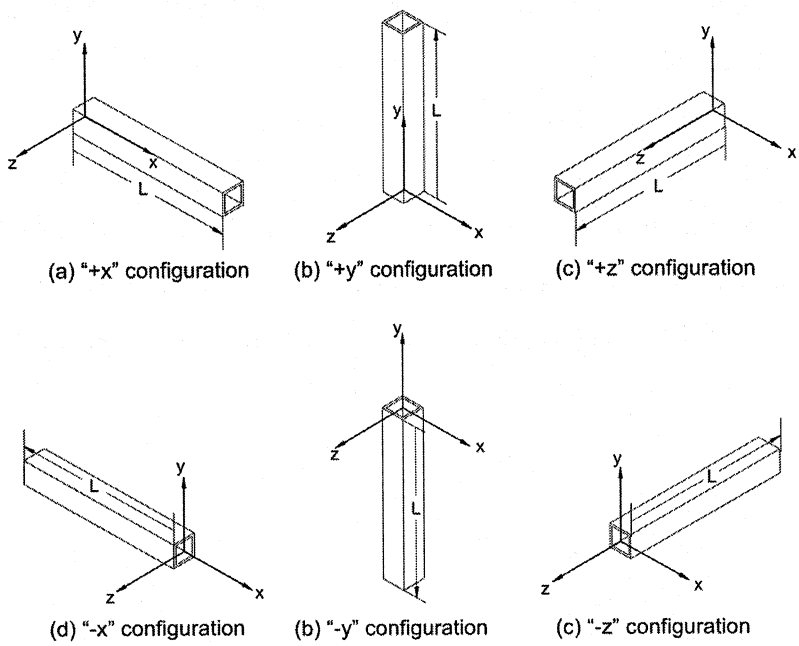


Figure 3-2: Directional configurability for a single link

The module's x-direction initially corresponds to the length of the module, and each of the six directional configurations are accomplished through the use of a configuration rotation matrix with known angular rotations for each configuration. The directional configurability is determined by using the rotation matrix found in equations (3.6) or (3.7) to calculate the configuration rotation matrix.

$$\mathbf{R}_{config} = \mathbf{R}(\theta_{x,config},\ \theta_{y,config},\ \theta_{z,config}) \tag{3.10}$$

where $\theta_{x,config}$, $\theta_{y,config}$, and $\theta_{z,config}$ are the known angles for each configuration. Each of these angles for the six configurations shown in Fig. 3-2 are listed in the table below for a space-fixed reference frame (using space-fixed rotations, equation (3.6)).

Table 3-1: Rotation input angles for each configuration

Configuration Direction	$\theta_{x,config}$ [deg]	$\theta_{y,config}$ [deg]	$\theta_{z,config}$ [deg]
x	0	0	0
y	0	0	90
z	0	-90	0
-x	0	180	0
-y	0	0	-90
-z	0	90	0

As shown in Fig. 3-2, the directional configurability method is ideal for modules that have six directions of configurability, such as cube-shaped modules. However if the modules do not have six orthogonal directions of configurability, or there exists angular offsets such as a base module resting on an uneven table or uneven ground, or there are connector plates that allow the modules to be at angular offsets, then these offset angles must be accounted for in the initial configuration setup. Therefore it is necessary to define three offset angles about the x, y and z axis of the selected reference coordinate frame, and another rotation matrix must be used in the kinematic equations. This rotation matrix is known as the *initial configuration setup rotation matrix* and is given as:

$$\mathbf{R}_{ICSU} = \mathbf{R}(\theta_{x,offset},\ \theta_{y,offset},\ \theta_{z,offset}) \tag{3.11}$$

where $\theta_{x,offset}$, $\theta_{y,offset}$, and $\theta_{z,offset}$ are the offset angles about the x, y and z axis of the reference coordinate system, respectively, and **R** is the rotation matrix found in equation (3.6) or (3.7). Throughout this thesis, the revolute joint length is the aligned with its axis of rotation for joint motion and can be taken as the length of the motor shaft, or the length

of the joint casing. Also, the prismatic joint length is aligned with its direction of extension (or retraction) and changes with motion. It is necessary to model the revolute joints as separate bodies to take into account possible long joint lengths in the position equations. To aid in the connectivity of the modules, a simple graphical representation for module connectivity was created using MATLAB and is shown in Fig. 3-3. Here, the user first inputs the base vector for the static base link, then inputs the configuration direction (from Table 3-1) and any offset angles for use in equation (3.11) beginning with the module connected to the base link, then continuing with the rest of the modules until the end of the serial chain is complete.

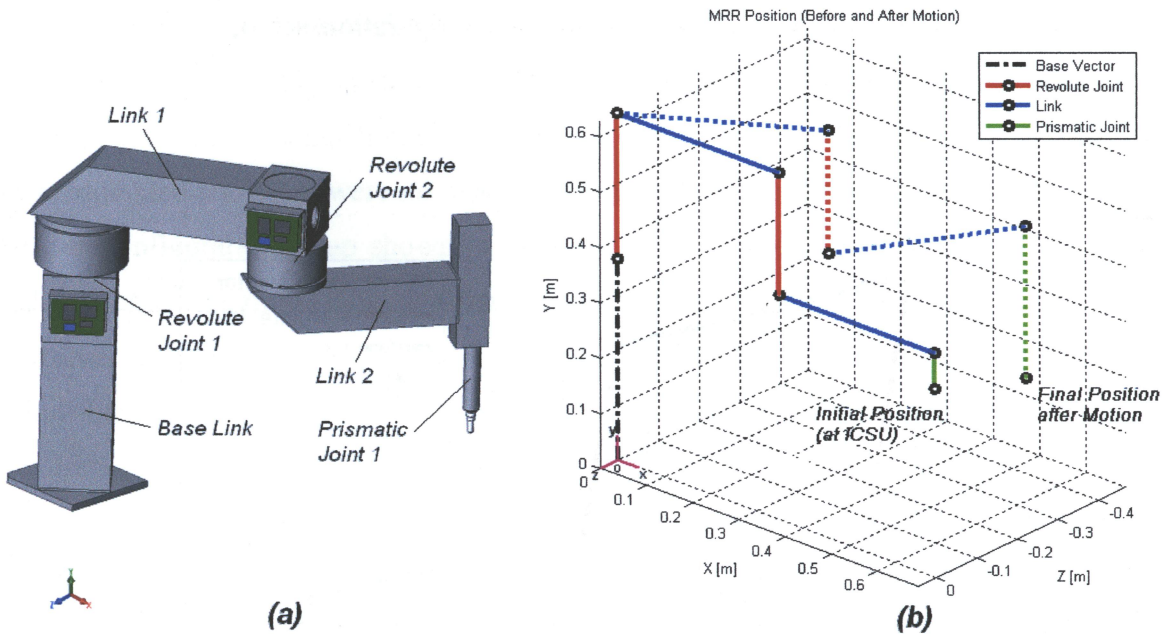


Figure 3-3: (a) CAD model of MRR, (b) Beam model of MRR in MATLAB

In Fig. 3-3 (b), each module is represented as a beam with two nodes and each module connects to the other modules at the nodes. The base link is represented as a black beam with dashed lines, revolute joint modules are represented as red beams, prismatic joint modules are represented as green beams and link modules are represented as blue beams. Also, the solid line model in Fig. 3-3 (b) represents the MRR at its initial configuration and the dotted line model represents the MRR pose after joint motion. In the following sections, three methods are described for MRR module connectivity and the equations to determine the static and motion pose for each method is presented. While

configuring each module, a plot like Fig. 3(b) can be created in MATLAB to show each module's local coordinate system using one of the three methods described below.

3.3 Position and Orientation using Relative Connectivity

The relative connectivity method assumes that each module has a fixed coordinate system at its origin, where the length of each module always coincides with its local x-direction. The next corresponding module is then connected to the previous module using the directional configurability method with the reference frame located at the previous module's origin. In a kinematics program, it is necessary to visualize each module's local coordinate system in order to aid with the initial configuration setup, as shown in an example in the figure below.

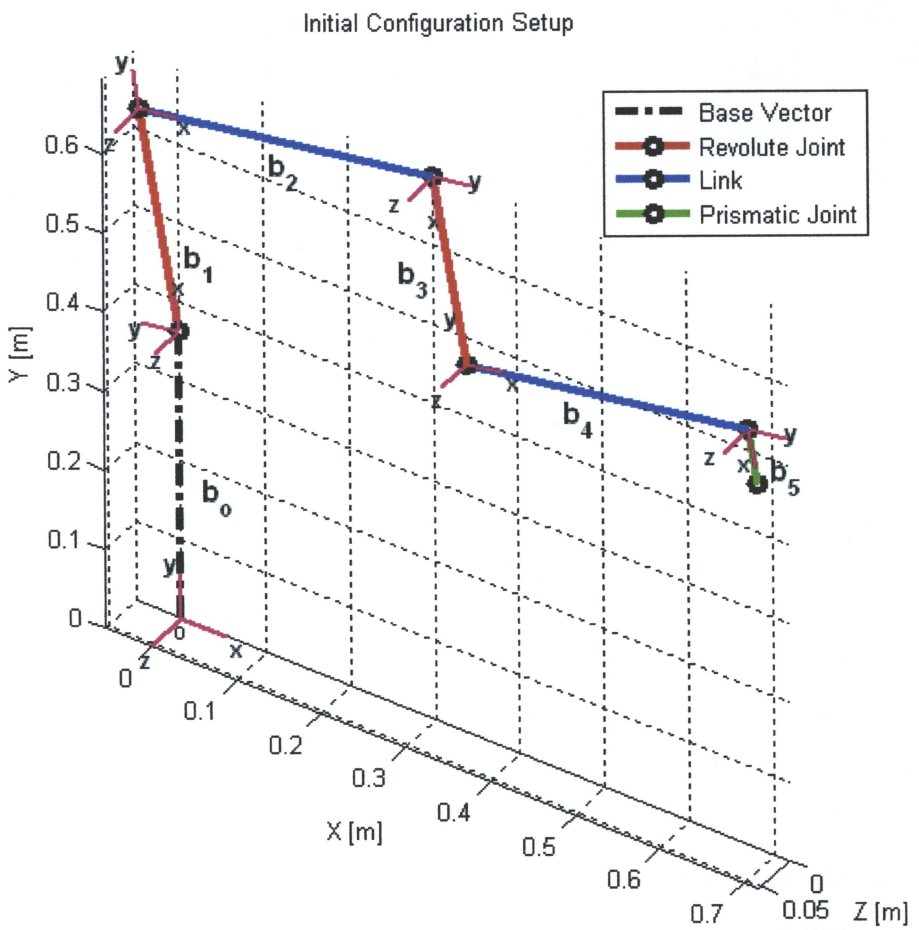


Figure 3-4: Schematic representation of initial configuration setup using relative connectivity

The first step for module connectivity is to determine the number of link and joint modules. The next step in setting up the initial configuration is to enter the base link body vector in global coordinates, which is denoted as $\mathbf{b}_0$ throughout this thesis. The kinematic equations in this thesis are for n number of bodies. As shown in Fig 3-4, the coordinate system of the base link always coincides with the global coordinates. The next revolute joint module (body vector $\mathbf{b}_1$) was then aligned in the "y" direction of the base link's reference frame. However, for whatever reasons, the first revolute joint was at an offset angle of 10° about the z-axis of the base link, hence an offset angle of $\theta_{z,offset}$ = 10° was entered, while all other offset angle inputs were zero for each module. Then, the next link ($\mathbf{b}_2$) was entered in the "-y" direction of the previous joint module's coordinates. Afterwards, the next revolute joint ($\mathbf{b}_3$) direction was entered as "-y" with reference to the previous link's coordinate frame. This pattern was continued until the complete MRR was assembled and there were no more joint or link modules available for assembly. The type of configuration setup shown in Fig. 3-4 can be classified as "y, -y, -y, y, -y" in order to easily determine what configuration each module is in using relative connectivity. Furthermore, the type of robot in Fig. 3-4 may be classified as "RLRLP", where "R" stands for revolute joint module, "L" is for link module and "P" stands for prismatic joint module.

If there exists n modules, excluding the base link, where i is the i^{th} module in the chain then each module's static body vector in its local coordinate system is given by:

$$\mathbf{b}_{s,i}' = \begin{bmatrix} L_i & 0 & 0 \end{bmatrix}^{\mathrm{T}} \tag{3.12}$$

where L_i is the length of each module along its local x-axis. For the static position (at the initial configuration setup) a recursive method can be used [42]:

$$\begin{aligned} \mathbf{P}_1 &= \mathbf{b}_0 \\ \mathbf{P}_2 &= \mathbf{P}_1 + \mathbf{R}_{01}\mathbf{b}_1' \\ \mathbf{P}_3 &= \mathbf{P}_2 + \mathbf{R}_{01}\mathbf{R}_{12}\mathbf{b}_2' \\ &\vdots \\ \mathbf{P}_i &= \mathbf{P}_{i-1} + \mathbf{R}_{0i-1}\mathbf{b}_i' \end{aligned} \tag{3.13}$$

Also, the recursive orientation of each module is given by:

$$\mathbf{R}_{0i} = \mathbf{R}_{01}\mathbf{R}_{12}\ldots\mathbf{R}_{i-1i} = \mathbf{R}_{0i-1}\mathbf{R}_{i-1i} \tag{3.14}$$

Equations (3.13) and (3.14) hold true for the general motion of multiple bodies. Now the next step is to include the offset angles and configuration matrix using equations (3.10) and (3.11). Note that the pose calculation involves a static part and a motion part. For the static part (when the robot does not move at the initial configuration setup) the static rotation matrix for body i is defined as:

$$\mathbf{R}_{s,i} = \mathbf{R}_{config,i}\mathbf{R}_{ICSU,i} \tag{3.15}$$

The static recursive rotation matrix can then be defined as:

$$\mathbf{R}_{0i,s} = \prod_{i=1}^{n}\mathbf{R}_{s,i} \tag{3.16}$$

For the base link, the above equation is the unity matrix since the base vector is defined in global coordinates. The above equation can determine the static orientation of the tip and the rotation angles can be determined using equation (3.9). The tip position can be determined using a recursive method as:

$$\mathbf{P}_{s,i+1} = \sum_{i=0}^{n}\mathbf{R}_{0i,s}\mathbf{b}_{s,i}' \tag{3.17}$$

where $\mathbf{P}_{s,i+1}$ is the position vector for tip of the base link in global coordinates. The next step is to determine the tip position of the MRR after it moves due to joint motion and the only modules that produce active motion are the revolute and prismatic joints. The revolute joints provide rotation with a joint input angle, and the prismatic joints provide linear motion with a joint input displacement. A Boolean operator, η, is defined to describe the type of motion each module undergoes. If the module is a prismatic joint, η = 1, otherwise η = 0. For a prismatic joint, the motion body vector can be defined as:

$$\mathbf{b}_{m,i}' = \begin{bmatrix}\eta_i\theta_{input,i} & 0 & 0\end{bmatrix}^{\mathrm{T}} \tag{3.18}$$

where $\theta_{input,i}$ is the local prismatic joint input for motion. Here, the prismatic joint is only allowed to move along its local x-axis (in the direction of its length). Also, if the module

is a link or a revolute joint, equation (3.18) becomes a zero-vector. The motion rotation matrix for a revolute joint module with a known joint input angle is given by:

$$\mathbf{R}_{m,i} = \mathbf{R}\left((1-\eta_i)\theta_{input,i},\quad 0,\quad 0\right) \tag{3.19}$$

where $\theta_{input,i}$ is the local revolute joint input for motion and $\mathbf{R}$ is obtained from equation (3.6) or (3.7) Here, the revolute joint module is only allowed to rotate about its local x-axis. For each module, the local body vector that includes both the static and motion parts is:

$$\mathbf{b}_i' = \mathbf{b}_{s,i}' + \mathbf{b}_{m,i}' \tag{3.20}$$

and the rotation matrix for each body which includes both static and motion components is defined by:

$$\mathbf{R}_i = \mathbf{R}_{s,i}\mathbf{R}_{m,i} \tag{3.21}$$

The recursive rotation matrix with motion included then becomes:

$$\mathbf{R}_{0i,m} = \sum_{i=1}^{n} \mathbf{R}_i \tag{3.22}$$

The above equation will give the orientation of the MRR after joint motion occurs. The tip position after motion can then be determined using:

$$\mathbf{P}_{m,i+1} = \sum_{i=0}^{n} \mathbf{R}_{0i,m}\mathbf{b}_i' \tag{3.23}$$

Equations (3.16), (3.17) and (3.21) to (3.23) are only valid if the relative connectivity method is used. This method allows for directional configurability and the use of angular offsets. Also, if the ground is uneven, then the offset angles only need to be accounted for in the first module, while the offset angles are all zero for the successive modules. On the other hand, if there is no way of visualizing the relative coordinates of each body as shown in Fig 3-4, the connectivity of each module can become increasingly difficult with an increasing number of modules, and can be prone to errors. Therefore, for a large number of modules, it is easier to use a global reference frame for connectivity.

3.4 Position and Orientation using ZRP Connectivity

The zero reference plane (ZRP) method for module connectivity is used to configure each module with respect to a single global coordinate system. The initial configuration setup for the system occurs at the zero reference plane and the joint motion inputs are then entered afterwards. This method was first developed by Xi and Fenton [34]; however, each module was initially configured by inputting the values for the position and orientation manually and lacked directional configurability, which is much easier to use. In this section, the ZRP equations were modified to include directional configurability, as well as angular offsets. Therefore, by using the directional configurability, each module is configured using a common Cartesian reference coordinate plane and its six available directions with offsets occurring about the axes of the ZRP. Fig. 3-5 shows the same MRR model example shown in Fig. 3-4 connected using the modified ZRP method.

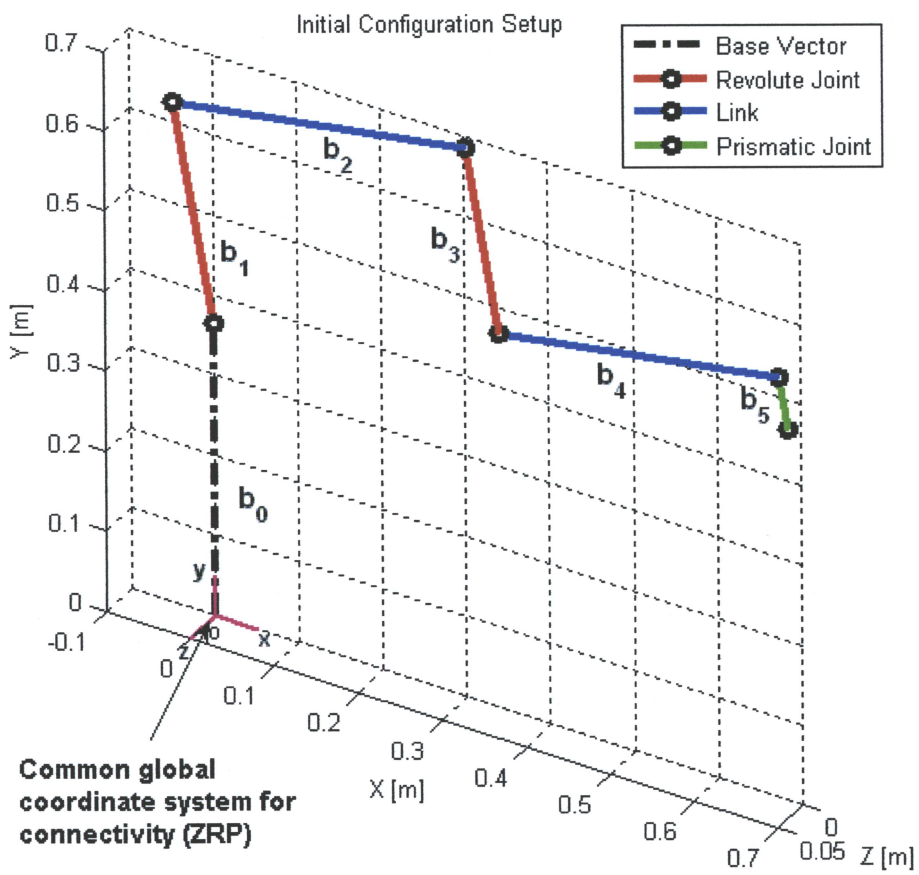


Figure 3-5: Schematic representation of initial configuration setup using ZRP connectivity

In Fig. 3-5, each module is configured and connected with respect to a common set of global coordinates (ZRP), located at the bottom end of the base link. The user first inputs the base link vector, and then enters the configuration direction for directional module connectivity along with offset angles for each corresponding module. Using the same example shown in Fig. 3-4, after the base link vector was entered, the next module ($\mathbf{b}_1$) was a revolute joint in the "y" direction, with an offset angle of 10° about the z-axis of the ZRP coordinate system ($\theta_{z,offset}$ = 10°). Afterwards, a link module ($\mathbf{b}_2$) was configured in the "x" direction. However, unlike the relative connectivity method in the previous section, an offset angle ($\theta_{z,offset}$ = 10°) was also required for this link module to produce the same configuration shown in Fig. 3-4 since the connectivity reference frame was fixed at the global coordinates. Otherwise, the link module would not be perpendicular to the previous joint module. The next module entered ($\mathbf{b}_3$) was a revolute joint in the "-y" direction, also with an offset angle of 10° about the global z-axis. Unlike the relative connectivity method, the user must input offset angles for corresponding modules once any preceding module is at an angular offset since a common global reference frame is used to configure each module. The connectivity pattern continued with the same offset angle applied to all modules until there were no link or joint modules left. Using the ZRP method, the type of configuration setup shown in Fig. 3-5 can be termed as "y, x, -y, x, -y", and the robot can once again be classified as "RLRLP" since the module types were not changed from the example shown in Fig. 3-4.

For the ZRP method, the static body vector for each module can be defined using equation (3.12), and the static rotation matrix for body *i* can be determined using equation (3.15) which includes the configuration and angular offset (ICSU) matrices. For the static part, there is no need for a recursive rotation matrix since all the modules can be defined using the same global coordinates. Thus, the static body vector in global coordinates for each module is given by:

$$\mathbf{b}_{s,i} = \mathbf{R}_{s,i}\mathbf{b}_{s,i}' \tag{3.24}$$

where $\mathbf{b}_{s,i}$ is the global static body vector of a single body with respect to the ZRP (not to be confused with $\mathbf{b}_{s,i}'$, which is in local coordinates). There is no need for a recursive

static rotation matrix since the rotations here are with respect to the global reference coordinates. The static tip position vector with respect to the origin then becomes:

$$\mathbf{P}_{s,i+1} = \sum_{i=0}^{n} \mathbf{b}_{s,i} \tag{3.25}$$

To determine the pose after motion occurs, the motion body vector for prismatic joint modules can be determined using equation (3.18) and the motion rotation matrix for revolute joint modules is given by equation (3.19), where the required joint input is about the local x-axis in both cases. However, using equation (3.19) for the ZRP method, the revolute joint in motion is constrained to rotate about the global x-axis only, thus the recursive rotation matrix given in equation (3.22) is not valid for the ZRP method. From equation (3.21), the rotation matrix with static and motion components for the first module is:

$$\mathbf{R}_1 = \mathbf{R}_{s,1}\mathbf{R}_{m,1} \tag{3.26}$$

For the second module, equation (3.26) can be written as:

$$\mathbf{R}_2 = \mathbf{R}_{s,1}\mathbf{R}_{m,1}\mathbf{R}_{s,12}\mathbf{R}_{m,2} \tag{3.27}$$

where $\mathbf{R}_{s,12}$ is the static rotation matrix from module 2 to module 1. At the initial configuration setup with no motion parts, the equation for static rotation of module 2 becomes:

$$\mathbf{R}_{s,2} = \mathbf{R}_{s,1}\mathbf{R}_{s,12} \tag{3.28}$$

Solving equation (3.28) for $\mathbf{R}_{s,12}$ and then substituting into equation (3.27), along with $\mathbf{R}_1$ from equation (3.26), gives the following recursive rotation matrix with motion parts included:

$$\mathbf{R}_2 = \mathbf{R}_1\mathbf{R}_{s,1}^{\mathrm{T}}\mathbf{R}_{s,2}\mathbf{R}_{m,2} \tag{3.29}$$

For n number of modules, equation (3.29) the recursive rotation matrix after joint motion can be written for module i as:

$$\mathbf{R}_{0i,m} = \mathbf{R}_{0(i-1),m}\mathbf{R}_{s,i-1}^{\mathrm{T}}\mathbf{R}_{s,i}\mathbf{R}_{m,i} \tag{3.30}$$

In equation (3.30), $\mathbf{R}_{0(i-1),m}$ is the recursive rotation matrix including joint motion for the previous module. Also, the static parts $\mathbf{R}^{T}_{s,i-1}$ and $\mathbf{R}_{s,i}$ are rotations with respect to the ZRP, and the motion parts $\mathbf{R}_{0i,m}$, $\mathbf{R}_{0(i-1),m}$ and $\mathbf{R}_{m,i}$ are due to local joint motion. Equation (3.30) can be used to determine the tip orientation, or the orientation at module *i*. Afterwards, the tip position with motion included can be determined using equation (3.23), with the local body vector (motion included) obtained from equation (3.20), with $\mathbf{b}_{s,i}$ obtained from equation (3.24).

The advantage of the ZRP method is that each module can be configured and connected with respect to a common reference plane and the MRR does not need any visualization of the coordinate systems of each module. However, the disadvantage is that once there is one offset angle in the open chain, this offset angle needs to be included for each successive module if the modules are to remain orthogonal to each other. Also, if there are more than one modules with offset angles (for instance, the ground is uneven and another module has a connector plate that is at an angle) then it becomes difficult to keep the other modules orthogonal to the offset modules because all offset angles must be accounted for. In other words, each module after the offset modules must include the addition of any previous offset angles in the initial configuration setup.

3.5 Position and Orientation using Quasi-Global Connectivity

The quasi-global connectivity method assumes that the reference plane of global coordinates shifts at each module according to each angular offset that is defined. Thus, if an angular offset is defined for one module, the reference plane coordinate system automatically shifts according to that angular offset for each consecutive module. Also, when using directional configurability, each module is configured using the global reference coordinate system until an angular offset occurs at a module. Once an angular offset occurs in a chain, each successive module will automatically take that offset into account and no more additional offset angles need to be defined for those modules. Fig. 3-6 shows the same MRR assembly shown in Fig. 3-4 and 3-5, but with quasi-global connectivity used. Here, the reference coordinate system shifts at the first revolute joint module since there is an angular offset, and every other module's reference coordinate

system coincides with the quasi-global coordinate system for the first revolute joint module.

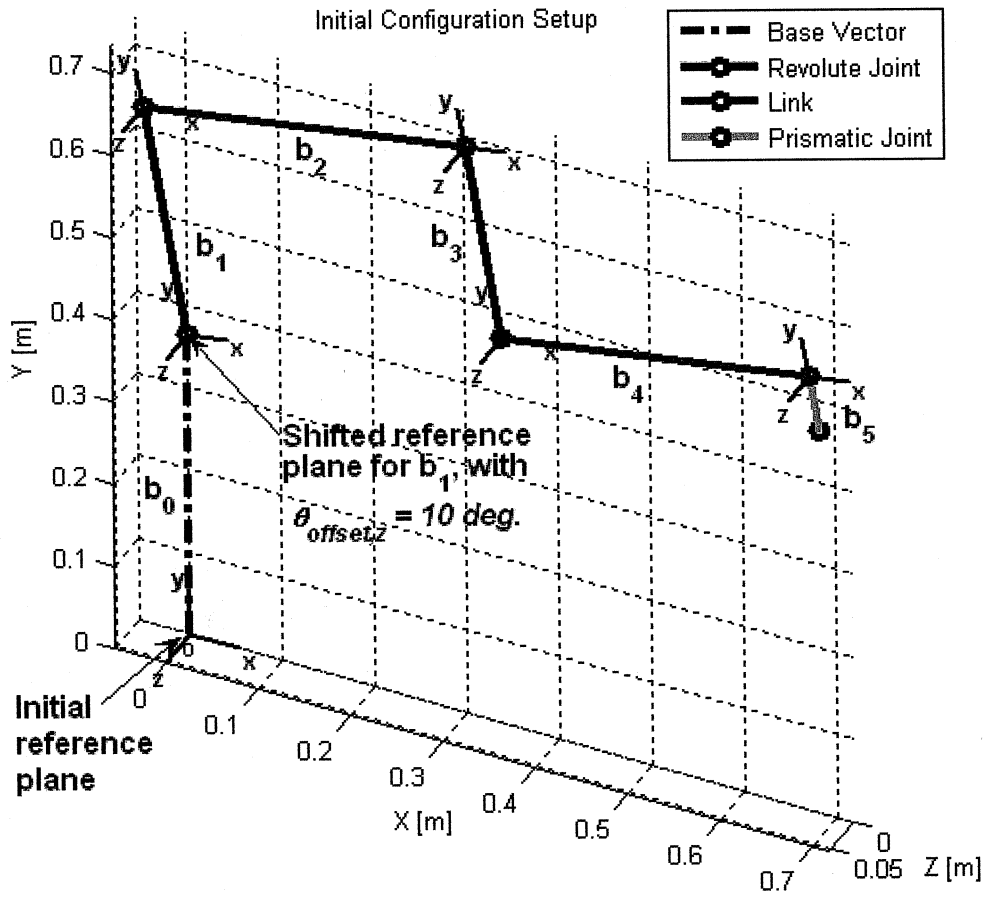


Figure 3-6: Schematic representation of initial configuration setup using quasi-global connectivity

In Fig. 3-6, the first revolute joint module ($\mathbf{b}_1$) was setup with respect to the reference global coordinate system at the bottom end of the base vector in the "y" direction of configurability, with an offset angle about the z-axis of 10°. Here, this module has a shifted global coordinate system due to its offset angle. The next link module ($\mathbf{b}_2$) was then setup in the "x" direction, with no need to input any offset angles due to the previous module's angular offset. Then, the next revolute joint module ($\mathbf{b}_3$) was configured in the "-y" direction, again without the need to input any offset angles due to the previous shift in the reference global coordinate system. The same connectivity pattern continued for the next link ($\mathbf{b}_4$) and prismatic joint modules ($\mathbf{b}_5$) without the need

to further define the offset due to the angular offset found for the first revolute joint module. Hence, all modules were orthogonal to each other when necessary. The robot configuration in Fig. 3-6 can then be classified as "y, x, -y, x, -y", and the type of robot is "RLRLP".

In the quasi-global connectivity method, each module's static body vector is first rotated with the configuration matrix from equation (3.10) and can be written as:

$$\mathbf{b}_{s,i}' = \mathbf{R}_{config,i}\begin{bmatrix} L_i & 0 & 0 \end{bmatrix}^{\mathrm{T}} \tag{3.31}$$

Therefore, the static rotation matrix for each body is only due to the offset angles, given by:

$$\mathbf{R}_{s,i_{body}} = \mathbf{R}_{ICSU,i} \tag{3.32}$$

In the above equation, the subscript "*body*" means that this rotation matrix is used to rotate the body vector given in equation (3.31) and does not define the actual static orientation. Thus, the static recursive rotation matrix for each body can then be defined as the following:

$$\mathbf{R}_{0i,s_{body}} = \prod_{i=1}^{n} \mathbf{R}_{ICSU,i} \tag{3.33}$$

and the static tip position can be determined using a recursive method as:

$$\mathbf{P}_{s,i+1} = \sum_{i=0}^{n} \mathbf{R}_{0i,s_{body}} \mathbf{b}_{s,i}' \tag{3.34}$$

Equation (3.34) gives the same result as the relative and ZRP connectivity methods for the static tip position. However, the actual orientation of each module is quite different from equation (3.33) when compared to the relative and ZRP connectivity methods because this method first rotates the static body vector with the configuration matrix included, as shown in equation (3.31). That is the reason why there is a "*body*" subscript in equation (3.33). Thus, if there are no offset angles for the modules, the orientation that can be calculated from equation (3.33) will simply be a zero vector since each module was set up using body vectors first. In order to obtain the true static

orientation of each module with respect to the global coordinates the following equation holds:

$$\mathbf{R}_{0i,s_{actual}} = \prod_{i=1}^{n} \mathbf{R}_{config,i} \mathbf{R}_{ICSU,i} \tag{3.35}$$

For the motion part, the prismatic joint module extends along its local length; hence the configuration matrix was used to define the direction of extension for the motion body vector as follows:

$$\mathbf{b}_{m,i}{}' = \mathbf{R}_{config,i} \begin{bmatrix} \eta_i \theta_{input,i} & 0 & 0 \end{bmatrix}^{\mathrm{T}} \tag{3.36}$$

where η_i is 1 for prismatic joints, 0 for link and revolute joint modules, and $\theta_{input,i}$ is the local prismatic joint input for motion. The direction of the rotation angle for revolute joint modules must also be defined. First, the revolute joint angle of motion needs to be defined about an axis that is parallel to its length and in the same direction. This was done by defining a vector of possible revolute joint input angles to transform the single joint input to one of three axes (positive and negative directions included) as follows:

$$\mathbf{A}_{input,i} = \mathbf{R}_{config,i} \begin{bmatrix} (1-\eta_i)\theta_{input,i} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \zeta_{x,i} \\ \zeta_{y,i} \\ \zeta_{z,i} \end{bmatrix} \tag{3.37}$$

where $\mathbf{A}_{input,i}$ is defined as the vector of joint motion rotations, and $\theta_{input,i}$ is the local revolute joint input for motion. Also, $\zeta_{x,i}$, $\zeta_{y,i}$, and $\zeta_{z,i}$ are the transformed joint motion angles to allow the revolute joint shaft to rotate at an axis parallel to its length in the same direction. Then the motion rotation matrix for the revolute joint becomes:

$$\mathbf{R}_{m,i_{body}} = \mathbf{R}\left(\zeta_{x,i},\ \zeta_{y,i},\ \zeta_{z,i}\right) \tag{3.38}$$

Afterwards, the local body vector that includes both static and motion parts can be calculated using equation (3.20) (with $\mathbf{b}_{s,i}$ obtained from equation (3.31) and $\mathbf{b}_{m,i}$ obtained from equation (3.36)) and the rotation matrix with motion parts included for each body can be defined as:

$$\mathbf{R}_i = \mathbf{R}_{ICSU,i} \mathbf{R}_{m,i_{body}} \tag{3.39}$$

The recursive rotation matrix with motion parts included then becomes.

$$\mathbf{R}_{0i,m_{body}} = \sum_{i=1}^{n} \mathbf{R}_i \tag{3.40}$$

Equation (3.40) does not give the actual orientation of each body, since each module's body vector was rotated at the initial configuration using the configuration matrix in equation (3.31). However, in the quasi-global method, the tip position vector calculation is based on body vectors and since each body vector was initially rotated, the tip position calculation (with motion) can use the above equation as:

$$\mathbf{P}_{m,i+1} = \sum_{i=0}^{n} \mathbf{R}_{0i,m_{body}} \mathbf{b}_i' \tag{3.41}$$

To correct the orientation of each module after motion and include the configuration matrix, a static rotation matrix correction can be defined as:

$$\mathbf{R}_{s,i_{actual}} = \mathbf{R}_{config,i} \mathbf{R}_{s,i_{body}} \tag{3.42}$$

Equation (3.42) gives the quasi-global method the same results as equation (3.21) for the relative connectivity method and the ZRP method. Since the motion rotation matrix in equation (3.39) involved initially transforming the revolute joint inputs by multiplying by the configuration matrix, the configuration matrix needs to be removed. By doing so, the same motion rotation matrix for each body in equation (3.19) can be obtained.

$$\mathbf{R}_{m,i_{actual}} = \mathbf{R}_{config,i}^{\mathrm{T}} \mathbf{R}_{m,i_{body}} \mathbf{R}_{config,i} \tag{3.43}$$

Using equations (3.42) and (3.43), the rotation matrix for the first module with static and motion parts that gives the actual orientation is given by equation (3.26) and can be written as:

$$\mathbf{R}_1 = \mathbf{R}_{s,1_{actual}} \mathbf{R}_{m,1_{actual}} = \left(\mathbf{R}_{config,1} \mathbf{R}_{s,1_{body}}\right)\left(\mathbf{R}_{config,1}^{\mathrm{T}} \mathbf{R}_{m,1_{body}} \mathbf{R}_{config,1}\right) \tag{3.44}$$

For the second module, equation (3.29) can be written as:

$$\mathbf{R}_2 = \mathbf{R}_1 \mathbf{R}_{s,1_{actual}}^{\mathrm{T}} \mathbf{R}_{s,2_{actual}} \mathbf{R}_{m,2_{actual}} \tag{3.45}$$

Substituting equations (3.42) and (3.43) into equation (3.45) gives:

$$\mathbf{R}_2 = \mathbf{R}_1 \left(\mathbf{R}_{config,1} \mathbf{R}_{s,1_{body}} \right)^{\mathrm{T}} \left(\mathbf{R}_{config,2} \mathbf{R}_{s,2_{body}} \right) \left(\mathbf{R}^{\mathrm{T}}_{config,2} \mathbf{R}_{m,2_{body}} \mathbf{R}_{config,2} \right) \qquad (3.46)$$

Thus, the recursive rotation matrix after joint motion that gives the actual orientation can be written for module *i* as:

$$\mathbf{R}_{0i,m_{actual}} = \mathbf{R}_{0(i-1),m_{actual}} \left(\mathbf{R}_{config,i-1} \mathbf{R}_{s,i-1_{body}} \right)^{\mathrm{T}} \left(\mathbf{R}_{config,i} \mathbf{R}_{s,i_{body}} \right) \left(\mathbf{R}^{\mathrm{T}}_{config,i} \mathbf{R}_{m,i_{body}} \mathbf{R}_{config,i} \right)$$

.....(3.47)

Equation (3.47) gives the same results as equation (3.22) when using the relative connectivity method, and equation (3.30) when using the ZRP method.

When there are no module angular offsets, the quasi-global method of connectivity has all of the same connectivity inputs as the ZRP method. However, when there are multiple offsets, the ZRP method becomes difficult since each offset has to be accounted for in each of the modules, whereas the quasi-global method allows for multiple offsets and there is no need to include any previous module offsets during the initial configuration setup. In other words, once offsets are entered for a module, the global coordinate system automatically shifts according to all of the previous angular offsets, so the next model can be configured using this shifted coordinate system. The position and orientation of the robot can then be determined for the original global coordinate reference plane.

4. Flexibility Analysis of MRR Joint-link Modules

In this chapter, an effective method of determining the tip deformation and vibration properties of a joint-link module for MRRs is presented based on flexibility analysis. A typical MRR module consists of a motor, joint and link, and a FEM-based joint-link model is applied to consider the joint and link flexibility. Since the joint modules found in most MRRs are large, instead of modeling the joint as a torsional spring, the joint is modeled as a finite element beam where the length of the joint shaft is taken into account to justify possible joint bending and axial deformation. Furthermore, the flexibility of the finite element beam depends on the beam length and cross section properties; hence two ratios are defined. The first ratio is the length of the link to the length of the joint shaft, and the second ratio is area moment of inertia of the link to that of the joint. By utilizing these two ratios, each joint-link module can be categorized into three cases: *a rigid joint-flexible link model*, *a rigid link-flexible joint model*, or *a hybrid model of flexible joint-link.* A static analysis was conducted using each of the three models by varying joint-to-link length and inertia ratios to determine the tip deflection and then identify exactly which model is appropriate to use for a given size and configuration. Afterwards, a free vibration analysis was performed to assess the flexibility of the joint-link module under dynamic conditions for the same purpose.

4.1 Static Flexibility Analysis

To determine the flexibility of a single joint-link module under an applied static load, three models were used to determine the tip deflection for each possible configuration, and then compared. The hybrid FEM model consists of two 3D finite element beams connected to each other where one beam represents the joint shaft (length of exposed motor shaft after the bearing) and the other beam represents the link. This model assumes complete flexibility of both the joint and link components, namely, with bending in two planes as well as axial and torsional deformation along the length of the beam element. Therefore, the hybrid FEM model takes into account all possible modes of joint deformation, including bending which is found in MRRs with longer joint shafts. The flexible link model is a simple cantilever beam and uses the same stiffness found in

the hybrid FEM model with boundary conditions applied to make the joint totally rigid and allow for flexibility in the link alone. Conversely, the flexible joint model assumes the link is a straight rigid beam and allows only joint flexibility.

4.1.1 Static Hybrid FEM Model

The method used in this model is the displacement-based finite element method proposed by Bathe [43]. A single MRR joint-link module is idealized as two interconnected 3D finite element beams, and the system is solved by formulating the force balance equations corresponding to the unknown displacements at the ends of each element. As shown in Fig. 4-1, it can be proven that the use of a single finite element over the beam length leads to the same tip deformation using beam theory. Hence, a single finite element beam is used in this thesis to represent a single module.

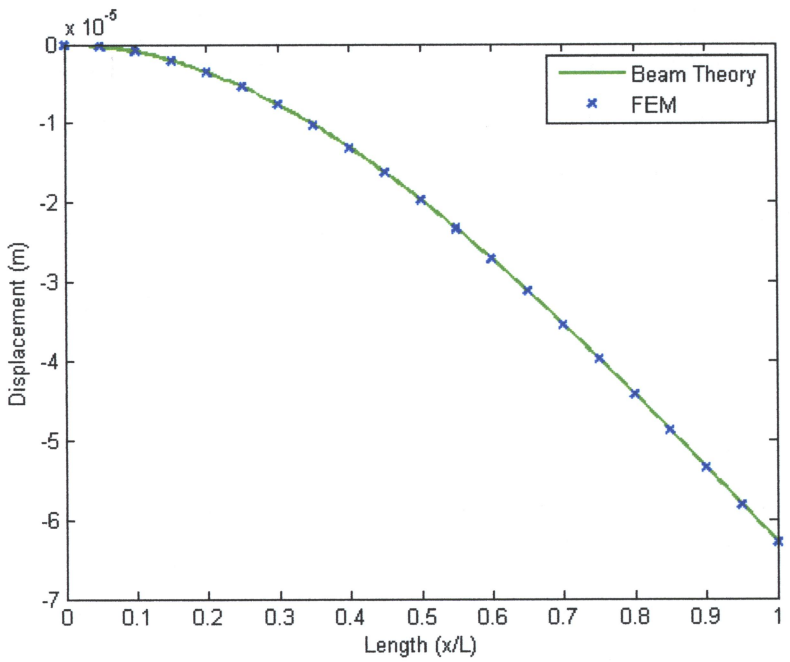


Figure 4-1: Comparison of Beam Theory with single FEM beam

The potential energy due to strain along a single beam element is given by:

$$PE = \frac{1}{2}\int_0^L \left(EA\left(\frac{\partial u}{\partial x}\right)^2 + EI_z\left(\frac{\partial^2 v}{\partial x^2}\right)^2 + EI_y\left(\frac{\partial^2 w}{\partial x^2}\right)^2 + GJ\left(\frac{\partial \theta}{\partial x}\right)^2 \right) dx \qquad (4.1)$$

Where L, E, A, I_z, I_y, G and J are the length, modulus of elasticity, cross section area, area moment of inertia about the z-, and y-axis, shear modulus and torsion constant of the beam element, respectively. The terms u, v, w and θ in equation (4.1) refer to the axial displacement, bending displacements about the y- and z-axis, and torsional deformation about the x-axis, respectively. Equation (4.1) assumes that the beam element has a constant uniform cross section and the material has linear-elastic properties. Discretizing equation (4.1) using shape functions with respect to the elastic displacements will lead to:

$$PE = \frac{1}{2}\int_0^L \{\mathbf{U}_e\}^{\mathrm{T}} [\mathbf{B}]^{\mathrm{T}} [\mathbf{D}][\mathbf{B}]\{\mathbf{U}_e\}\, dx \tag{4.2}$$

where $\{\mathbf{U}_e\}$ is the local vector of displacements for a beam element with two nodes, i and j, as shown in Fig. 4-2 and is written as:

$$\{\mathbf{U}_e\} = \{u_i' \quad v_i' \quad w_i' \quad \theta_i' \quad \phi_i' \quad \psi_i' \quad u_j' \quad v_j' \quad w_j' \quad \theta_j' \quad \phi_j' \quad \psi_j'\}^{\mathrm{T}} \tag{4.3}$$

where θ' is the local torsional displacement angle about the x-axis, φ' and ψ' are the bending angles about the y-axis and z-axis, respectively.

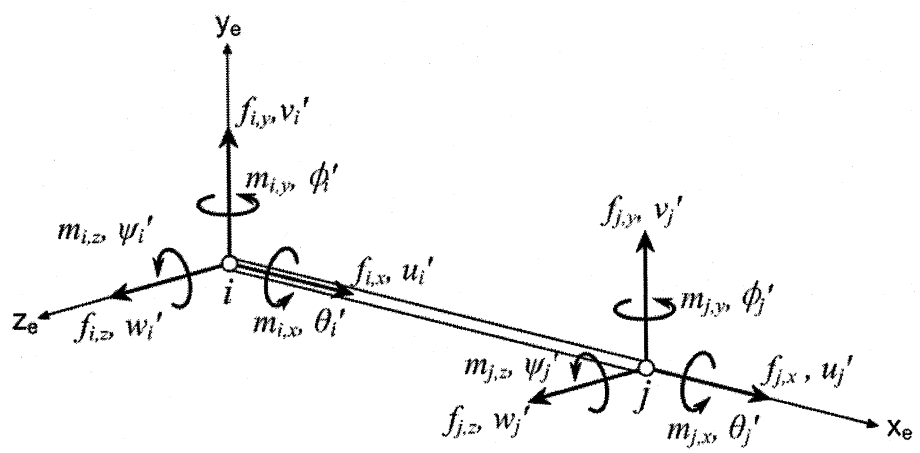


Figure 4-2: Local displacements and forces at each node in a finite element beam

From equation (4.2), the local stiffness matrix for a single element can be expressed as:

$$[\mathbf{k}_e] = \int_0^L [\mathbf{B}]^T [\mathbf{D}][\mathbf{B}]\, dx \tag{4.4}$$

The matrix [**D**] in equation (4.4) relates the cross section and material properties pertaining to the type of deformation and is given by:

$$[\mathbf{D}] = \begin{bmatrix} AE & 0 & 0 & 0 \\ 0 & EI_z & 0 & 0 \\ 0 & 0 & EI_y & 0 \\ 0 & 0 & 0 & GJ \end{bmatrix} \tag{4.5}$$

The matrix [**B**] in equation (4.4) contains the first derivatives of the shape functions with respect to x for the axial and torsional displacements, along with the second derivatives of the bending shape functions with respect to x and is given by:

$$[\mathbf{B}] = \begin{bmatrix} \frac{dN_1}{dx} & 0 & 0 & 0 & 0 & 0 & \frac{dN_2}{dx} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{d^2N_3}{dx^2} & 0 & 0 & 0 & \frac{d^2N_4}{dx^2} & 0 & \frac{d^2N_5}{dx^2} & 0 & 0 & 0 & \frac{d^2N_6}{dx^2} \\ 0 & 0 & \frac{d^2N_7}{dx^2} & 0 & \frac{d^2N_8}{dx^2} & 0 & 0 & 0 & \frac{d^2N_9}{dx^2} & 0 & \frac{d^2N_{10}}{dx^2} & 0 \\ 0 & 0 & 0 & \frac{dN_{11}}{dx} & 0 & 0 & 0 & 0 & 0 & \frac{dN_{12}}{dx} & 0 & 0 \end{bmatrix}$$

.....(4.6)

In equation (4.6), N_1 and N_2 are shape functions for the axial displacements. A linear displacement function is assumed since this type of deformation acts along a single axis with only two DOFs (one axial displacement at each of the two nodes). The axial displacement shape functions are given by:

$$[\mathbf{N}]_{axial} = [N_1, \quad N_2] = [1 - \tfrac{x}{L}, \quad \tfrac{x}{L}] \tag{4.7}$$

The shape functions due to bending in the x-y plane assume a cubic displacement function since there are four DOFs (one transverse and one rotational displacement at each node) associated with this type of bending. The shape functions for bending about the z-axis are obtained from a Hermitian cubic polynomial and are given by:

$$[\mathbf{N}]_{bend,z} = [N_3, \ N_4, \ N_5, \ N_6]$$
$$= \left[1-3\left(\tfrac{x}{L}\right)^2+2\left(\tfrac{x}{L}\right)^3, \ x-2\tfrac{x^2}{L}+\tfrac{x^3}{L^2}, \ 3\left(\tfrac{x}{L}\right)^2-2\left(\tfrac{x}{L}\right)^3, \ -\tfrac{x^2}{L}+\tfrac{x^3}{L^2}\right] \tag{4.8}$$

Similarly, the shape functions for bending in the x-z plane (about the y-axis) can be determined using a cubic displacement function and are given by:

$$[\mathbf{N}]_{bend,z} = [N_7, \ N_8, \ N_9, \ N_{10}]$$
$$= \left[1-3\left(\tfrac{x}{L}\right)^2+2\left(\tfrac{x}{L}\right)^3, \ -x+2\tfrac{x^2}{L}-\tfrac{x^3}{L^2}, \ 3\left(\tfrac{x}{L}\right)^2-2\left(\tfrac{x}{L}\right)^3, \ \tfrac{x^2}{L}-\tfrac{x^3}{L^2}\right] \tag{4.9}$$

For torsional deformation along the length of the beam element, a linear torsion angle variation is assumed, hence the shape functions can be determined as:

$$[\mathbf{N}]_{torsion} = [N_{11}, \ N_{12}] = \left[1-\tfrac{x}{L}, \ \tfrac{x}{L}\right] \tag{4.10}$$

Upon substitution of equations (4.7) to (4.10) into equation (4.6), then substituting equations (4.5) and (4.6) into equation (4.4), the stiffness matrix for a single 3D beam element is created with two nodes and six displacement DOFs at each node; therefore, equation (4.4) is a 12 × 12 matrix and is shown explicitly in Appendix A.

For a single MRR joint-link module with a flexible joint and flexible link, the joint and link components are represented by two interconnected finite element beams in the hybrid FEM model. Fig. 4-2 shows the hybrid FEM model with the local coordinate systems for the link and the joint for three different joint configurations. Assuming the joint has a uniform circular cross section with flat ends and the link has a uniform rectangular cross section, one flat end of the joint can be reconfigured about one of the five flat faces at the end of the link to provide the necessary shaft rotation about the x-, y-, or z-axis of the link's local coordinate origin in order to actuate the entire link, as shown in Fig. 4-2.

The reconfigurability of the joint relative to the link is accomplished using a transformation matrix to align the joint shaft rotation axis (which is the joint's local x-axis) with the local x-, y-, or z-axis of the link. So for each joint-link module, a body-fixed coordinate system is used with the frame of reference originating at the link's local

coordinate system origin. In this chapter, the rotation matrix used for body-fixed rotations is given by equation (3.8) from the previous chapter.

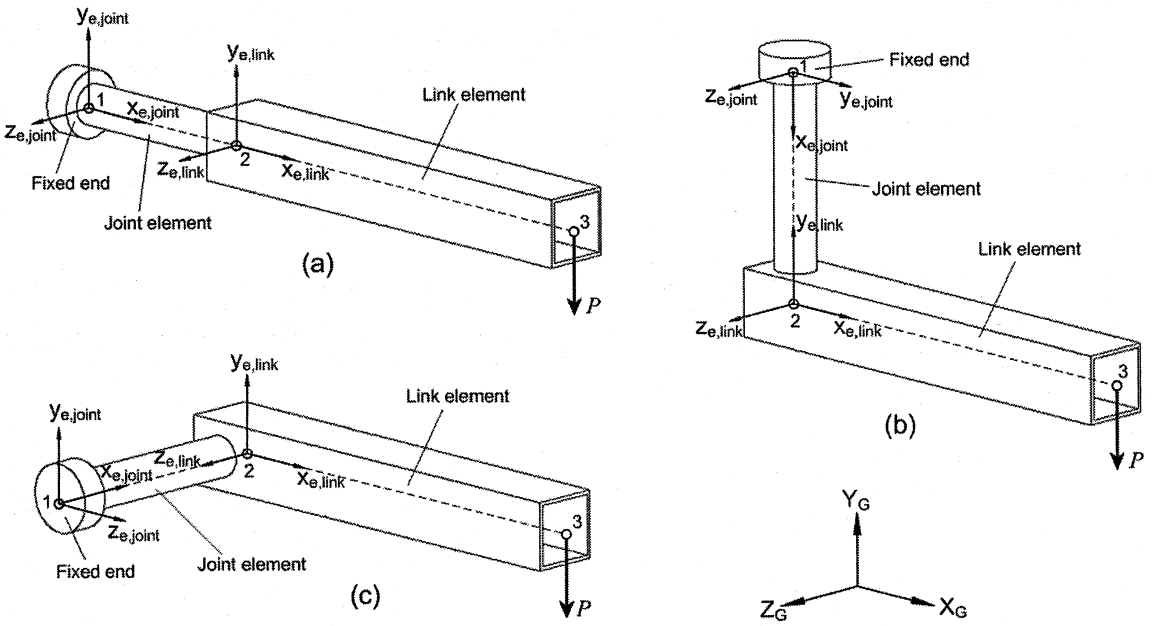


Figure 4-3: MRR Joint-link module in three different body-fixed joint configurations

Each joint configuration can be configured about the link using the directional configurability method presented in Chapter 3.2. Note that the local input torque for the joint element is positive counter clockwise about its local x-axis. For the configurations shown in Fig 4-3(b) and (c), the local positive input torque from the joint will become negative in the global coordinate system since the local x-axis of the joint is in the opposite direction of the link's positive y- and z-axes, respectively. The transformation matrix used to orient the displacement DOFs for the joint configuration relative to the link consists of four identical body-fixed rotation matrices from the configuration matrix (equation (3.10)) and is given by:

$$[\mathbf{T}]_{12\times12} = diag\left(\left[\mathbf{R}_{config}\right]_{3\times3}\right) \tag{4.11}$$

Assuming that the link's local coordinates coincide with the system's global coordinates, the joint stiffness can be transformed relative to the link stiffness using:

$$[\mathbf{k}]_j = [\mathbf{T}]^T [\mathbf{k}_e]_j [\mathbf{T}] \tag{4.12}$$

Where $[\mathbf{k}_e]_j$ is the joint local stiffness and can be calculated using equation (4.4). The link local stiffness is also calculated using equation (4.4) but no transformation to global coordinates is necessary.

After the joint is transformed to represent one of the five joint configurations relative to the link, the hybrid FEM joint-link element can be assembled using the direct stiffness method given by Logan [44]. Here, the system global stiffness, $[\mathbf{K}]_G$, is created by adding terms in the transformed joint and link stiffness matrices ($[\mathbf{k}]_j$ and $[\mathbf{k}]_l$) associated with the same nodal DOFs, and directly placing these added terms into the corresponding DOF locations in $[\mathbf{K}]_G$. By doing so, the assembled hybrid FEM joint-link stiffness is an 18 × 18 matrix with no boundary conditions applied.

For a single beam element with nodes i and j as shown in Fig. 4-2, the vector of local element forces with two nodes, i and j, is:

$$\{\mathbf{f}_e\} = \{f_{i,x} \quad f_{i,y} \quad f_{i,z} \quad m_{i,x} \quad m_{i,y} \quad m_{i,z} \quad f_{j,x} \quad f_{j,y} \quad f_{j,z} \quad m_{j,x} \quad m_{j,y} \quad m_{j,z}\}^{\mathrm{T}}$$

.....(4.13)

When the hybrid FEM model is assembled for a given joint configuration, a tip load, P, at the end of the link, i.e. node 3, is applied in the negative y-direction (in the direction of gravity) to represent the weight of an end-effector, as shown in Fig. 4-3. Also, for each configuration, the weights of the joint and link elements are lumped at the end nodes of each element and replaced by equivalent nodal forces and moments, as shown in Fig. 4-4 where the joint is configured to align with the -z axis.

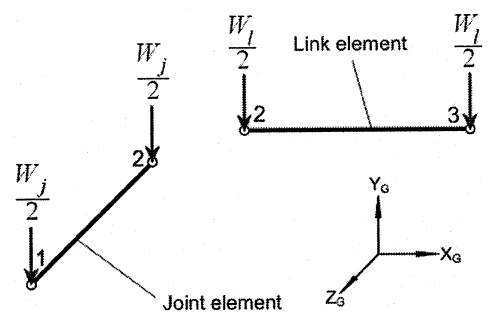


Figure 4-4: Lumped weights at nodes of joint and link for the configuration in Fig. 3(c)

Since the joint element configuration is oriented with respect to the link, the vector of local forces in equation (4.13) for the joint must be transformed to the link, keeping in mind that the forces due to the weight of the joint at the end nodes of the joint element must always act in the negative y-direction. The local joint input torques can be transformed to global coordinates and added to the forces due to the weight of the joint in the direction of gravity using:

$$\{\mathbf{f}\}_j = [\mathbf{T}]^T \{\mathbf{f}_{e,torque}\}_j + \{\mathbf{f}_{weight}\}_j \tag{4.14}$$

Here, the subscript j denotes joint. The local force vector for the link requires no transformation since its local coordinate system coincides with the global coordinates. Afterwards, the global force vector $\{\mathbf{F}\}_G$ is assembled by adding the forces and moments from each element with associated global DOFs at each node. When the global force vector is assembled, the boundary conditions can then be applied to the entire joint-link model. As shown in Fig. 4-3, one end of the joint is fixed at node 1. Therefore, in the global displacement vector, $\{\mathbf{U}\}_G$:

$$u_1 = v_1 = w_1 = \theta_1 = \phi_1 = \psi_1 = 0 \tag{4.15}$$

After the boundary conditions are applied, the system of equations can be solved for the unknown global displacements using:

$$\{\mathbf{U}\}_G = [\mathbf{K}]_G^{-1} \{\mathbf{F}\}_G \tag{4.16}$$

4.1.2 Static Flexible Link Model

The flexible link model assumes that the link is flexible and the joint is totally rigid. Thus, the same global system of equations found in equation (4.16) can be used with the exception that the boundary conditions are fixed at joint nodes 1 and 2, consequently immobilizing the joint entirely to create a simple cantilever beam model. Hence, the displacements associated with nodes 1 and 2 in the global displacement vector are set to zero.

$$u_1 = v_1 = w_1 = \theta_1 = \phi_1 = \psi_1 = u_2 = v_2 = w_2 = \theta_2 = \phi_2 = \psi_2 = 0 \tag{4.17}$$

With the boundary conditions applied to make the joint totally rigid, the unknown tip displacements at the end of the flexible link then can be solved using equation (4.16).

4.1.3 Static Flexible Joint Model

The static flexible joint model assumes that only the joint is flexible and the link is relatively rigid, so only the joint stiffness in equation (4.12) is considered. When the joint is configured about the ± x- or ± y-axis with the tip load applied in the negative y-direction, as shown in Fig 4-3(a) and (b), the joint experiences axial stretching and bending with minimal torsion; while the link remains rigid as shown in Fig. 4-5(a) and (b). In these two configurations, the applied tip load, P, and the weight of the link at its centroid are lumped to node 2 of the joint element as a single force and couple.

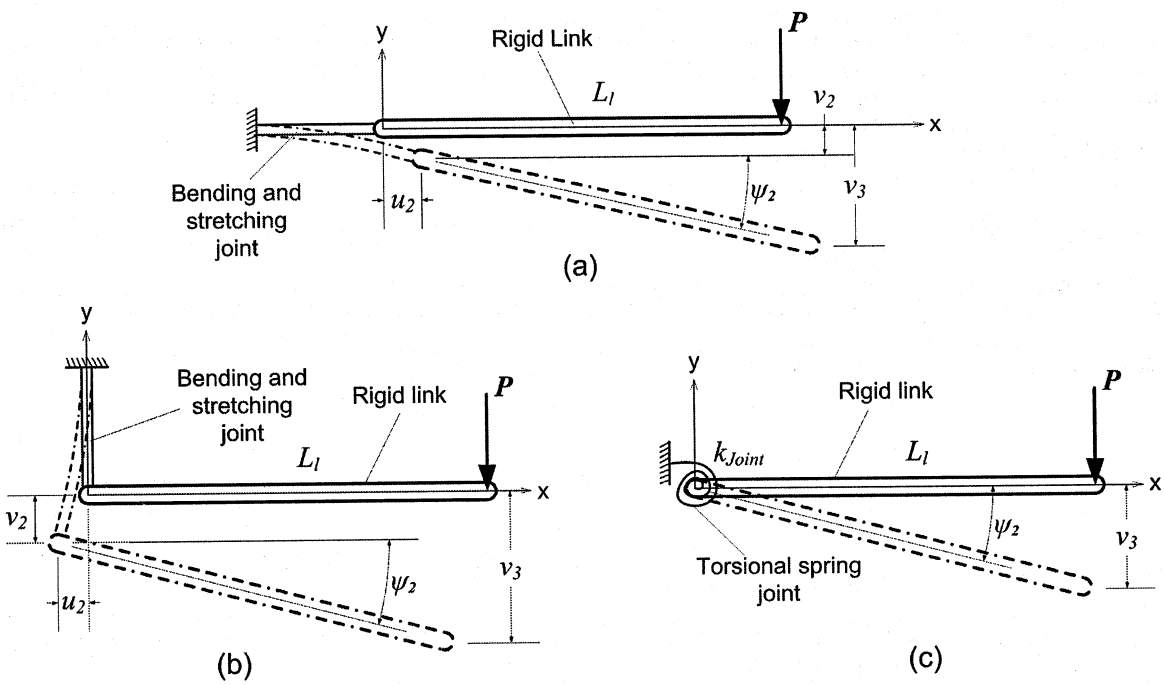


Figure 4-5: Flexible joint models in three different joint configurations

Therefore, when the local joint forces are transformed to global coordinates using equation (4.14), the force and couple replacement due to the loads acting on the link are added to the force in the y-direction and the moment about the z-axis at node 2 of the joint element to form the system global force matrix.

$$F_{2,y} = f_{2,y} + (P + W_l)$$
$$M_{2,z} = m_{2,z} + \left(PL_l + \frac{W_l L_l}{2} \right) \tag{4.18}$$

Where the subscript l denotes link and the terms in the brackets are the force and moment replacements due to the link loads.

If the link is in a position where the joint will experience no torsional deformation along its local x-axis, and if bending occurs only in the global x-y plane, the flexible joint model can be simplified to a 2D problem. Therefore, with the joint fixed at node 1, the boundary conditions for the joint stiffness in equation (4.12) become:

$$u_1 = v_1 = w_1 = \theta_1 = \phi_1 = \psi_1 = w_2 = \theta_2 = \phi_2 = 0 \tag{4.19}$$

The unknown displacements at the end of the joint at node 2 are then solved using equation (4.16). Afterwards, when the global vertical displacement and bending angle about the z-axis are determined for the joint, the vertical tip deflection of the link can be calculated using geometry for the configurations shown in Fig. 4-5(a) and (b):

$$v_3 = v_2 + L_l \sin \psi_2 \tag{4.20}$$

When the joint shaft is aligned about the ± z-axis of the link, as shown in Fig. 4-3(c), it undergoes high torsional deformation. In this configuration, the joint can be modeled as a torsional spring with negligible bending and axial deformation as shown in Fig. 4-5(c). Thus, all of the displacement DOFs become zero with the exception of ψ_2. Equation (4.16) then reduces to a single equation, which resembles a general torque-twist equation:

$$\psi_3 = \left(\frac{L_j}{G_j J_j} \right) T = \left(\frac{L_j}{G_j J_j} \right) \left(PL_l + \frac{W_l L_l}{2} \right) \tag{4.21}$$

where T is the external torque acting on the joint shaft. Assuming that the link is totally rigid, the tip deflection can once again be calculated from geometry:

$$v_3 = L_l \sin \psi_3 \tag{4.22}$$

4.1.4 Static Flexibility Simulation and Results

In a single MRR joint-link module, to determine if the joint or link is either flexible or rigid under static loading conditions, a flexibility analysis is performed by varying joint and link sizes for each configuration shown in Fig. 3. Yang and Sadler [28] noted that the ratio of joint to link stiffness determines the overall deflection of the system and that there exists ranges where either the joint torsion, link bending, or combined torsion and bending responses are dominant. Also, Xi and Fenton [29] showed that the link to joint stiffness and mass moment of inertia ratios play a role in determining the overall system flexibility, and there exists three regions of system *softening, stiffening,* and *mixed flexibilities*.

In the hybrid FEM model, the stiffness of the beam element described in equation (4.4) depends on the length (L), material properties (E and G) and cross section properties (A, I_y, I_z, and J) of the component. Rather than test the flexibility of each component by varying the material properties, or the stiffness ratio as a whole, the length and cross section properties were varied to show the effect of link and joint sizing on the flexibility of MRR joint-link modules. Thus, the length ratio (R_L) is defined as the length of the link to the length of the joint in a single joint-link module:

$$R_L = \frac{L_l}{L_j} \tag{4.23}$$

where L_l and L_j are the lengths of the link and joint shaft, respectively. The length of the revolute joint element can be interpreted as the joint casing since it carries the motor, gear drive, and bearing, or the length of the joint element can be taken as the length of the joint shaft that is exposed after the bearing if the bearing is assumed to take the majority of the load. In this thesis, the latter assumption is made. Define the inertia ratio (R_I) as the area moment of inertia of the link to that of the joint:

$$R_I = \frac{I_l}{I_j} \tag{4.24}$$

where I_l and I_j are the area moment of inertias of the link and joint, respectively. Note that if the link has a uniform square cross section, $I_{y,l} = I_{z,l} = I_l$, and if the joint has a solid uniform circular cross section, $I_{y,j} = I_{z,j} = J_j = I_j$.

For the static flexibility analysis, R_L and R_I were varied to represent joint and link sizes ranging from a smaller desktop-sized robot to a larger industrial-sized MRR. This was done in order to determine the flexibility of a single joint-link module and the tip deflection in the positive y-direction for different link and joint sizes for each of the configurations, as shown in Fig. 3. The link tested was made of aluminum and the joint shaft was alloy steel. The material properties of the link and joint components are shown in table 4-1 below.

Table 4-1: Material properties of link and joint shaft

Component	Material	E [GPa]	G [GPa]	ρ [kg/m^3]	σ_y [MPa]
Link	Al-6061	69	26	2700	55.1485
Joint	Alloy Steel	210	79	7700	620

The link was assumed to have a hollow square cross section and the joint shaft was assumed to have a solid circular cross section. A tip load at the end of the link (at node 3) of P = 100 N was applied in the negative y-direction to simulate the load of a heavy end-effector attached to the free end of the link. Furthermore, the length of the joint shaft was fixed to 0.02 m, and the area moment of inertia of the link was fixed to 7.6945×10^{-6} m^4. The length ratio, R_L, was varied from 1 to 100, thus varying the link length from 0.02 m to 2 m. Also, R_I was varied from 0.01 to 100, thus the joint area moment of inertia (which is equal to the torsion constant in this case) and cross section area also changed. With the sizes of the joint and link components changing, the weight of each component was also updated. The static tip deflection in the positive y-direction was then determined using the static hybrid FEM model, flexible link model and flexible joint model for different joint and link sizes that varied with the given ranges of R_L and R_I.

For all three configurations tested in Fig. 4-3, a large value of R_L indicates that the link is much longer than the joint. When R_L equals 1, the link is of the same length as the joint shaft. Also, when R_I is very small, the joint has a larger cross section area than the

link, and a larger value of R_I means that the link has a larger cross section area than the joint. It is interesting to note that when the joint is configured to provide shaft rotation about the link's negative y-axis (or z-axis), the results of the static deflection analysis are exactly the same when the joint is configured about the link's positive y-axis (or z-axis) due to symmetry.

Figs. 4-6 to 4-9 show the results of the tip displacement by varying R_L and R_I for each of the configurations tested. Fig. 4-6(a) shows the comparison between the flexible link and FEM hybrid model when the joint is configured for shaft rotation about the link's x-axis. When R_L increases, the link becomes much longer than the joint shaft, thus becoming more flexible in comparison to the joint. When R_I decreases, the joint cross section becomes much larger than the link, thus the joint becomes more rigid and experiences minimal static deflection. Also, when R_I increases, and R_L remains less than 10, the joint cross section becomes much smaller than that of the link and the joint is more flexible in terms of bending deformation. Fig 4-6(b) shows the comparison between the flexible joint and FEM hybrid model for the same configuration used in Fig. 4-6(a). Here, when R_L is less than 10 and R_I is less than 1, the link exhibits low flexibility and the joint is relatively rigid. However, when R_I increases to 100 and R_I is less than 1, the joint becomes more flexible in bending and the link becomes relatively rigid.

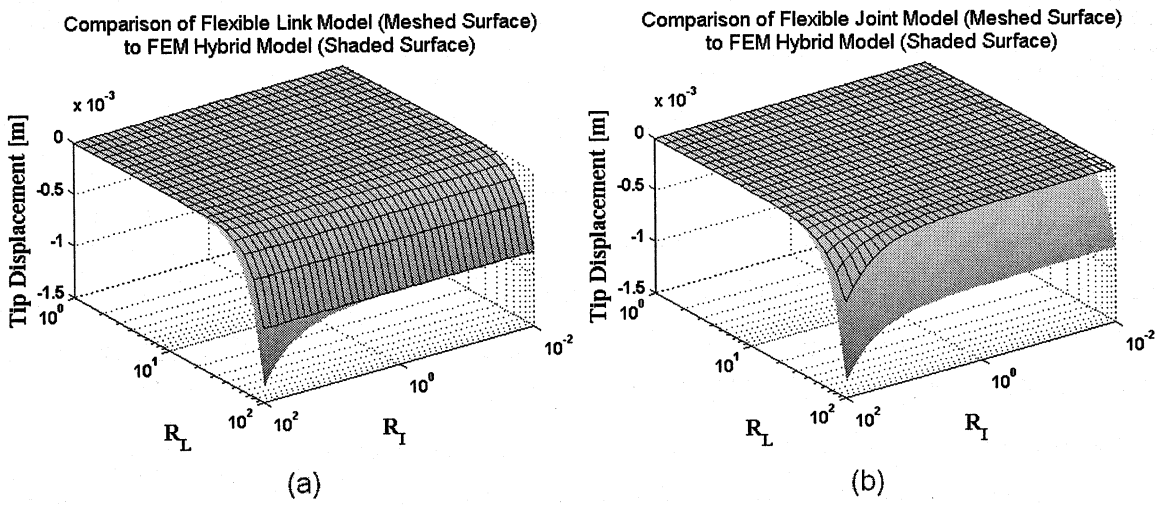


Figure 4-6: Static analysis results when the joint is configured about the link's x-axis

Similar results are shown in Fig. 4-7(a) and (b) when the joint is configured to provide shaft rotation about the link's y-axis, with the exception that the link is flexible and the joint is relatively rigid over a smaller range of R_L and R_I values because both the axial and bending displacements at the joint-link interface determined from the hybrid FEM model are rather large.

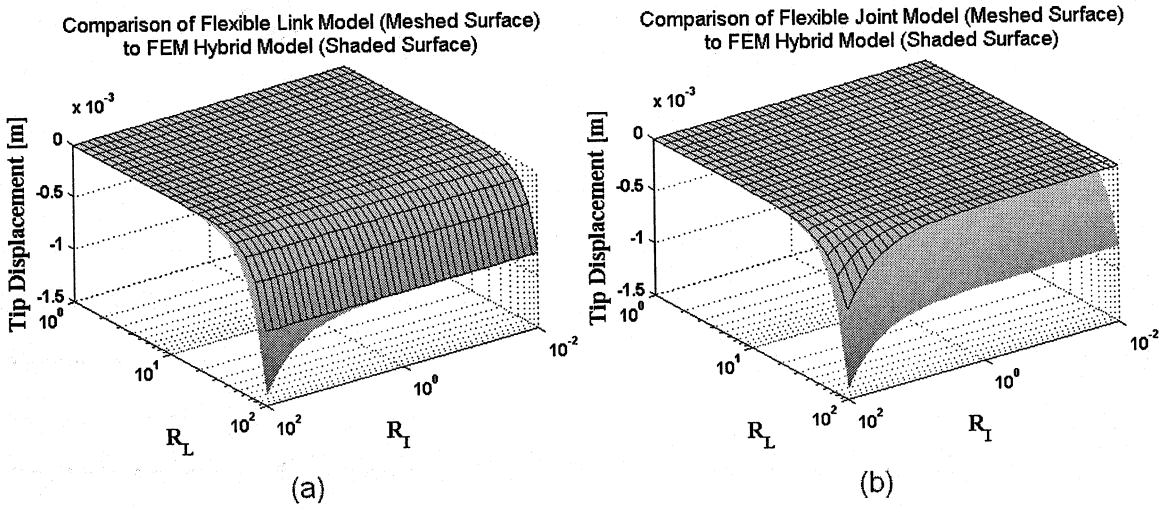


Figure 4-7: Static analysis results when the joint is configured about the link's y-axis

Fig. 4-8(a) shows the comparison between the hybrid and flexible link models when the joint is configured to provide shaft rotation about the link's z-axis. Here, when R_I is less than 0.03 and when R_L is between 1 and 100, the joint cross section is much larger than that of the link and the joint is torsionally rigid. In Fig. 4-8(b), when R_I is increased to 100 and when R_L is between 1 and 10, the joint cross section becomes much smaller than the link's and the joint becomes more flexible in torsion while the link is rigid in comparison. In all static cases shown, when both R_L and R_I increase to 100, both the joint and link become fully flexible, and the maximum tip deflection occurs in that region. The configuration that has the highest calculated tip deflection is when the joint shaft is positioned to rotate the link's z-axis. Here, the dominant mode of deformation for the joint is torsion, which is combined with the link's high bending flexibility when R_L and R_I are equal to 100.

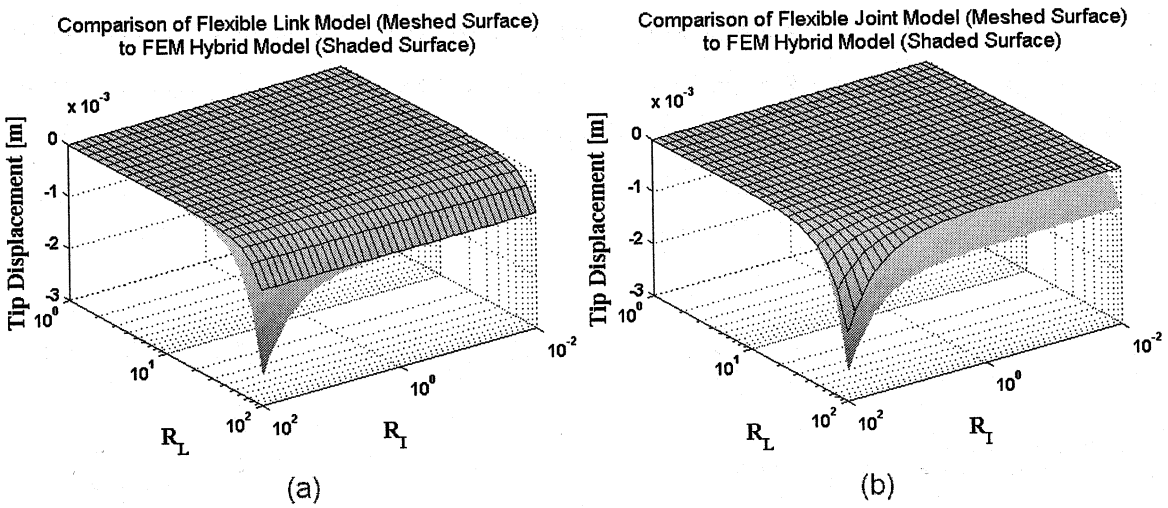


Figure 4-8: Static analysis results when the joint is configured about the link's z-axis

Afterwards, the regions of static flexibility were graphed for the given ranges of R_L and R_I to show the flexibility of the joint and link for different sizes. Each contour plot was created to show the differences between the flexible joint and flexible link models to the hybrid FEM model. For reconfigurable link-joint modules of different sizes, graphs like Figs. 4-9 to 4-11 can be plotted and the user can simply calculate ratios of R_L and R_I to determine the flexibility of modules with different link and joint sizes and determine exactly which model is appropriate to use depending on the link or joint static flexibility. Fig. 4-9 shows the regions of static flexibility when the joint is configured to rotate the about the link's x-axis. Here, the flexible link model can be used for a majority of R_L and R_I regions, specifically, when R_L increases from 1 to 100 and R_I increases from 0.04 to 100. The flexible joint model can be used when R_L is increased from 1 to 6 and R_I is increased from 4 to 100. In Fig. 4-10, the regions of static flexibility are shown when the joint is aligned to rotate the link's y-axis. When R_L is a bit larger than 1 and is increased to 100, and R_I is increased from 0.01 to 11, the flexible link model can be used. When R_L increases from 1 to 5, and R_I is increased from 4 to 100, the flexible joint model can be used.

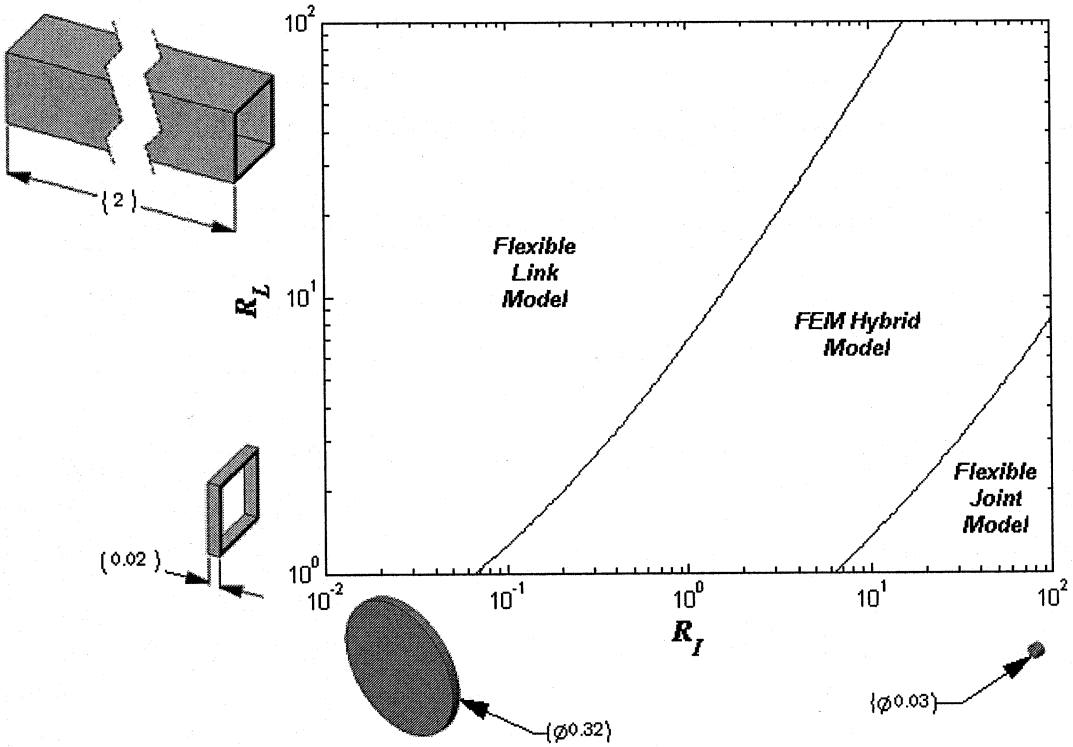


Figure 4-9: Regions of static flexibility when the joint is configured about the link's x-axis

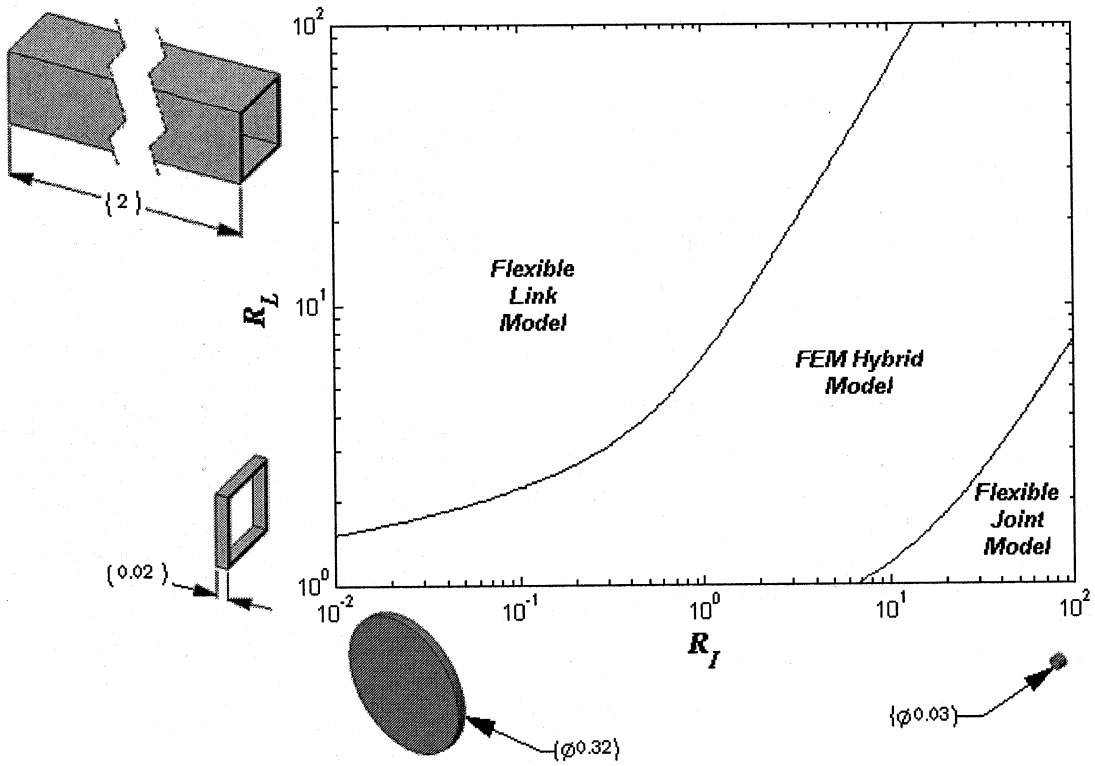


Figure 4-10: Regions of static flexibility when the joint is configured about the link's y-axis

Fig. 4-11 shows the regions of static flexibility when the joint is positioned to rotate about the link's z-axis. When R_L is increased from 1 to 100, and R_I is increased from 0.03 to 3, the flexible link model can be used. Also, when R_L is increased from just above 1 to 13, and R_I is increased from 12 to 100, the flexible joint model can be used. Figs. 4-9 to 4-11 represent the flexibility regions for the joint-link module in a specified configuration; however, they do not represent the entire flexibility regions of the joint-link module for all configurations combined.

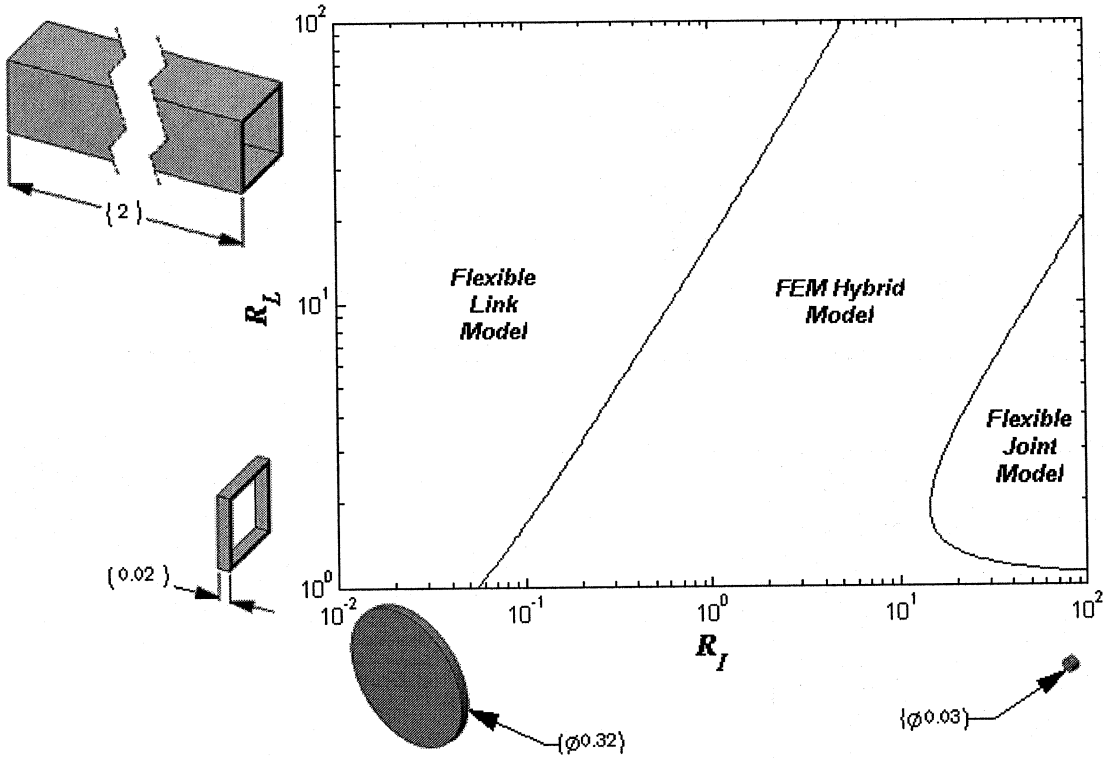


Figure 4-11: Regions of static flexibility when the joint is configured about the link's z-axis

The respective flexibility regions that are valid for all configurations are the regions where each of the flexibility models in Figs. 4-9 to 4-11 intersect each other. Thus, the regions of static flexibility for all configurations combined were plotted in a single graph in Fig. 4-12, and the regions where the joint, link, or both components are flexible for all configurations are shown. In the flexible link model region on the left hand side of Fig. 4-12, the joint can be considered rigid and the link flexible for all three

configurations. Similarly, the flexible joint region in Fig. 4-12 shows where the link can be considered rigid and the joint flexible for all three configurations.

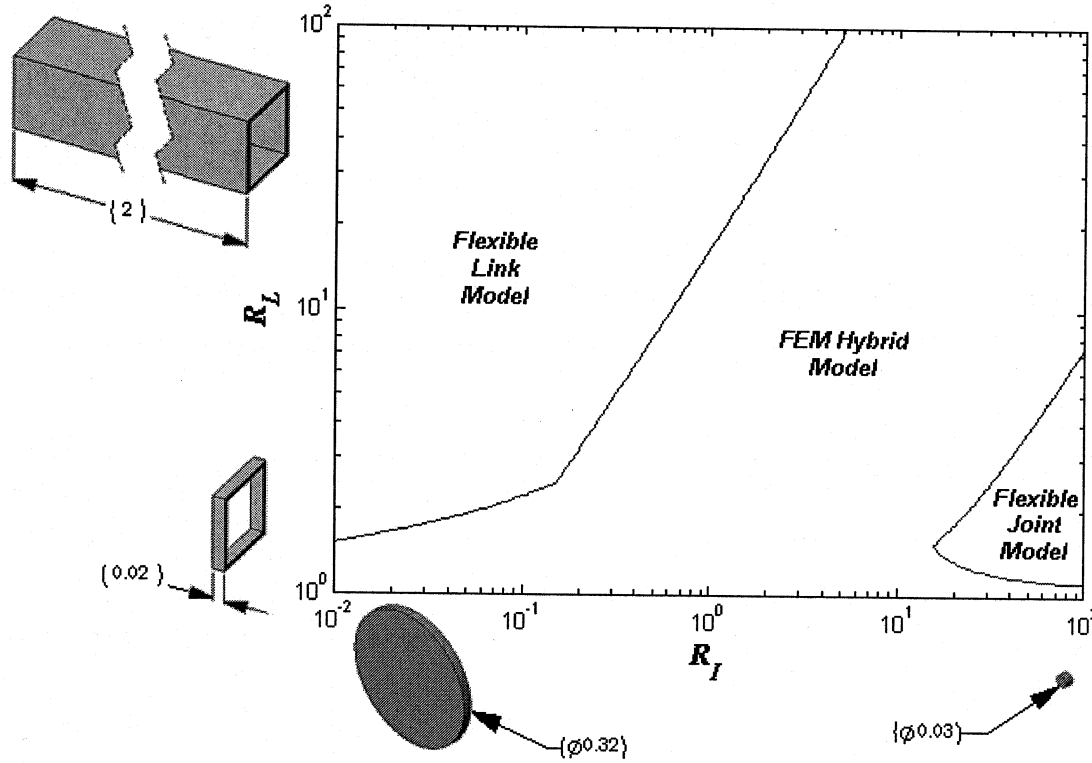


Figure 4-12: Regions of combined static flexibility for all configurations tested

4.2 Dynamic Flexibility Analysis

In order to understand the behaviour of the reconfigurable joint-link module under dynamic conditions, a free-vibration analysis is conducted to determine the natural frequencies for the system. The lowest natural frequency, or fundamental frequency, can be used as a means to estimate the dynamic flexibility of the system. In addition, a lower fundamental frequency means that the system is more flexible and a higher fundamental frequency means that the system is more rigid. Three dynamic models were compared to each other to determine the dynamic flexibility of the joint and the link: the hybrid FEM model (where both components are flexible), the flexible link model (where only the link is flexible) and the flexible joint model (where only the joint is flexible).

4.2.1 Dynamic Hybrid FEM Model

For the dynamic analysis, a single joint-link module can be idealized as two finite element beams in a similar fashion used in the static hybrid FEM model described in section 3.1. According to Thomson and Dahleh [45], using a single finite element gives good results for the lowest mode shape because the static deflection curve is close to the lowest dynamic mode shape. The same stiffness matrix used in equation (4.4) can be used for each element and the joint element can be configured about the link using equation (4.12). Thus, the same assembled global stiffness matrix derived in equation (4.16) can also be used. The dynamic hybrid FEM model uses a consistent mass matrix that coincides with the 12 DOFs found in equation (4.3). A consistent mass matrix was used instead of a lumped mass matrix because consistent mass matrices usually tend to have more accurate results for beams because the rotary inertial effects are not ignored as in lumped mass matrices. The kinetic energy for a single beam element is:

$$KE = \frac{1}{2}\int_0^L \left(\rho A(\dot{u})^2 + \rho A(\dot{v})^2 + \rho A(\dot{w})^2 + I_p(\dot{\theta})^2 \right) dx \tag{4.25}$$

where ρ is the density of the material used in the beam element and I_p is the polar mass moment of inertia. The shape functions along the local x-axis of the beam are given by:

$$u(x) = [\mathbf{N}]\{U_e\} \tag{4.26}$$

where **[N]** is a matrix containing the shape functions N_1 to N_{12} used in the stiffness derivation in section 4.1.1, and is arranged as:

$$[\mathbf{N}] = \begin{bmatrix} N_1 & 0 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_3 & 0 & 0 & 0 & N_4 & 0 & N_5 & 0 & 0 & 0 & N_6 \\ 0 & 0 & N_7 & 0 & N_8 & 0 & 0 & 0 & N_9 & 0 & N_{10} & 0 \\ 0 & 0 & 0 & N_{11} & 0 & 0 & 0 & 0 & 0 & N_{12} & 0 & 0 \end{bmatrix} \tag{4.27}$$

In equation (4.27), the shape functions N_1 to N_{12} are given by equations (4.7) to (4.10). Taking the first time derivative of equation (4.26) and substituting into equation (4.25) will yield:

$$KE = \frac{1}{2}\rho \int_0^L \{\dot{\mathbf{U}}_e\}^{\mathrm{T}} [\mathbf{N}]^{\mathrm{T}} [\mathbf{C}][\mathbf{N}]\{\dot{\mathbf{U}}_e\} dx \tag{4.28}$$

where the matrix [**C**] contains the cross section properties relating to the type of inertial forces found in equation (4.25) for each mode of deformation and can be written as:

$$[\mathbf{C}] = \begin{bmatrix} A & 0 & 0 & 0 \\ 0 & A & 0 & 0 \\ 0 & 0 & A & 0 \\ 0 & 0 & 0 & J \end{bmatrix} \tag{4.29}$$

Thus, in equation (4.28) the local mass matrix for a single beam element is given by:

$$[\mathbf{M}_e] = \rho \int_0^L [\mathbf{N}]^{\mathrm{T}} [\mathbf{C}][\mathbf{N}] dx \tag{4.30}$$

This mass matrix is given in its explicit form in Appendix A. For the joint, the mass matrix in equation (4.30) can be transformed to global coordinates by using the transformation matrix in equation (4.11) according to the joint configuration about the link, and the global mass matrix for the joint element is:

$$[\mathbf{M}]_j = [\mathbf{T}]^{\mathrm{T}} [\mathbf{M}_e]_j [\mathbf{T}] \tag{4.31}$$

Afterwards, the global mass matrix, $[\mathbf{M}]_G$, can be assembled in using the direct stiffness method described at the end of section 4.1.1. The equations of motion for the assembled joint-link element can be written as:

$$[\mathbf{M}]_G \{\ddot{\mathbf{U}}\}_G + [\mathbf{K}]_G \{\mathbf{U}\}_G = \{\mathbf{F}\}_G \tag{4.32}$$

where $\{\mathbf{F}\}_G$ is the vector of generalized moments and forces, and for free-vibration $\{\mathbf{F}\}_G$ is equal to $\{\mathbf{0}\}$. If the joint-link element is in harmonic motion, equation (4.32) takes the following form:

$$\left([\mathbf{K}]_G - \omega^2 [\mathbf{M}]_G\right)\{\hat{\mathbf{U}}\}_G = \mathbf{0} \tag{4.33}$$

where $\{\hat{\mathbf{U}}\}$ is the vector containing the natural mode shapes and ω is the natural frequency. The characteristic equation for the system in free-vibration becomes:

$$\left|[\mathbf{K}]_G - \omega^2 [\mathbf{M}]_G\right| = \mathbf{0} \tag{4.34}$$

The same boundary conditions used in equation (4.15) are used for the dynamic hybrid FEM model for the stiffness, as well as the mass matrices. After the boundary conditions are applied, the total number of DOFs for the hybrid FEM element will reduce to 12, and equation (4.34) will give 12 roots of the characteristic equation and the corresponding natural frequencies can then be determined.

4.2.2 Dynamic Flexible Link Model

The flexible link model used for the dynamic analysis uses the same boundary conditions found in equation (4.17), thus simplifying the system to a cantilever beam model attached to a rigid joint. Equation (4.34) is then solved for its eigenvalues and the natural frequencies can then be found.

4.2.3 Dynamic Flexible Joint Model

The dynamic flexible joint model assumes the joint is flexible and the link is rigid. In this model, which is shown in Fig. 4-5, the mass terms of the rigid link are lumped to the end of the joint and the joint acts as a spring with possible bending about the joint's local z-axis, axial, and torsional displacements. Assuming that the nodal displacements for the three modes deformation are the generalized coordinates, Lagrange's equation can be written as:

$$\frac{d}{dt}\left(\frac{\partial KE}{\partial\{\dot{\mathbf{U}}_e\}}\right) - \frac{\partial KE}{\partial\{\mathbf{U}_e\}} + \frac{\partial PE}{\partial\{\mathbf{U}_e\}} = \{\mathbf{F}\} \tag{4.35}$$

Using equation (4.35) the equations of motion for axial, bending and torsional modes of deformation then become:

$$\begin{aligned} m_l\ddot{u}_j + k_{u,j}u_j &= 0 \\ m_l\ddot{v}_j + k_{v,j}v_j &= 0 \\ I_{p,l}\ddot{\theta}_j + k_{t,j}\theta_j &= 0 \end{aligned} \tag{4.36}$$

where m_l is the mass of the link; u_j, v_j, and θ_j are the local axial, transverse and torsional displacements of the joint, respectively. $I_{p,l}$ is the polar mass moment of inertia for the

link and is taken about the link's x-axis for the configuration shown in Fig. 4-3(a) at the centroid of the link. For the configuration in Fig. 4-3(b) and 4-3(c) I_p is taken about the link's z- and y-axis, respectively. The natural frequencies for equation (4.36) can be determined as:

$$\omega_{n,axial} = \sqrt{\frac{A_j E_j}{m_l L_j}}$$
$$\omega_{n,bending} = \sqrt{\frac{3E_j I_{z,j}}{m_l L_j^3}} \qquad (4.39)$$
$$\omega_{n,torsion} = \sqrt{\frac{G_j J_j}{I_{p,l} L_j}}$$

In equation (4.39), the dominant flexible mode of deformation will have the lowest natural frequency, and the stiffest mode will have the highest.

4.2.4 Dynamic Flexibility Simulation and Results

For the dynamic flexibility analysis of a single reconfigurable joint-link module, R_L and R_I from equations (4.23) and (4.24) were used to vary the length of the link and cross section area of the joint, thus fixing the link inertia and joint length to the values defined in section 4.1.4. The values of R_L and R_I were varied from 1 to 100 and 0.01 to 100, respectively. Also, the same material properties used in the static analysis were applied to the joint and the link. A free vibration analysis was then performed using the hybrid FEM, flexible link, and flexible joint models for the given range of link and joint sizes for each of the configurations shown in Fig. 4-3. The lowest natural frequency was determined, based on which the flexible link and flexible joint models were each compared to the fully flexible hybrid FEM model. The results were then used as a basis for determining the dynamic flexibility of the link or joint components.

Figure 4-13(a) shows a comparison of the lowest natural frequencies obtained from the hybrid model and the flexible link model. When R_I decreases to below 0.1, there are small differences between the natural frequencies for both models, because the joint becomes rigid. When R_L decreases to 1 and R_I increases to 100, there exists a larger

difference between the two models since the joint is no longer rigid in that region. Figure 4-13(b) shows the comparison of the lowest natural frequencies obtained from the hybrid and flexible joint models. When R_I decreases to 0.01 and R_L decreases to 1, there exist large differences between the two models. In this region, the joint is rigid because of a higher joint inertia, and the link is more rigid because its length is shorter. When R_I is increased to 100, and R_L is increased to 10, the link becomes more rigid, and the joint is relatively flexible. However, when both R_L and R_I are increased to 100, some bending deformation occurs in the link, and the joint remains flexible in comparison to the link.

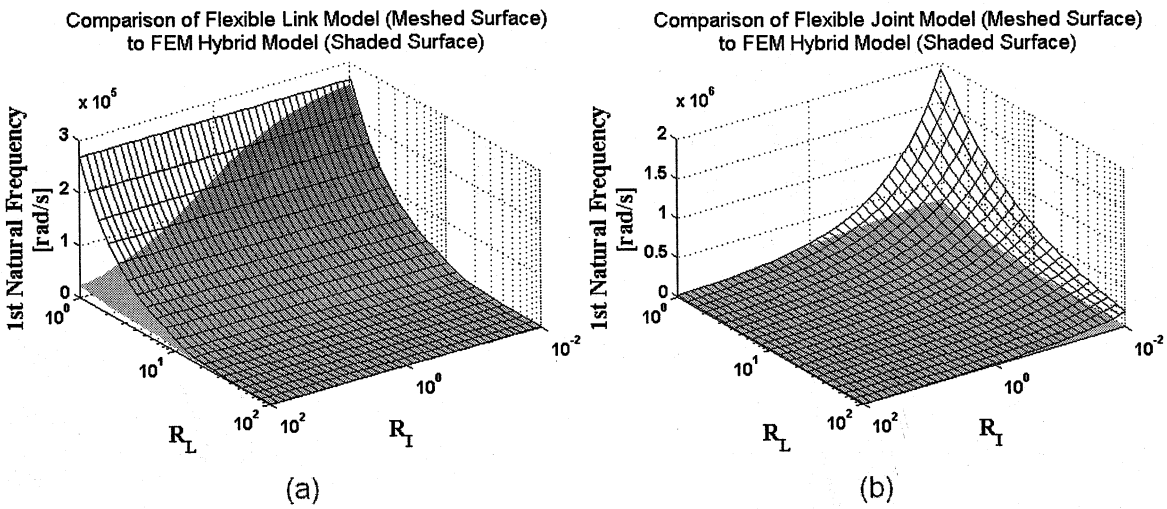


Figure 4-13: Dynamic analysis results when the joint is configured about the link's x-axis

The same pattern for the flexible link and flexible joint models exist when the joint is configured about the link's y- or z-axis as shown in Fig. 4-14 with the exception that when R_L equals 1 and R_I approaches 0.01, there is less discrepancy between the flexible link and hybrid models. The y- and z- configurations give the exact same results because the configurations are orthogonal to each other, hence they will have the same natural frequencies. Also, since the link has a square cross section, the bending modes in the y- and z- direction yield the same results.

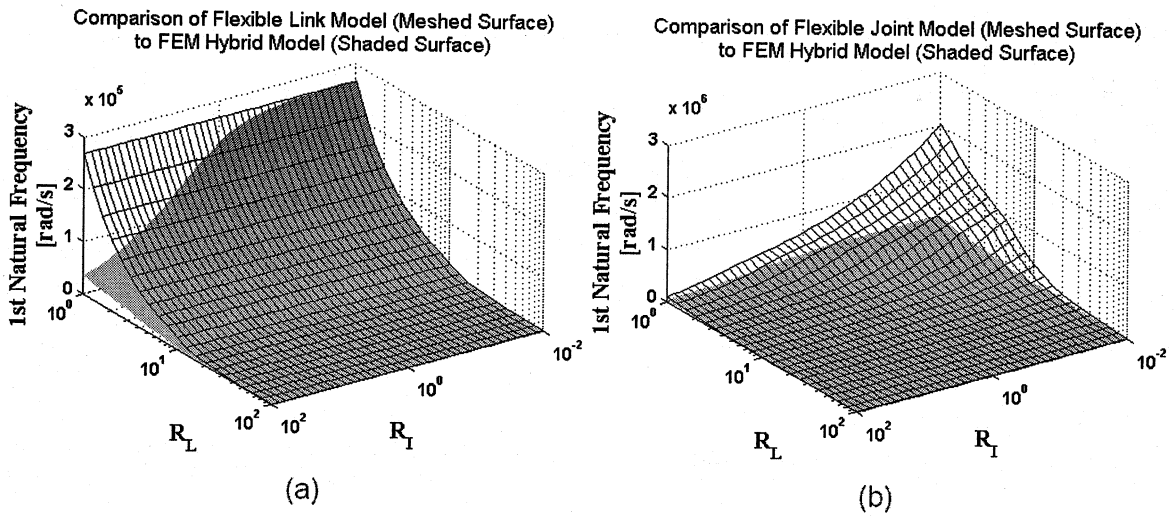


Figure 4-14: Dynamic analysis results when the joint is configured about the link's y- or z-axis

To coincide with the static flexibility analysis, the regions of dynamic flexibility were then graphed for the reconfigurable joint-link module with varying link and joint sizes. Fig. 4-15 shows the regions of dynamic flexibility with varying values of R_I and R_L when the joint rotation axis is aligned with the link's x-axis. In the flexible link model region, the zigzag area in the dividing curve is due to the change in lowest natural frequencies corresponding to different DOFs in the link-joint hybrid model. When R_L is greater than 10, and R_I is close to 1, the lowest fundamental frequency dominates due to link bending. When R_L is less than 10, and R_I is close to 1, the lowest natural frequency is due to another DOF, such as link torsion, or bending about the x-z plane. In this graph, the flexible link model can be used when R_L increases from 1 to 100, and R_I increases from 0.01 to 10. When R_L increases from 1 to 9, and R_I increases from 12 to 100, the joint is flexible and the link becomes rigid, thus the flexible joint model can be used. Fig. 4-16 shows the dynamic flexibility results when the joint is configured to rotate about the link's y-axis. Again, the results are the same when the joint is configured about the link's z-axis since both configurations are axisymmetric about the link's x-axis. In Fig. 4-16 the zig-zag area in the dividing line of the flexible link model region is more profound than that of Fig. 4-15. Here, when R_L is increased from 1 to 100, and R_I is increased from 0.2 to 2, the link is more flexible than the joint, thus the flexible link model can be used. When R_L is increased from 2 to 30, and R_I is increased from 10 to 100, the link becomes rigid relative to the joint, thus the flexible joint model can be used.

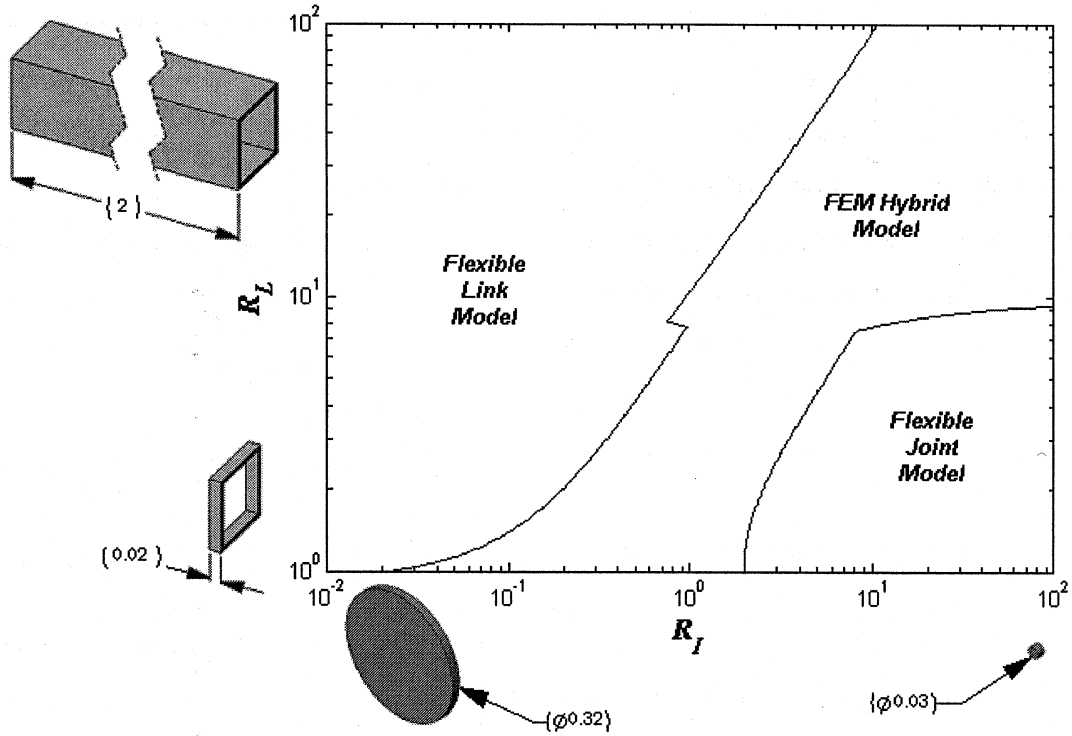


Figure 4-15: Regions of dynamic flexibility when the joint is aligned with the link's x-axis

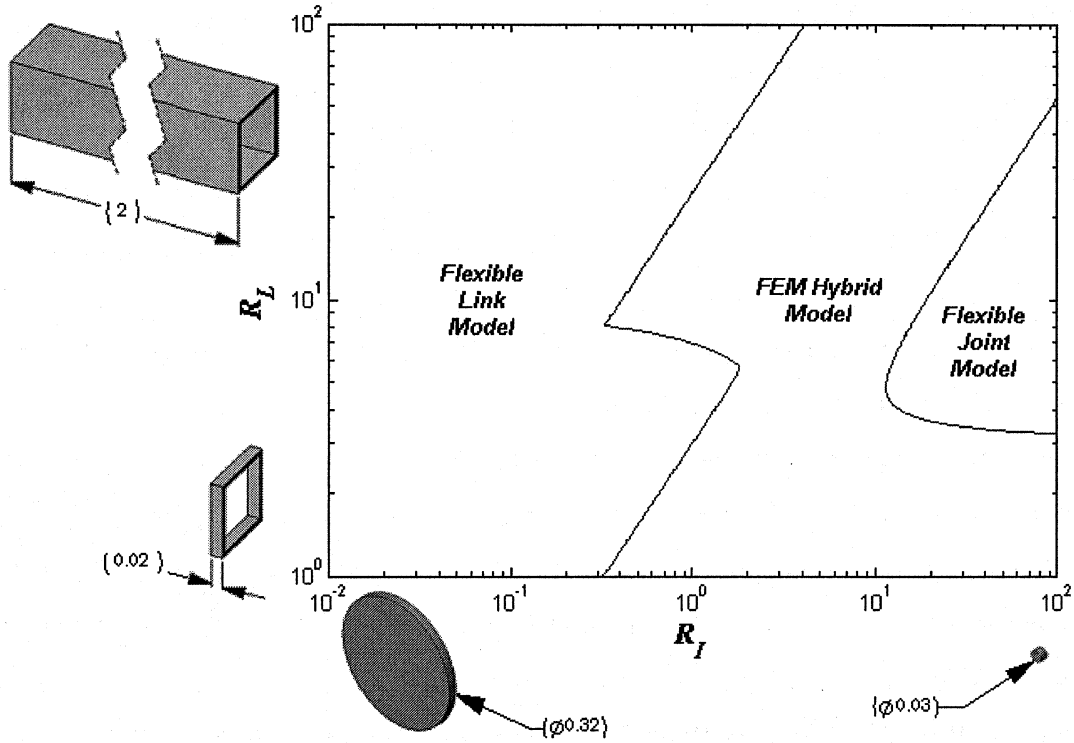


Figure 4-16: Regions of dynamic flexibility when the joint is aligned with the link's y- or z-axis

Afterwards, Fig. 4-15 and 4-16 were plotted together and the regions where all of the configurations can use either the flexible link, flexible joint or hybrid FEM models are shown in Fig. 4-17. Here, the flexible link model region is where the link is dynamically flexible and the joint is rigid for all configurations tested. Similarly, the flexible joint model region is where the joint is dynamically flexible and the link is rigid in all configurations tested.

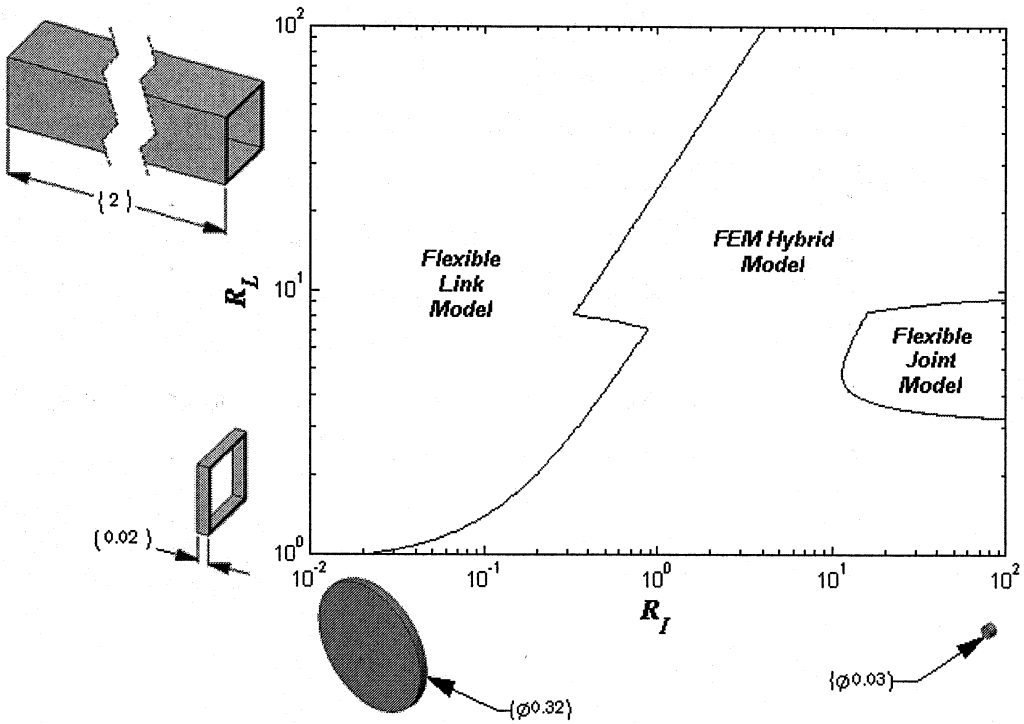


Figure 4-17: Regions of combined dynamic flexibility for all configurations tested

By knowing the flexibility of each component, the appropriate model can be used and the overall positional accuracy of the joint-link module can be improved by including the effects of static and dynamic flexibility. Flexibility contours like those in Figs. 4-12 and 4-17 can be plotted for static and dynamic test cases and designers can simply determine the flexibility of the joint and link modules based for a range of joint and link sizes. This method can aid in the design and control of link-joint modules by accounting for joint and/or link flexibility. Furthermore, the flexibility of the joint-link models presented in this chapter can be used as a basis to perform the kinetostatic analysis of an entire MRR.

5. Kinetostatic Analysis

This chapter explains the kinetostatic analysis of reconfigurable revolute joint, prismatic joint and link modules. The modules described in Chapter 3 were rigid body models. Here, the applied loads on the robot are accounted for and the orientation equations described in Chapter 3 are used in conjunction with the flexibility (compliance and stiffness) to determine the effects of the tip deformation in any possible robot position. Chapter 4 showed that both revolute joint and link modules could become more flexible for a certain range of sizes. Also, it becomes necessary to model the revolute joint lengths since the revolute joints may become flexible under torsion, axial deformation and bending. Thus, in this chapter, the flexibility of each module is taken into account in order to determine the elastic deformation under an applied load. The flexible modules are assumed to be undergoing low-speed (quasi-static) motion with negligible acceleration, thus the dynamic inertia effects may be ignored. Therefore, kinetostatic analysis can be facilitated, and the robot tip deflections can be determined along with the flexible tip position in the robot's workspace. The kinetostatic modeling in this thesis includes the flexibility of each component and can later be used to determine the flexible tip workspace, and other factors such as the maximum allowable payload a robot can take, and the position of the robot that gives the maximum deflection.

In the following sections, two kinetostatic models are presented. The first model uses the compliance and Jacobian in order to solve for the global tip deflections of the end module for a given pose of the MRR. The second model is based on the finite element model described in Chapter 4 and is used in conjunction with the recursive rotation matrices found in Chapter 3 in order to determine the global deflections at the end nodes of each module. A solid body with lower compliance, or higher stiffness, will undergo less deformation. This is important for robotic applications such as automated manufacturing, where accuracy is a main concern. Thus, it is important to include the flexibility of reconfigurable robot modules. In this chapter, the Compliance-Jacobian method was first developed, and compared to a modified finite element kinetostatic

method. For both models, post-processing methods were also defined in order to solve for the local forces, displacements and stresses of each module.

5.1 Compliance-Jacobian Method

The Compliance-Jacobian method can be used to solve for the tip deflection of an assembled MRR. The compliance is a measure of the ability of a solid body to deform under applied loads, whereas the stiffness is a measure of the ability of a solid body to resist deformation under applied loads. This method was used in references [39-40] to perform a kinetostatic analysis of a flexible parallel kinematic machine. In this thesis, this method was modified to facilitate the kinetostatic analysis of a serial MRR. The method first involves determining the appropriate compliance matrix, based on Euler beam theory, to determine the tip deformation of each module with allowable local axial displacement, bending about the local y- and z- axes, and torsional deformation. Afterwards, a Jacobian matrix can be used to transform each module's local forces and displacements to global coordinates. Unlike the finite element used in Chapter 4, each element does not need to be transformed before the global system equations are assembled. However, this method initially does not solve for the global tip deformations for the modules before the tip module (last module in the open chain), but these global as well as local deformations can be determined after the last module's tip deformation is solved during the post-processing stage.

5.1.1 Local Compliance Derivation

A second-order method for determining the tip deformation at one end of a beam with an applied force and moment [46] was used in order to determine the compliance terms for bending about the y- and z- axis. Chapter 4 demonstrated that a single finite element beam is suitable to determine the tip deformation of a single beam. Thus a single beam will also be used in this model, and the results can also be compared to beam theory. Consider a cantilever beam of length L, with a single transverse load f_y in the positive y-direction and a moment m_z about the z-axis at one end, as shown in Fig. 5-1(a).

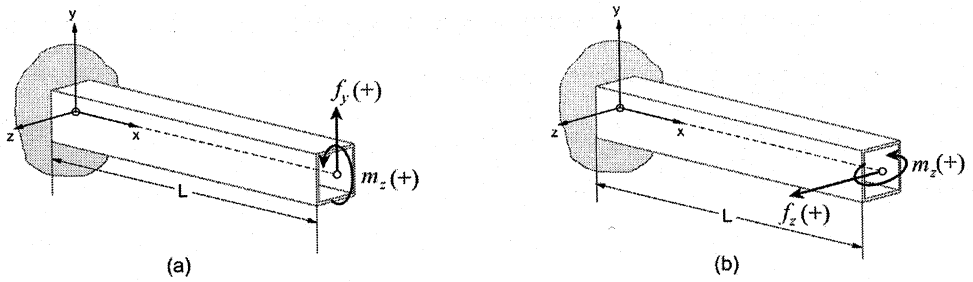


Figure 5-1: Sign conventions for beam theory

For the cantilever beam in Fig. 5-1(a), taking the sum of moments for a small increment of the beam about the z-axis gives:

$$m_z(x) - f_y(L-x) - m_z = 0 \tag{5.1}$$

Thus the Moment-Curvature equation is given by:

$$EI_z \frac{d^2v}{dx^2} = m_z(x) = f_y(L+x) + m_z \tag{5.2}$$

where E is the modulus of elasticity for the material. Integrating equation (5-2) once, and then twice gives the following:

$$EI_z \frac{dv}{dx} = \psi(x) = m_z x + f_y L x - f_y \frac{x^2}{2} + c_1 \tag{5.3}$$

$$EI_z v = m_z \frac{x^2}{2} + f_y L \frac{x^2}{2} - f_y \frac{x^3}{6} + c_1 x + c_2 \tag{5.4}$$

Since the beam in Fig 5-1(a) is fixed at one end, it is subject to the boundary conditions $\frac{dv(0)}{dx} = 0$ and $v(0) = 0$. Thus, both constants become zero in equations (5.3) and (5.4) and the angular and transverse deflections due to bending at the tip of the beam (where $x = L$) can be written as:

$$\psi(L) = \frac{1}{EI_z}\left(m_z L + f_y \frac{L^2}{2}\right) \tag{5.6}$$

$$v(L) = \frac{1}{EI_z}\left(m_z \frac{L^2}{2} + f_y \frac{L^3}{3}\right) \tag{5.7}$$

The stiffness can be defined as a ratio of the load to the deflection. Thus, the compliance is the ratio of the deflection to the load, and a single compliance term is the inverse of the corresponding stiffness term. Therefore, the deflections at the tip of a single beam can be related to the force and compliance as:

$$Cf_{tip} = U_{tip} \tag{5.8}$$

In equations (5-6) and (5-7), when the applied moment about the z-axis is set to zero, the equations become:

$$\frac{L^2}{2EI_z} f_y = \psi \tag{5.9}$$

$$\frac{L^3}{3EI_z} f_y = v \tag{5.10}$$

Also, in equations (5-6) and (5-7), when the applied force in the y-direction is set to zero, the equations become:

$$\frac{L}{EI_z} m_z = \psi \tag{5.11}$$

$$\frac{L^2}{2EI_z} m_z = v \tag{5.12}$$

Using the definition in equation (5-8) the compliance terms that contribute to the transverse deflection in the y-direction v and angular bending deflection about the z-axis ψ in equations (5-9) and (5-10) are:

$$C_{bend,\psi_1} = \frac{L^2}{2EI_z} \tag{5.13}$$

$$C_{bend,v_1} = \frac{L^3}{3EI_z} \tag{5.14}$$

Also, the compliance terms that contribute to the transverse deflection in the y-direction v and angular bending deflection about the z-axis ψ in equations (5.11) and (5.12) become:

$$C_{bend,\psi_2} = \frac{L}{EI_z} \tag{5.15}$$

$$C_{bend,v_2} = \frac{L^2}{2EI_z} \tag{5.16}$$

The second order method used in equations (5.1) to (5.16) to determine the tip deflection can also be applied to determine the compliance terms for bending about the y-axis shown in Fig. 5-1(b). When the applied moment about the y-axis m_y is set to zero, the compliance terms that contribute to the transverse deflection in the z-direction w and angular bending deflection about the y-axis φ become:

$$C_{bend,\phi_1} = -\frac{L^2}{2EI_y} \tag{5.17}$$

$$C_{bend,w_1} = \frac{L^3}{3EI_y} \tag{5.18}$$

The negative sign in equation (5-17) is because the sign convention in the x-z plane is different from that in the x-y plane. When the applied force along the y-axis f_y is set to zero, the compliance terms that contribute to the transverse deflection in the z-direction w and angular bending deflection about the y-axis φ are:

$$C_{bend,\psi_2} = \frac{L}{EI_y} \tag{5.19}$$

$$C_{bend,w_1} = -\frac{L^2}{2EI_y} \tag{5.20}$$

For the axial compliance terms, a linear displacement function is assumed since this type of deformation acts along a single axis. The axial compliance term becomes:

$$C_{axial,u} = \frac{L}{AE} \tag{5.21}$$

Also, for torsional deformation along the length of the beam element, a linear torsion angle variation is assumed and the torsion compliance term is:

$$C_{torsion,\theta} = \frac{L}{GJ} \tag{5.22}$$

Writing out equation (5-8) in matrix form, and substituting the compliance components from equations (5-13) to (5-22) will give the tip deformation for a single beam:

$$\begin{bmatrix} \frac{L}{AE} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{L^3}{3EI_z} & 0 & 0 & 0 & \frac{L^2}{2EI_z} \\ 0 & 0 & \frac{L^3}{3EI_y} & 0 & -\frac{L^2}{2EI_y} & 0 \\ 0 & 0 & 0 & \frac{L}{GJ} & 0 & 0 \\ 0 & 0 & -\frac{L^2}{2EI_y} & 0 & \frac{L}{EI_y} & 0 \\ 0 & \frac{L^2}{2EI_z} & 0 & 0 & 0 & \frac{L}{EI_z} \end{bmatrix} \begin{Bmatrix} f_x \\ f_y \\ f_z \\ m_x \\ m_y \\ m_z \end{Bmatrix}_{tip} = \begin{Bmatrix} u \\ v \\ w \\ \theta \\ \phi \\ \psi \end{Bmatrix}_{tip} \tag{5.23}$$

The 6×6 matrix in equation (5.23) is the local compliance matrix for a single module, denoted as $[\mathbf{C}_e]_i$. This beam compliance can be used to represent the compliance of each of the modules described in Chapter 3. To check the validity of the compliance model, equation (5.23) was solved for a single beam with force applied to the tip, and the results were compared to beam theory as shown in Fig. 5-2 below.

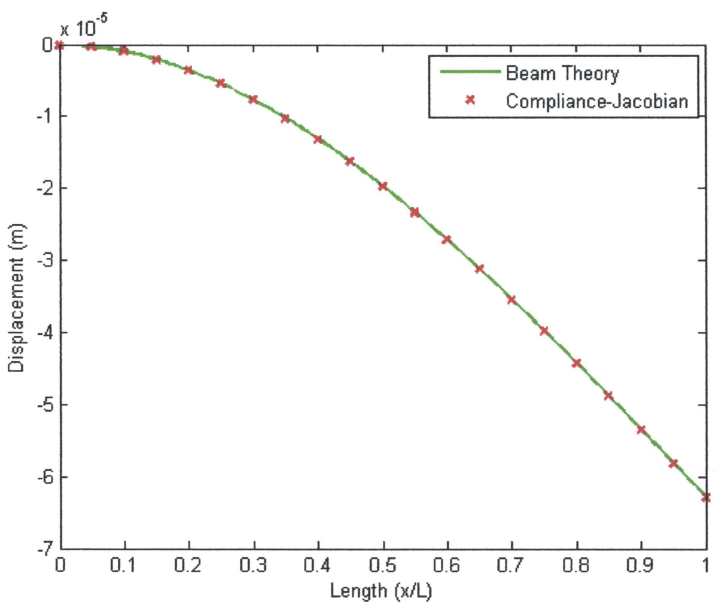


Figure 5-2: Comparison of Compliance model with Beam Theory

Fig. 5-2 shows that the tip deformation at x = L was the same for the compliance model and beam theory, and the results also agree with Fig. 4-1 in Chapter 4 for the finite element beam. Therefore, a single compliance beam model can accurately predict the tip deformation of a single module. Here, the revolute and prismatic joint modules can be represented as beams with hollow or solid circular cross sections and the link modules can be represented as beams with hollow or solid rectangular cross sections. For prismatic joint modules, the length terms in equation (5.23) can be taken as the final length of the prismatic joint after extension. Now that the local compliance matrix was defined for a single module, the next step was to transform the compliance matrix from local to global coordinates.

5.1.2 Deformation Jacobian

The deformation Jacobian matrix is used to transform the local module forces and displacements to the global frame of reference. This Jacobian matrix also accounts for reconfigurability since the rotation matrices which define the configuration of each module (found in Chapter 3) are included. Furthermore, since the interest is only in the tip deformations of each module, the tip forces and moments on one module must be converted backwards to the tips of the previous modules. This was done by the use of a rotation matrix and moment vector within the Jacobian matrix to convert the forces to moments where appropriate, and convert the local variables to global variables. For a single module the deformation Jacobian matrix is given by:

$$\left[\mathbf{J}_{d,i}\right]=\begin{bmatrix} \mathbf{z}_{1,i} & \mathbf{z}_{2,i} & \mathbf{z}_{3,i} & \mathbf{z}_{1,i}\times\left(\mathbf{P}_{n+1}-\mathbf{P}_{i+1}\right) & \mathbf{z}_{2,i}\times\left(\mathbf{P}_{n+1}-\mathbf{P}_{i+1}\right) & \mathbf{z}_{3,i}\times\left(\mathbf{P}_{n+1}-\mathbf{P}_{i+1}\right) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{z}_{1,i} & \mathbf{z}_{2,i} & \mathbf{z}_{3,i} \end{bmatrix}_{6\times 6} \qquad \text{.....(5.24)}$$

where $\mathbf{z}_{1,i}$, $\mathbf{z}_{2,i}$, and $\mathbf{z}_{3,i}$, are the first, second and third columns, respectively, of the recursive rotation matrix after motion occurs ($\mathbf{R}_{0i,m}$) given in equations (3.22), (3.30) and (3.47) in Chapter 3 for the relative, ZRP, and quasi-global connectivity methods, respectively. This Jacobian matrix is different from a velocity Jacobian for the fact that it transforms the deformations backwards from the tip of the n^{th} module to the tip of the i^{th} module, where a velocity Jacobian matrix transforms velocities from the i^{th} module to the

n^{th} module. Also, the deformation Jacobian allows infinitesimal motion in three linear directions and three axes of rotation, where the velocity Jacobian allows infinitesimal motion in three linear directions and only one axis of rotation for each module. The **z** terms in equation (5.24) are the unit directional vectors to transform linear displacements and the cross-product terms are the moment vectors to transform angular displacements. The ($\mathbf{P}_{n+1}$ - $\mathbf{P}_{i+1}$) terms in the moment vectors denote the position vector from the tip of the end module to the tip of the i^{th} module. Note that equation (5.24) holds only for a single module and the deformation Jacobian for an entire robot with n total modules is:

$$\left[\mathbf{J}_{d,n}\right]=\left[\mathbf{J}_{d,1} \quad \mathbf{J}_{d,2} \quad \mathbf{J}_{d,3} \quad \cdots\cdots\cdots \quad \mathbf{J}_{d,n}\right]_{6\times 6n} \tag{5.25}$$

Equation (5.25) can be used to determine the vector of local forces for all modules given the force at the tip of the last (or n^{th}) module. However, if it is required to transform a vector such as the local displacements at the tip of module i in an open chain of modules, equation (5.24) cannot be used alone for any module after the first module. Instead, equation (5.25) should be modified with the exception that $n = i$, so that the size of the Jacobian becomes $6\times 6i$. This modified Jacobian can be referred to as:

$$\left[\mathbf{J}_{d,(1\ to\ i)}\right]=\left[\mathbf{J}_{d,1} \quad \mathbf{J}_{d,2} \quad \mathbf{J}_{d,3} \quad \cdots\cdots\cdots \quad \mathbf{J}_{d,i}\right]_{6\times 6i} \tag{5.26}$$

where the subscript (1 *to i*) means that the Jacobian is taken from module 1 to module i.

5.1.3 System Equations and Assembly

In this chapter, the system equations were created in order to use the space-fixed reference frame rotations, given in equation (3.6). The equation that relates the local forces at the tips of each module to the known global forces at the tip of the n^{th} module can be determined as:

$$\left\{\mathbf{f}_{e,tip}\right\}=\left[\mathbf{J}_{d}\right]^{\mathrm{T}}\left\{\mathbf{F}_{tip}\right\}_{G} \tag{5.27}$$

where $\{\mathbf{f}_{e,tip}\}$ is the vector containing the local forces at the tip of each module in the body-fixed frame and $\{\mathbf{F}_{tip}\}_G$ is the vector containing the known global forces acting on the tip of the n^{th} module in the global coordinate frame. Also, the global to local displacement relation is given by:

$$\{\mathbf{U}_{tip}\}_G = [\mathbf{J}_d]\{\mathbf{U}_{e,tip}\} \tag{5.28}$$

where $\{\mathbf{U}_{e,tip}\}$ is a vector containing the local displacements at the tip of each module in the body-fixed coordinate frame and $\{\mathbf{U}_{tip}\}_G$ is the vector containing the global displacements at the tip of the n^{th} module in the global coordinate frame. Equation (5.23) can be written in terms of the assembled compliance for n modules as:

$$\{\mathbf{U}_{e,tip}\} = [\mathbf{C}]\{\mathbf{f}_{e,tip}\} \tag{5.29}$$

where [**C**] is the assembled compliance for n total modules and can be written as:

$$[\mathbf{C}]_{6n\times 6n} = diag\left([\mathbf{C}_e]_1, [\mathbf{C}_e]_2, \ldots\ldots, [\mathbf{C}_e]_n\right) \tag{5.30}$$

Unlike the finite element method stiffness matrix, the assembled compliance in this method requires no transformations from local to global coordinates for each module before the compliance matrix is assembled for the system. Also, since this method is based at one of the end tips of each module, whereas the finite element method is node-based at both end tips, there is no overlapping of any individual local compliance terms required for the global compliance assembly; unlike the finite element method, where there are stiffness terms that overlap for each element at a common node. In addition, no further boundary conditions need to be applied to the compliance matrix before solving for the unknown tip displacements. In the Compliance-Jacobian method presented, the assembled compliance matrix is created first, and then transformed while solving for the unknown displacements. Substituting equation (5.29) into (5.28) gives:

$$\{\mathbf{U}_{tip}\}_G = [\mathbf{J}_d][\mathbf{C}]\{\mathbf{f}_{e,tip}\} \tag{5.31}$$

To determine the global displacements at the tip of the n^{th} module, given the applied global forces at tip of the n^{th} module, equation (5.27) can be substituted into equation (5.31) to convert the local forces to global forces, along with the Jacobian in equation (5.25) to give:

$$\{\mathbf{U}_{tip,n}\}_G = [\mathbf{J}_{d,n}][\mathbf{C}][\mathbf{J}_{d,n}]^{\mathrm{T}}\{\mathbf{F}_{tip,n}\}_G \tag{5.32}$$

where $\{\mathbf{U}_{tip,n}\}_G$ is a 6×1 vector containing the global displacements at the tip of the assembled MRR where the payload is applied, $[\mathbf{J}_{d,n}]$ is the $6\times6n$ deformation Jacobian from equation (5.25), $[\mathbf{C}]$ is the $6n\times6n$ assembled compliance assembled compliance matrix given in equation (5.30) and $\{\mathbf{F}_{tip,n}\}_G$ is the 6×1 global force vector containing the applied forces at the tip of the n^{th} module.

If it is necessary to include the weights of each module, along with the weights of each joint's motor, equation (5.32) can be modified as:

$$\left\{\mathbf{U}_{tip,n}\right\}_G = \left[\mathbf{J}_{d,n}\right]\left[\mathbf{C}\right]\left(\left[\mathbf{J}_{d,n}\right]^{\mathrm{T}}\left\{\mathbf{F}_{tip,n}\right\}_G + \left\{\mathbf{f}_{e,weight}\right\} + \left\{\mathbf{f}_{e,motor}\right\}\right) \tag{5.33}$$

where $\{\mathbf{f}_{e,weight}\}$ is the vector containing the lumped weights at the tips of each module in local coordinates, and $\{\mathbf{f}_{e,motor}\}$ is the vector containing the weights of the prismatic and revolute joint motors in local coordinates. Since the Compliance-Jacobian model is based on only one node for each module, the weight of one module is located at the tip node in the negative y-direction in global coordinates, and that node also includes the weight of the next consecutive module. Also, the motor weights for the prismatic and joint modules are located at the first node, or base node, of each element. However, since no first node exists in this method, motor weights are placed at the tip nodes of the preceding modules. Furthermore, since the self-weights and motor weights are known in global coordinates, it is necessary to convert the weight vectors to local coordinates in order to use equation (5.33). The local weight vector containing the self-weights of all modules, and the vector that contains the weights of the joint motors can be written as:

$$\left\{\mathbf{f}_{e,weight}\right\} = \sum_{i=1}^{n}\left[\mathbf{J}_{d,(1\,to\,i)}\right]^{\mathrm{T}}\left\{\mathbf{f}_{weight,i}\right\} \tag{5.34}$$

where $[\mathbf{J}_{d,(1\ to\ i)}]$ is obtained from equation (5.26) and $\{\mathbf{f}_{weight,i}\}$ is the vector containing the lumped global weights for module i and module $i+1$. Note that the summation in equation (5.34) does not imply a simple summation because the vectors produced are not equal length. For example, when $i = 1$ the vector is 6×1, and then $i = 2$ the vector is 12×1. Thus, the first six rows of the first vector are added to the first six rows of the second

vector, and so on until the n^{th} module is reached. The self-weight vector for each module can be written in global coordinates as:

$$\{\mathbf{f}_{weight,i}\} = \left\{0 \quad -\tfrac{W_i}{2} - \tfrac{W_{i+1}}{2} \quad 0 \quad 0 \quad 0 \quad 0\right\}^{\mathrm{T}} \tag{5.35}$$

The vector containing the weights of each motor in the local frame can be written as follows:

$$\{\mathbf{f}_{e,motor}\} = \sum_{i=1}^{n} \left[\mathbf{J}_{d,(1\,to\,i)}\right]^{\mathrm{T}} \{\mathbf{f}_{motor,i+1}\} \tag{5.36}$$

where $\{\mathbf{f}_{motor,i+1}\}$ is the vector containing the global motor weights for module $i+1$. In other words, since the motor weight is located at the base node instead of the tip node of each joint element, if module $i+1$ is a joint, the motor weight will be transferred to the tip of module i since this method has single tip nodes for each element. Also, the summation rule is the same used for equation (5.34).

5.1.4 Post-Processing

The post-processing stage can be used to determine the local unknowns such as local forces, displacements, and stresses. After the global tip displacements are solved using equations (5.32) or (5.33), the next step is to determine the vector of local tip forces for all modules, which is given by:

$$\{\mathbf{f}_{e,tip}\} = \left[\mathbf{J}_{d,n}\right]^{\mathrm{T}} \{\mathbf{F}_{tip}\}_G + \{\mathbf{f}_{e,weight}\} + \{\mathbf{f}_{e,motor}\} \tag{5.37}$$

If the self-weights of each module and the motor weights are to be ignored, equation (5.37) reduces to equation (5.27). If it is required to determine the global displacements at the tip of the i^{th} module, equation (5.31) can be modified as:

$$\{\mathbf{U}_{tip,i}\}_G = \left[\mathbf{J}_{d,(1\,to\,i)}\right]\left[\mathbf{C}_{(1\,to\,i)}\right]\{\mathbf{f}_{e,tip}\} \tag{5.38}$$

Afterwards, the local tip displacements can be calculated using equation (5.28) rewritten as:

$$\{\mathbf{U}_{e,tip,i}\} = \left[\mathbf{J}_{d,i}\right]^{\mathrm{T}} \{\mathbf{U}_{tip,i}\}_G \tag{5.39}$$

The next step is to calculate the local stresses at the base of each module. There are six stress cases considered: axial, bending in the x-y plane, bending in the x-z plane, shear in the y-direction, shear in the z-direction and torsion stresses. This means that one end of each module undergoes stresses due to the combined loads. The axial stress at one end of module *i* is given by:

$$\sigma_{axial,i} = \frac{f_{x,i}}{A_i} \tag{5.40}$$

The stresses due to bending about the z-axis and y-axis at one end of module *i*, respectively, can be written as:

$$\sigma_{bend,z} = \frac{m_{z,i} y}{I_{z,i}} \tag{5.41}$$

$$\sigma_{bend,z} = \frac{m_{y,i} z}{I_{y,i}} \tag{5.42}$$

Usually the transverse shear stresses due to bending loads have negligible contributions to the failure of the beams as compared to the bending, axial and torsional stresses. However, they are included in this thesis in order to provide a more conservative model. For modules with a circular (solid or hollow) cross section, such as prismatic and revolute joints, the transverse shear stress is given by [46]:

$$\tau_{shear,y} = \frac{2 f_{y,i}}{A_i} \tag{5.43}$$

$$\tau_{shear,z} = \frac{2 f_{z,i}}{A_i} \tag{5.44}$$

If the link modules are assumed to have a hollow rectangular cross section, the transverse shear stresses in the y- and z- directions can be determined from:

$$\tau_{shear,y} = \frac{f_{y,i} Q_i}{I_z t_i} \tag{5.45}$$

$$\tau_{shear,z} = \frac{f_{z,i} Q_i}{I_y t_i} \tag{5.46}$$

where Q_i is the first moment of the area with respect to the neutral axis above the level y' or z' [46], and t_i is the thickness of the member. If the link modules are assumed to have a solid cross section, the maximum transverse shear stresses in the y- or z- direction can be calculated using [46]:

$$\tau_{shear,y} = \frac{3 f_{y,i}}{2 A_i} \tag{5.47}$$

$$\tau_{shear,z} = \frac{3 f_{z,i}}{2 A_i} \tag{5.48}$$

The stress due to torsion for a solid, or hollow joint module with circular cross section is given by:

$$\tau_{torsion,i} = \frac{m_{x,i} r_{outer,i}}{I} \tag{5.49}$$

where $r_{outer,i}$ is the outer radius of the joint module, and $I = I_y = I_z$. If the link modules have a hollow rectangular cross section, the stress due to torsion is:

$$\tau_{torsion,i} = \frac{m_{x,i}}{2 t_i A_{m,i}} \tag{5.50}$$

where $A_{m,i}$ is the area within the median curve defined in [46]. If the link modules have a solid rectangular cross section, the stress due to torsion is:

$$\tau_{torsion,i} = \frac{m_{x,i}}{\alpha_i b_{outer,i} h_{outer,i}^2} \tag{5.51}$$

where $b_{outer,i}$ and $h_{outer,i}$ are the outer base length and height of the module's solid cross section, respectively, and α_i is a dimensionless torsional stress constant found in [46]. Using equations (5.40) to (5.51) the stress resultants can be determined for each module and the Von Mises equivalent stress can be calculated using the following equation [47]:

$$\sigma_{VM,i} = \frac{1}{\sqrt{2}}\left[\left(\sigma_x - \sigma_y\right)^2 + \left(\sigma_y - \sigma_z\right)^2 + \left(\sigma_z - \sigma_x\right)^2 + 6\left(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{xz}^2\right)\right]^{\frac{1}{2}} \tag{5.52}$$

Afterwards, a factor of safety based on the Von Mises yield criterion can be determined for each module as:

$$FS_{y,i} = \frac{\sigma_{yield,i}}{\sigma_{VM,i}} \tag{5.53}$$

where $\sigma_{yield,i}$ is the yield stress of the module's material. If $FS_{y,i}$ is less than 1, the module structurally fails.

5.2 Finite Element Kinetostatic Method

The finite element system equations in this chapter differ from those of Chapter 4 because space-fixed rotations instead of body-fixed rotations were used in order to be compatible with the kinematic equations presented in Chapter 3, and coincide with the Compliance-Jacobian method presented in the previous section. Unlike the Compliance-Jacobian method, the finite element method is node-based with each beam element having two nodes, and information such as global displacements are obtained at each node instead of just the tip node alone. Here, the link modules are modeled as solid or hollow rectangular beams, and the joint modules are modeled as solid or hollow circular beams. Also, the length of the prismatic joint is taken as the final length after joint extension.

5.2.1 System Equations and Assembly

This method uses the same stiffness matrix which was derived in Chapter 4, using equation (4.4). For convenience, this stiffness matrix is shown in its explicit form in Appendix A. The transformation matrix presented in equation (4.11) can be modified to include the rotation matrices found in equations (3.22), (3.30) and (3.47) in Chapter 3 for the relative, ZRP, and quasi-global connectivity methods in order to use space-fixed rotations. Thus, the transformation matrix used in this chapter is:

$$\left[\mathbf{T}_i\right]_{12\times 12} = diag\left(\left[\mathbf{R}_{0i,m}\right]_{3\times 3}\right) \tag{5.54}$$

For a single module, the local force vector can be written using space-fixed rotations found in equation (5.54) as:

$$\{\mathbf{f}_{e,i}\} = [\mathbf{T}_i]^{\mathrm{T}} \{\mathbf{f}_i\} \tag{5-55}$$

where $\{\mathbf{f}_i\}$ is a vector containing the global forces at each node of module *i*. Also, the relation between the local and global displacements for a single module can be written as:

$$\{\mathbf{U}_{e,i}\} = [\mathbf{T}_i]^{\mathrm{T}} \{\mathbf{U}_i\} \tag{5-56}$$

where $\{\mathbf{U}_i\}$ is a vector containing the global displacements found at both nodes of the module. The local force-displacement relation can be determined using:

$$\{\mathbf{f}_{e,i}\} = [\mathbf{k}_{e,i}]\{\mathbf{U}_{e,i}\} \tag{5.57}$$

where $\{\mathbf{k}_{e,i}\}$ is the local stiffness matrix for a single module, obtained from equation (4.4). Substituting equation (5.55) and (5.56) into (5.57) gives:

$$[\mathbf{T}_i]^{\mathrm{T}} \{\mathbf{f}_i\} = [\mathbf{k}_{e,i}][\mathbf{T}_i]^{\mathrm{T}} \{\mathbf{U}_i\} \tag{5.58}$$

Since the transformation matrix in equation (5.54) is orthogonal, equation (5.58) can be written as:

$$\{\mathbf{f}_i\} = [\mathbf{T}_i][\mathbf{k}_{e,i}][\mathbf{T}_i]^{\mathrm{T}} \{\mathbf{U}_i\} \tag{5.59}$$

Consequently, the transformed stiffness matrix using space-fixed rotations for a single module is given by:

$$[\mathbf{k}_i] = [\mathbf{T}_i][\mathbf{k}_{e,i}][\mathbf{T}_i]^{\mathrm{T}} \tag{5.60}$$

Like the FEM hybrid model in Chapter 4, each stiffness matrix can be assembled using the direct stiffness method [44] to obtain the system global stiffness equation, $[\mathbf{K}]_G$. Afterwards, the system of equations in global coordinates can be solved using:

$$\{\mathbf{U}\}_G = [\mathbf{K}]_G^{-1} \{\mathbf{F}\}_G \tag{5.61}$$

where $\{\mathbf{U}\}_G$ and $\{\mathbf{F}\}_G$ are of the size $(6+6n)\times 1$, and $[\mathbf{K}]_G$ is $(6+6n)\times(6+6n)$. Equation (5.61) is subject to the boundary conditions in equation (4.15). Computationally, the

boundary conditions can be applied by replacing the first six diagonal terms in the global stiffness equation with a large number such as 10^{20}, and replacing the corresponding rows and columns with zeros. Also, $\{\mathbf{U}\}_G$ gives the global displacements at each node along the assembly, unlike the Compliance-Jacobian method where the global displacements must be calculated afterwards. If the self-weights of each module along with the motor weights are to be included, equation (5.61) becomes:

$$\{\mathbf{U}\}_G = [\mathbf{K}]_G^{-1}\left(\{\mathbf{F}\}_{tip} + \{\mathbf{f}\}_{weight} + \{\mathbf{f}\}_{motor}\right) \tag{5.62}$$

where $\{\mathbf{f}\}_{weight}$ and $\{\mathbf{f}\}_{motor}$ are vectors containing the global forces due to self-weight and joint motor weights. The forces due to self-weight for each module are lumped at the two end nodes as shown in Fig. 4-4 in Chapter 4. Also, the weights due to the joint motors can be placed at the first node of each joint module instead of placing them at the tips of the previous modules like the Compliance-Jacobian method. Furthermore, the global tip displacements can be found in the last six rows of $\{\mathbf{U}\}_G$ in equations (5.61) or (5.62).

5.2.2 Post-Processing

Once the global displacements are determined, the local displacement vector for each module can be determined using equation (5.56). Afterwards, the local forces can be obtained using equation (5.57). The nodal stresses given in equations (5.40) to (5.51) can then be applied to the first node of each element in this model, and the Von Mises equivalent stress along with the factor of safety for each module can be calculated using equations (5.52) and (5.53).

5.3 Simulation and Results

To compare the two kinetostatic methods described in the previous sections in this chapter, a kinetostatic analysis was performed on a MRR at a single position and configuration after moving from its initial configuration setup. The goal of this analysis was to determine the tip deflection at the end module in order to test the validity of the two models. The robot tested consisted of three revolute joint modules and three links and was configured according to the quasi-global method presented in Chapter 3. Two of revolute joint and link modules were equal sized, and the revolute joint and link located

near the base of the robot were larger than the other two joint and link modules. Also, the base link, which was considered as a rigid body, had a height of 0.9 m.

The sizes of each module are shown in table 5-1. Using the notation provided in Chapter 3.3, the robot can be specified as "RLRLRL" and its configuration can be classified as "y, x, z, x, -y, x". Also, there were no angular offsets defined for each of the modules at their initial configuration setup. Furthermore, all modules were made of aluminum, with the material properties listed in table 4-1 in Chapter 4. In this simulation, each revolute joint casing, instead of the motor shaft, was modeled as a hollow cylinder and each link was modeled as a straight, hollow beam with a uniform square cross section.

Table 5-1: Module sizes and joint input angles for kinetostatic test case

Module	Configuration	Length, L_i [m]	Inner cross section base length, b_i [m]	Inner radius, r_i [m]	Wall thickness going outwards, t_i [m]	Motor mass [kg]	Joint input angle [deg]
Revolute Joint 1	y	0.3	-	0.15	0.005	3	40
Link 1	x	0.5	0.13	-	0.004	-	-
Revolute Joint 2	z	0.23	-	0.1	0.004	1.5	30
Link 2	x	0.4	0.09	-	0.003	-	-
Revolute Joint 3	-y	0.23	-	0.1	0.004	1.5	-20
Link 3	x	0.4	0.09	-	0.003	-	-

After the initial configuration setup, the first, second and third joint modules were moved with local input angles of 30°, 40° and -20°, respectively, and the link module furthest from the base was loaded with a force of *-100 N* in the global y-direction to simulate the weight of a payload at the end of the robot. The flexible tip position vector includes the effects of the displacements and can be defined as:

$$\mathbf{P}_{i+1,flex} = \mathbf{P}_{m,i+1} + \begin{bmatrix} u_{tip,n} & v_{tip,n} & w_{tip,n} \end{bmatrix}^{\mathrm{T}} \tag{5.63}$$

Where $\mathbf{P}_{m,i+1}$ is the rigid body position vector after motion obtained from the position analysis in Chapter 3, $u_{tip,n}$, $v_{tip,n}$,and $w_{tip,n}$ are the linear deflections in the x, y and z directions, respectively. The kinetostatic analysis was conducted to determine and compare the tip deflections and flexible tip position vectors for both methods, with and without the inclusion of each module's self-weight. Fig. 5-3 shows the schematic beam representation of the MRR in its initial and final position after motion.

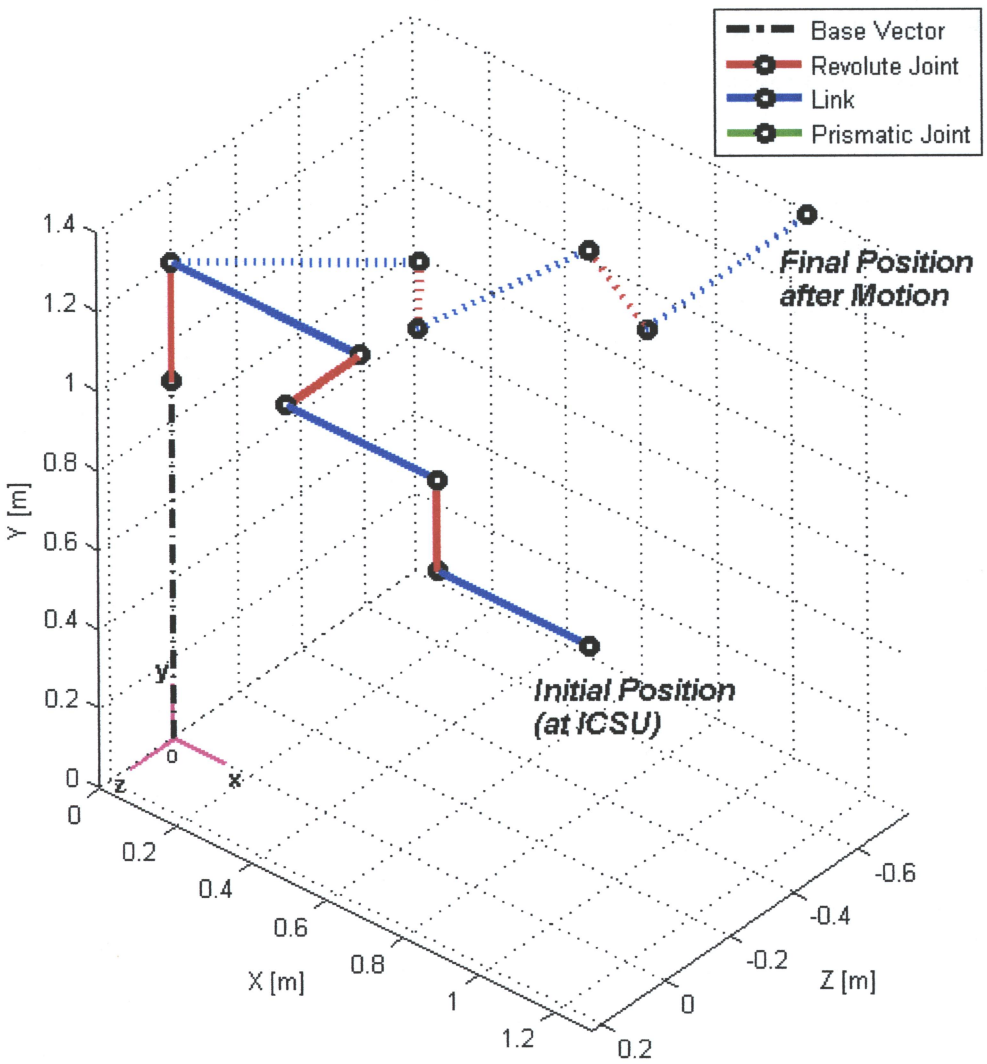


Figure 5-3: Schematic beam diagram of MRR tested at final position after motion

Table 5-2 shows the results of the kinetostatic analysis using the Compliance-Jacobian method as well as FEM for with each module's self-weight included.

Table 5-2: Results of the kinetostatic analysis with module self-weight included

Method	Flexible tip position vector, $P_{i+1,flex}$			Tip deflections at end of n^{th} module, $U_{tip,G}$					
	x [m]	y [m]	z [m]	u 10^{-3} [m]	v 10^{-3} [m]	w 10^{-3} [m]	θ 10^{-3} [rad]	φ 10^{-3} [rad]	ψ 10^{-3} [rad]
Compliance	1.04581	1.38833	-0.75588	0.06460	-0.41817	-0.04318	-0.39769	-0.01612	-0.37413
FEM	1.04581	1.38833	-0.75588	0.06460	-0.41817	-0.04318	-0.39769	-0.01612	-0.37413

From table 5-2, both the Compliance-Jacobian and FEM kinetostatic methods give the same results. Upon further inspection in MATLAB, the results differ by about 10^{-15}, which is close to the computer's epsilon accuracy. Hence, both methods give essentially the same results. In the test case where the motor and module weights were included, the highest deflection occurs in the y-direction, which is -0.41817 mm. This is mostly due to the large payload applied at the tip of the MRR and the weights of the modules and motors which are all applied in the same direction. Large deflections are a concern for robots with high accuracy requirements, such as those that perform precision manufacturing operations, because this deflection is too large.

Both kinetostatic methods presented in this chapter have their advantages and disadvantages. To begin, the assembly of the compliance matrix for the total number of modules in the Compliance-Jacobian method is much easier than the assembly process of the stiffness matrix in finite element method. This is because each module's compliance matrix has terms pertaining to the tip of the module and is not based on two nodes like the finite element method, and there is no overlapping of compliance terms within the assembled compliance matrix. However, since the Compliance-Jacobian method is based on first determining the tip deflections at the n^{th} module in global coordinates, the other module global deflections must be determined during the post-processing stage; whereas the FEM model determines all global deflections simultaneously.

In both methods, the local forces can be determined simultaneously once the global forces are known. However, the local displacements must be solved individually

for each module and involves using the Jacobian or transformation matrices for each individual module. Furthermore, it is more difficult to include the weights of each component in the Compliance-Jacobian method because they must first be transformed to local coordinates, while on the contrary, the finite element method can directly use the component weights in global coordinates to solve the system equations. In addition, if the lengths of the revolute joints are negligible, the compliance terms can accept zero-length modules, but the stiffness terms in the FEM model cannot accept zero-length modules since it will lead to division by zero. Hence, a small number such as 10^{-20} must be entered, which will lead to almost zero deflections for the zero-length member. Moreover, the Jacobian used in the compliance method can also be used to assess the kinetostatic performance of a MRR, which is discussed in the next chapter.

6. Applications of Kinetostatic Analysis

In the previous chapter, two kinetostatic models were introduced. In this chapter, the kinetostatic models were applied to determine the maximum payload that a given MRR can withstand. Before the maximum payload can be determined, the MRR position where the maximum deflection exists must be determined. This can be done in two ways. The first method is to conduct a forward workspace search for given ranges of joint angles and store the maximum deflection and the joint angles where the maximum deflection occurs. This method can be rather tedious, especially for small increments of joint angles and a large number of MRR configurations. However, using this brute force method, a forward workspace visualization can be obtained, along with the volume (or area for planar motions) of the workspace boundary. Also, the number of configurations can be drastically reduced using an enumeration process, where only the feasible configurations are considered. The other method involves determining parameters related to the Jacobian matrix in the Compliance-Jacobian method presented in Chapter 5.

Since the deformation Jacobian can be used to transform the local forces and infinitesimal displacements to global coordinates and vice-versa, it must be capable of assessing when the forces and infinitesimal displacements are at a maximum for given module configurations and orientations. Once the position and orientation of an MRR that gives the maximum deflection is determined, the maximum payload that the robot can carry for a given tip deflection can be found using the bisection method. Furthermore, it is shown that the maximum deflection position of an MRR does not change for different values of the payload, as long as the payload forces are acting in the same direction.

6.1 Maximum Deflection Position and Orientation

The determination of the position and orientation that gives the maximum deflection of an MRR with a payload applied at the tip of the robot is necessary in order to determine the maximum possible payload that robot can carry. The following sub-

sections describe two methods to determine the maximum deflection position and orientation.

6.1.1 Workspace Search Method

This method involves searching the entire workspace in order to find the maximum deflection position for a given MRR configuration. This is done using forward kinematics for a given range of joint input angles for each joint, while the other method involves enclosing the allowable movement of the robot's tip position within a given volume and determining the range of motion within that volume using inverse kinematics. The first step in searching the workspace using forward kinematics is to enter the range of motion and the number of angular increments for each joint. Smaller increments lead to higher computational times. Next, the position and orientation equations (presented in Chapter 3) and the tip deflections (from Chapter 4) must be determined for each possible set of joint input angles. Since the norm of a vector can be used to determine its length, for each iteration the highest norm of the linear deflections of the tip of the robot can be stored. Thus, the maximum linear deflection is represented by:

$$\mathbf{U}_{tip,\max} = \left\| \begin{pmatrix} u & v & w \end{pmatrix} \right\|_{tip,\max} \tag{6.1}$$

It is important to note that only the linear deflections are considered in the vector norm, and not the angular deflections because of unit inconsistency. The joint input angles, position, orientation, and deformation Jacobian are also stored once the maximum deflection is determined. If the current iteration has a higher tip displacement than the previous maximum, then the current maximum displacement and its pose are stored. The main drawback of this method is that it takes considerable computation times, especially for small joint angular increments. However, if the flexible tip positions obtained from the kinetostatic analyses are also stored, a workspace visualization can be created, and the volume (or area) of the forward workspace can be determined.

There exist several methods of representing the workspace volume. Two methods are the dextrous workspace and the reachable workspace [48]. The dextrous workspace

point in any orientation. The reachable workspace consists of the Cartesian points that are reachable by at least one orientation of the robot's end-effector. Agrawal et al. determine the dextrous workspace of a MRR by partitioning the robot chain and finding the workspaces for each part of the chain. Then, there is a search within the known local dextrous workspaces to determine if there are any common points, which means that the robot can reach that common point while satisfying all joint limits. Botturi and Fiorini [49] described a method of creating the workspace by successively sweeping the curves made by successive joint angular motions, which represent the individual joint workspaces. Each of the smaller workspaces were then combined and the boundary of the workspace can be determined using software such Mathematica and MATLAB.

In this thesis, the *workspace outer boundary* is determined for an open-chain MRR. This workspace is the outer boundary in which the robot's end-effector is constrained to move within. Consequently, the robot's end-effector cannot move outside of the workspace outer boundary. For spatial robot motion, the workspace outer boundary is represented by a volume, and for planar motion it is an area. This is essentially important for determining the size of the room needed to allow the robot to undergo its full range of motion. Since all of the end-effector points can be stored while performing the search for the maximum deflection for a given MRR configuration, the outer workspace boundary can be represented as the boundary containing the outermost points. This can be represented by determining the convex hull for the set of end-effector points gathered during the workspace search.

A convex hull of a set of points $\boldsymbol{S}$ is the smallest convex set that contains those points [50]. In other words, the convex hull of set $\boldsymbol{S}$ is the smallest polygon (for planar points) or polyhedron (for spatial points) that contains $\boldsymbol{S}$. The convex hull can be imagined as a rubber band that surrounds $\boldsymbol{S}$ that contains planar points, or a balloon in tension that encloses a solid object in 3-D space. Also, the convex hull is the smallest area (or volume) that encloses $\boldsymbol{S}$. The difference between a convex hull and a polygon enclosing a set of points is shown below. Barber et al. [50] created the Quickhull algorithm to determine the convex hull for a set of 2-D, 3-D and 4-D points. This

algorithm searches for the extreme points and creates facets, or ridges of the convex hull. If a point is outside of the facet boundaries, it is an extreme point; otherwise it is not part of the convex hull. The code has been implemented in MATLAB and is used to determine the set of points which represents the convex hull, which is taken as the outer workspace boundary in this thesis. The MATLAB function *convhull* finds the convex hull points for a set of 2-D points and calculates its area, and *convhulln* determines the convex hull for a set of 3-D points along with its volume. These two functions were used to compute the convex hull of the gathered flexible tip points, hence determining the workspace outer boundary areas or volumes.

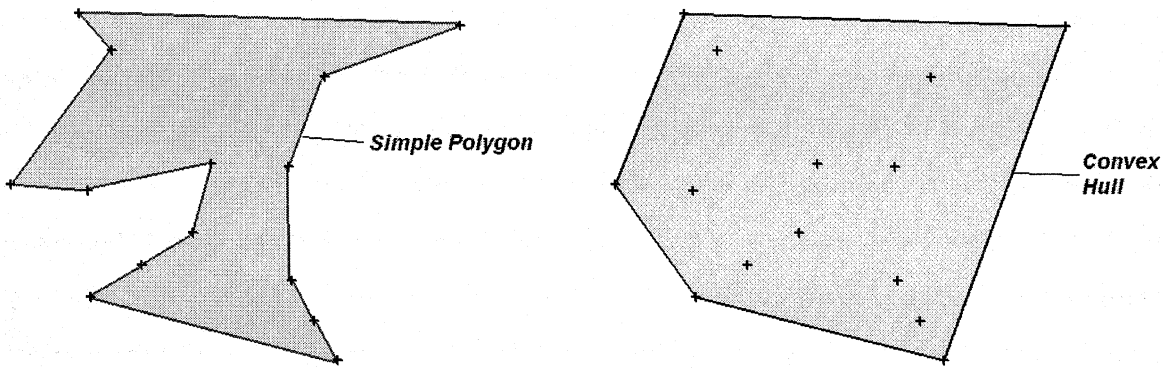


Figure 6-1: Simple polygon and convex hull surrounding a set of points

6.1.2 Jacobian Index Method

The second method to determine the maximum deflection position and orientation involves using the deformation Jacobian matrix in of Chapter 5 in order to assess when the global tip displacements are at the highest for a given range of joint input angles. It is well known that higher forces give larger displacements; thus, when the robot is fully extended to a point that is furthest from the base of the robot, the largest deflection due to the payload can occur, as opposed to any other points, since the moment arm that is created is the longest.

Previously researchers created performance indices using the velocity Jacobian to assess the kinematic performance of a robot. Yoshikawa [51] determined the

manipulability index for a redundant manipulator by taking the product of the singular values of the velocity Jacobian which is the same as taking the square root of the product of the Jacobian and transpose of the Jacobian. A redundant manipulator has a higher number of DOFs required to perform a certain range of motion, so the extra joints are redundant. For non-redundant manipulators, the manipulability index is simply the determinant of the velocity Jacobian. Afterwards, Yoshikawa [51] showed that his manipulability index was proportional to the manipulability ellipsoid volume which represents the distance between the manipulator configuration and a kinematic singularity. When the manipulability index approaches zero, the manipulator approaches a singular position, so the manipulability is a measure of the robot's ability to move freely in any direction. Another performance index using the velocity Jacobian is the dexterity [52]. This is the reciprocal of the velocity Jacobian condition number, or the maximum to minimum singular value ratio. The dexterity gives an indication of the shape of the manipulability ellipsoid because the maximum and minimum singular values of the Jacobian correspond the major and minor axis of the ellipsoid. When the condition number is equal to one, the ellipsoid shape becomes a circle and the end-effector of the robot can move with the same performance in any direction. On the other hand, a larger condition number means that the robot is closer to a singular configuration.

Although the two previously mentioned velocity performance measures use different Jacobians than deformation Jacobian described in Chapter 5, they can still be used to determine when the deformation Jacobian allows the robot to be at a fully stretched position in a direction perpendicular to the payload force, where the maximum displacement is most likely to occur. Referring to the deformation Jacobian in equations (5.24) and (5.25), the cross product terms in the upper right hand part of the Jacobian are used to transform the moments and forces from the tip of the robot to the tip of the i^{th} module. When the distance from the tip to the i^{th} module becomes larger, the forces become higher, consequently increasing the tip displacement. The phenomenon is similar to a single cantilever beam in bending with a load applied at the tip in the direction of gravity. If the beam is tested at its full length along with another similar beam, but with half the original length, under the same applied load, the beam with the longer length will

have the higher deflection because it has the largest moment arm. Since the dexterity can measure the ability of a robot to arbitrarily change its position and orientation, it can be used as a means to measure a robot's ability to apply forces in arbitrary directions. Define the *kinetostatic manipulability index* as:

$$w_k = \sqrt{\det\left(\mathbf{J}_{d,n}\mathbf{J}_{d,n}^{\mathrm{T}}\right)} \tag{6.2}$$

Alternatively, equation (6.2) can be re-written as the product of the singular values of the deformation Jacobian for n modules:

$$w_k = \lambda_1\lambda_2 \ldots\ldots \lambda_n \tag{6.3}$$

In a similar scenario with the kinematic manipulability index, the kinetostatic manipulability index gives the shape of an ellipsoid, but it has no indication of its size [52]. However, the dexterity can give an idea of shape of the ellipsoid. Thus, using a form of the dexterity, the *kinetostatic dexterity* can be defined as:

$$\kappa_k = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{6.4}$$

where λ_{max} is the maximum and λ_{min} are the maximum and minimum singular values of the deformation Jacobian matrices for modules i to n. When the reciprocal of the kinetostatic dexterity is larger, the manipulator approaches a singular configuration. Therefore, to find the maximum deflection position, equation (6.4) can be minimized for a given range of joint input angles. The minimum value of the kinetostatic dexterity can occur at the maximum deflection position, because the robot will be stretched out furthest from its base, closer to a singular position. However, if a robot has a large number of joints, there may be more than one singular position, hence the solution that gives the longest reach perpendicular to the direction of the payload force should be used.

6.2 Maximum Allowable Payload

The maximum allowable payload is the largest payload that the MRR can hold at its end-effector. Once the maximum deflection position is determined, the maximum allowable payload can be determined at this position and orientation, since this position

will not change for different magnitudes of loads applied in the same direction. For a given deflection, the maximum allowable payload can be determined using a number of optimization methods. In this thesis, a modified bisection method is used. Usually, the bisection method is used to determine the roots of a function at x = 0. Here, the function is given by equation (5.33) to solve for the maximum allowable force in the negative y-direction at a constrained tip deflection in the same direction. Thus, the "zero" line in the original bisection method is shifted to the constrained tip deflection. After choosing the maximum allowable displacement, v_{max}, the next step in the bisection method involves choosing an initial guesses for the upper and lower limit to the maximum allowable payload, F_u and F_l. In this version of the bisection method, it is quite safe to assume a large positive number and a large negative number for the upper and lower initial force bounds, respectively. Afterwards, the upper and lower limits are bisected using:

$$F_{y,final} = \frac{F_u + F_l}{2} \tag{6.5}$$

Then, equation (6.5) is substituted into equation (5.33) to determine the tip deflections. The calculated tip deflection, $v(F_{y,final})$, is then used to check for the following bisection conditions:

$$\begin{aligned}
&\text{if } v\left(F_{y,final}\right) - v_{\max} > 0 \\
&\qquad \text{then set } F_u = F_{y,final} \\
&\text{else if } v\left(F_{y,final}\right) - v_{\max} < 0 \\
&\qquad \text{then set } F_l = F_{y,final} \\
&\text{else } v\left(F_{y,final}\right) - v_{\max} = 0 \\
&\qquad \text{and } F_{y,final} \text{ is the maximum allowable force.}
\end{aligned} \tag{6.6}$$

The algorithm in equation (6.6) is repeated until a specified absolute maximum error (ε) is obtained. The absolute maximum tolerance can be written as:

$$\varepsilon_{abs} = \frac{F_u - F_L}{F_u + F_L} \times 100\% \tag{6.7}$$

6.3 Module Enumeration

For an increasing number of modules, the number of configurations can increase exponentially. For the example, an MRR with three links and three joints each with six possible configurations, the total number of configurations is $6^6 = 46656$. Although this is a large number of total configurations, not all of them are feasible. The purpose of the enumeration is to eliminate infeasible configurations with regards to certain criteria in order to reduce the total number of configurations required to analyze. Chen and Burdick [53] created an enumeration method which was based on the geometric symmetry properties of a set of modules and a graphical approach to represent the robot's topology in order to eliminate configurations that represent symmetric kinematics.

In this thesis, the enumeration was performed by using the ZRP and quasi-global connectivity methods presented in Chapter 3 to represent the configurations of the joints and links. A number of steps were created to enumerate the revolute joint modules, but not the link modules. This is because each link module is constrained to be in a configuration with a positive direction perpendicular to the previous joint in order to reduce the chances of module collision. For example, if a revolute joint is configured in the "x", "y", "z", "-x", "-y" or "-z" direction, the next link will either be in the "z" or "x" directions only. The following MRR assembly constraints are the basis for the enumeration. First, the total possible number of joint configurations can be determined and placed in a table, and then the infeasible configurations can be eliminated with the use of the following rules:

1. *No two configurations can have a symmetric topology about the same plane.*

This rule implies that no two configurations can have symmetric shapes about the same plane because two symmetric configurations not only give the same kinematic results during motion, but they also produce the same maximum deflection position. For example, consider an "RLRL" type MRR in the "z, x, z, x" and "-z, x, -z, x" configurations. Both cases give the same position and orientation about the plane of symmetry after movement and the same magnitude of the tip deflection is also obtained. The same results are obtained for "y, x, -y, x" and "-y, x, y, x" configurations. Although

the directions of some of the displacement terms can be different between the two configurations, their magnitudes remain equal.

2. *The first joint can have only positive-axis configurations.*

This rule is because once the first joint has negative directional configurations, there are high possibilities that symmetric configurations about the base link can be formed. Also, referring to the system described in Chapter 2, the first joint cannot have any "-y" configurations because it will clash with the base link. This brings forth the constraint to prevent module clashing for all other modules.

3. *Each module cannot be configured in the negative direction of the previous module.*

This enumeration constraint relates most to the connectivity of both link and joint modules. Since the link modules are assumed to be configured in only the "x" or "z" configurations, special attention must be made to the configurations of the joints and link modules to ensure that the following modules do not clash or overlap with the previous modules. If a joint is in the "y" configuration, the next joint cannot be in the "-x" direction, since the preceding link after the first joint is automatically configured in the "x" direction. The following table shows the possible configurations a joint and a preceding link.

Table 6-1: Joint and successive link configuration directions

Joint Direction	Next Link Direction
x	z
y	x
z	x
-x	z
-y	x
-z	x

4. *Each module must be perpendicular to each other.*

As shown for the MRR in Chapter 2, each link is configured with the axis of its length perpendicular to the preceding revolute joint's axis of rotation. Therefore, no two

successive modules can be configured in the same direction. This can produce axial joint-link modules and for the system used in this thesis, there are no available connection interfaces to allow these types of configurations.

Using these four rules, a table with the total number of configurations can be drastically reduced to contain only feasible configurations. Additionally, configurations that allow redundant planar motion can also be removed. However it can be shown that some planar configurations can give higher tip displacements than some spatial configurations. It is important to note that when these enumeration rules are carried out in the above order, the smallest set of feasible configurations are produced. If the rules are carried out in the reverse order, they produce a larger set of feasible configurations, which can be more time consuming to perform analyses, but is a more conservative path to take when the known number of feasible configurations are higher.

6.4 Simulation and Results

A simulation was conducted to determine the maximum deflection position and maximum allowable payload for the example robot given in Chapter 5. The workspace search method was used in order to provide a visualization of the workspace outer boundary and determine its volume using the *convhulln* function in MATLAB and also check the kinetostatic dexterity using equation (6.4). Unlike the analysis in Chapter 5, the module self-weights and motor weights were ignored because equation (6.4) does not include the possibilities of loads applied elsewhere from the tip. With three revolute joints and three link modules, there are $6^3 = 216$ possible joint configurations. Using the enumeration rules from the previous section, the smallest number of feasible joint configurations is 18. These unique feasible configurations are listed in Table 6-2. All configurations undergo spatial motion except configurations 3, 7, 14, and 18.

For each configuration, the MRR initially moved from the initial configuration setup position and a payload force of $P = 100\ N$ was applied in the negative y-direction. The robot was kinematically positioned using the quasi-global connectivity method presented in Chapter 3. Each joint had a range of motion from -90° to +90° in increments

of 10°, so there were 6859 total positions tested for each of the configurations. For each position tested, the tip deflection vector was calculated using equation (5.32) and the kinetostatic dexterity was calculated and stored also.

Table 6-2: Enumeration table for an MRR with three revolute joints and link modules

Config. #	Joint 1 Configuration	Link 1 Configuration	Joint 2 Configuration	Link 2 Configuration	Joint 3 Configuration	Link 3 Configuration
1	x	z	x	z	y	x
2	x	z	y	x	y	x
3	y	x	y	x	y	x
4	z	x	y	x	y	x
5	x	z	-x	z	y	x
6	x	z	-y	x	y	x
7	y	x	-y	x	y	x
8	z	x	-y	x	y	x
9	y	x	-z	x	y	x
10	x	z	y	x	z	x
11	y	x	y	x	z	x
12	z	x	y	x	z	x
13	y	x	z	x	z	x
14	z	x	z	x	z	x
15	y	x	-y	x	z	x
16	z	x	-y	x	z	x
17	y	x	-z	x	z	x
18	z	x	-z	x	z	x

Once the tip deflections were determined, they were added to the rigid-body tip position vector obtained from the kinematics (as in equation (5.63)) and stored to later calculate the workspace. Also, for each iteration, the calculated tip displacement vector was compared to the maximum tip displacement vector. If the calculated vector was higher than the maximum, it was stored as the maximum displacement vector, along with the corresponding Jacobian and recursive rotation matrices, rigid-body tip position and local body vectors, along with the joint input angles for the maximum deflection position. Afterwards, when the maximum deflection position was determined for each

configuration, the maximum allowable payload for a maximum deflection constraint of *0.0001 m* was determined, with an error tolerance of 10^{-10}. Table 6-3 shows the maximum deflection position results and Table 6-4 shows the workspace, minimum dexterity and maximum allowable force for a tip deflection in the y-direction of -0.0001 m.

Table 6-3: Joint angles, maximum tip deflection and maximum allowable payload results

Config .#	Local Joint Input Angles for Maximum Deflection Position			Global Maximum Tip Deflections			Maximum Allowable Payload (for $v = -0.1$ mm) [N]
	Joint 1 [deg]	Joint 2 [deg]	Joint 3 [deg]	u 10^{-3} [m]	v 10^{-3} [m]	w 10^{-3} [m]	
1	0	20	-50	0.01304	-0.37392	-0.00560	-26.74306
2	30	-20	-20	0.02278	-0.36317	0.00248	-27.53525
3	*-90*	*0*	*0*	*-0.06249*	*-0.31468*	*0.11430*	*-31.77777*
4	-30	-70	20	0.00248	-0.36317	0.02278	-27.53525
5	0	-20	-90	0.002742	-0.35828	-0.01441	-27.91109
6	20	20	-20	-0.00994	-0.32903	-0.00710	-30.39158
7	*-90*	*0*	*0*	*0.04182*	*-0.31468*	*0.12216*	*-31.77777*
8	-20	70	20	-0.00710	-0.32903	-0.00994	-30.39158
9	0	20	40	-0.00467	-0.36337	-0.01334	-27.52000
10	10	-20	10	0.02972	-0.37384	0.01251	-26.74923
11	-90	20	20	-0.00283	-0.32399	0.08978	-30.86497
12	-10	70	10	0.01251	-0.37228	-0.02958	-26.86147
13	60	0	0	0.00061	-0.39214	-0.00052	-25.50058
14	***0***	***0***	***0***	***0***	***-0.40361***	***0***	***-24.77600***
15	-40	-20	-20	-0.06574	-0.32399	-0.05894	-30.86497
16	10	-70	-10	-0.01251	-0.37228	0.02958	-26.86147
17	-70	0	0	0.00026	-0.34772	0.00071	-28.75874
18	0	0	0	0	-0.35865	0	-27.88230

From Table 6-3 above, the maximum tip deflection occurs for configuration #14 which also has the lowest allowable payload for the given deflection constraint. Also, this configuration has zero deflections in the x- or z-directions because the maximum deflection position is perpendicular to the payload force. The maximum deflection position for configuration #14 occurs at its initial setup position and is shown in Fig. 6-2. On the other hand, configurations #3 and #7 have the lowest maximum deflections in the negative y-direction and they can take the highest allowable payloads, as expected. These two configurations undergo planar motion and configuration #3 is shown in Fig. 6-3. Although these configurations have the highest y-deflections, they do not have the highest resultant deflection. Instead, configurations #6 and #8 have the highest resultant deflections and configuration #8 is shown in Fig 6-4. Furthermore, it is worth noting that

two configurations undergo planar motion and configuration #3 is shown in Fig. 6-3. Although these configurations have the highest y-deflections, they do not have the highest resultant deflection. Instead, configurations #6 and #8 have the highest resultant deflections and configuration #8 is shown in Fig 6-4. Furthermore, it is worth noting that for configuration #13 (shown in Fig. 6-5), the same displacement values occur when joint 1 is rotated from -90° to 90° and the other two joints remain at 0°. The same kinetostatic dexterity values were also present at those positions. The result is most likely due to the machine precision used in MATLAB.

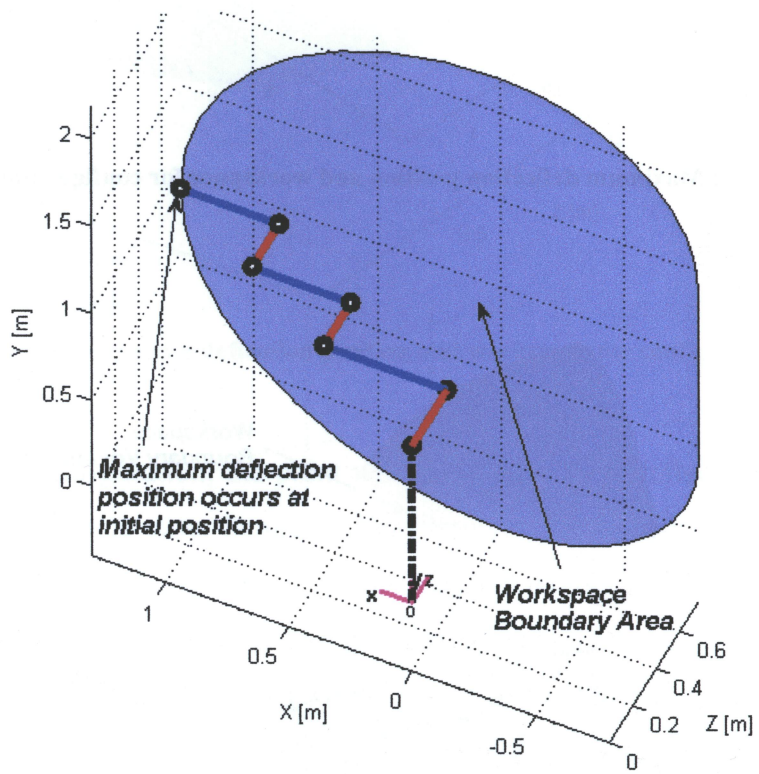


Figure 6-2: Maximum deflection position and workspace for configuration #14

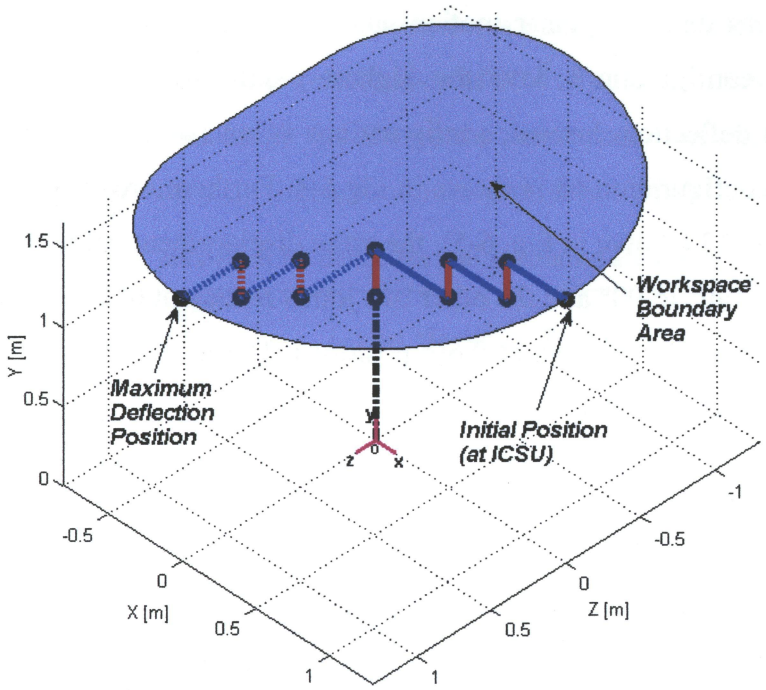


Figure 6-3: Maximum deflection position and workspace for configuration #3

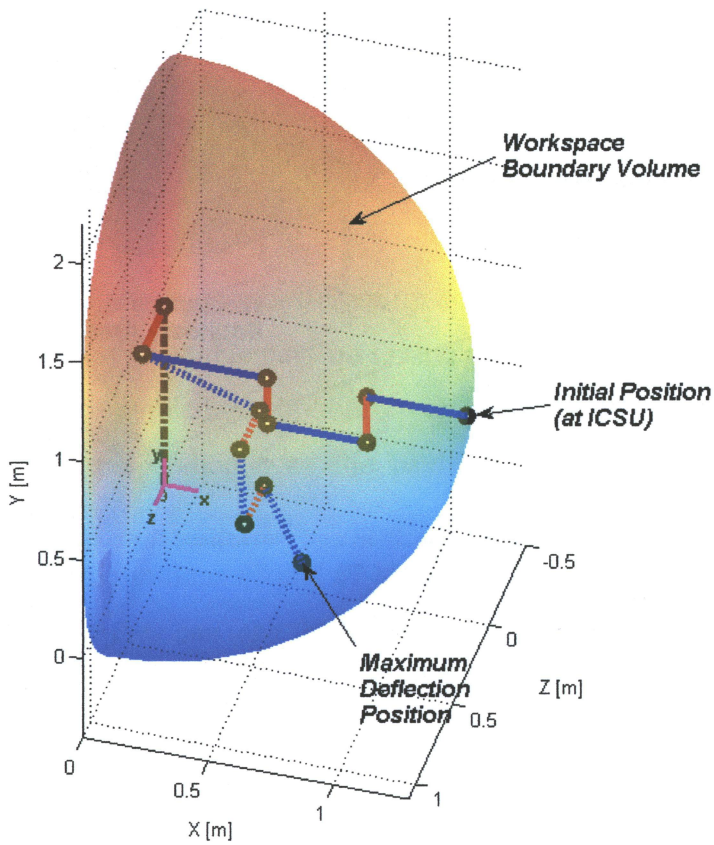


Figure 6-4: Maximum deflection position and workspace for configuration #8

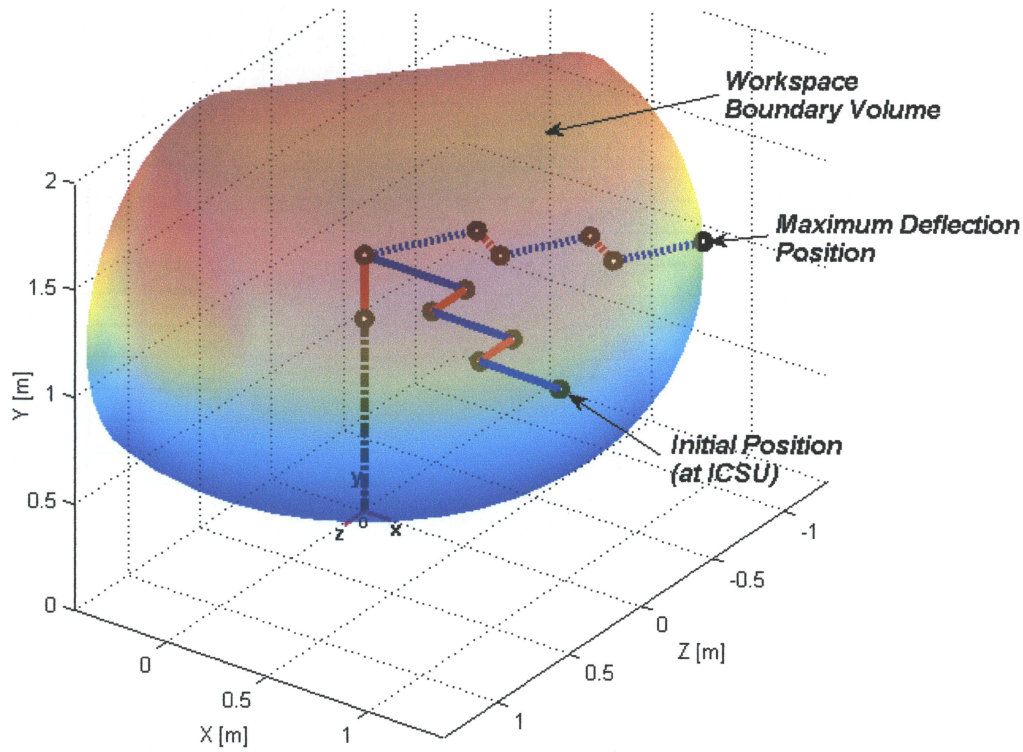


Figure 6-5: Maximum deflection position and workspace for configuration #13

For each configuration, the point where the maximum deflection position occurs intersects the workspace outer boundary. Interestingly, the tip position of configuration #14 had the furthest reach away from the base vector, so this was the configuration with the highest moment about the tip of the base vector, thus producing the maximum deflection. Table 6-4 shows the coinciding workspace and kinetostatic dexterity results. The maximum workspace volume occurs in configurations #9, #12 and #16. In each of their initial setup configurations, the middle joint is perpendicular to the other two joints. The smallest workspace volume occurs for configurations #1 and #5 because the first joints have collinear axes of rotation, and the third joint has an axis of rotation that is perpendicular to the first two joints during the initial configuration setup. Configuration #5 is shown in Fig. 6-6.

Table 6-4: Workspace volume (or area) and kinetostatic dexterity results

Configuration #	Workspace Outer Boundary [m^3] for spatial motion, [m^2] for planar motion	Kinetostatic Dexterity, κ
1	*1.62927 m^3*	0.54553
2	3.72052 m^3	0.55165
3	4.44140 m^2	0.55439
4	4.51160 m^3	0.55165
5	*1.62927 m^3*	0.55044
6	2.86361 m^3	***0.57054***
7	4.44140 m^2	0.55439
8	3.24709 m^3	***0.57054***
9	***5.13303 m^3***	0.54553
10	3.54744 m^3	0.55165
11	3.25854 m^3	0.56010
12	***5.13303 m^3***	0.55204
13	4.51160 m^3	*0.53413*
14	4.44140 m^2	*0.53413*
15	3.25854 m^3	0.56010
16	***5.13303 m^3***	0.55204
17	3.24709 m^3	0.55044
18	4.44140 m^2	0.55044

As expected, configurations #13 and #14 have the lowest kinetostatic dexterity which coincide with the highest deflections and lowest allowable payload results from Table 6-3. Configurations #6 and #8 have the highest minimum kinetostatic dexterity parameters, but they do not have the lowest deflection in the negative y-direction. However, their resultant deflections, obtained from equation (6.1), are the lowest of all configurations. Also, configurations #13 and #14 have the two lowest kinetostatic dexterities, and the highest tip deflections, proving that the kinetostatic dexterity can be used to determine the positions and configurations with the highest deflections. Furthermore, The maximum allowable and maximum deflection position can be tested by applying the maximum allowable payload for a y-deflection of -0.0001 mm and performing the workspace search once again. When this was conducted, all maximum deflections were -0.0001 m and the maximum deflection positions remained the same.

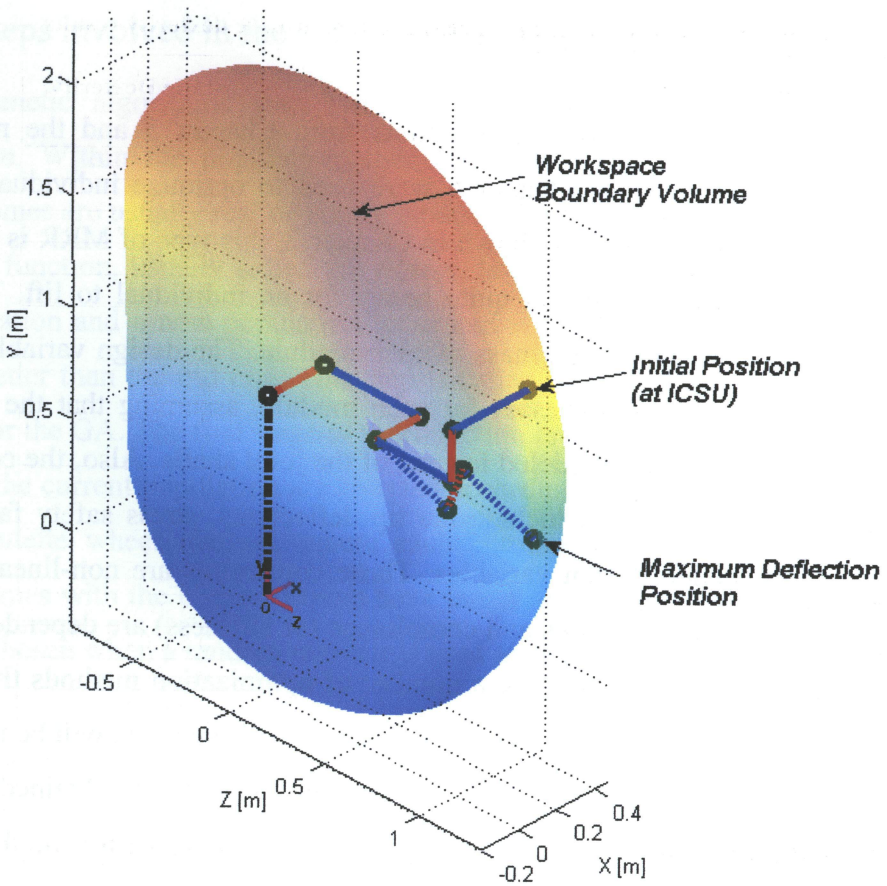


Figure 6-6: Maximum deflection position and workspace for configuration #5

7. Module-Based Optimization

In this chapter, the kinetostatic equations from Chapter 5 and the maximum deflection position methods from Chapter 6 were used to optimize individual module sizes, namely their cross sections. As shown in Chapter 2, this type of MRR is manually reconfigured and the modules can get quite heavy for an individual to lift. Thus, the objective function is to minimize the mass of each module. The design variables in this chapter are the cross section thicknesses for each module, assuming that the links are hollow and the joint casings are modeled instead of the joint shafts. Also, the constraints acting on this design optimization include the tip deflections, stress safety factors and bounds on the cross section design variables. These constraints are non-linear and are implicitly solved since the deflections and compliance (or stiffness) are dependent on the cross section design variables. There are a number of optimization methods that can be used to solve this problem, but in this chapter a genetic algorithm (GA) will be used since it can be applied to a majority of optimization problems. The results obtained from the GA were compared to a simple, yet effective, direct method to solve for the thicknesses of each module for known deflection constraints. This method involved solving the compliance terms directly and approximating the local deflections using a pseudoinverse form for the inverse of the deformation Jacobian matrix.

7.1 Optimization using a Genetic Algorithm

There are several advantages to using a genetic algorithm for a given optimization problem. First, GAs only use function values in the search process to progress toward a solution with disregards as to how the functions are evaluated [54]. Also, it does not matter whether or not the functions are differentiable or continuous because the GA does not consider this. Also, GAs can be applied to a variety of problems with any kind of objective functions and constraints, and they determine global optima instead of local optima. However, the main drawbacks are possible large computational times and there is no absolute guarantee that the global optimal solution has been obtained [55].

7.1.1 Steps involved in the Genetic Algorithm

Genetic algorithms start with an initial set of solutions, called the *initial population*. Within the population, there is a set of individual *chromosomes*. These chromosomes are usually real or binary bit-strings, expressed as a sequence of *genes*. The objective function, usually called the *fitness*, is then evaluated for each chromosome in the population and a new population is created with the hopes that this new population will be better than the old one. This step is called *reproduction* and begins the iterative process for the GA. The first stage of reproduction is called *selection*, where designs from the from the current population are selected according to the selection process used, such as the roulette wheel method, where each chromosome is assigned an area and the chromosomes with the highest fitness have larger areas, hence they have a better chance of being chosen when a random number is used to select the chromosome [55].

Afterwards, the *crossover* stage occurs, where designs are selected with a specified crossover probability and random sites on each of the bit-strings are swapped. The next step in the process is called *mutation* where a mutation probability is chosen for the newly selected chromosomes and a randomly selected position on the bit string is switched, thus changing the fitness of the chromosome. Afterwards, the newly created offspring are placed into the population and the process is repeated until an end condition is met. Each iteration is called a *generation* and usually, the end condition is a specified tolerance between the past few successive generations, such as a very small change in the fitness function between generations. If the progress towards the best fitness is too slow, completely new designs can be introduced into the population after a set number of iterations in order to increase the diversity in a process called *migration* [54].

7.1.2 MRR Implementation

Genetic algorithms can be applied to a number of problems addressed in this thesis, for example, the search for the maximum deflection position in Chapter 6. For that problem, the upper and lower bounds for the joint input angles can act as constraints, and the fitness function would be to maximize the deflection in the negative y-direction with the design variables being the joint input angles. In this chapter, the configuration and

position of the MRR with the highest maximum tip deflection from Chapter 6 was used as a design point in order to minimize the tip deflections incurred under the applied payload of *-100 N*. It is also necessary to reduce the mass of the entire MRR in order to easily lift and position the modules during manual reconfiguration. Therefore, the fitness function to minimize is given by:

$$m_{total} = \rho_1 A_1 L_1 + \rho_2 A_2 L_2 + \ldots\ldots + \rho_n A_n L_n \tag{7.1}$$

In order to decrease the mass of the MRR, the cross sections of each module can be modified to have the smallest cross section area possible. Here, it is assumed that the links as well as the joint casings are hollow and the thickness of each module (t_i .... t_n) are the design variables. The design variables also act as constraints since they can have upper and lower bounds and are given by:

$$t_{lower} \le t_i \le t_{upper} \tag{7.2}$$

Also, some robot designers prefer to have some equal-sized links and joints, while others simply do not care. Thus, there exists some preferred size ratios and the first link and first joint in the open-chain can act as a reference size to the consecutive links and joints. Define the size ratio for the sizes of each link relative to the first link as $R_{sl,i}$ and the size ratio for the sizes of each joint after the first joint as $R_{sj,i}$. The size ratio constraints can then be written as:

$$\begin{aligned}
&\text{For links:} \\
&t_{l,1} = (known),\ \ t_{l,2} \le R_{s,l,1} t_{l,1},\ \ t_{l,3} \le R_{s,l,3} t_{l,1},\ \ \cdots\ t_{l,n_l} \le R_{s,l,n_l} t_{l,1} \\
&\text{For joints:} \\
&t_{j,1} = (known),\ \ t_{j,2} \le R_{s,j,1} t_{j,1},\ \ t_{j,3} \le R_{s,j,3} t_{j,1},\ \ \cdots\ t_{j,n_j} \le R_{s,j,n_j} t_{j,1}
\end{aligned} \tag{7.3}$$

where $t_{l,i}$ and $t_{j,i}$ are the link and joint thicknesses for the i^{th} module. Thus, a size ratio for each successive module can be defined with reference to the first module of its kind. Also, there are structural constraints involved, such as the maximum tip deflection and the stress factors of safety which are given by:

$$\begin{aligned}
&\left|v_{tip}\right| \le \left|v_{\max,tip}\right| \\
&FS_{y,i} \ge FS_{y,\min}
\end{aligned} \tag{7.4}$$

where $v_{max,tip}$ is the maximum allowable tip deflection in the y-direction, $FS_{y,i}$ can be calculated for each module from equation (5.53) in Chapter 5, and $FS_{y,max}$ is the minimum allowable factor of safety.

7.1.3 Simulation and Results

Using the three-link and three-joint MRR provided in sections 5.3 and 6.4, each link and joint module was optimized in order to reduce the system weight at the maximum deflection position and configuration found in section 6.4, which was configuration #14 with all joint input angles of 0°. However, it is desirable to load the robot with a payload of *-100 N* instead of using the maximum allowable payload forces determined in section 6.4. Also, the maximum allowable tip deflection in the y-direction was constrained to *-0.0001 m*. The upper and lower bound thickness constraints from equation (7.2) for each module were 0.03 m and 0.003 m, respectively and the minimum allowable factor of safety for each module was 1.5. Furthermore, the size ratios for joints 2 and 3 were relative to the first joint were both $R_{s,j} = 0.75$, so these joints could not have a thickness of more than 75% of the first joint's thickness. Similarly, the size ratios for links 2 and 3 relative to the first link were both 0.8. Another test case was conducted without the size ratio constraints to test the effectiveness of the GA. Typically, the module thicknesses should get larger going from the tip of the robot to the base, because the local displacements on each module decrease down the chain in that direction. The GA was implemented using the *Genetic Algorithm and Direct Search* toolbox from MATLAB with the parameters listed in Table 6-1 below.

Table 7-1: Parameters used in the genetic algorithm

Parameter	Value
Initial population size	30
Crossover fraction	0.85
Crossover function	Two point method
Fitness scaling function	Rank method
Maximum number of generations	100
Migration fraction	0.1
Migration interval	10
Selection Function	Roulette wheel method
Maximum number of stall generations	15
Tolerance for fitness	10^{-11}
Feasibility with respect to non-linear constraints	10^{-11}

The initial population size was chosen as 30 because if it was too small or too large, the GA would take longer to converge to a final solution. The crossover fraction was chosen to be 0.85, because typically, the crossover probability ranges from 0.8 and 0.95 for most studies [40]; and the crossover function chosen was the two point method, which first selects two random integers from 1 to the number of design variables, then selects a range of genes between those two integers from the first chromosome to replace the same range of genes in the second chromosome. The fitness scaling used was the rank method, which ranks the chromosomes in order according to their fitness [55]. Also, a migration function was used for this optimization because it can prevent the GA from having stalled iterations, where the fitness does not change after a few iterations. The migration interval was therefore set to 10 iterations.

In MATLAB, the migration scheme chooses the best individuals from one sub-population to replace the worst set of individuals from another sub-population, with a specified fraction of individuals allowed to migrate. Also, the maximum population size was chosen to be 100 in order to prevent the GA from running too long, and the fitness and convergence tolerances were each set to 10^{-11}. Table 6-2 below shows the final results of the optimization for the test cases with and without the size constraints and Figure 7-1 shows the fitness statistics for the optimization.

The *Genetic Algorithm and Direct Search Toolbox* allowed for quick solutions for both cases with only eight generations each. When the size ratio constraint was active, the results show that the entire weight of the MRR was slightly higher than the results produced when the size ratio constraint was ignored. Furthermore, the factor of safety for each module was much larger than the minimum recommended. Upon determining the new tip deflections, both cases had transverse deflections in the negative y-direction slightly less than the known deflection constraint.

Table 7-2: Module thicknesses obtained from GA

Module:	Joint 1	Link 1	Joint 2	Link 2	Joint 3	Link 3	Total Mass [kg]
t_i [m] *(using "≤" size ratios)*	0.00703	0.01571	0.00527	0.01226	0.00350	0.00523	*29.60123*
t_i [m] *(ignoring size ratios)*	0.00626	0.01442	0.00747	0.01261	0.00358	0.00481	*28.79922*

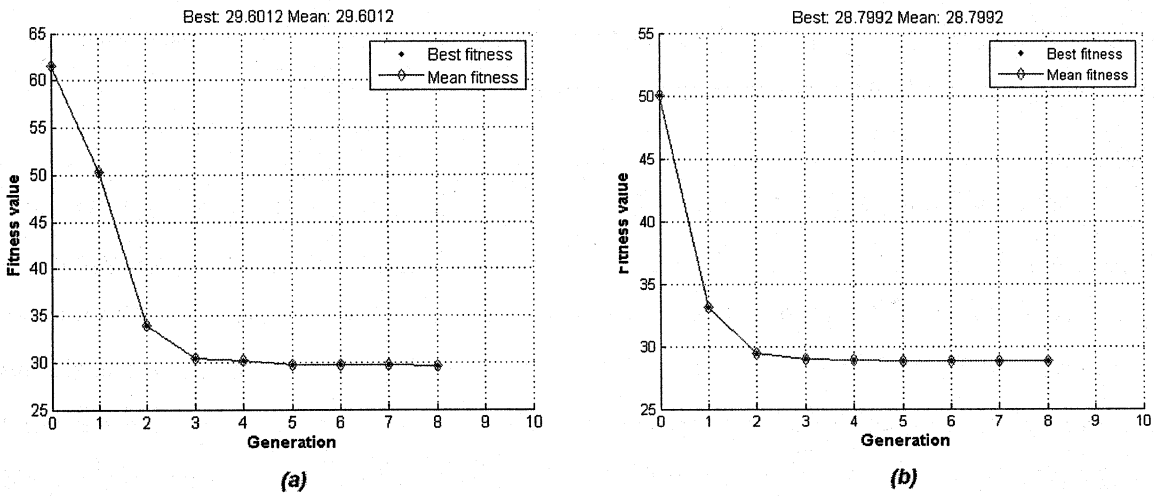


Figure 7-1: Fitness statistics for test case (a) using size ratios, (b) ignoring size ratios

7.2 Direct Module-Based Method

Instead of using an optimization algorithm, this method can solve for the module cross section sizes by directly using the compliance equations developed in Chapter 5 for a given payload and global tip deflection vector. However, if the module weights are included, than an iterative process is necessary. This involves solving for each module's cross section properties explicitly using the local compliance matrix from equation (5.23) in expanded form and rearranging the equations to determine the unknown variables for each mode of deformation, and the largest thickness or cross section area for each case can be chosen as the final module size. Consequently, the solutions are obtained in the

local frame for each module. This is a conservative approach, but should yield results that satisfy the constrained deflections since the unknown variables are solved directly.

7.2.1 System Equations

After the maximum deflection position and configuration are determined using one of the methods in Chapter 6, the next step is to specify a global tip displacement vector which will act as the maximum deflection constraints, along with known allowable payload. Then, the following step includes determining the local forces using equation (5.27) instead of equation (5.39) because the global tip forces for each individual module cannot be determined due to the unknown compliance terms. Afterwards, the local displacements must be determined. This can be done using equation (5.28), however, since the inverse of the deformation Jacobian for n modules cannot be found, an approximation must be applied.

Previously, the pseudoinverse of the Jacobian was applied to solve robot kinematic equations [56, 57]. Here, the pseudoinverse will be utilized to determine an approximation for the inverse of the deformation Jacobian for n modules. In equation (5.26) the Jacobian is an m × n matrix, where m < n. Thus, the pseudoinverse of the deformation Jacobian can be written as:

$$\mathbf{J}_{d,n}^{*} = \mathbf{J}_{d,n}^{\mathrm{T}}\left(\mathbf{J}_{d,n}\mathbf{J}_{d,n}^{\mathrm{T}}\right)^{-1} \tag{7.5}$$

The above equation provides the least-squares solution to the system of linear equations in found in equation (5.28), and its modified form then becomes:

$$\left\{\mathbf{U}_{e,tip}\right\} = \mathbf{J}_{d,n}^{\mathrm{T}}\left(\mathbf{J}_{d,n}\mathbf{J}_{d,n}^{\mathrm{T}}\right)^{-1}\left\{\mathbf{U}_{tip}\right\}_{G} \tag{7.6}$$

Equation (7.6) is used to determine the unknown local displacement vectors using only the global tip displacement vector, which must be specified beforehand. A good estimate of the tip displacement vector can be obtained from performing a kinetostatic analysis of the original model with known compliance terms and obtaining the tip displacement vector. Then, using this known tip displacement vector, another set of tip displacements, which can act as the maximum allowable, can be specified using the same directions of

the displacements found in the kinetostatic analysis. Equation (7.6) distributes the allowable global tip displacement vector to local coordinates for each module, and once each allowable local tip displacement vector is determined, the local compliance equations for each module can be expanded from equation (5.23) and the cross section variables can be determined. The axial force-displacement relation can be rewritten as:

$$\frac{L_i}{A_i E_i} f_{x,i} = u_i \tag{7.7}$$

Assuming the link is a hollow rectangular beam with a known inner cross section base length of b_{inner} and cross section height of h_{inner}, the thickness due to axial forces can be determined using equation (7.7) as:

$$\frac{L_i}{u_i' E_i} f_{x,i} = 4t_i^2 + 2t_i b_{inner,i} + 2t_i h_{inner,i} \tag{7.8}$$

Since equation (7.8) provides more than one solution, the largest real number should be chosen for the thickness. If the joint casing is a hollow cylinder, equation (7.7) can be rewritten to solve the thickness as:

$$\frac{L_i}{u_i E_i} f_{x,i} = 2\pi t_i \left(r_{inner,i} + t_i\right) \tag{7.9}$$

To determine a module's thickness due to torsional deformations, the torsion relation can be written as:

$$\frac{L_i}{G_i J_i} m_{x,i} = \theta_i' \tag{7.10}$$

If the module is a hollow rectangular link, its thickness can be solved using the following equation:

$$\frac{L_i}{G_i \theta_i'} m_{x,i} = \frac{2t_i \left(t_i + b_{inner,i}\right)^2 \left(t_i + h_{inner,i}\right)^2}{\left(2t_i + b_{inner,i} + h_{inner,i}\right)} \tag{7.11}$$

Similarly, if the module is a joint, the thickness of its casing can be determined from:

$$\frac{L_i}{G_i\theta_i^{'}} m_{x,i} = \frac{\pi}{2}\left[\left(r_{inner,i}+t_i\right)^4 - r_{inner,i}^4\right] \tag{7.12}$$

The terms due to bending in the x-y plane can be written directly from equation (5.23) as:

$$\begin{aligned} \frac{L_i^3}{3E_iI_{z,i}} f_{y,i} + \frac{L_i^2}{2E_iI_{z,i}} m_{z,i} &= v_i^{'} \\ \frac{L_i^2}{2E_iI_{z,i}} f_{y,i} + \frac{L_i}{E_iI_{z,i}} m_{z,i} &= \phi_i^{'} \end{aligned} \tag{7.13}$$

In equation (7.13) there are two terms due to the coupling of the forces, hence two terms had to be determined to solve for the thicknesses of both the link and joint modules. For a hollow link module, the thickness can be determined using:

$$\begin{aligned} \left(\frac{L_i^3}{3E_i} f_{y,i} + \frac{L_i^2}{2E_i} m_{z,i}\right)\frac{1}{v_i^{'}} &= \frac{1}{12}\left[\left(b_{inner,i}+2t_i\right)\left(h_{inner,i}+2t_i\right)^3 - b_{inner,i}h_{inner,i}^3\right] \\ \left(\frac{L_i^2}{2E_i} f_{y,i} + \frac{L_i}{E_i} m_{z,i}\right)\frac{1}{\phi_i^{'}} &= \frac{1}{12}\left[\left(b_{inner,i}+2t_i\right)\left(h_{inner,i}+2t_i\right)^3 - b_{inner,i}h_{inner,i}^3\right] \end{aligned} \tag{7.14}$$

The upper term in equation (7.14) is due to the linear displacement in the y-direction and the lower term is due to angular displacement about the z-axis, therefore two distinct thickness values are to be determined. For a hollow joint module, equation (7.14) becomes:

$$\begin{aligned} \left(\frac{L_i^3}{3E_i} f_{y,i} + \frac{L_i^2}{2E_i} m_{z,i}\right)\frac{1}{v_i^{'}} &= \frac{\pi}{2}\left[\left(r_{inner,i}+t_i\right)^4 - r_{inner,i}^4\right] \\ \left(\frac{L_i^2}{2E_i} f_{y,i} + \frac{L_i}{E_i} m_{z,i}\right)\frac{1}{\phi_i^{'}} &= \frac{\pi}{2}\left[\left(r_{inner,i}+t_i\right)^4 - r_{inner,i}^4\right] \end{aligned} \tag{7.15}$$

For bending in the x-z plane the force-displacement relations can be written as:

$$\begin{aligned} \frac{L_i^3}{3E_iI_{y,i}} f_{z,i} - \frac{L_i^2}{2E_iI_{y,i}} m_{y,i} &= w_i^{'} \\ -\frac{L_i^2}{2E_iI_{y,i}} f_{z,i} + \frac{L_i}{E_iI_{y,i}} m_{y,i} &= \psi_i^{'} \end{aligned} \tag{7.16}$$

Using equation (7.16) for a hollow link, the thicknesses due to the linear displacement in the z-direction and rotational displacement about the y-axis can be determined as:

$$\left(\frac{L_i^3}{3E_i}f_{z,i}-\frac{L_i^2}{2E_i}m_{y,i}\right)\frac{1}{w_i'}=\frac{1}{12}\left[\left(h_{inner,i}+2t_i\right)\left(b_{inner,i}+2t_i\right)^3-h_{inner,i}b_{inner,i}^3\right]$$
$$\left(-\frac{L_i^2}{2E_i}f_{z,i}+\frac{L_i}{E_i}m_{y,i}\right)\frac{1}{\psi_i'}=\frac{1}{12}\left[\left(h_{inner,i}+2t_i\right)\left(b_{inner,i}+2t_i\right)^3-h_{inner,i}b_{inner,i}^3\right] \quad (7.17)$$

Correspondingly, for a hollow joint module, equation (7.17) becomes:

$$\left(\frac{L_i^3}{3E_i}f_{z,i}-\frac{L_i^2}{2E_i}m_{y,i}\right)\frac{1}{\phi_i'}=\frac{\pi}{2}\left[\left(r_{inner,i}+t_i\right)^4-r_{inner,i}^4\right]$$
$$\left(-\frac{L_i^2}{2E_i}f_{z,i}+\frac{L_i}{E_i}m_{y,i}\right)\frac{1}{\phi_i'}=\frac{\pi}{2}\left[\left(r_{inner,i}+t_i\right)^4-r_{inner,i}^4\right] \quad (7.18)$$

Therefore, once the local displacements are determined, equations (7.7) to (7.18) can be used to determine the recommended thickness for each module for the six different types of local tip deformations. When the components are assumed to be hollow, each equation gives more than one solution for the thickness, so the highest positive real thickness should be chosen for each module. Afterwards, the highest thicknesses from each solution can be compared and the overall highest thickness can be chosen for the final module design in order to produce a more conservative global tip deflection result.

7.2.2 Simulation and Results

Using the MRR from the example in section 7.2.1, with the same maximum deflection position and applied payload, the recommended thickness of each module was determined by using the Direct Module-Based Method. However, instead of constraining the vertical displacement alone, this method required that the entire maximum allowable global tip displacement vector is defined. It is worth noting that some of the allowable local displacement vectors determined using equation (7.6) might have zero terms, and in order to prevent division by zero, if any of the local displacements found in equations (7.7) to (7.18) are zero, the corresponding equation is not calculated. Also, the module and motor weights were neglected, and if they were included, equations (7.7) to (7.18) must be solved iteratively since the component weights change with different thicknesses.

For configuration #14 in section 6.4, the linear global displacement in the y-direction was negative and the other linear global displacements were zero. Also, it was determined that the rotational displacement about the x-axis was positive, and the rotational displacement about the z-axis was negative, while there was no rotational displacement present about the y-axis. Thus, for this example the same directions were used and the constrained global tip displacement vector for the n^{th} module in equation (5.28) is defined as:

$$\{\mathbf{U}_{tip}\}_G = \{0 \quad -0.0001 \quad 0 \quad 0.00007 \quad 0 \quad -0.0001\}^T \tag{7.19}$$

For each module, the maximum thickness required for each mode of deformation was calculated and listed in the following table.

Table 7-3: Module thicknesses obtained from Direct Module-Based Method

Module	Maximum thicknesses for each mode of deformation						Dominant mode of deformation
	Thickness due to f_x [m]	Thickness due to f_y [m]	Thickness due to f_z [m]	Thickness due to m_x [m]	Thickness due to m_y [m]	Thickness due to m_z [m]	
Joint 1	0	0.00065	0	***0.00363***	0	0.00100	Torsion about x-axis
Link 1	0	***0.025760***	0	0.020832	0	0.019493	Transverse bending in y-direction
Joint 2	0	0.00075	0	***0.00598***	0	0.00153	Torsion about x-axis
Link 2	0	***0.02420***	0	0.02174	0	0.02221	Transverse bending in y-direction
Joint 3	0	0.00030	0	***0.00335***	0	0.00055	Torsion about x-axis
Link 3	0	***0.01340***	0	0	0	0.01122	Transverse bending in y-direction

From the each of the six possible thickness choices for each module, the final design will include the highest possible thickness in order to satisfy the deflection constraints. For the link modules, the highest thickness was due to the local transverse force in the y-direction and the dominant mode of deformation was due to the linear displacement in the local y-directions. For all of the joint modules, the highest thicknesses are due to the torsional deformation. Using the maximum thickness values available for each of the modules in Table 7-3, the overall mass can be calculated as

46.1271 kg. Comparing the results to the GA used in Chapter 7.1.3, the Direct Module-Based method gives much higher component weights, due to the conservative choices of module thicknesses and the pseudoinverse approximation of the allowable local displacements. However, since the thicknesses were higher, the tip displacements can be much lower than those obtained from the GA and lower than the maximum allowable deflections. Afterwards, a kinetostatic analysis was performed to determine the tip deflections and all of the absolute values of tip deflections were indeed considerably lower than the constrained tip deflections in equation (7.19), due to the higher thickness values, which made each component stiffer. In addition, the factors of safety based on equation (5.53) for each module were well over the recommended factor of safety of 1.5. Also, the maximum deflection position for configuration #14 was tested using the new thicknesses determined using the GA and the Direct Module-Based method, and the position at maximum deflection remained the same for all cases. While the Direct Module-Based method does not provide the optimal solution, designers can simply select different feasible thickness values once they are solved in order to provide different flexibility requirements.

8. Conclusions and Future Work

8.1 Conclusions

This thesis presented a kinetostatic model of a modular reconfigurable robot, which was implemented to perform a module-based kinetostatic analysis and optimization of an MRR. But before the model was developed, kinematic models were created to simplify the connectivity of MRR modules to form an open-chain robot, and a flexibility analysis was performed to determine the static and dynamic flexibilities of joint and link modules.

A method based on relative coordinates was created for those who prefer to connect modules relative to a body-fixed coordinate system found on another module. The modified zero reference plane method allowed the modules to connect using a single global coordinate system, and the quasi-global method allowed the global coordinate frame to be shifted according to any possible offset angles. In each of these methods, the possibilities of angular offsets were accounted for, and it can be shown that the quasi-global method allows for the easiest connectivity and reconfiguration of modules since the global coordinate system is shifted for the connectivity of each module after any angular offsets are present. However, this method requires an extra set of calculations to determine the true orientations of each module since it configures the body vectors using the configuration matrix, unlike the other two methods where the configuration matrix is implemented in the static rotation matrix calculation. When there are no offsets present, the quasi-global connectivity method is the same as the ZRP method.

A flexibility analysis of a link connected to a reconfigurable joint model was conducted based on varying joint and link sizes. Three models were created for the static and dynamic cases to represent a flexible link-rigid joint module, a flexible joint-rigid link module, and a hybrid flexible link-flexible joint module. Also, the effects of joint bending and axial deformation were included in the hybrid model to account for larger joints most commonly found in MRRs. Since the flexibility of each component is dependent on its size, two ratios (R_L and R_J) were defined to show the effect of flexibility

on joint and link sizing for different configurations. For various joint and link sizes, one of the components may become flexible while the other becomes rigid, or both components can be considered flexible. For this reason, the regions of flexibility for varying joint and link sizes are plotted for each configuration of the joint-link module, from which the type of flexibility can easily be identified for a specified joint or link size. By knowing the flexibility of each component, the appropriate model can be used and the overall positional accuracy of the joint-link module can be improved by including the effects of static and dynamic flexibility. This method can aid in the design and control of link-joint modules by accounting for joint and/or link flexibility. Furthermore, the flexibility of the joint-link models presented in this thesis were used as a basis to perform the kinetostatic analysis of an entire MRR.

Two kinetostatic models were then developed based on the flexible joint and link modules and compared to test their relative accuracies against each other. Both models incorporated the forward kinematics and connectivity methods developed in Chapter 3. The Compliance-Jacobian model used the compliance of each module and the deformation Jacobian, which transforms global forces and deformations to each module's local coordinate frame. Therefore, this model easily determines the global displacements at the tip of the end module for a given kinematic position and orientation. On the other hand, the Compliance-Jacobian model requires extra steps if the module and motor weights are to be accounted for in the deflection calculations, and the global tip deflections for each module can only be solved once the global tip deflection at the end module is determined. The other method was a finite element method with modified equations to account for the space-fixed kinematic equations previously developed. The stiffness matrix used in this method was more difficult to assemble, but the system assembly allows for all of the global tip deflections to be solved simultaneously. Also, if the joint modules are assumed to have a negligible length, the finite element model equations cannot solve stiffness terms with zero lengths or else the stiffness matrix becomes singular, whereas the Compliance-Jacobian method can still be solved with some zero-length terms. When the two models were compared, the results show negligible differences.

Afterwards, the Compliance-Jacobian kinetostatic model was utilized to determine the position and configuration of the robot where the tip deflection is the largest. This was done using an iterative process to search for the joint positions that contained the highest tip deflections for each configuration of a MRR between the known joint ranges of motion. By performing the forward workspace search, the workspace boundary can be visualized by determining the outermost points using a convex hull. It can also be shown that by using a dexterity index, called the kinetostatic dexterity, the joint positions that give the highest tip deflections have the lowest index, and vice-versa. Therefore, this index can be used in an optimization scheme to determine the position and configuration where the highest tip deflection occurs. However, since the equation is based on the singularity of the deformation Jacobian matrix, there can be more than one optimal kinetostatic dexterity index, so the robot position where the longest reach away from the base in a direction perpendicular to the payload force should be chosen. After the maximum deflection position and configuration was determined, the maximum allowable payload for a given deflection constraint can be easily calculated and it can also be shown that the maximum deflection position does not change if the magnitude of the payload force changes. In addition, an enumeration method was developed to test only the feasible configurations, drastically reducing the total number of configurations that requires testing.

The Compliance-Jacobian kinetostatic model was also used in an optimization scheme in order to determine the best cross section properties for a given number of modules in order to reduce the weight of each module ant the maximum deflection position and configuration. For this problem, a genetic algorithm in MATLAB was implemented and was proven useful in determining the cross section thicknesses of each module tested. The results from the optimization were then compared to the Direct Module-Based Method, which involves a direct solution to determining the cross sectional properties of each module once the global tip deformation at the free end of the robot is known.

8.2 Contributions

The contributions presented in this thesis are summarized as follows:

1. *Directional Configurability and Connectivity*

   The concept of directional configurability was introduced to allow for the simple initial configuration setup and reconfiguration for each MRR module in kinematics programs by simple entering the length of each module and the direction that length goes relative to a coordinate frame of reference. There were three methods of connectivity presented to allow for directional configurability: the relative connectivity method, zero reference plane, and quasi-global method. Each method included the possibilities of angular offsets such as an uneven base link.

2. *Flexibility Analysis of Reconfigurable Modules*

   A method to determine the static and dynamic flexibility of link and joint modules was presented to show that for various sizes of modules, rigid-body assumptions are not applicable. Two size ratios were define and three models were created to determine the flexibilities of either the joint and link in different configurations. Then, the three regions of static and dynamic flexibility can be plotted and designers can use these graphs to determine which component is more flexible based on its length and cross section area.

3. *Compliance-Jacobian Kinetostatic Method for MRRs*

   This method was developed to determine the tip deflections at the end module of a MRR. Reconfigurability was incorporated by using one of the three connectivity methods, and equations were developed to enable each unknown variable to be determined. The Compliance-Jacobian model was then applied to determine the maximum payload a MRR can take, along with a typical structural optimization problem. The Jacobian used in this method can also be utilized in the kinetostatic dexterity index to determine the maximum deflection position and configuration for an MRR.

4. *Direct Module-Based Method to Determine Unknown Compliance Variables*
   This method was created to directly determine the cross section properties of an MRR under a given payload with a constrained global tip deflection vector. It was shown that the equations present in the Compliance-Jacobian Method can be directly solved to determine any unknown geometric variables, and can be used to produce conservative MRR designs with lower deflections, but higher module weights.

8.3 Future Work

The kinetostatic modeling presented in this thesis deals with simple-shaped modules such as cylindrical-shaped joints and rectangular links with uniform cross sections, but in reality, not all MRR components have such simple shapes. Thus, it is recommended that three-dimensional finite elements with higher-order shape functions, such as tetrahedral and hexahedral elements, be used to create meshed 3-D models of MRR modules which can better represent complicated shapes. The only downside to this is the large computational times required to perform the kinetostatic analysis, especially when searching the entire workspace of a robot, but with computing advancements increasing rapidly, this can become a realization in the near future.

In this thesis, it was assumed that the robot moves at a quasi-static speed and the dynamics can be ignored, however, many MRRs can move at high speeds. Hence, the dynamic effects should also be considered. In addition, a method to determine the inverse kinematics for MRRs should be investigated and used in the determination of the MRR dynamics. Furthermore, a better graphical representation of each module can be created for kinematics programs, and instead of representing the modules as simple beams, solid models can be visualized. This can give way to more accurate kinematic equations because the solid geometric models can include the possibility of module collisions more accurately. Also, beam finite element models can be derived using Timoshenko beam theory to account for shear deformation in beams with low aspect ratios (length to thickness). Another future task is to verify the results of the kinetostatic models with experiments.

9. References

[1] I.M. Chen, G. Yang, "Kinematic Calibration of Modular Reconfigurable Robots Using Product-of-Exponentials Formula". *Journal of Robotic Systems* 14 (11), (1997), pp. 807-821.

[2] L. Kelmar, P.K. Khosla, "Automatic Generation of Kinematics for a Reconfigurable Modular Manipulator System". *Proc. IEEE International Conference on Robotics and Automation* (1988), pp. 663-668.

[3] Y. Fei, X. Zhao, L. Song, "A Method for Modular Robots Generating Dynamics Automatically". *Robotica* 19 (1), (2001), pp. 59-66.

[4] A. Castano, P. Will, "Mechanical Design of a Module for Reconfigurable Robots". *Proc. IEEE International Conference on Intelligent Robots and Systems* (2000), pp. 2203-2209.

[5] Qiang (John) Sun, "Modeling and Simulation of Reconfigurable Systems with Application to the Polishing Process", MASc. Thesis, Ryerson University, (2004).

[6] Y. Fei, X. Zhao, W.L. Xu, "Kinematics and Dynamics of Reconfigurable Modular Robots". *Proc. IEEE International Conference on Systems, Man and Cybernetics*, (1998), pp. 3325-3334.

[7] G.J. Hamlin, A.C. Sanderson, "Tetrobot: A Modular System for Hyper-Redundant Parallel Robotics". *Proc. IEEE International Conference on Robotics and Automation* (1995), pp. 863-870.

[8] G. Yang, I.M. Chen, I.G. Kang, "Task-based optimization of modular robot configurations: Minimized degree-of-freedom approach". *Mechanism and Machine Theory*, 35 (4), (2000), pp. 517-540.

[9] I.M. Chen, and G. Yang, "Numerical Inverse Kinematics for Modular Reconfigurable Robots". *Journal of Robotic Systems* 16 (4), (1999), pp. 213-225.

[10] M. Hafez, D. Lichter, S. Dubowsky, "Optimized Binary Modular Reconfigurable Robotic Devices". *Proc. IEEE International Conference on Robotics and Automation* (2002), pp. 335-340.

[11] B. Li, S. Ma, J. Liu, Y. Wang, "Development of a Shape Shifting Robot for Search and Rescue". *Proc. IEEE International Workshop on Safety, Security and Rescue Robotics* (2005), pp. 31-35.

[12] M. Yim, D.G. Guff, K.D. Roufas, "Polybot: A Modular Reconfigurable Robot". *Proc. IEEE International Conference on Robotics and Automation* (2000), pp. 514-520.

[13] Q. Wu, G. Cao, Y. Fei, "Described model of a Modular Self-Reconfigurable Robot". *Proc. of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou*, (2005), pp. 18-21.

[14] C. Ünsal, P.K. Khosla, "Mechatronic Design of a Modular Self-Reconfigurable Robotic System". *Proc. IEEE International Conference on Robotics and Automation* (2000), pp. 1742-1747.

[15] K. Tomita, S. Murata, E. Yoshida, H. Kurokawa, S. Kojaki, "Reconfiguration Method for a Distributed Mechanical System". *Distributed Autonomous Robotic System*, (1996), pp. 17-25.

[16] J.W. Suh, S.B. Homans, M. Yim, "Telecubes: Mechanical Design of a Module for Self-Reconfigurable Robots". *Proc. IEEE International Conference on Robotics and Automation* (2002), pp. 4095-4101.

[17] Y. Fei, X. Zhao, "Modules Classification and Automatic Generation of Kinematics on Self-Reconfigurable Modular Machines". *Journal of Intelligent and Robotic Systems* 43, (2005), pp. 147–159.

[18] I. Chen, G. Yang, W.K. Lim, S.H. Yeo, "Design and Kinematic Analysis of Modular Reconfigurable Robots". *Proc. IEEE International Conference on Robotics and Automation* (1999), pp. 2501-2506.

[19] G.L. Anderson, "Stability of a Manipulator with Resilient Joints". *Journal of Sound and Vibration* 101 (4), (1985), pp. 463-480.

[20] S.C. Tang, C.C. Wang, "Computation of the Effects of Link Deflections and Joint Compliance on Robot Positioning". *Proc. IEEE International Conference on Robotics and Automation* (1987), pp. 910-915.

[21] J. Martins, M.A. Botto, J. Sá da Costa, "Modeling of Flexible Beams for Robotic Manipulators". *Multibody System Dynamics*, 7 (1), (2002), pp. 79-100.

[22] L.L Howell, A. Midha, "Parametric Deflection Approximations for End-Loaded Large-Deflection Beams in Compliant Mechanisms". *ASME Journal of Mechanical Design*, 117 (1), (1995), pp. 156-165.

[23] S. Tosunoglu, "Modeling of Flexible Elements in Robotic Systems for Modular Environments". *IECON (Industrial Electronics Conference) Proceedings*, (1995), pp. 138-143.

[24] B.J. Torby, I. Kimura, "Dynamic Modeling of a Flexible Manipulator with Prismatic Links". *ASME Journal of Dynamic Systems, Measurement and Control*, 121 (1), (1999), pp. 691-696.

[25] B. Fallahi, S. Lai, C. Venkat, "A Finite Element Formulation of a Flexible Slider Crank Mechanism Using Local Coordinates". *ASME Journal of Dynamic Systems, Measurement and Control*, 117 (2), (1995), pp. 329-335.

[26] Z.C. Lin, D.T. Lin, "Dynamic Deflection Analysis of a Planar Robot". *Computers and Structures*, 53 (4), (1994), pp. 947-960.

[27] Y. Shigang, Y. Yueqing, B Shixan, "Flexible Rotor beam Element for the Manipulators with Joint and Link Flexibility". *Mechanism and Machine Theory*, 32 (4), (1997), pp. 209-219.

[28] Z. Yang, J.P. Sadler, "Finite Element Analysis of Revolute Manipulators with Link and Joint Compliance by Joint-Beam Elements". *ASME Design Engineering Division (DE), Robotics Spatial Mechanisms and Mechanical Systems*, 45 (1) (1992), pp. 619-625.

[29] F. Xi, R.G. Fenton, "Coupling Effect of a Flexible Link and a Flexible Joint". *International Journal of Robotics Research*, 13 (5), (1994), pp. 443-453.

[30] S.G. Kim and J. Ryu, "New Dimensionally Homogeneous Jacobian Matrix Formulation by Three End-Effector Points for Optimal Design of Parallel Manipulators". *IEEE Transactions on Robotics and Automation*, 19 (4), (2003), pp. 731-737.

[31] D. Chablat, P. Wenger, F. Majou, J.P. Merlet, "An Interval Analysis Based Study for the Design and the Comparison of Three-Degrees-of-Freedom Parallel Kinematic Machines". *International Journal of Robotics Research*, 23 (6), (2004), pp. 615-624.

[32] D. Chablat, J. Angeles, "On the kinetostatic optimization of revolute-coupled planar manipulators". *Mechanism and Machine Theory*, 37 (4), (2002), pp. 209-219.

[33] W. Khan, J. Angeles, "The Kinetostatic Optimization of Robotic Manipulators: The Inverse and Direct problems". *ASME Journal of Mechanical Design*, 128 (1), (2006), pp. 168-178.

[34] F. Xi, R.G. Fenton, "A Quasi-Static Motion Planner for Flexible Manipulators Using the Algebra of Rotations". *Proc. IEEE International Conference on Robotics and Automation* (1991), pp. 2363-2368.

[35] F. Xi, R.G. Fenton, "Point-To-Point Quasi-Static Motion Planning for Flexible-Link Manipulators". *IEEE Transactions on Robotics and Automation* (1995), pp. 770-776.

[36] H.S. Kim, Y.J. Choi, "The Kinetostatic Capability Analysis of Robotic Manipulators". *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems* (1999), pp. 1241-1247.

[37] J. Roy, R.P. Goldberg, L.L. Whitcomb, "Structural Design, Analysis and Performance Evaluation of a New Semi-Direct Drive Robot Arm: Theory and Experiment". *IEEE/ASME Transactions on Mechatronics*, 9 (1), (2004), pp. 10-18.

[38] G. Mroz, L. Notash, "Design and Prototype of Parallel, Wire-Actuated Robots with a Constraining Linkage". *Journal of Robotic Systems*, 21 (12), (2004), pp. 677–687.

[39] F. Xi, D. Zhang, C.M. Mechefske, S.Y.T. Lang, "Global kinetostatic modeling of tripod-based parallel kinematic machine". *Mechanism and Machine Theory*, 39 (4), (2004), pp. 517-540.

[40] D. Zhang, Z. Xu, C.M. Mechefske, F. Xi, "Optimum design of parallel kinematic toolheads with genetic algorithms". *Robotica*, 22 (1), (2004), pp. 77-84.

[41] A.A. Shabana, "Computational Dynamics". John Wiley & Sons Inc., (2001).

[42] F. Xi, "Computational Dynamics Lecture Notes". Ryerson University, (2004).

[43] K.J. Bathe," Finite Element Procedures". Prentice Hall, New Jersey (1996).

[44] D.L. Logan, "A First Course in the Finite Element Method". 3rd Ed. Brooks/Cole, Pacific Grove, CA (2002).

[45] W.T. Thomson, M.D. Dahleh, "Theory of Vibration with Applications". Prentice Hall, New Jersey (1998).

[46] R.R. Craig Jr., "Mechanics of Materials". 2nd Ed. John Wiley & Sons Inc., (2000).

[47] G. Dieter, "Mechanical Metallurgy". 3rd Ed. McGraw-Hill, New York, (1986).

[48] S.K. Agrawal, L. Kissner, M.Yim, "Joint Solutions of Many Degrees-of-freedom Systems Using Dextrous Workspaces". *Proc. IEEE International Conference on Robotics and Automation* (2001), pp. 2480-2485.

[49] D. Botturi, P. Fiorini, " A Geometric Method for Workspace Computation". *Proc. International Conference on Advanced Robots (ICAR)* (2003).

[50] C.B. Barber, D.P. Dobkin, H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls". *ACM Transactions on Mathematical Software*, 22 (4), (1996), pp. 469-483.

[51] T. Yoshikawa, "Manipulability of Robotic Mechanisms". *International Journal of Robotics Research,* 4 (3), 1985, pp. 3-9.

[52] N.A. Aspragathos, S. Foussias. "Optimal location of a robot path when considering velocity performance". *Robotica*, 20 (1), 2002.

[53] I.M. Chen, J.W. Burdick, "Assembly Configurations of Modular Robotic Systems". *Proc. IEEE/RSJ International Conference on Intelligent Robotics and Systems* (1993), pp. 1985-1992.

[54] J. S. Arora, "Introduction to Optimum Design". 2nd Ed. Elsevier Academic Press, London, (2004).

[55] M. Mitchell, "An Introduction to Genetic Algorithms". MIT Press, Boston, (1998).

[56] J. Yuan, "Local SVD inverse of robot Jacobians". *Robotica,* 19(1), (2001), pp.79-86.

[57] J. Lenarcic, "Alternative Computational Scheme of Robot Kinematics". *Proc. IEEE International Conference on Robotics and Automation* (1998), pp. 3235-3240.

Appendix A - FEM Stiffness and Mass Matrices

The local stiffness equation for a single beam element that can be calculated using equation (4.4) in Chapter 4 can be explicitly expressed as:

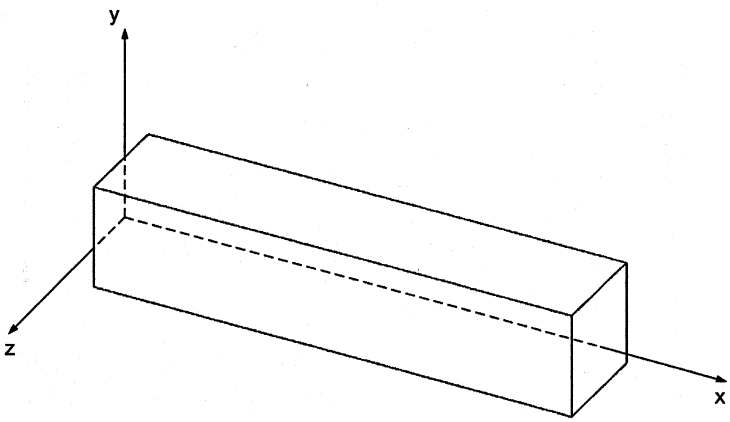
$$[\mathbf{k}_e]=\begin{bmatrix}
\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\
0 & 0 & \frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & 0 & -\frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 \\
0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\
0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\
0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\
-\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\
0 & 0 & -\frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\
0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\
0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\
0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L}
\end{bmatrix}$$

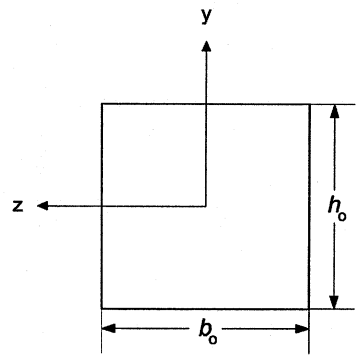
The local consistent mass matrix for a single beam element that can be calculated using equation (4.30) in Chapter 4 can be explicitly expressed as:

$$[\mathbf{M}_e]=\begin{bmatrix}
\frac{140\rho AL}{420} & 0 & 0 & 0 & 0 & 0 & \frac{70\rho AL}{420} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{156\rho AL}{420} & 0 & 0 & 0 & \frac{22\rho AL^2}{420} & 0 & \frac{54\rho AL}{420} & 0 & 0 & 0 & -\frac{13\rho AL^2}{420} \\
0 & 0 & \frac{156\rho AL}{420} & 0 & -\frac{22\rho AL^2}{420} & 0 & 0 & 0 & \frac{54\rho AL}{420} & 0 & \frac{13\rho AL^2}{420} & 0 \\
0 & 0 & 0 & \frac{140\rho JL}{420} & 0 & 0 & 0 & 0 & 0 & \frac{70\rho JL}{420} & 0 & 0 \\
0 & 0 & -\frac{22\rho AL^2}{420} & 0 & \frac{4\rho AL^3}{420} & 0 & 0 & 0 & -\frac{13\rho AL^2}{420} & 0 & \frac{-3\rho AL^3}{420} & 0 \\
0 & \frac{22\rho AL^2}{420} & 0 & 0 & 0 & \frac{4\rho AL^3}{420} & 0 & \frac{13\rho AL^2}{420} & 0 & 0 & 0 & \frac{-3\rho AL^3}{420} \\
\frac{70\rho AL}{420} & 0 & 0 & 0 & 0 & 0 & \frac{140\rho AL}{420} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{54\rho AL}{420} & 0 & 0 & 0 & \frac{13\rho AL^2}{420} & 0 & \frac{156\rho AL}{420} & 0 & 0 & 0 & -\frac{22\rho AL^2}{420} \\
0 & 0 & \frac{54\rho AL}{420} & 0 & -\frac{13\rho AL^2}{420} & 0 & 0 & 0 & \frac{156\rho AL}{420} & 0 & \frac{22\rho AL^2}{420} & 0 \\
0 & 0 & 0 & \frac{70\rho JL}{420} & 0 & 0 & 0 & 0 & 0 & \frac{140\rho JL}{420} & 0 & 0 \\
0 & 0 & \frac{13\rho AL^2}{420} & 0 & \frac{-3\rho AL^3}{420} & 0 & 0 & 0 & \frac{22\rho AL^2}{420} & 0 & \frac{4\rho AL^3}{420} & 0 \\
0 & -\frac{13\rho AL^2}{420} & 0 & 0 & 0 & \frac{-3\rho AL^3}{420} & 0 & -\frac{22\rho AL^2}{420} & 0 & 0 & 0 & \frac{4\rho AL^3}{420}
\end{bmatrix}$$

Appendix B - Cross section Properties for Common Shapes

1) Solid Rectangular Beam





Area moment of inertia about the y- and z-axis:

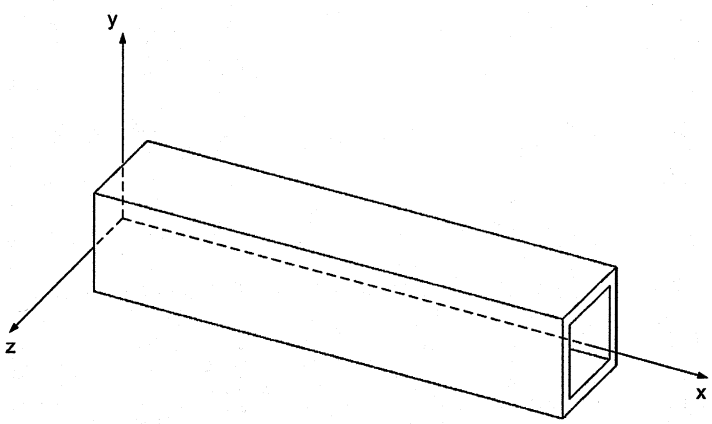
$$I_y = \frac{h_o b_o^3}{12}, \quad I_z = \frac{b_o h_o^3}{12} \tag{A-1}$$

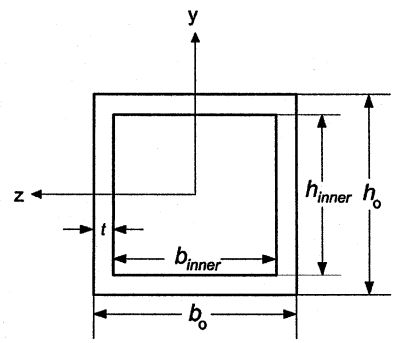
Torsion constant:

$$J = \beta b h^3 \tag{A-2}$$

Where β is a dimensionless constant and can be determined from Craig [46].

2) Hollow Rectangular Beam with Constant Thickness





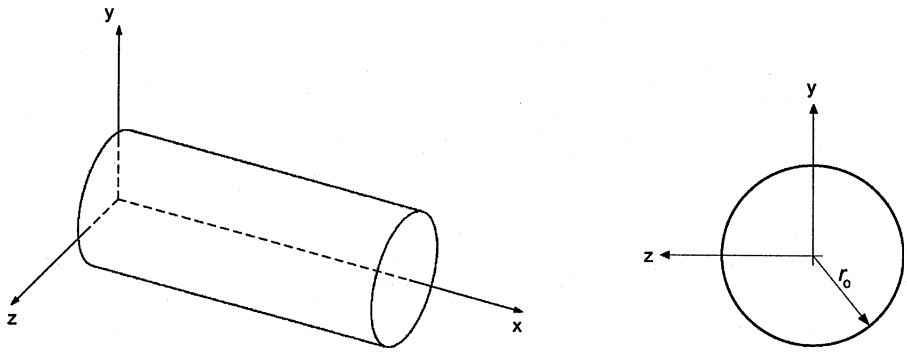
Area moment of inertia about the y- and z-axis:

$$I_y = \frac{1}{12}\left(h_o b_o^3 - h_{inner} b_{inner}^3\right), \quad I_z = \frac{1}{12}\left(b_o h_o^3 - b_{inner} h_{inner}^3\right) \tag{A-3}$$

Torsion constant:

$$J = \frac{2t\left(b_o - t\right)^2\left(h_o - t\right)^2}{b_o + h_o - 2t} \tag{A-4}$$

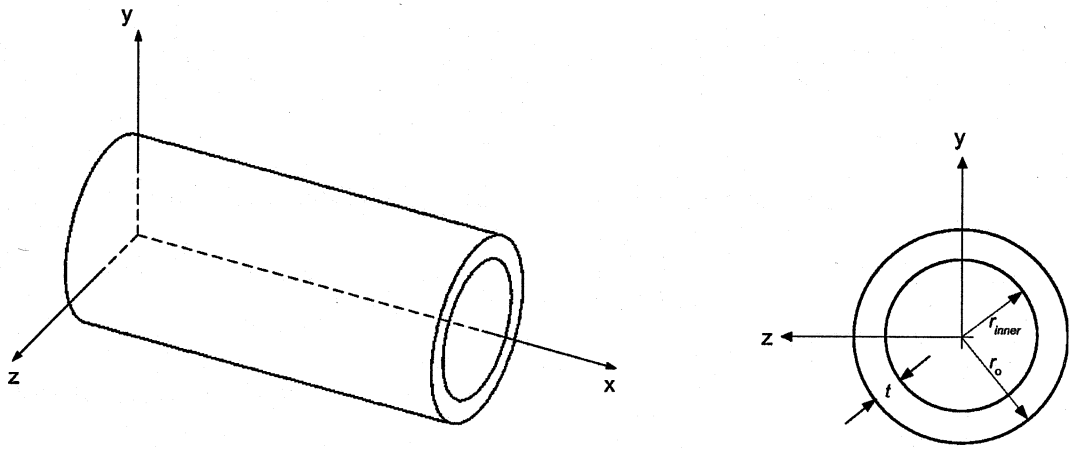
3) Solid Circular Shaft



Area moment of inertia about the y- and z-axis, and torsion constant:

$$I_y = I_z = \frac{\pi r_o^4}{2} = J \tag{A-5}$$

4) Hollow Circular Shaft



Area moment of inertia about the y- and z-axis:

$$I_y = I_z = \frac{\pi}{2}\left(r_0^4 - r_{inner}^4\right) \tag{A-6}$$