

Model Based Deformable Object Manipulation Path
Planning and Position Control

MODEL BASED DEFORMABLE OBJECT MANIPULATION PATH
PLANNING AND POSITION CONTROL

BY
JERZY SMOLEN , B.Eng.

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
AND THE SCHOOL OF GRADUATE STUDIES
OF MCMASTER UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

© Copyright by Jerzy Smolen , December 2009

All Rights Reserved

Master of Applied Science (2009)
(Electrical & Computer Engineering)

McMaster University
Hamilton, Ontario, Canada

TITLE: Model Based Deformable Object Manipulation Path Planning and Position Control

AUTHOR: Jerzy Smolen
B.Eng., (Department of Electrical and Computer Engineering)
McMaster University, Hamilton, Canada

SUPERVISOR: Dr. Alexandru Patriciu

NUMBER OF PAGES: xv, 112

Dedicated to my loving parents.

Abstract

A deformable object is a body that can undergo an alteration in shape and dimension as a result of external interactions. This characteristic makes deformable object handling considerably more difficult than handling rigid objects, which allow for the independent control of object grasping and manipulation. In the case of a deformable object, the grasping and manipulation interfere with one another, since a large enough force on the object may change its shape and location. One solution to this problem is the use of deformable object models. Models can provide an insight to the behavior of a deformable object subjected to various interactions with its environment. Lately, the use of deformable object models has received much attention from the industrial and medical communities.

This thesis addresses the use of a deformable object model, developed using the mesh free Reproducing Kernel Particle Method RKPM, to aid in the accurate control of soft object manipulation path planning procedures and target stabilization during needle insertion procedures. Also, the deformable object is used as a part of an integrated system to control the manipulation of a physical object, using a robotic tool under position feedback from a stereoscopic camera.

The results present a comparison between two different integration methods used to solve the deformable object model; The Gaussian Quadrature and a more efficient method referred to as the Collocated method, which is explained in Chapter 3. In both cases,

the accuracy values of the results were comparable. A preliminary study was successfully completed using a coarse and refined object model to show a virtual concept of the physical system prior to its development. Using markers to dictate the performance of a physical object corresponding to the simulated object, experimental results were attained for a planar deformable object. Successful path following tasks were accomplished using the physical deformable object.

Further, the target stabilization method proved to be successful in reducing the movement of a target with respect to the insertion direction of the needle in the tissue. The static paddle approach used the criteria of choosing the optimal paddle based on the highest reduction in target lateral movement, which showed to be case specific. The dynamic paddle approach was formulated to reposition the target back to its original position and was more robust to object characteristic changes.

Acknowledgements

My thanks and appreciation to my supervisor, Dr. Alexandru Patriciu, for his extraordinary guidance, support and patience throughout my time at McMaster.

A special thanks to my family for their constant belief in me and encouragement that helped me excel in my academic career.

A sincere gratitude to all my friends for sharing many laughs.

Contents

Abstract	iv
Acknowledgements	vi
1 Introduction	2
1.1 Motivation	2
1.2 Problem Statement and Thesis Contribution	5
1.2.1 Deformable Object Manipulation Path Planning	5
1.2.2 Target Stabilization During Needle Insertion Procedures	6
1.3 Organization of Thesis	7
1.4 Related Publications	7
2 Literature Review	8
2.1 Rigid Objects	9
2.2 Deformable Objects	10
2.2.1 Deformable Object Modeling and Simulation	10
2.2.2 Deformable Objects and Robotic Interactions	15
3 Description of Deformable Object Model	19

3.1	Summary	19
3.2	Meshless Methods	20
3.3	Deformable Object Model	21
3.4	Discretization of the Continuum Equation Using RKPM	25
3.5	Derivation of Discretized Equations	28
3.6	Numerical Implementation	31
4	Experimental Setup	34
4.1	Summary	34
4.2	System Setup	35
4.2.1	Vision System	40
4.2.2	System Coordinate Transformation	43
4.2.3	Robot to Object Space	47
4.2.4	Physical Object	48
4.2.5	Robot and Platform System	48
5	Deformation Planning for Robotic Soft Object Manipulation	51
5.1	Summary	51
5.2	Object Deformation Control Method	52
5.3	Results	54
5.3.1	Preliminary Object Model Simulations	54
5.3.2	Physical Validation Study	61
5.4	Discussion	84
6	Target Stabilization in Needle Insertions	86
6.1	Summary	86

6.2	Procedure	87
6.3	Simulation Results	88
6.3.1	Target Point Stabilization with Static Paddle	89
6.3.2	Stabilization of Targeted Point with Dynamic Paddle	93
6.4	Discussion	95
7	Conclusion and Future Work	97
7.1	Conclusion	97
7.2	Future Work	99
A	Appendix	101
A.1	Vision System	101
A.2	Physical Object	101
A.3	Matlab Robot Control	102

List of Figures

3.1	Deformable object with immobile connection being manipulated by a set of robotic arms, following a set of defined paths for surface control points.	20
3.2	Object coordinate system in un-deformed and deformed states.	22
3.3	Graphical representation of RKPM discretization. Illustration obtained from Chen (1996).	32
4.1	System overview for object control using a robotic manipulator, deformable object model and a stereoscopic camera.	35
4.2	System overview showing the data flow sequence between the simulated object model, vision system and robotic manipulator.	37
4.3	Section outlining the robotic manipulator commands to receive data from, actuate and send data to the object model.	38
4.4	Section outlining the interaction between the vision system, object model and robotic manipulator.	39
4.5	Information flow for acquiring an image using the stereoscopic camera.	40
4.6	Example of the left and right images obtained from a stereoscopic camera of the same environment.	42
4.7	Illustrating the geometry of a stereoscopic camera.	42
4.8	Illustration of coordinate systems belonging to the camera, tool and object.	43

4.9	Tool tip localization.	45
4.10	Image guided system platform set up.	49
4.11	Tool manipulating object model.	50
5.1	Paddle acts on the bottom 2/3 of object face at position $x = -2.5$, and the entire back face is immobile at position $x = -1.5$	55
5.2	Each step of the reduced model corresponds to a deviation from that of the desired control point path in Table 1.	56
5.3	Deformation step 1 in Table 5.1 of reduced model following control point trajectory.	57
5.4	Deformation step 3 in Table 5.1 of reduced model following control point trajectory.	57
5.5	Deformation step 6 in Table 5.1 of reduced model following control point trajectory.	58
5.6	Each step of the refined model corresponds to a deviation from that of the desired control point path in Table 5.1.	59
5.7	Deformation steps 1, 2 and 3 in Table 5.1 of refined model following control point trajectory.	59
5.8	Deformation steps 4, 5 and 6 in Table 5.1 of refined model following control point trajectory.	60
5.9	Paddle actuates on entire object face at position $x = -2.5$ and entire back face is immobile at position $x = -1.5$	60
5.10	Compression in the X Axis by displacing the essential boundary by 0%, 10% respectively for Gaussian Quadrature method.	63

5.11	Compression in the X Axis by displacing the essential boundary by 20%, 30% respectively for Gaussian Quadrature method.	63
5.12	Compression in the X Axis by displacing the essential boundary by 40%, 43% respectively for Gaussian Quadrature method.	64
5.13	Compression in the X and Y Axis by displacing the essential boundary by 0%, 10% respectively for Gaussian Quadrature method.	64
5.14	Compression in the X and Y Axis by displacing the essential boundary by 20%, 30% respectively for Gaussian Quadrature method.	65
5.15	Compression in the X and Y Axis by displacing the essential boundary by 40%, 50% respectively for Gaussian Quadrature method.	65
5.16	Compression in the X and Y Axis by displacing the essential boundary by 58%, 59% respectively for Gaussian Quadrature method.	66
5.17	Compression in the X Axis by displacing the essential boundary by 0%, 10%, and 15% respectively for Collocated method.	66
5.18	Compression in the X and Y Axis by displacing the essential boundary by 0%, 10%, and 14% respectively for Collocated method.	67
5.19	Set point regulation of control point via manipulation point control using the Gaussian Quadrature Method.	68
5.20	Set point regulation of control point via manipulation point control using the Collocated Method.	68
5.21	Simulating the tracking of a desired control point trajectory in Table 5.3 via object manipulation using the Gaussian Quadrature Method.	70
5.22	Simulated control point tracking error of trajectory in Table 5.3 using the Gaussian Quadrature Method.	71

5.23	Simulating the tracking of a desired control point trajectory in Table 5.4 via object manipulation using the Gaussian Quadrature Method.	71
5.24	Simulated control point tracking error of trajectory in Table 5.4 using the Gaussian Quadrature Method.	72
5.25	Simulating the tracking of a desired control point trajectory in Table 5.5 via object manipulation using the Gaussian Quadrature Method.	72
5.26	Simulated control point tracking error of trajectory in Table 5.5 using the Gaussian Quadrature Method.	73
5.27	Simulating the tracking of a desired control point trajectory in Table 5.3 via object manipulation using the Collocated Method.	73
5.28	Simulated control point tracking error of trajectory in Table 5.3 using the Collocated Method.	74
5.29	Simulating the tracking of a desired control point trajectory in Table 5.4 via object manipulation using the Collocated Method.	75
5.30	Simulated control point tracking error of trajectory in Table 5.4 using the Collocated Method.	75
5.31	Simulating the tracking of a desired control point trajectory in Table 5.5 via object manipulation using the Collocated Method.	76
5.32	Simulated control point tracking error of trajectory in Table 5.5 using the Collocated Method.	76
5.33	Set point regulation of control point via manipulation point control.	78
5.34	Indication that there is a potential problem when small manipulator movements are involved.	78

5.35	Path comparison between the simulated object and recorded control points in set point regulation problem.	79
5.36	Control of physical object in tracking a box shape trajectory.	80
5.37	Path comparison between desired control point and recorded control point in tracking a box shape trajectory.	80
5.38	Path comparison between desired control point and simulated object control point in tracking a box shape trajectory.	81
5.39	Path comparison between the simulated object and recorded control points in tracking a box shape trajectory.	81
5.40	Recorded force data in space acting on the tool tip.	82
5.41	Calculated force on the object model control point for each successive path step.	82
5.42	Calculated force on the object model manipulation points for each successive path step.	83
6.1	Deformed object model indicating immobile boundary, needle insertion, stabilizing paddle and a non-homogeneous region.	87
6.2	Control point error in the X-axis of a homogeneous object as a function of paddle position.	90
6.3	Simulated test object after a needle compression of 15% without the use of a static paddle support. Note the control point begins at position (0.5,0.4) .	91
6.4	Simulated test object after a needle compression of 15% using static paddle support in position index 4. Note the control point begins at position (0.5,0.4)	92

6.5	Control point error in the X-axis of an object with one non-homogeneous region located at indices $(0.6, 0.4), (0.7, 0.4), (0.7, 0.3), (0.6, 0.3)$ as a function of paddle position.	92
6.6	Control point error in the X-axis of an object with non-homogeneous region at indices $(0.3, 0.4), (0.4, 0.4), (0.4, 0.5), (0.3, 0.5)$ as a function of paddle position.	93
A.1	Central control for communication between the robotic manipulator, stereo camera and object information.	103
A.2	TCP/IP communication block between robot and object model.	104
A.3	Tool transformation from robot output coordinates to object input coordinates.	104
A.4	Tool transformation from object output coordinates to robot input coordinates.	104

List of Tables

- 5.1 Desired path for object control point in simulated system. 55
- 5.2 Values for A and C tensors. All values not indicated are zero. Obtained
from Chapter 15 Tissue Mechanics Cowin (2006). 62
- 5.3 Desired path for control point following a box shape trajectory. 69
- 5.4 Desired path for control point following a figure 8 shape trajectory. 69
- 5.5 Desired path for control point following a straight line. 70

- 6.1 Values for A and C tensors. All values not indicated are zero. Obtained
from Chapter 15 Tissue Mechanics Cowin (2006). 89
- 6.2 Index relating to paddle positions. 89
- 6.3 δx_{min} as a result of calculated Jacobian Matrices. 95

Chapter 1

Introduction

1.1 Motivation

The development of modern robotic systems used in industrial, space or medical applications require sophisticated and adaptable designs to perform complex manipulation tasks in unknown and variably structured environments. Having such flexibility and tolerance in dealing with unpredictable environments, presents an alternative for highly specialized and specific workspace setups. Imitating basic human vision-touch capabilities makes this possible, where the vision and touch sensory systems allow the robot to make decisions based on multiple feedback information. This approach maybe useful in tasks where robots interact with soft deformable environments.

A deformable object is a body that can undergo an alteration in shape and dimension as a result of external interactions. Handling these deformable objects requires a consideration of more factors than handling rigid objects. Grasping objects that are rigid requires the independent control of grasping forces, while the position and orientation is changed simply by manipulating the free moving object. In the case of a deformable object, the

grasping and manipulation interfere with one another since a large enough force applied on the object may change its shape and location. It becomes important to merge the grasping and manipulation control in order to accurately account for the deformation and position changes.

Today, the interaction of a robot with an environment that is entirely composed of rigid objects or obstacles is atypical. Many of the objects that robotic manipulators are called to handle in industry are in fact not rigid. Objects can often flex or significantly vary in their characteristics, which can make the handling process quite complex. Even a stiff material can become quite flexible or delicate if its structure allows for it, such as an extended pipe or a large thin sheet. Manufacturing errors themselves can have variable effects on the product, forcing for multiple exceptions in pre-programmed processes to handle various anomalies.

Currently, objects within a robotic environment can be separated into several groups and are handled accordingly. Objects that require more sophisticated handling techniques typically call upon the dexterity of a human worker. Reasons for this include the fact that a robot system is typically in a stationary position and often requires the knowledge of its local environment if it is to do repetitive work efficiently. Also, the object behavior can be quite unpredictable and difficult for a robotic system to adjust to without proper adaptability. Further, objects that are rigid, or those that undergo negligible deformations upon being subjected to external forces, can be assumed and treated as rigid objects. Handling rigid objects is typically addressed by recognizing the object's distinct features, and developing highly sophisticated, special-purposed robot end effectors specific for dealing with that particular object. It could be more economical to design a form of hardware that is more general and accommodating for dealing with variable objects. In this case, more time

can be spent on developing robot sensory systems, simulation quality and robot prediction techniques.

This extension of rigid object manipulation approaches to deformable object handling has attracted various investigative studies for a wide range of applications. Early studies and prototypes include methods of handling and processing of textiles and systems for maintaining high voltage lines in order to removing operators from dangerous work conditions Henrich (2000). Also, new techniques for handling a wide range of components to be processed and assembled in shipbuilding, aerospace and automobile industries. Most of which are flexible and irregular in shape. More recently, more of this type of research is finding its way into the medical field. With increased demands to perform less invasive surgical procedures like laparoscopic techniques, medical professionals require training systems of higher complexity aside from the current training on living animals or cadavers. Simulated surgery can become a critical development tool for the medical professional Cotin (1999). The study of surgical procedures performed on a deformable environment mimicking a realistic one is expected to help create control algorithms that will aid significantly in complicated surgical tasks. With accurate models of various tissues and their behaviour when subjected to external forces, autonomous tasks can be developed to perform planned procedures. These tasks can involve assistive surgical manipulations, surgical incisions or internal target tracking for drug delivery or biopsy procedures. This has the potential to increase patient recovery time and provide more reliable procedures where repetition is required.

1.2 Problem Statement and Thesis Contribution

Currently, a large portion of this type of research is designated to finding an improved way of accurately modeling large structural deformations of various materials, with key considerations for accuracy and proper material characteristics. In this thesis, two problems are addressed with theoretical and experimental validation. The problem outline and solution proposals are provided below.

1.2.1 Deformable Object Manipulation Path Planning

Deformable object operations rely on fundamental positioning techniques to maneuver them in a desired manner. Due to the nature of these objects, an applied external force can alter the original shape, making the operations more difficult than simple grasping force control. Currently, deformable object manipulation is proposed as an indirect positioning operation. This includes tracking chosen control points on an object Wada (1999), Mallapragada (2008), and providing these points a trajectory to follow. These control points are used to gauge the accuracy of the progress of the object manipulation, based on a simulated model which helps in the development of a proper control law for the physical system. In these current proposed methods, an assumption is made that the deformable objects are accessible from any position on the object. Simulated models are created from a mass-spring system. Multiple manipulators are allowed to perform the task, and in the case of Wada (1999), on multiple chosen control points.

In this thesis, the above issue is addressed by utilizing a different modeling method. The mesh free Reproducing Kernel Particle Method (RKPM) is employed. While previous approaches to deformation control based on mass-spring networks proved to be adequate for simple shapes and structures, the deformable model used in this work is expected to

allow deformation control on objects with complex shapes and strain energy functions. The model is based on the incremental nonlinear elastic formulation. RKPM was chosen for this study as it has been proven to accurately handle large deformations and requires no re-meshing algorithms Chen (1996). This model has the potential to be more realistic in modeling various material characteristics by using appropriate strain energy functions.

1.2.2 Target Stabilization During Needle Insertion Procedures

Currently, the placement of needles for percutaneous intervention is usually performed manually using straight-line needle trajectories. This is mainly due to the absence of sufficient feedback and knowledge of needle steering mechanics required to make trajectory compensation feasible. If the results are unsatisfactory, the procedure is repeated which can lead to unnecessary tissue trauma. Several approaches have been undertaken to remedy this issue, including complex algorithms for needle steering Alterovitz (2003), DiMaio (2003) and Glozman (2007) and tissue target manipulation in line with the straight needle insertion Mallapragada (2008). Most of the current research in this area is still in its infancy stages.

To address this problem, a novel approach was developed to hold the target in place while the tissue was deformed due to the inserted needle. In other words, the tissue target is stabilized to intersect with the inserted needle trajectory. This gives an alternative approach to using expensive and complex maneuverable needles, with the potential for less preoperative planning. The same RKPM model is utilized to simulate object deformations.

1.3 Organization of Thesis

The rest of this work is organized in the following manner. A literature review is presented in Chapter 2 pertaining to the integration of image guided robotic procedures and deformable object modelling and manipulation. Further, chapter 3 outlines a detailed description of the deformable object model. In chapter 4, the experimental equipment set up is outlined and described in detail. Chapters 5 describes the deformable object manipulation path planning experiments, including various simulations, experimental results and discussion. Similarly, chapter 6 describes the target stabilization application including simulations, experimental results and discussion. The final chapter is designated for conclusions and possible future works.

1.4 Related Publications

- J. Smolen and A. Patriciu, “Deformation Planning for Robotic Soft Tissue Manipulation,” Second International Conferences on Advances in Computer-Human Interactions, Feb 2009, Cancun, Mexico, pp. 199 – 204. (*Best Paper Award*)
- J. Smolen and A. Patriciu, “Model Based Stabilization of Soft Tissue Targets in Needle Insertion Procedures,” in Proc. Int. Conf. of the IEEE Engineering in Medicine and Biology, Sept 2009, Minneapolis, USA.

Chapter 2

Literature Review

Prior to object deformation research, robotic systems were mostly built under the assumption that the interactive environment of the robot was composed entirely of rigid objects. These systems were hard coded to perform precise tasks, requiring complete information about the working environment. This included the manipulated objects shape, size, rigidity and location within the workspace. The main objective in terms of interaction with these objects was stable grasping techniques which allowed for object relocation. Even though most objects are rigid or enough to be assumed as such, many objects currently dealt with are in fact not rigid with unsymmetrical, compliant and alterable shapes which significantly complicate the entire handling process.

In the case of autonomous robotics systems, the main challenge stems from the several different interconnected problems that must be considered simultaneously during the process. These problems include the development of an accurate deformation model of the environment, its simulation and comparison to physical estimates of the environment being modeled, as well as the establishment of a control law to carry out the manipulation process based on multi-sensory feedback, such as vision or force sensors Khalil (2007).

This chapter presents an overview of the modeling developments and current applications of deformable objects. Also, modern concepts adopted by the robotics community addressing the issues involved with deformable object manipulation, will be discussed.

2.1 Rigid Objects

Understanding robotic interactions with rigid objects is a good beginning for understanding the complexity involved in interacting with deformable objects. To perform most tasks, the manipulator must be able to securely restrain and manipulate the rigid object. As a result, establishing reliable contact and grasping models such as form closure and force closure Khalil (2007) are required. Form closure is the ability to prevent motions of the object, relying on frictionless contact constraints Bicchi (2000), which can become challenging when finding the optimal locations and number of contact placements for arbitrary geometric objects Markenscoff (1990) and Markenscoff (1989) and Liu (2000). On the other hand, force closure deals with completely restraining the grasped object despite whatever external disturbances, typically by using suitably large forces by the gripper on the object. The analysis of force closure can be seen in Chen (1993) and Nguyen (1988). In general, the main idea is to choose a grasping technique which will avoid or minimize the chance of slippage of an object during the process of manipulation. This may include finer point contact points or more robust handling by wrapping around the object using several contact points and palm. Dexterous manipulation is an area where multiple manipulators coordinate to perform grasping and manipulations operations on objects. This method distinguishing characteristic is that it is object centered Okamura (2000), in other words the problem is formulated around the expected behavior the object.

With these techniques, the process of controlling the interactive forces at the contact

points requires methods such as the Hybrid position/force control Craig (1981) or the Impedance control scheme Hogan (1987). The Hybrid model breaks the force and position into two orthogonal components, where the force act perpendicular to the surface on which the position is tracked. The Impedance model keeps the force position components together and attempts to reduce the rigidity of the manipulator by assigning a desired impedance to the tool.

2.2 Deformable Objects

The automatic handling of deformable objects typically relies on models and simulations to perform thorough evaluations of the object deformation characteristics prior to conducting physical experiments. This allows for accurate predictions of the object behavior under stress and can lead to optimized control strategies for efficient object-manipulator interactions.

2.2.1 Deformable Object Modeling and Simulation

Early deformation modeling emerged in computer animation Terzopoulos (1987), where the development of models for deformable curves, surfaces, and solids based on simplifications of elasticity theory were made, to overcome the more tedious kinematic modeling methods used. These were successful in demonstrating complex, realistic motions arising from the interaction of deformable models with its environment. Terzopoulos *et al.* were further able to develop alternative formulations of their deformable models by unifying both rigid and non-rigid dynamics, know as their hybrid model Terzopoulos (1988). More specifically, the deformations were decomposed into a reference component, which could

be represented as an arbitrary shape, and a displacement component allowing deformation away from the reference shape. Here the reference component was modeled according to the laws of rigid body dynamics while, the deformable model followed the laws of linear elasticity. Although, no particular attempt to model specific materials accurately was made, complicated concepts of material cracking and fractures under increased stress were addressed. Similarly, research done in Platt (1988) also helped to show the advantages of physical models over kinematic models for computer animations. In the survey on deformable objects in computer graphics Gibson (1997), a broad range of approaches for dealing with deformable object simulations were presented for shape editing, cloth modeling, object and character animation, image analysis, and surgical simulation in computer graphics.

A commonly used modeling technique is the mass-spring system. An object is modeled as a collection of point masses connected by springs and dampers in a lattice structure Howard (1997), Paloc (2002) and Tokumoto (1999). Mass-spring systems are a simple physical model with well understood dynamics. They are easy to construct and are well suited for parallel computation due to the local nature of the interactions between nodes. Although, the parameters for these models Lloyd (2007) are often hard to estimate and complex networks exhibit a localization of the deformation as stated in Zhong (2005) where the propagation of external forces is dictated by a model which draws an analogy between heat conduction and elastic deformation. Along the same lines, a similar concept model drew an analogy between elastic deformation and cellular neural networks Zhong (2006). Other popular methods include finite element method and boundary element methods. The boundary element methods differ from finite elements in that only the contour of an object needs to be meshed, unlike the complete object meshing for finite elements.

This makes boundary elements attractive for computer vision problems, where object edges can be tracked upon deformation and compared to modeled force calculations as seen in Greminger (2003). In Pincibono (2001), the finite element method was used to model soft tissue for a haptic simulator, and similarly in Chanthasopeephan (2007) for a soft tissue prior to cutting analysis. In each case the goal of the study was to improve the realism of environment. As mentioned in Gibson (1997) that surgical systems may become the new standard for training for surgeries, it is especially important that tissue modeling be not only physically accurate, but also stress the real-time interactions during surgical simulations require feedback to be instantaneous upon tissue stimulations Cotin (1999) regardless its complexity. The model in Sundaraj (2002), addresses the problem of real-time simulation of large deformations in interactive applications involving deformable objects. A virtual surgical environment based on fracture mechanics for modeling surgical cutting of tissue was developed in Mahvash (2001), and later implemented in a haptic environment. The authors identified two interaction modes between tool and medium cutting and deformation. The authors also evaluated experimentally the forces that are exerted on the blade during the cutting process. Doran *et al.* Doran (2004) developed a model for thin biological membranes cutting based on a strain energy based failure model. A failure model for the skin was developed and the critical value for the resistance to fracture was experimentally identified for chicken skin. This result has particular significance as it provides experimental data from biological samples.

Mesh free particle methods were developed since they were seen as a logical step to avoid mesh constraints, which are produced in finite element methods Li (2002). It was found that it was computationally efficacious to discretize a continuum by only a set of nodal points, excluding a mesh. There are numerous advantageous of using the meshfree

approach, including easily handling large deformations since the connectivity of the computed nodes can change over time, can easily handle damage of its components which makes it suitable for fracture modeling, as well as the discretization can provide accurate representation of geometric objects Li (2002). Examples of using these techniques include the ability to simulate crack growth and propagation in a surface Belytschko (1994) and the handling of large deformation in rubber materials Chen (1997).

In addition to the significant amount of research on deformation for computer graphics and virtual environments, other research efforts in robotics have been dedicated to the modeling problem of objects in the process of the physical manipulation. In this case, general and adaptable approaches are needed to handle a wide range of object characteristics. Due to nonlinear attributes and other unknown variables, deformable objects are subject to significantly change in shape during a manipulation process, and can react differently to a range of applied forces Khalil (2007). As a result, having an accurate object model is indispensable to determine deformation characteristics during an object-manipulator interaction. There have been many various approaches to deformable object modeling. According to Saadat (2002), the majority of these models focused on the *2D* problems, with a few attempts at addressing *1D* and *3D* models. Due to the fact that industrial demand dictates these statistics, there should be a large trend to work towards *3D* models as more attempts are made to automate parts of the food and medical industries.

Deformable linear objects are typically studied as a precursor or simple case before tackling the more complex *3D* model. Many manipulative operations deal with them such as wires, cords, and threads with flexural, torsional, and extensional deformations in three-dimensional *3D* space. Their applications extend to maintenance of live line work during power outages Maruyama (2000), communications systems, electrical apparatuses, auto

and space industries, and medical suturing. Deformation models of linear objects have been modeled using internal energy of the object under geometric constraints, allowing calculation of deformation by minimizing the internal energy shown by Hirai in Henrich (2000) and Wakamatsu (2004). In this case, Hirai used rubber bands and paper sheets as phantoms to validate the model to handle bends, twists and extensions in 3D space. Further, in the compilation book Henrich (2000), Remede and Henrich simulate the behavior of linear objects numerically.

Further, a very common issued addressed by studying the behavior of deformable objects involves the insertion of these objects into a rigid hole. Meer *et al.* Meer (1994), developed a controller for implementing an insertion task in which the object had to be deformed within certain specifications in order to perform the assembly operation. The model required an explicit initial model of the object which included mass object dynamics and a model of the object's flexibility, and complex sensors were required to measure the position and force of the object. Several other strategies were developed for this application by Nakagaki (1997), Chen (1991), Zheng (1991) and Yue (2002). Beam trajectory planning and the vibration-free handling of a beam during object transfer were addressed using beam dynamic models in Zheng (1993) and Chen (1993) respectively. Also, a deformable object was discretized using finite elements and treated as an underactuated mechanical system Tanner (1999), where the object was tested under axial load and bending. These methods are all extendable to the 2D case.

In Howard (1997) a generalized approach was developed to automatically manipulate 3D deformable objects by developing a firm grasp of the material modeled as a mass-spring system. The premise of the study was to relate the grasping force characteristics to the deformability of the material. This study was followed up by the implementation of neural

networks Howard (1999) to generalize the grasping force for lifting a wide range of object materials. There is a need for 3D object handling in the food and agricultural industry, as most of these operations deal with viscoelastic objects and are prone to deformation during operational processes Saadat (2002).

2.2.2 Deformable Objects and Robotic Interactions

The typical way to automate robotic manipulators in handling deformable objects is through the merger of visual and force sensing. Visual information will provide the guidance of the manipulator to locate the object as well as help in the evaluation the manipulator accuracy during task performance or object slippage. In line with visual information, data collected from force measurements can provide important clues about the object physical characteristics and to perform skillful manipulations of deformable objects which may be irregular shape and composition. Provided that this information is elegantly synthesized into a complete model, object manipulation can be fined tuned through feedback systems. To make these processes feasible, modeling of the behavior of the deformable object under certain handling conditions are required. Sometimes due to the deformation of the objects the operations may result in failure because of unexpected deformation of the object during the manipulation process. Evaluation of these objects through simulations are therefore necessary so that we can perform the manipulative operations successfully.

The indirect deformation of a deformable object can be described as the movement of a set of points interior to the object, by a separate set of points on the object boundary. In general, the set of points to be guided as a result of object manipulation, are referred to as the controlled points. The set of points at which the object is interacting with the environment, are referred to as the manipulation points. This approach seems the most

realistic in imitating the way humans interact with most objects, and for cases where objects are in difficult to access areas with little available manipulation surface.

An early approach to the indirect deformation method was undertaken in Wada (1998), to manipulate extensional fabrics for the garment industry. Due to the difficulty of building an exact model of a deformable object, resulting from objects typical nonlinearities and parameter variations, a $2D$ coarse model was developed from a mesh of masses and springs. Based on this idea, an iterative control scheme was proposed. The objective of the control method was to indirectly control multiple points simultaneously on textile fabrics, and to position them in desired locations with the aid of visual feedback and a coarse object model. In follow up studies, Wada (2001) the previous discrete control scheme which made grasping stability difficult was replaced by the used of a continuous PID feedback control. Simulations showed the position error converging from deformations that were not large.

This application is a very important preliminary set to the development of image-guided surgical procedures. Among the most common procedures undertaken during clinical practise is the subcutaneous insertion of needles and catheters. The complexity of these procedures varies greatly, from superficial needle sticks to biopsies of deep-seated tumors. Currently, physicians rely on their own three dimensional spatial orientation of anatomical structures and perform the procedures by hand. Since tissue deformations are ignored, errors in needle tip placements occur.

Closely resembling Wada *et al.*, Mallapragada (2007) performed indirect deformable object manipulations to simulate breast biopsy procedures. In their preliminary study, external actuator were used to position a lesion or inhomogeneity located within the object

in line with a straight needle insertions. This procedure was performed as a proof of concept in real-time under the guidance of visual feedback. A similar mass-spring model was used to model the deformable object. Further studies extended the idea to several different approaches of indirect tumor manipulation. In Mallapragada (2008), two new trials were performed. In the first, they demonstrated the movement of the inclusion during needle insertion. The phantom used for the procedure was inhomogeneous as to mimic the deflection of the inclusion during external disruptions. Secondly, the inclusion was placed at a desired location. In this case the needle was inserted into the tissue creating a disruption, and the external manipulators were used to compensate and move the inclusion to the needle in the $2D$ plane. The method was successful, although the fact that the object was accessible from all directions in the $2D$ plane makes the model less realistic. A similar model approach addressing this issue was undertaken in Torabi (2009), where only one manipulator was used as a support during needle insertion to bring the needle tip to the target. Having limited accessibility to the object requires additional preoperative planning.

Alternatively, the trajectory modeling of the needle path has attractive applications. The ability to steer the needle through complex environments such as organ, bone or blood vessel, based on the needle's ability to follow curved paths through tissue using different mechanisms, can provide speedier recovery times for patients undergoing certain procedures. In DiMaio (2002b), DiMaio (2002a) and DiMaio (2003), a haptically-enabled virtual insertion environment was developed which allowed for a flexible symmetric-tip needle to be steered in a 3D inhomogeneous linear elastostatic tissue model. This allows for giving tissue motion resulting from needle penetration. Further, more complicated concepts were introduced in DiMaio (2005) to develop the first needle trajectory motion planning technique that incorporated soft tissue motion, needle flexibility, and physically based contact

model. Needle tip trajectories were planned with the use of planning guides which were oval-shaped potential fields representing point obstacles, for which the corresponding gradients were modeled as repulsive for obstacles and attractive for targeted areas. The needle tip was set to follow the direction of the field gradient at each iterative set in the simulation. Similarly, in Glozman (2007) a flexible needle-tip trajectory is calculated, using an inverse kinematics algorithm, to avoid obstacles on its way to the target. Assuming small displacements, the needle is modeled as a linear beam supported by virtual springs. Finally, Alterovitz *et al.* addressed planning optimal insertion locations and insertion distances for rigid symmetric tip needles to compensate for 2D tissue deformations Alterovitz (2003). As well, a 2D planner was developed to explicitly compensate for the effects of tissue deformation by combining finite element simulation with numeric optimization Alterovitz (2005).

Chapter 3

Description of Deformable Object Model

3.1 Summary

A deformable object model is created to accurately simulate a physical object subjected to externally applied forces. The idea behind the model is to try to emulate the behavior of the physical object during a specified task. Having this capability allows the model to be used to develop a control scheme for the physical object manipulation. It is assumed that the deformable object model can be manipulated using several robotic manipulators. Considered in this work, the object deformation is induced through boundary displacements, which are defined as the essential boundary, while other sections of the object can be set to be immobile or contain different physical properties to create a non-homogeneity. Symbolically, it is assumed that the deformable object is manipulated at a set of points called manipulation points MP_i where $i = 1 \dots N_{MP}$ and its deformation is measured on a finite number of control points CP_i where $i = 1 \dots M_{CP}$. This is shown in Figure 3.1, where a set of three robotic manipulators attached at various MP locations, indirectly handle the object whose deformation path is dictated by the three CP positions. The model tries to mimic

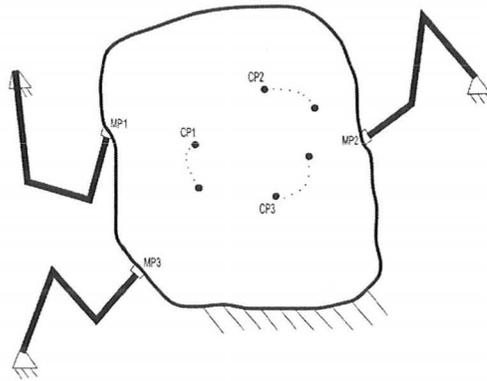


Figure 3.1: Deformable object with immobile connection being manipulated by a set of robotic arms, following a set of defined paths for surface control points.

objects in the real world by adding certain constraints on its boundaries.

3.2 Meshless Methods

Many engineering problems in solid mechanics consider a body undergoing large deformations, which lead to considerable variations in the original shape that cannot be neglected. This requires the development of a robust model which can accurately handle such large deformations. Although the Finite Element Method (FEM) has dominated in terms of analysis of structural linear and nonlinear responses, it has several drawbacks, some of which can be addressed using the Meshfree methods. To include a few of these, FEM numerical compatibility is only an approximation of the real physical compatibility of a continuum, where for Lagrangian type of computations, mesh distortions can either end the computation altogether or result in drastic deterioration of accuracy Li (2004). Also, very fine mesh in problems with high gradients are usually needed, which become computationally expensive. Further, adaptive remeshing for complex problems such as large deformations,

or procedures for simulations of impact and penetration problems have become a common task to undertake, although remeshing and remapping the state variables from the old mesh to a new mesh, is a process often prone to numerical errors Li (2004). The Meshfree particle methods are a different form of discretization and were developed to improve the inadequacies of the FEM Li (2004). They are able to easily handle large deformations since the connection between nodes are generated as part of the computation and can change in time, as well supporting a flexible refinement procedure which allows more new particle to be added to areas requiring higher precision Li (2004).

Over the past few decades, much of the Meshfree methods research was devoted to weak-form methods compared to the strong-form methods Liu (2005). From this many different approaches came into existence, including the reproducing kernel particle methods (RKPM) Liu (1995). The main idea of RKPM was to improve the Smooth Particle Hydrodynamics (SPH) approximation to satisfy consistency requirements using a correction function Liu (2005). Among its wide range of applications, RKPM was used in nonlinear and large deformation problems for structural analysis Chen (1996).

3.3 Deformable Object Model

Throughout this thesis the following conventions will be followed. The initial coordinate system is represented by upper case X and the deformed configuration is represented by lower case x . The region occupied by the body in the initial configuration is Ω_X and it has a boundary Γ_X . The deformed configuration is Ω_x with the boundary Γ_x . The deformation of the body is a one to one function ϕ , $x = \phi(X, t)$ as shown in Figure 3.2. The determinant of the deformation gradient is $J = \det(\partial x_i / \partial X_j) \neq 0$. The directional derivative is represented using comma notation, and repeated indices indicate a sum over

the number of dimensions.

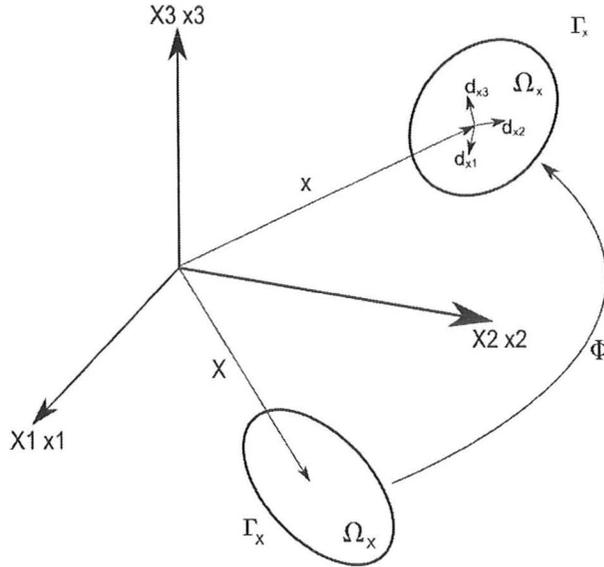


Figure 3.2: Object coordinate system in un-deformed and deformed states.

The deformation model is formulated as a boundary problem using nonlinear elasticity tools as follows. The body in the deformed state Γ_x is subject to body forces b_i , boundary traction h_i on the natural boundary $\Gamma_x^{h_i}$ and boundary displacement g_i on the essential boundary $\Gamma_x^{g_i}$. Then, the task is to find $u_i(X, t)$ such that

$$\tau_{ij,j} + b_i = 0 \quad (3.1)$$

where the comma is used to represent a spatial derivative with respect to j in equation 3.1.

$$\frac{\delta \tau_{ij}}{\delta x_j} + b_i = 0$$

is the equivalent to equation 3.1. This equation is subject to boundary conditions.

$$\tau_{ij}n_j = h_i \quad \text{on} \quad \Gamma_x^{h_i} \quad (3.2)$$

$$u_i = g_i \quad \text{on} \quad \Gamma_x^{g_i} \quad (3.3)$$

with n_i the outward surface normal in the deformed configuration, $u(X,t) = \phi(X,t) - X$ is the material displacement, and τ_{ij} the Cauchy stress. The stress is computed from the constitutive equation that connects the stress to the current strain. Biological tissue doesn't obey linear elasticity constitutive laws and are modeled as hyperelastic materials. Hyperelastic materials are characterized by a strain energy density function W . The second Piola-Kirchhoff stress is computed from the strain energy density as

$$S_{ij} = \frac{\partial W}{\partial E_{ij}} \quad (3.4)$$

and the Cauchy stress is computed as

$$\tau_{ij} = \frac{1}{J} F_{im} S_{mn} F_{jn} \quad (3.5)$$

where F is the deformation gradient and E is the Green Lagrangian strain.

$$F_{ij} = \frac{\partial x_i}{\partial X_j} = \frac{\partial u_i}{\partial X_j} + \delta_{ij} \quad (3.6)$$

and

$$E = \frac{1}{2} (F^T F - I). \quad (3.7)$$

The standard variational (weak) form of equation 3.1 is posed as follows Liu (2005). This is a weak-form defined over the global problem domain Ω .

$$\int_{\Omega_x} \delta u_{i,j} \tau_{ij} d\Omega - \int_{\Omega_x} \delta u_i b_i d\Omega - \int_{\Gamma_x^{h_i}} \delta u_i h_i d\Gamma = 0 \quad (3.8)$$

where δu_i represent the variations which are zero on the essential boundary, and u_i represent the test functions which satisfy the essential boundary conditions. The previous equation can be expressed in body coordinates X as

$$\int_{\Omega_X} \delta u_{i,\bar{j}} F_{ik} S_{kj} d\Omega - \int_{\Omega_X} \delta u_i(X), b_i(x(X)) J(X) d\Omega - \int_{\Gamma_X^{h_i}} \delta u_i(X) h_i^0(X) d\Gamma = 0 \quad (3.9)$$

where h_0 is the surface force per unit of undeformed area on the undeformed natural boundary $\Gamma_X^{h_i}$, $u_{i,j} \equiv \partial u_i / \partial x_j$, $u_{i,\bar{j}} \equiv \partial u_i / \partial X_j$.

The previous equations are nonlinear with respect to the unknown function u . This equation will be converted into a incremental version using a Taylor series expansion of $F_{ik} S_{kj}$. The function $F_{ik} S_{kj}$ is a function of $u_{i,\bar{j}}$. Assuming incremental deformations as defined in Ogden (2007) the following relation holds

$$F_{ik} S_{kj}(D) = F_{ik} S_{kj}(D_0) + \frac{\partial F_{ik} S_{kj}}{\partial D} \Big|_{D_0} \Delta D \quad (3.10)$$

with $D = u_{l,\bar{m}} = F - I$ the displacement gradient, $\Delta D = D - D_0$.

$$\frac{\partial F_{ik} S_{kj}}{\partial u_{l,\bar{m}}} \Delta u_{l,\bar{m}} = \left(\frac{\partial F_{ik}}{\partial u_{l,\bar{m}}} S_{kj} + F_{ik} \frac{\partial S_{kj}}{\partial E_{qm}} F_{lq} \right) \Delta u_{l,\bar{m}} \quad (3.11)$$

$$\frac{\partial F_{ik}}{\partial u_{l,\bar{m}}} S_{kj} = \frac{\partial u_{i,\bar{k}}}{\partial u_{l,\bar{m}}} S_{kj} = \delta_{il} \delta_{km} S_{kj} = \delta_{il} S_{mj} = D_{ijlm} \quad (3.12)$$

$$F_{ik} \frac{\partial S_{kj}}{\partial E_{qm}} F_{lq} = F_{ik} F_{lq} \frac{\partial W}{\partial E_{kj} \partial E_{qm}} = T_{ijklm} \quad (3.13)$$

and

$$\frac{\partial F_{ik} S_{kj}}{\partial u_{l,\bar{m}}} \Delta u_{l,\bar{m}} = \left(\delta_{il} S_{mj} + F_{ik} F_{lq} \frac{\partial W}{\partial E_{kj} \partial E_{qm}} \right) \Delta u_{l,\bar{m}} \quad (3.14)$$

Using equations 3.10 through 3.14 the incremental version of 3.9 becomes

$$\begin{aligned} \int_{\Omega_X} \delta u_{i,\bar{j}} (D_{ijklm} + T_{ijklm}) \Delta u_{l,\bar{m}} d\Omega &= \int_{\Omega_X} \delta u_i(X), b_i(x(X)) J(X) d\Omega \\ &+ \int_{\Gamma_X^{h_i}} \delta u_i(X) h_i^0(X) d\Gamma - \Theta^k \end{aligned} \quad (3.15)$$

where $\Theta_k = \int_{\Omega_X} \delta u_{i,\bar{j}} F_{ik} S_{kj} d\Omega$ computed using the deformation and stress at the k^{th} iteration. Equation 3.15 will be used to compute the the deformation and stress at $k + 1^{th}$ iteration.

3.4 Discretization of the Continuum Equation Using RKPM

In this work meshless discretization of the continuum body using reproducing kernel particle methods (RKPM) is used. The continuum is discretized using NP particles distributed in Ω_X and each particle has an associated shape function. The displacement u is approximated as a linear combination of the particles shape functions.

$$u_i^a(X) = \sum_{I=0}^{NP} \Psi_I(X) d_i \quad (3.16)$$

In this work the following shape function is used

$$\Psi_I(X) = H(0)^T M(X)^{-1} H\left(\frac{X - X_I}{a}\right) \Phi_a(X - X_I) \Delta V_I \quad (3.17)$$

$$M(x) = \sum_{J=0}^{NP} H\left(\frac{X - X_J}{a}\right) H\left(\frac{X - X_J}{a}\right)^T \Phi_a(X - X_J) \Delta V_J \quad (3.18)$$

where n_{sd} is the space dimensions, X_I is the coordinate of particle I , $\Omega_I = \prod_{i=1}^{n_{sd}} [X_{Ii} - a_{Ii}, X_I + a_{Ii}]$, a_{Ii} is the dilation parameter in direction i for the shape function associated with particle I , H is the multidimensional basis function vector, and Φ_I is the multidimensional kernel function.

$$H(Y) = [1, Y_1, Y_2, \dots, Y_{n_{sd}}] \quad (3.19)$$

$$\Phi_I(X) = \prod_{i=1}^{n_{sd}} \frac{1}{a} \Phi\left(\frac{X_i - X_{Ii}}{a}\right) \quad (3.20)$$

where $\Phi(x) : \mathbf{R} \rightarrow \mathbf{R}$ is

$$\Phi(x) = \begin{cases} \frac{2}{3} - 4x^2 + 4|x|^3, & 0 \leq |x| \leq \frac{1}{2}; \\ \frac{4}{3} - 4|x| + 4x^2 - \frac{4}{3}|x|^3, & \frac{1}{2} \leq |x| \leq 1; \\ 0, & \text{otherwise.} \end{cases} \quad (3.21)$$

The RKPM functions do not have the Kronecker delta property, therefore, they do not automatically satisfy the essential boundary conditions. This can be resolved if the set of particles is divided in two subsets. The set of particles belonging to the essential boundary $\Lambda_b = \{i_1^b, \dots, i_{N_b}^b\}$ and the non-boundary particles $\Lambda_{nb} = \{i_1^{nb}, \dots, i_{N_{nb}}^{nb}\}$

$$\begin{aligned}
u_i^h(X) &= \sum_{I \in \Lambda_{nb}} \Psi_I^{nb}(X) d_{iI} + \sum_{I \in \Lambda_b} \Psi_I^b(X) g_{iI} \\
\delta u_i^h(X) &= \sum_{I \in \Lambda_{nb}} \Psi_I^{nb}(X) d_{iI}
\end{aligned} \tag{3.22}$$

where

$$\Psi_I^{nb}(X) = \Psi_I(X) - (N^b(X)(D^b)^{-1}D^{nb})_I \tag{3.23}$$

$$\Psi_I^b(X) = (N^b(X)(D^b)^{-1})_I \tag{3.24}$$

$$N^b(X) = \left(\Psi_{i_1^b}(X) \quad \dots \quad \Psi_{i_{Nb}^b}(X) \right); \Lambda_b = \{i_1^b, \dots, i_{Nb}^b\} \tag{3.25}$$

$$N^{nb}(X) = \left(\Psi_{i_1^{nb}}(X) \quad \dots \quad \Psi_{i_{Nnb}^{nb}}(X) \right); \Lambda_{nb} = \{i_1^{nb}, \dots, i_{Nnb}^{nb}\} \tag{3.26}$$

$$\frac{\partial \Psi_I^{nb}(X)}{\partial X_j} = \frac{\partial \Psi_I(X)}{\partial X_j} - \left(\frac{\partial N^b(X)}{\partial X_j} (D^b)^{-1} D^{nb} \right)_I \tag{3.27}$$

$$\frac{\partial \Psi_I^b(X)}{\partial X_j} = \left(\frac{\partial N^b(X)}{\partial X_j} (D^b)^{-1} \right)_I \tag{3.28}$$

$$\frac{\partial N^b(X)}{\partial X_j} = \left(\frac{\partial \Psi_{i_1^b}(X)}{\partial X_j} \quad \dots \quad \frac{\partial \Psi_{i_{Nb}^b}(X)}{\partial X_j} \right); \Lambda_b = \{i_1^b, \dots, i_{Nb}^b\} \tag{3.29}$$

$$D^b = \begin{pmatrix} \Psi_{i_1^b}(X_{i_1^b}) & \dots & \Psi_{i_{Nb}^b}(X_{i_1^b}) \\ \vdots & \ddots & \vdots \\ \Psi_{i_1^b}(X_{i_{Nb}^b}) & \dots & \Psi_{i_{Nb}^b}(X_{i_{Nb}^b}) \end{pmatrix} \quad (3.30)$$

$$D^{nb} = \begin{pmatrix} \Psi_{i_1^{nb}}(X_{i_1^b}) & \dots & \Psi_{i_{Nb}^{nb}}(X_{i_1^b}) \\ \vdots & \ddots & \vdots \\ \Psi_{i_1^{nb}}(X_{i_{Nb}^b}) & \dots & \Psi_{i_{Nb}^{nb}}(X_{i_{Nb}^b}) \end{pmatrix} \quad (3.31)$$

By replacing 3.22 in equation 3.15 the discrete incremental equation is obtained as presented in next section.

3.5 Derivation of Discretized Equations

The displacement u is a linear combination of the modified kernel functions

$$\begin{aligned} u_i^h(X) &= \sum_{I \in \Lambda_{nb}} \Psi_I^{nb}(X) d_{iI} + \sum_{I \in \Lambda_b} \Psi_I^b(X) g_{iI} \\ \delta u_i^h(X) &= \sum_{I \in \Lambda_{nb}} \Psi_I^{nb}(X) d_{iI}^h \end{aligned} \quad (3.32)$$

$$\begin{aligned} u_{i,\bar{j}}^h(X) &= \sum_{I \in \Lambda_{nb}} \frac{\partial \Psi_I^{nb}(X)}{\partial X_j} d_{iI} + \sum_{I \in \Lambda_b} \frac{\partial \Psi_I^b(X)}{\partial X_j} g_{iI} \\ \delta u_{i,\bar{j}}^h(X) &= \sum_{I \in \Lambda_{nb}} \frac{\partial \Psi_I^{nb}(X)}{\partial X_j} d_{iI}^h \end{aligned} \quad (3.33)$$

$$\Delta u_{i,j}^h = \sum_{I \in \Lambda_{nb}} \frac{\partial \Psi_I^{nb}(X)}{\partial X_j} \Delta d_{iI} + \sum_{I \in \Lambda_b} \frac{\partial \Psi_I^b(X)}{\partial X_j} \Delta g_{iI} \quad (3.34)$$

By replacing the previous equations into 3.15 and rearranging terms we obtain

$$\begin{aligned} \sum_{M \in \Lambda_{nb}} d_{iM}^h \int_{\Omega_X} \frac{\partial \Psi_M^{nb}(X)}{\partial X_j} (D_{ijlm} + T_{ijlm}) \left(\sum_{N \in \Lambda_{nb}} \frac{\partial \Psi_N^{nb}(X)}{\partial X_m} \Delta d_{lN} + \sum_{I \in \Lambda_b} \frac{\partial \Psi_I^b(X)}{\partial X_m} \Delta g_{lI} \right) d\Omega & \quad (3.35) \\ \sum_{M \in \Lambda_{nb}} d_{iM}^h \left(\int_{\Omega_X} \Psi_M^{nb}(X) b_i(x(X)) J(X) d\Omega + \int_{\Gamma_X^{h_i}} \Psi_M^{nb}(X) h_i^0(X) d\Gamma - \right. \\ \left. \int_{\Omega_X} \frac{\partial \Psi_M^{nb}(X)}{\partial X_j} F_{ik} S_{kj} d\Omega \right) & \end{aligned}$$

The previous equation should hold for any d_{iM}^h therefore

$$\begin{aligned} \sum_{N \in \Lambda_{nb}} \int_{\Omega_X} \frac{\partial \Psi_M^{nb}(X)}{\partial X_j} (D_{ijlm} + T_{ijlm}) \frac{\partial \Psi_N^{nb}(X)}{\partial X_m} d\Omega \Delta d_{lN} = & \quad (3.36) \\ \int_{\Omega_X} \Psi_M^{nb}(X) b_i(x(X)) J(X) d\Omega + \int_{\Gamma_X^{h_i}} \Psi_M^{nb}(X) h_i^0(X) d\Gamma - \\ \int_{\Omega_X} \frac{\partial \Psi_M^{nb}(X)}{\partial X_j} F_{ik} S_{kj} d\Omega - \sum_{I \in \Lambda_b} \int_{\Omega_X} \frac{\partial \Psi_M^{nb}(X)}{\partial X_j} (D_{ijlm} + T_{ijlm}) \frac{\partial \Psi_I^b(X)}{\partial X_m} d\Omega \Delta g_{lI} & \end{aligned}$$

for $i = 1 \dots n_{sd}$; $M = i_1^{nb} \dots i_{Nnb}^{nb}$

This is a linear system of equations in Δd_{lN} . The system is converted in a matrix form as follows

$$K = K_{1\dots n_{sd} \times Nnb, 1\dots n_{sd} \times Nnb}; K_{MN} = K_{MN}^H + K_{MN}^G; \quad (3.37)$$

$$K_{MN}, K_{MN}^H, K_{MN}^G \in R^{n_{sd} \times n_{sd}}; M, N \in 1 \dots Nnb$$

$$(K_{MN}^H)_{il} = \int_{\Omega_X} \frac{\partial \Psi_M^{nb}}{\partial X_j}(X) D_{ijlm} \frac{\partial \Psi_N^{nb}(X)}{\partial X_m} d\Omega$$

$$(K_{MN}^G)_{il} = \int_{\Omega_X} \frac{\partial \Psi_M^{nb}}{\partial X_j}(X) T_{ijlm} \frac{\partial \Psi_N^{nb}}{\partial X_m} d\Omega$$

using a matrix notation the previous equations become

$$K_{MN}^G = \int_{\Omega_X} (B_M^F)^T \tilde{T} B_N^F d\Omega \quad (3.38)$$

$$\tilde{T} = \begin{pmatrix} (T_{1j1m})_{j,m=1\dots 3} & (T_{1j2m})_{j,m=1\dots 3} & (T_{1j3m})_{j,m=1\dots 3} \\ (T_{2j1m})_{j,m=1\dots 3} & (T_{2j2m})_{j,m=1\dots 3} & (T_{2j3m})_{j,m=1\dots 3} \\ (T_{3j1m})_{j,m=1\dots 3} & (T_{3j2m})_{j,m=1\dots 3} & (T_{3j3m})_{j,m=1\dots 3} \end{pmatrix}$$

$$B_K^F = \begin{pmatrix} \frac{\partial \Psi_K^{nb}}{\partial X_1} & \frac{\partial \Psi_K^{nb}}{\partial X_2} & \frac{\partial \Psi_K^{nb}}{\partial X_3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial \Psi_K^{nb}}{\partial X_1} & \frac{\partial \Psi_K^{nb}}{\partial X_2} & \frac{\partial \Psi_K^{nb}}{\partial X_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \Psi_K^{nb}}{\partial X_1} & \frac{\partial \Psi_K^{nb}}{\partial X_2} & \frac{\partial \Psi_K^{nb}}{\partial X_3} \end{pmatrix}^T$$

$$K_{MN}^H = \mathbf{I}_{3 \times 3} \int_{\Omega_X} \frac{\partial \Psi_M^{nb}}{\partial X_j}(X) S_{mj}(X) \frac{\partial \Psi_N^{nb}}{\partial X_m} d\Omega \quad (3.39)$$

$$K \Delta d = \Delta f \quad (3.40)$$

where

$$\Delta f = \left(\Delta f_{i_1}^{nb} \quad \dots \quad \Delta f_M \quad \dots \quad \Delta f_{i_{Nnb}}^{nb} \right)^T ; \Delta f_M \in R^{n_{sd} \times 1} \quad (3.41)$$

$$\begin{aligned} \Delta f_M = & \int_{\Omega_X} \Psi_M^{nb}(X) b_i(x(X)) J(X) d\Omega + \\ & \int_{\Gamma_X^{h_i}} \Psi_M^{nb}(X) h_i^0(X) d\Gamma - \int_{\Omega_X} \frac{\partial \Psi_M^{nb}}{\partial X_j}(X) F_{ik} S_{kj} d\Omega - \\ & \sum_{I \in \Lambda_b} \int_{\Omega_X} \frac{\partial \Psi_M^{nb}}{\partial X_j}(X) (D_{ijlm} + T_{ijlm}) \frac{\partial \Psi_I^b(X)}{\partial X_m} d\Omega \Delta g_{II}; i = 1 \dots 3 \end{aligned} \quad (3.42)$$

Note: There is no summation over i in equation 3.41. A solution is computed iteratively starting with an initial approximation. The equation 3.40 is solved and Δd is used to update the current deformation state through d . Using d the strain and stress are computed at the current step are computed and subsequently used to build the matrices for the next iteration.

3.6 Numerical Implementation

Using Gaussian quadrature rules, the problem domain is subdivided into integration zones that are defined independent of the RKPM particles as shown in Figure 3.3. The integration points are defined in each of these integration zones. Each integration point is covered by the supports of several shape functions. Figure 3.3 illustrates the integration point A covered by shape functions associated with particles I, J, and K. Let S^A represent a set of particles associated with integration point A.

Another integration method used in this thesis is the Collocated method developed in Chen (2002). In this case, the integration points and particles are located at the same position, unlike the particles in the Gaussian Quadrature method. Also, a strain smoothing

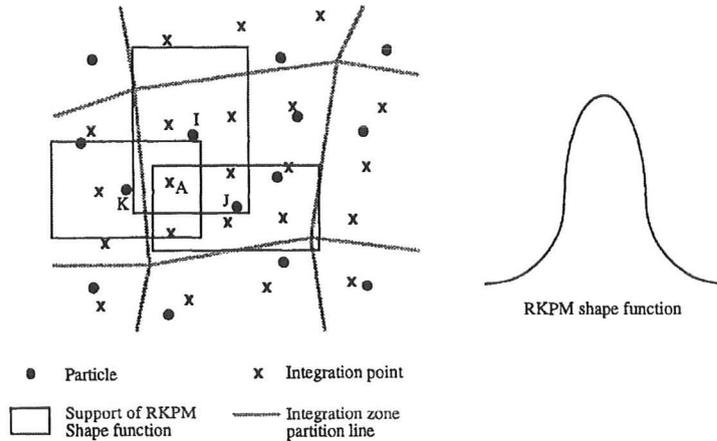


Figure 3.3: Graphical representation of RKPM discretization. Illustration obtained from Chen (1996).

that meets integration constraints is required to eliminate instabilities that results from the direct nodal integration of the weak form.

Next, the integration of the stiffness K matrix and the force f vector are computed in the algorithm described below:

The global stiffness matrix K is assembled using the corresponding nodal matrices, and the 2×2 nodal stiffness matrix $K_{MN} \neq 0$ only when the nodes I and J are covered by the support domain of at least one quadrature point. If nodes I and J are far apart and they do not share the same support domain of any quadrature point, K_{MN} becomes zero. Therefore, as long the support domain is compact and does not cover too widely the problem domain, many K_{MN} will be zero, and the global stiffness matrix K will be sparse.

Algorithm 1 Computation of stiffness matrix and force vector

```
for all Integration zones do
  for Each integration point  $X_A$  do
    Compute deformation gradient and 2nd Piola-Kirchhoff stress
    Form  $D(X_A)$  and  $T(X_A)$ 
    for Each particle M associated with current integration point do
      Form  $\Psi_M(X_A)$ ,  $B_M^G(X_A)$ , and  $B_M^H(X_A)$ 
      Assemble  $\delta f$ 
      for Each particle N associated with current integration point do
        Form  $\Psi_N(X_A)$ ,  $B_N^G(X_A)$ , and  $B_N^H(X_A)$ 
        Form  $K_{MN}^G$  and  $K_{MN}^H$ 
        Update K
      end for
    end for
  end for
end for
```

Chapter 4

Experimental Setup

4.1 Summary

The experimental system setup is an integration of several components. This includes a visual feedback system, a control object simulation system, a physical object, a method of calibration and a robotic manipulator. In this chapter, an overview of the system is provided as well as details of each system component.

4.2 System Setup

The overall system is based on position feedback, as shown in Figure 4.1, where J is the

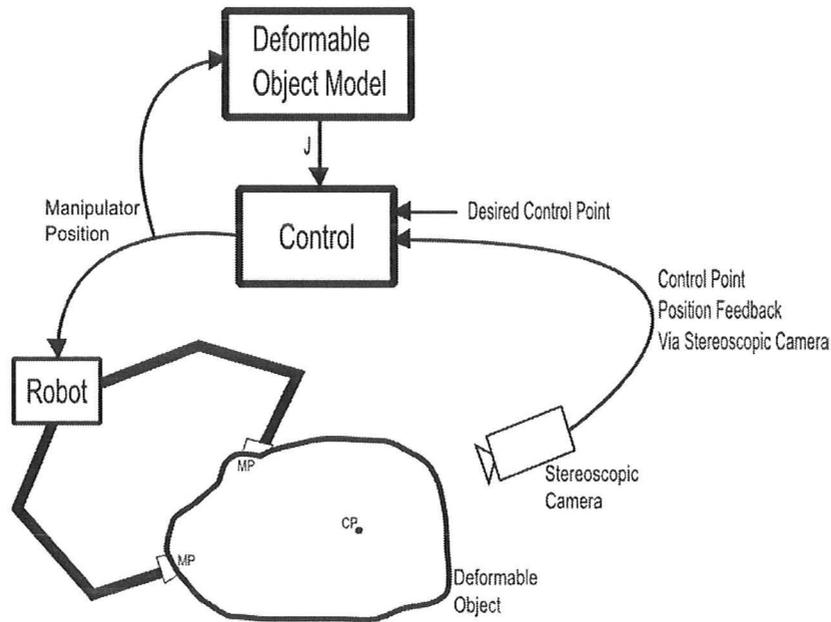


Figure 4.1: System overview for object control using a robotic manipulator, deformable object model and a stereoscopic camera.

Jacobian matrix, relating the manipulator-object interacting manipulation points MP and the control point CP on the object model. A physical object is equipped with markers tracked by a camera to record the object movement. These markers correspond to the control points chosen on the simulated object model. Relaying these marker positions back to the object model provides information to update the model and subsequently send information to the robotic manipulator to perform the next task on the physical object. The robotic manipulators act on manipulation points of the object boundary to indirectly maneuver the control points through object deformation. The camera detects new marker point positions, resulting in another system iteration.

The integrated system depicted in Figure 4.2, and its components in Figures 4.3 and 4.4, show the data flow for a system designed to validate the deformable object model using a physical object. All relevant specifications for the equipment used is provided in Appendix A.

The simulation of the object during system testing was performed on the same computer as that supporting the camera and the robot manipulator. An option was implemented into the system to allow the simulation of the object to be performed on a separate machine, while running in parallel with the camera and robot. This can be justified in case a better processor is used to improve model computations. The communication between the two computers was performed via TCP/IP.

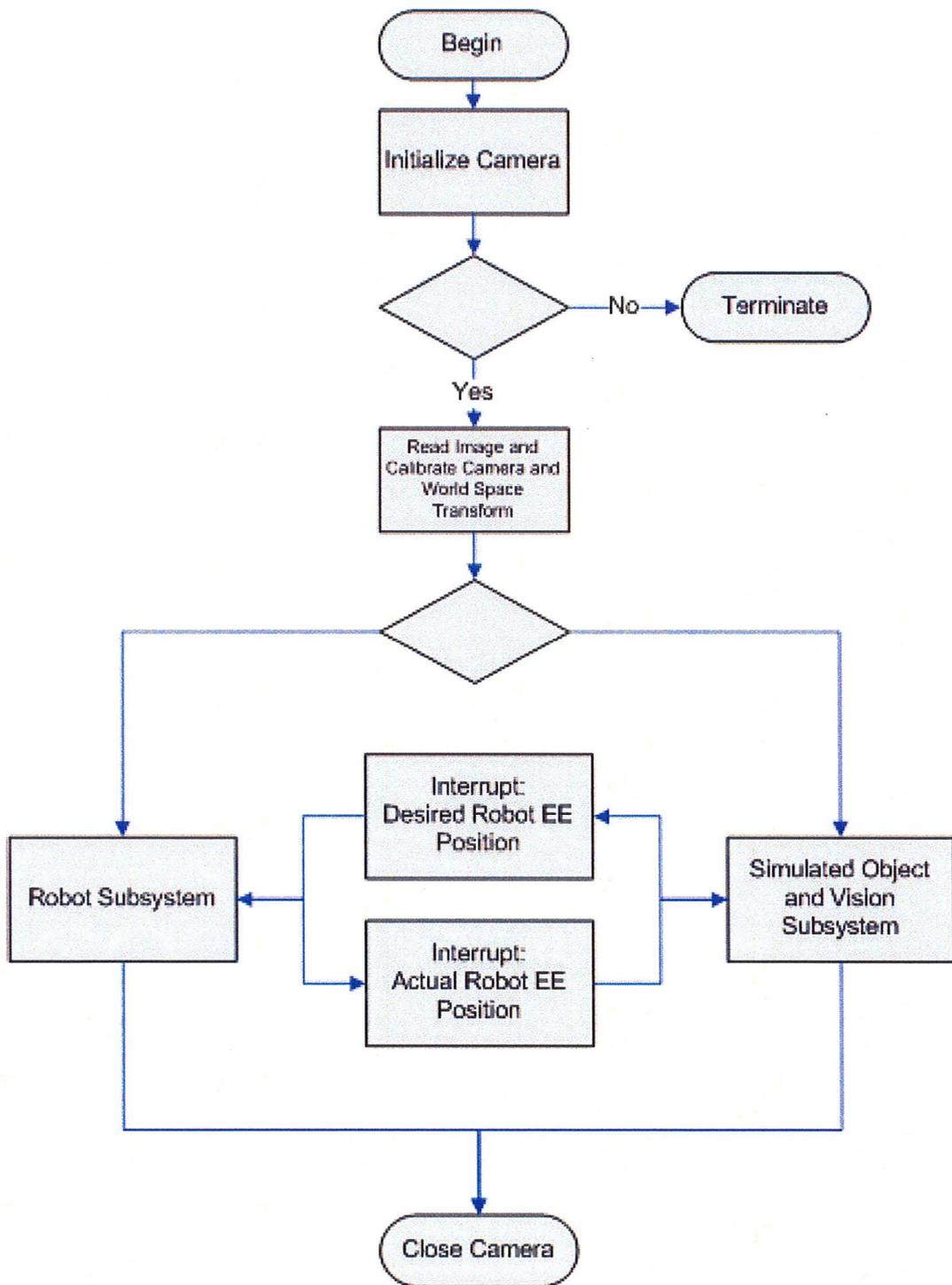


Figure 4.2: System overview showing the data flow sequence between the simulated object model, vision system and robotic manipulator.

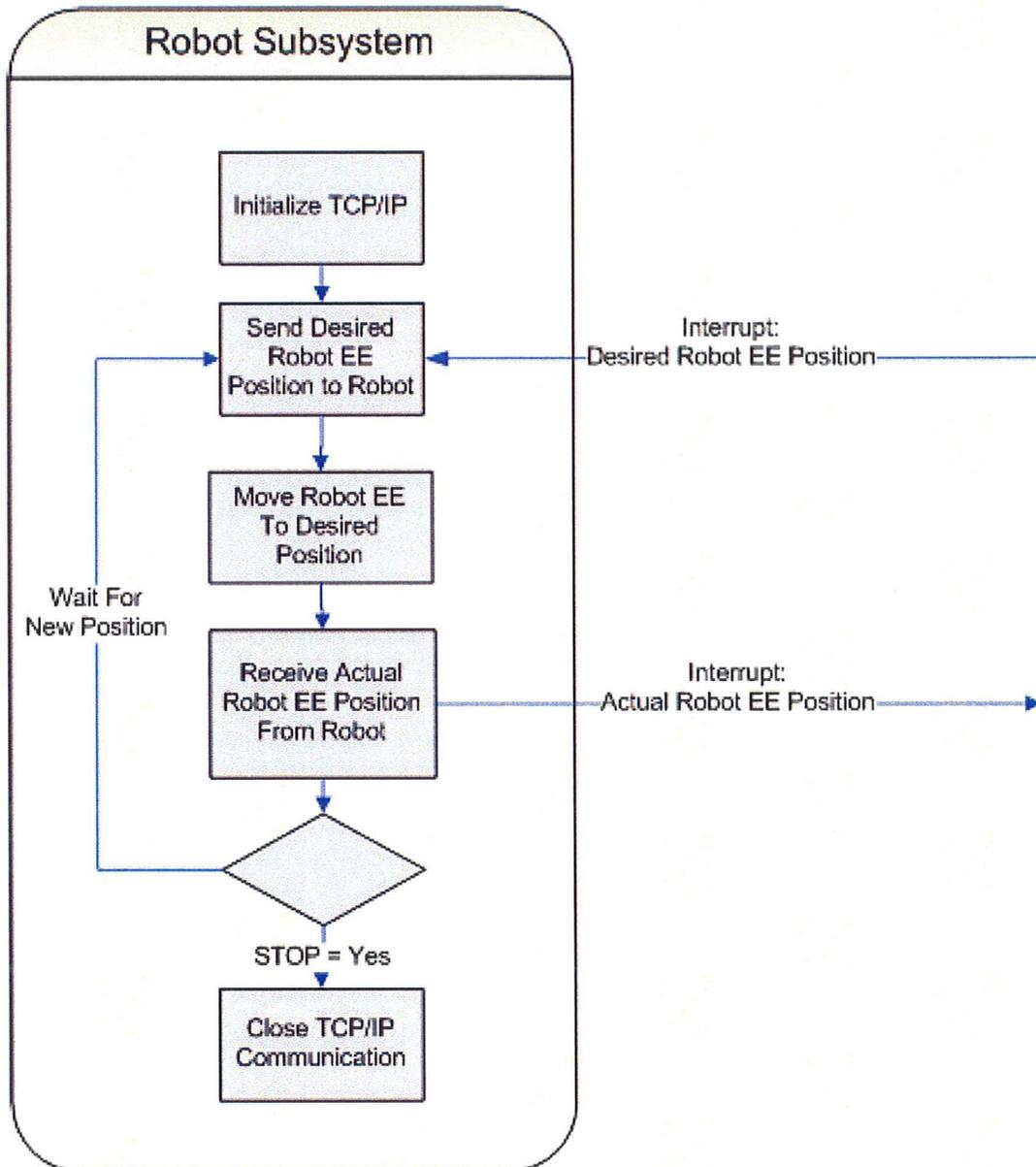


Figure 4.3: Section outlining the robotic manipulator commands to receive data from, actuate and send data to the object model.

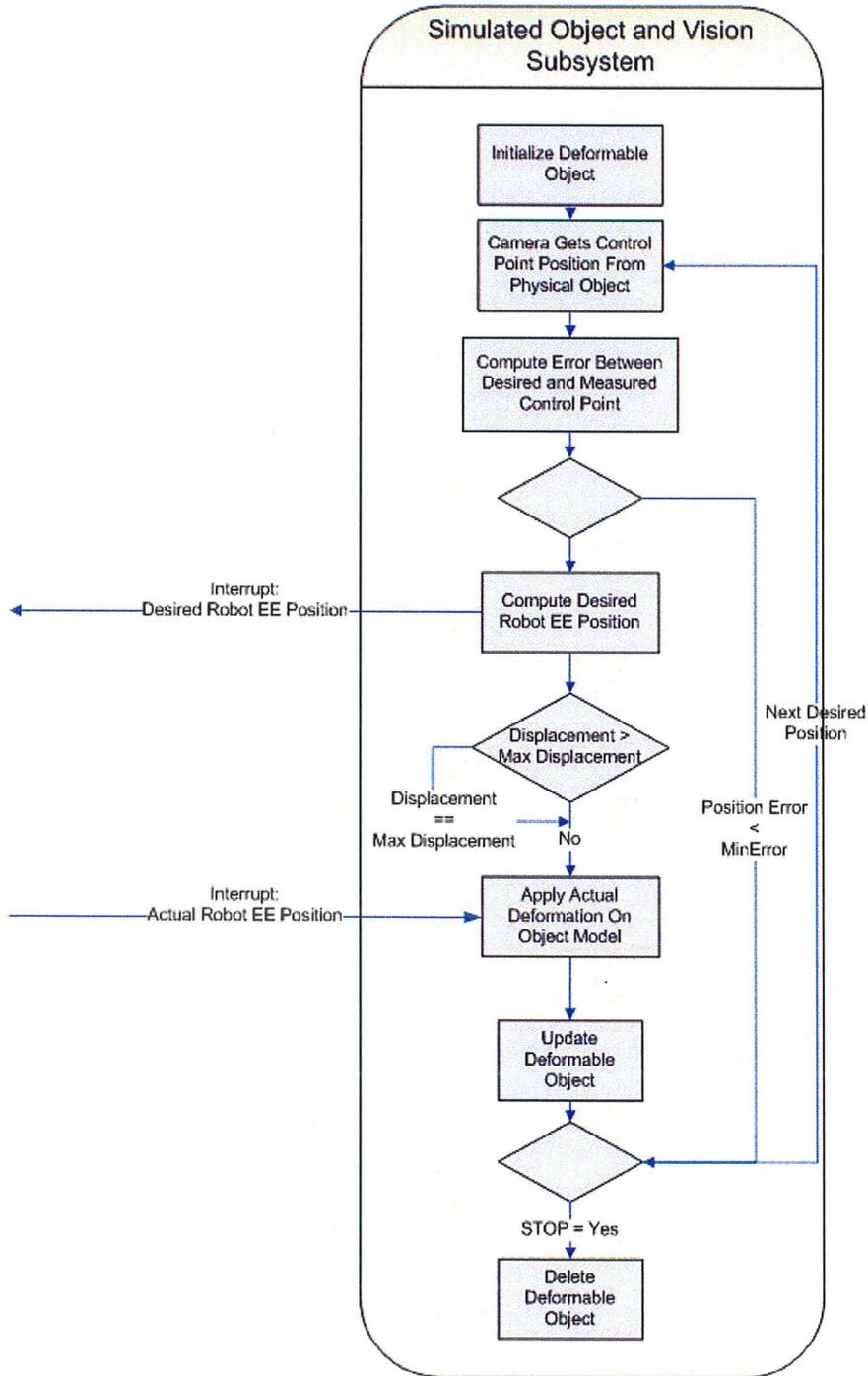


Figure 4.4: Section outlining the interaction between the vision system, object model and robotic manipulator.

4.2.1 Vision System

The vision system consists of a stereoscopic camera, strategically placed IR light emitting diodes (LEDs) and IR lens filters for the camera. The IR LEDs were used as the control points placed on the object, as well as a registration marker for the cameras calibration system. The use of a stereoscopic camera in the sequence shown in Figure 4.5, allowed for accurate 3D positioning of these markers.

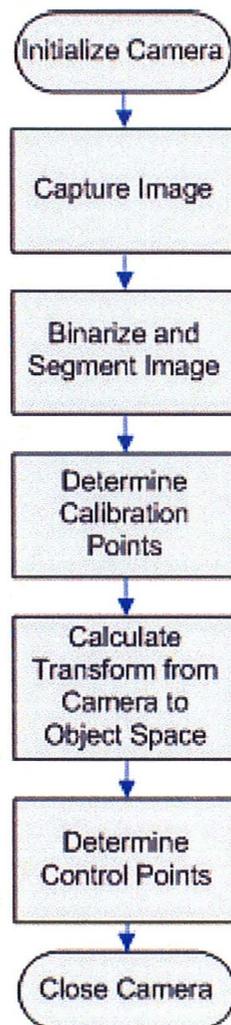


Figure 4.5: Information flow for acquiring an image using the stereoscopic camera.

Once the camera is initialized, images are acquired at a rate which is dependant on the speed of the attaining the object solution for a particular deformation. The images from the left and right cameras are binarized and segmented into distinct entities to filter out all image blobs other than the control and calibration markers. The 3D position of these markers is computed using image disparity, or the difference between the left and right images. The disparity calculation algorithm is described below:

Algorithm 2 Disparity Calculation in Stereo Camera

```

procedure DISPARITY
  for  $i = 1 \dots N$  do
    Find Center of  $X_{LeftImage}^i$  and  $Y_{LeftImage}^i$ 
    Find Center of  $X_{RightImage}^i$  and  $Y_{RightImage}^i$ 
  end for
  for  $i = 1 \dots N$  do
    Find  $\min |Y_{LeftImage}^i - Y_{RightImage}^i|$ 
     $PositionIndex = i$ 
  end for
  Calculate Image Horizontal Disparity:  $Disparity = X_{LeftImage}^{PositionIndex} - X_{RightImage}^{PositionIndex}$ 
end procedure

```

- $N \rightarrow$ Number of segmented regions
- $X \rightarrow$ X coordinate position of segmented region
- $Y \rightarrow$ Y coordinate position of segmented region

To illustrate this, Figure 4.6 shows an image taken from a stereoscopic camera. The segmented regions are shifted by a certain distance, the disparity, in the two images. In the three dimensional case, a larger disparity is interpreted as a larger separation between the two corresponding points. This would mean that point is a shorter distance away from the cameras.

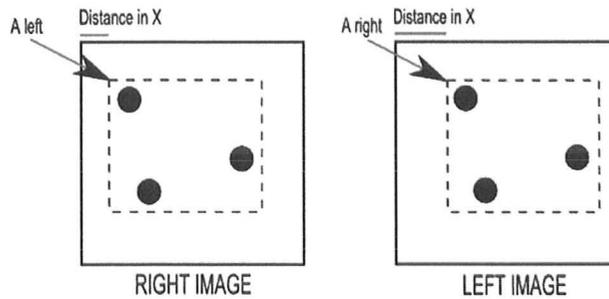


Figure 4.6: Example of the left and right images obtained from a stereoscopic camera of the same environment.

To determine the location of a point in space, a triangulation method is employed as shown in the camera representation in Figure 4.7. Here, M and N are shown as two pinhole cameras with a parallel optical axis, a baseline of b and a focal length of f . The baseline is perpendicular to the optical axes. The optical axes lie in the XZ plane and the XY plane is parallel to the image plane of both cameras. Note that the reference frame has an origin at the center of the left camera. Therefore to calculate the space position of point P the following equations are used. Let W_p represent the width between pixels.

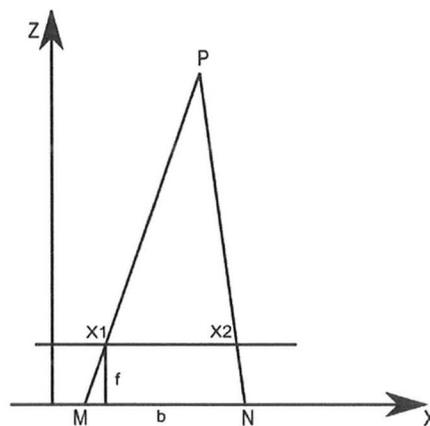


Figure 4.7: Illustrating the geometry of a stereoscopic camera.

$$Z = \frac{b \times f}{(x_1 - x_2) \times W_p} \quad (4.1)$$

$$X = x_1 \times \frac{Z}{f} \quad (4.2)$$

$$Y = y_1 \times \frac{Z}{f} \quad (4.3)$$

4.2.2 System Coordinate Transformation

In order to correctly manipulate the object with the robotic tool and track the object movements with the camera, the system must be set to operate in one coordinate system. To illustrate this, Figure 4.8 shows three independent coordinate system, including the robotic tool tip, the camera and the object.

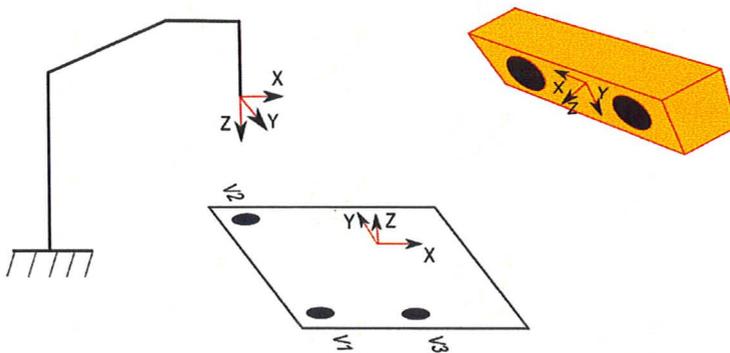


Figure 4.8: Illustration of coordinate systems belonging to the camera, tool and object.

Camera Space to Object Space Transformation

To transform the coordinate frame of the camera to the object coordinate frame, three independent markers, set in an L shape are used as shown in Figure 4.8. The lengths are

$(V_2 - V_0) = 2 \times (V_1 - V_0)$. The markers are always labels as shown, such that a basis vectors can be created as follows:

$$V_x = ||V_1 - V_0|| \quad (4.4)$$

$$V_y = ||V_2 - V_0|| \quad (4.5)$$

$$V_z = ||V_x \times V_y|| \quad (4.6)$$

The frame rotation between the camera and object coordinate system is:

$$Rotation(V_x, V_y, V_z) = \begin{pmatrix} V_{x0} & V_{x1} & V_{x2} \\ V_{y0} & V_{y1} & V_{y2} \\ V_{z0} & V_{z1} & V_{z2} \end{pmatrix} \quad (4.7)$$

and the translation is calculated as:

$$V_{Translation} = -Rotation(V_x, V_y, V_z) * V_0 \quad (4.8)$$

Finally, the world points for the entire system are calculated as:

$$ControlPoint_{World} = Rotation * (V_x, V_y, V_z) * ControlPoint_{Image} + V_{Translation} \quad (4.9)$$

Tool Localization

The robotic tool consisted of a 1/4 inch in diameter aluminum rod, approximately 8 inches in length. This rod was firmly attached end point of the robotic arm. In order to calculate the exact position of the tool tip relative to the robot's world coordinate frame, the tip position was located using the pivoting point calibration method. The general problem is depicted in Figure 4.9. A point is chosen arbitrarily in space at which the tip of the tool

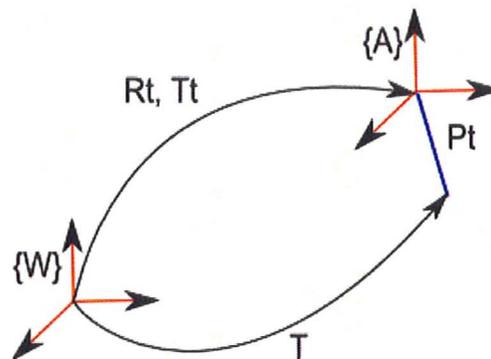


Figure 4.9: Tool tip localization.

will be rotated about. In this way, the rotation matrix R_t and translation vector T_t of the robot arm to which the tool is attached can be recorded from the robotic integrated sensors at its base, or the world frame labeled W , as shown in Figure 4.9. For all different recorded orientations of the tool tip, the translation vector T , between the world and tool tip frames, always remains the same. The objective now becomes to solve for the tool tip position P_t . Eight different orientations were recorded and used to locate the tool tip.

The calibration was performed by recording several different orientation values of the

arm around the tool tip point.

$$\begin{pmatrix} R_t & T_t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P_t \\ 1 \end{pmatrix} = \begin{pmatrix} T \\ 1 \end{pmatrix} \quad (4.10)$$

$$\begin{pmatrix} R_{t1} & T_{t1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P_{t1} \\ 1 \end{pmatrix} = \begin{pmatrix} T \\ 1 \end{pmatrix} \quad (4.11)$$

$$\begin{pmatrix} R_{t2} & T_{t2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P_{t2} \\ 1 \end{pmatrix} = \begin{pmatrix} T \\ 1 \end{pmatrix} \quad (4.12)$$

$$\begin{pmatrix} R_{tn} & T_{tn} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P_{tn} \\ 1 \end{pmatrix} = \begin{pmatrix} T \\ 1 \end{pmatrix} \quad (4.13)$$

It should be noted that using eight different orientations was sufficient to accurately locate the tool tip. From these measurements, an average rotation matrix and translation vector was calculated, where

$$R_{AVG} = \sum_{i_1}^n R_i \quad (4.14)$$

$$T_{AVG} = \sum_{i_1}^n T_i \quad (4.15)$$

Therefore, if the recorded values are compared against the average values it becomes

$$T = R_{AVG}P_{AVG} + T_{AVG} \quad (4.16)$$

$$\underline{-T = R_i P_i + T_i} \quad (4.17)$$

$$0 = (R_{AVG} - R_i)P + T_{AVG} - T_i \quad (4.18)$$

$$(R_{AVG} - R_i)P = -T_{AVG} + T_i \quad (4.19)$$

and the vector P_i can be solved for $Ax = b$

$$A(i) = R_{AVG} - R_i \quad (4.20)$$

$$b(i) = -T_{AVG} + T_i \quad (4.21)$$

4.2.3 Robot to Object Space

In order for the robot to move properly in the object space, a transform between the robot and object coordinate systems was required. The least squares estimation of the transformations between two sets of points Arun (1987) was used. The ultimate goal was to find a form such as $y_i = Rx_i + T$.

To obtain the set of matching points, the tool tip was recorded at three designated points on the object. Which then could be assembled into a matrix form

$$Y = \begin{pmatrix} Y_{x0} & Y_{x1} & Y_{x2} \\ Y_{y0} & Y_{y1} & Y_{y2} \\ Y_{z0} & Y_{z1} & Y_{z2} \end{pmatrix} \quad (4.22)$$

recorded in tool coordinates. Since the object was planar, the Z component in object space was assumed zero. The three points can be chosen arbitrarily, as long as they are known in object space. In this case, the calibration points were used as the reference points. These included the origin and one point strictly a distance in the X axis and one point a distance in the Y axis.

To more accurately obtain these points in object coordinates, the object X axis point

was set to equal

$$\|Y(:,x2) - Y(:,x0)\| \quad (4.23)$$

and the object Y axis point was set to equal

$$\|Y(:,x1) - Y(:,x0)\| \quad (4.24)$$

to obtain

$$X = \begin{pmatrix} 0 & 0 & \|Y(:,x2) - Y(:,x0)\| \\ 0 & \|Y(:,x1) - Y(:,x0)\| & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (4.25)$$

where the a and b in $Y(a,b)$ represent the row and column of matrix 4.22.

4.2.4 Physical Object

The physical object was made from a mixture of $\frac{1}{2}$ super soft mixture and $\frac{1}{2}$ plastic softener from M-F Manufacturing Co., Inc. See appendix for complete information about company. The object was created to fit the dimensions of $6 \times 6 \times \frac{1}{2}$ inches or $0.1524 \times 0.1524 \times 0.0127$ meters.

4.2.5 Robot and Platform System

The system platform used is composed of several components. The base on which the object lays on was manufactured using a $\frac{1}{4}$ inch aluminium plate with dimensions 12×12 inches. The plate included 4 $\frac{1}{2}$ through holes on the corners for base support and an array of 11×11 threaded holes separated by 1 inches increments in each direction. The tool is simply a 8 inch $\frac{1}{4}$ inch aluminium rod, which is attached to the robotic arm force sensor at

two positions. Further, the tool is situated in a predefined position in the object, where it is assumed that the contact between tool and the object is non-slippery. To prevent friction between the object and the base plate, the object is situated on top of many small balls that allow the object to move freely in the plane. For this simple Copper coated BB's were used. Next, to simulate an restricted boundary, one entire side of the object was screwed down onto the base plate using five screws and a piece of sheet metal to distribute the force.

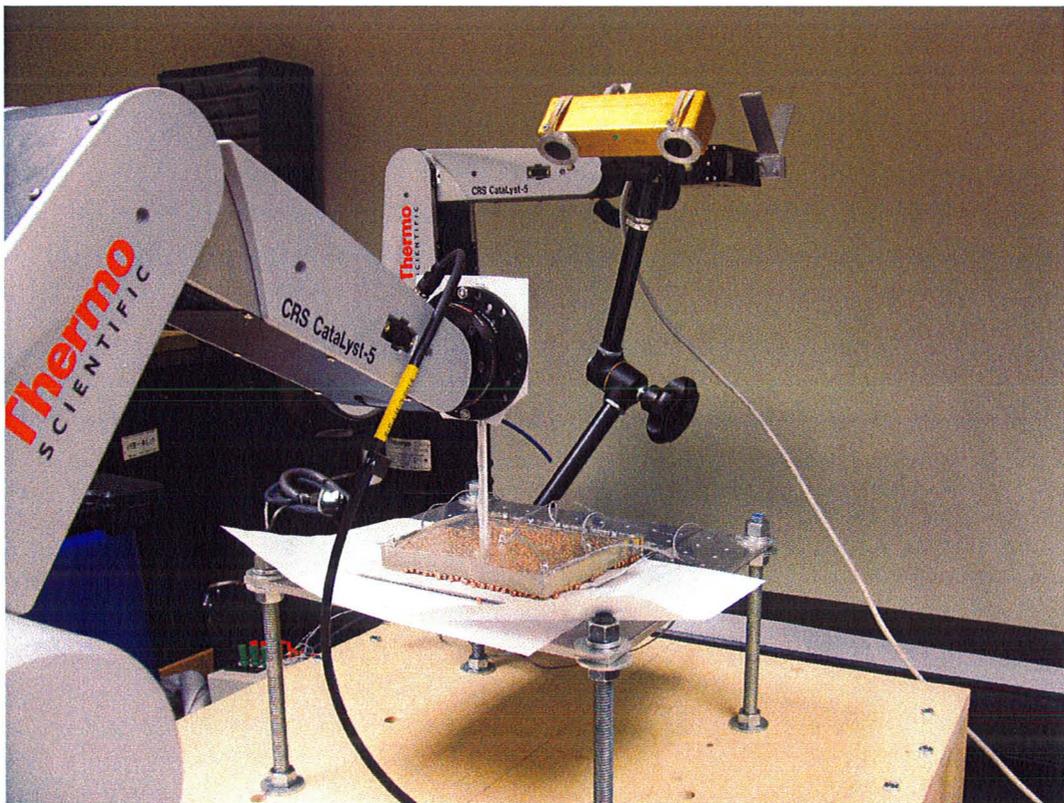


Figure 4.10: Image guided system platform set up.

Figure 4.10 depicts the tool attached to the robotic arm deforming the physical object, with the opposite side made immobile and Figure 4.11 shows the deformed object as a result of tool manipulation. Also, it shows the stereoscopic camera tracking the four LEDs situated on the object, and the object itself on top of a frictionless surface as described

above. The tool and the base platform are covered by white paper as to reduce interfering glare points coming from the base and tool smooth surfaces.

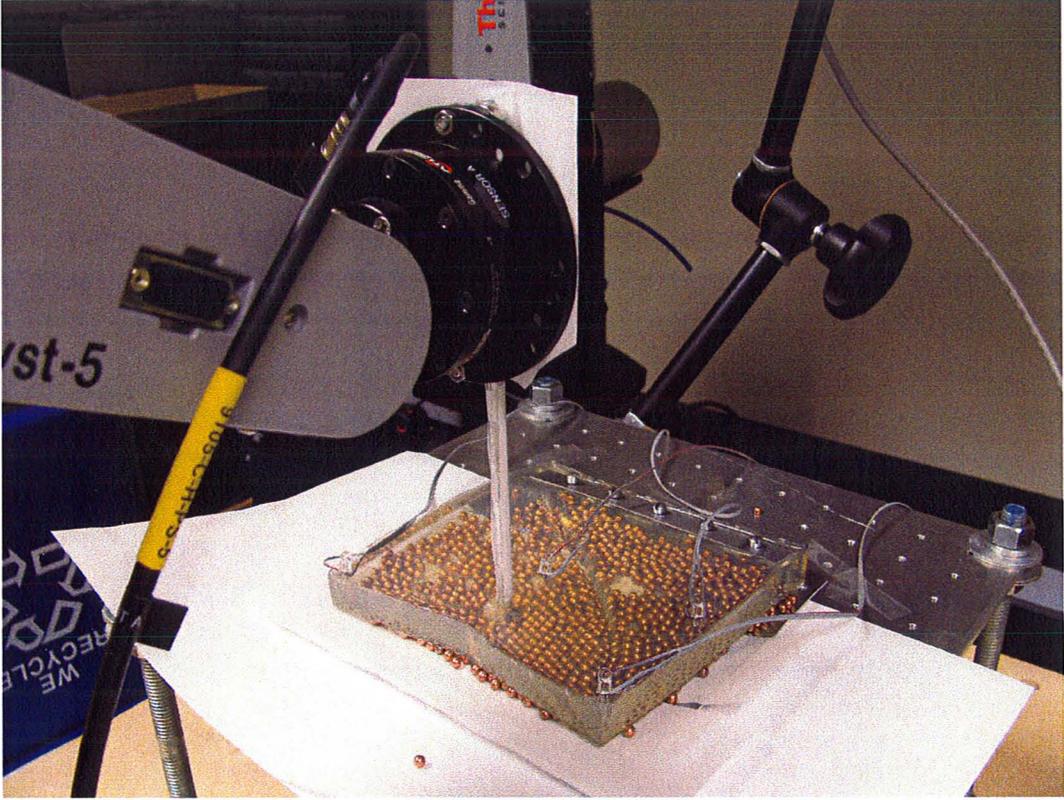


Figure 4.11: Tool manipulating object model.

Chapter 5

Deformation Planning for Robotic Soft Object Manipulation

5.1 Summary

Modeling the deformation response is an integral part for accurate robotic manipulations of deformable objects. This is exemplified in recent developments in surgical planning applications, which require reliable methods to perform functions such as surgical incisions or controlled needle insertions for biopsy procedures. Achieving physically realistic replicas of soft tissues or objects, for such and other applications is a difficult and computationally intensive task. As a result, accurate planning for surgical procedures employing automatic manipulation requires models that closely mimic the characteristics of the operating environment.

This chapter outlines a model based deformation control method for soft object manipulation path planning. The deformable object is manipulated through boundary displacements induced by robot manipulators using position control. The manipulated boundaries

are maneuvered such that the control points defined on the deformable object converge to their desired locations. The proposed control method is based on a Jacobian transformation between a set of manipulated point displacements and a set of control point displacements. The deformable object model is based on nonlinear elasticity, which allows the model to handle large deformations and nonlinear materials characteristics. Simulated results showing the tracking of a control path are presented for both Collocated and Gaussian Quadrature integration rules. Similar experiments are performed on a physical objects and compared to simulated model results.

5.2 Object Deformation Control Method

The deformable model connects the control point displacements to the manipulation point displacements, from which a control law is derived using a Jacobian matrix. A robot end-effector is modeled as an induced displacement on these manipulation points. The deformation planning algorithm is summarized in the algorithm below: The variables used are

Algorithm 3 Tracking Control Point Positions

```

procedure TRACKING CPP
  for  $CPP_i^d \dots CPP_N^d$  do
    repeat
       $e = CPP_i^d - CPP^c$ 
      Compute J
       $MPP^c = MPP^c + KJ^{-1}e$ 
    until  $\|e\|_2 < e_{max}$ 
  end for
end procedure

```

- $CPP_i^d \rightarrow$ Desired control point position vector
- $CPP^c \rightarrow$ Current control point position vector

- $MPP^c \rightarrow$ Current manipulation point position vector
- $e \rightarrow$ Current control point position error
- $N \rightarrow$ Number of desired control point positions
- $K \rightarrow$ Gain matrix
- $e_{max} \rightarrow$ Convergence condition. The tracking algorithm does not move to the next control points position until the current position is within a radius e_{max} from the current desired position.

Note that the data and algorithm from this point outline the results for a single group of manipulation points and a single control point in the model.

Assuming that the number of degrees of freedom of control points (N_D) are equal with the number of degrees of freedom of the manipulated points, the Jacobian is a square matrix and can be inverted if it is nonsingular. The Jacobian is estimated using an incremental equation described next.

The displacements of the control particles and the displacements of the manipulated particles are stacked respectively in arrays $CPA \in \mathbf{R}^{N_D \times 1}$ and $MPA \in \mathbf{R}^{N_D \times 1}$. The Jacobian is estimated by inducing an ε displacement in every dimensional direction of MPA and solving for CPA . This procedure is summarized in the algorithm below:

- $\varepsilon \rightarrow$ Unit increment set for the robot manipulator
- $N_D \rightarrow$ Dimension of the object

Algorithm 4 Jacobian Estimation

procedure COMPUTE J

Save current object configuration

for $i = 1 \dots N_D$ **do** $MPA = 0_{N_D \times 1}$ $MPA_i = \epsilon_i$ *Solve Equation 3.40 with MPA as displacements and retrieve CPA* $J(:, i) = CPA / \epsilon_i$ **end for**

Restore object configuration

end procedure

5.3 Results

The results below present an evaluation of the deformable object model for object path planning. Computer simulations followed by experiments performed on a physical object are provided to help with model validation. No modification were performed on the experimental setup described in chapter 4. The subsequent results use only one control and one manipulation point to show the method in practice.

5.3.1 Preliminary Object Model Simulations

A validation study was performed to show that the control of a physical object using the model could be successful. Initial results begin with a virtual system set up to perform path tracking on a 3 dimensional object. To mimic a physical system, a control scheme was developed involving a reduced and refined model. The idea was to use a reduced model, with fewer particles, on which to perform control calculations and then actuate the control commands on a more refined model. Proving this idea would show that an object model with finite particles could be used to accurately control any object.

To show this, two simulations were performed following a similar path for comparison,

using the Gaussian Quadrature integration methods. In the first simulation, a single control point in the reduced model was controlled to follow a desired path with the use of a specified manipulator paddle. The paddle specifies an area on the object which is in direct contact with the robot end-effector and uniformly moves all object particles attached to it. In the case of the reduced model, the paddle interacted with bottom 2/3 of the object face area at its original position $x = -2.5$. Figure 5.1, illustrates the object interaction with the

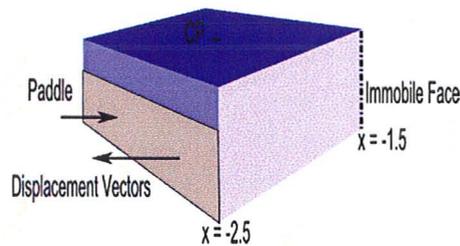


Figure 5.1: Paddle acts on the bottom 2/3 of object face at position $x = -2.5$, and the entire back face is immobile at position $x = -1.5$.

paddle. The entire opposite side of the object was firmly attached to an immobile surface at $x = -1.5$. The desired path for the control point, as listed in Table 5.1, starts with the

<i>Step</i>	<i>X</i>	<i>Y</i>	<i>Z</i>
1	-2.0	0.0	2.5
2	-2.0	0.0	2.7
3	-2.2	0.0	2.7
4	-2.3	0.0	2.9
5	-2.5	0.0	2.9
6	-2.6	0.0	3.0

Table 5.1: Desired path for object control point in simulated system.

control point resting at its initial position in the middle of the top surface of the object at $(-2.0, 0.0, 2.5)$. It should be noted that all values are based on object defined coordinates. In this case, a value of 0.01 indicates an increment of 1% on the object. The reduced

object is discretized into three symmetric layers of 3 by 3 particle arrays. To display the effectiveness of the Jacobian control law, Figure 5.2 illustrates the simulation results for

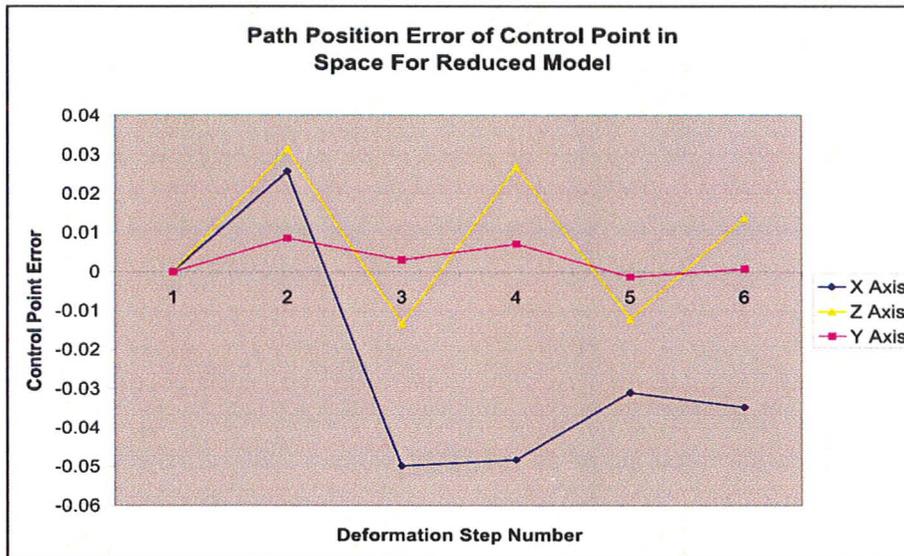


Figure 5.2: Each step of the reduced model corresponds to a deviation from that of the desired control point path in Table 1.

control point position deviation from the desired path plan.

In a graphical representation, the objects are separated into horizontal planes corresponding to the original symmetric layers of points formed along the z axis. For each simulated step, individual points are moved accordingly, which collectively change the shape of the planes, as to move the control point along the desired path. This is seen in the progression of Figures 5.3 through 5.5. Noticeably, the manipulated face of the object was pulled in the $-x$ and $+z$ axes to accommodate the movement of the labeled control point on the top face, while the immobile face is fixed at its original position. Also, the variable color scheme depicted on each plane is a function of amplitude, changing as the paddle maneuvers the manipulation points.

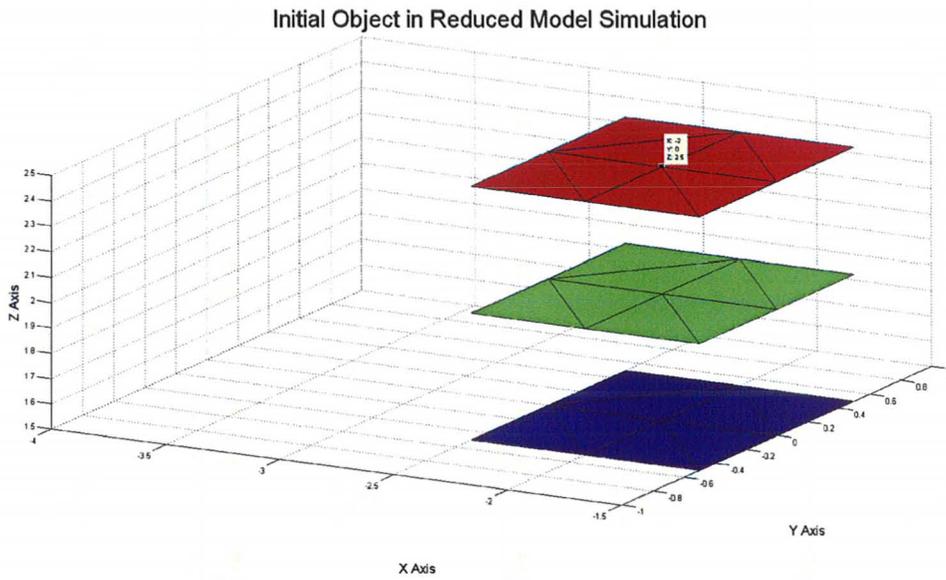


Figure 5.3: Deformation step 1 in Table 5.1 of reduced model following control point trajectory.

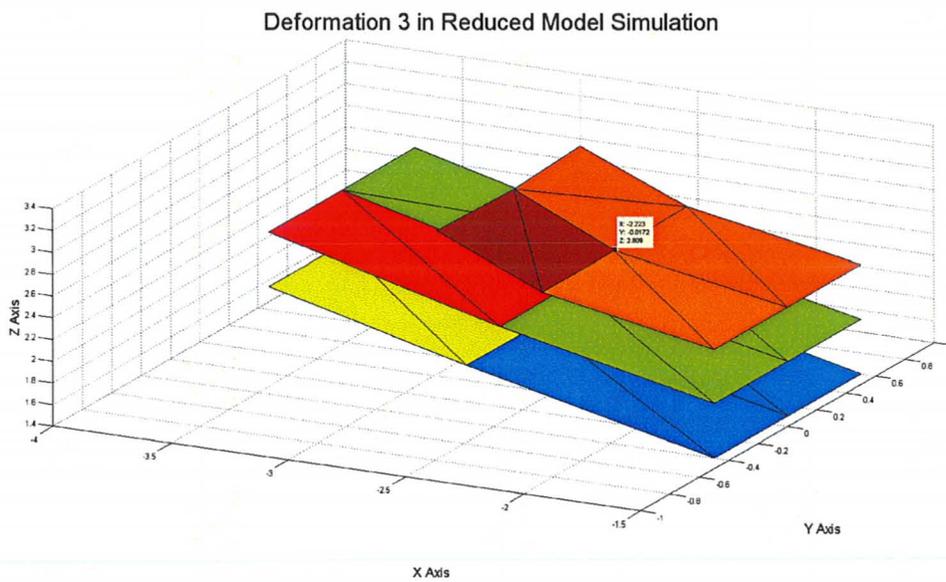


Figure 5.4: Deformation step 3 in Table 5.1 of reduced model following control point trajectory.

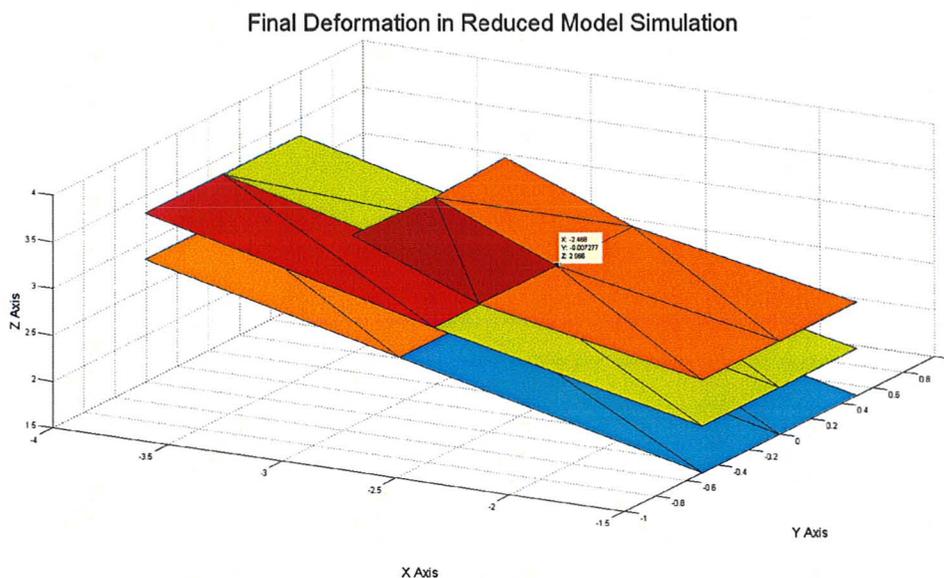


Figure 5.5: Deformation step 6 in Table 5.1 of reduced model following control point trajectory.

Similarly in the next case the same control path and point are used as above. The refined model contains particles half the step size apart as in the reduced model. Therefore, there are 125 particles in this model discretized into 5 symmetric layers of 5 by 5 arrays. Again, the error in path position is presented in Figure 5.6, showing the control point deviations in the more refined model.

Following Figures 5.7 and 5.8 depict the progression of object deformations. It is seen that the entire face of the object is grounded at $x = -1.5$ and that the paddle maneuvers the whole opposite face as depicted in Figure 5.9. The paddle manipulates the interacting face in the $-x$ and $+z$ axes to accommodate the control point desired path.

The computed Jacobian path planning algorithm provided accurate tracking of the control point position. The maximum absolute errors recorded during the simulations were 0.090322 for the refined model object, and 0.049859 for the reduced model object. For the

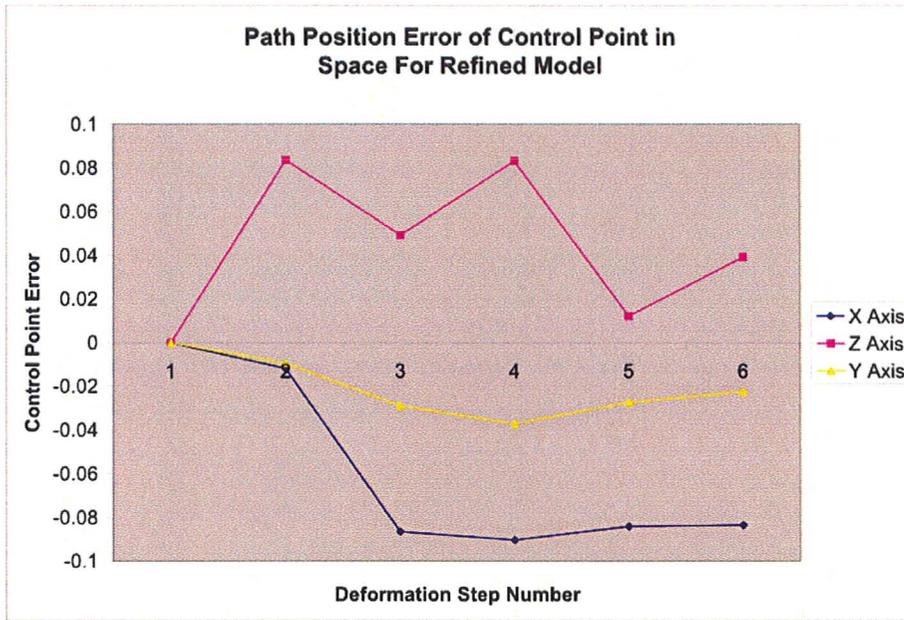


Figure 5.6: Each step of the refined model corresponds to a deviation from that of the desired control point path in Table 5.1.

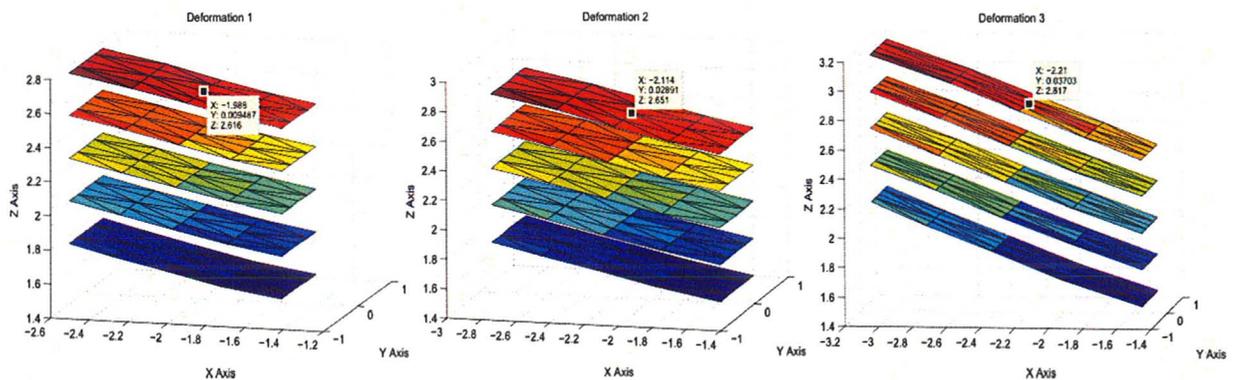


Figure 5.7: Deformation steps 1, 2 and 3 in Table 5.1 of refined model following control point trajectory.

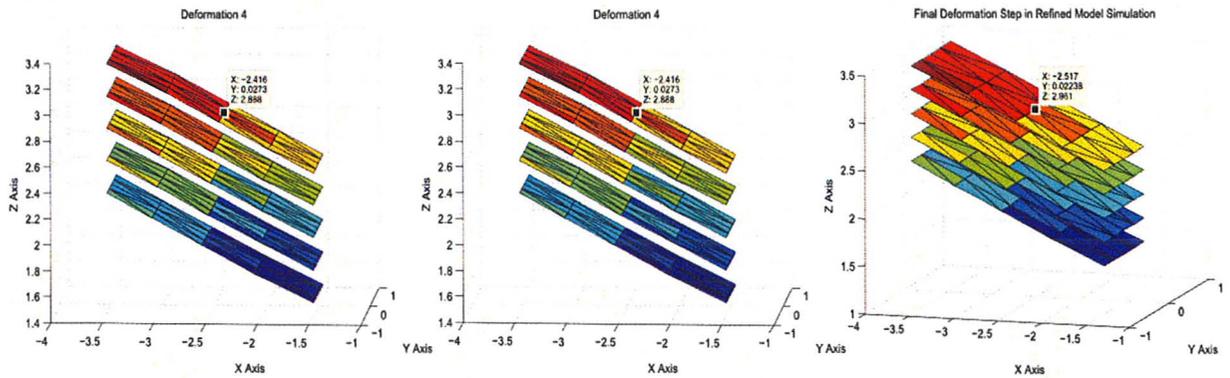


Figure 5.8: Deformation steps 4, 5 and 6 in Table 5.1 of refined model following control point trajectory.

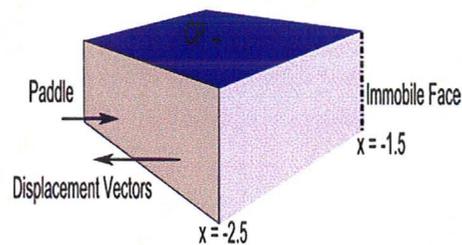


Figure 5.9: Paddle actuates on entire object face at position $x = -2.5$ and entire back face is immobile at position $x = -1.5$.

above simulations, the Strain Energy Density Function used was

$$W = \frac{1}{2}\alpha E^T E \quad (5.1)$$

where α is a positive constant. These successful results allowed for a physical validation study to be carried out.

5.3.2 Physical Validation Study

Physical validation of the model path planning method was performed on a planar physical object, created from a soft plastic. Object simulations were carried out on 9×9 particle objects, using both the Collocated and Gaussian Quadrature integration rules. Both were simulated in order to compare their accuracy, amount of deformation handling and run time. It is shown below that the accuracy of both methods is comparable for smaller deformations. The ability to handle larger deformations is largely favored by using Gaussian Quadrature. Although, its run time compared to the Collocated integration rules is drastically larger. The long run time makes the use of the Gaussian Quadrature method for this application unfeasible, given its current computation method. Below are the results from both simulated methods. Based on these results only the Collocated integration rules was integrated into the physical validation system due to simulation time requirements. For the solutions obtained below Fung's Strain Energy function is used.

$$W = \alpha e^{E_{ij}C_{ijkl}E_{kl}} + \beta E_{ij}A_{ijkl}E_{kl} \quad (5.2)$$

was employed help define the object characteristics, where α and β are positive constants, E is the strain tensor represented as a vector and A and C are symmetric second-rank

tensors. The values employed for A and C are listed in table 6.1 determined through tissue measurement in Cowin (2006).

A_{ijkl}	Value	C_{ijkl}	Value
$A(0,0,0,0)$	1020	$C(0,0,0,0)$	3.5
$A(1,1,1,1)$	1020	$C(1,1,1,1)$	3.5
$A(0,1,0,1)$	254	$C(0,1,0,1)$	0.5
$A(1,0,1,0)$	254	$C(1,0,1,0)$	0.5
$A(0,0,1,1)$	383	$C(0,0,1,1)$	1.5
$A(1,1,0,0)$	383	$C(1,1,0,0)$	1.5
$A(0,1,1,0)$	383	$C(0,1,1,0)$	1.5
$A(1,0,0,1)$	383	$C(1,0,0,1)$	1.5

Table 5.2: Values for A and C tensors. All values not indicated are zero. Obtained from Chapter 15 Tissue Mechanics Cowin (2006).

Comparison of Gauss Quadrature and Collocated Integration Rules

Initially to show the ability to handle large deformations, the results using the Gaussian Quadrature and Collocated integrations rules are compared in a compression and shear compression deformation simulation. In both cases the models were compressed until object model failure occurred. The manipulator size included three boundary point which were uniformly displaced at equal intervals of 0.01 or 1% in the X axis, followed by a shear compression in the X and Y axis using equivalent intervals. Results for the Gaussian Quadrature Methods are shown in Figures 5.10 through 5.12 for straight compression and Figures 5.13 through 5.16 for the shear compression. Notice the model begins to breakdown at 50% for the shear compression and at 43% for the straight compression.

As a comparison, similar simulations were performed using the Collocated model, shown in Figure 5.17 for straight compression and Figure 5.18 for shear compression. The simulations break down at 15% and 14% respectively.

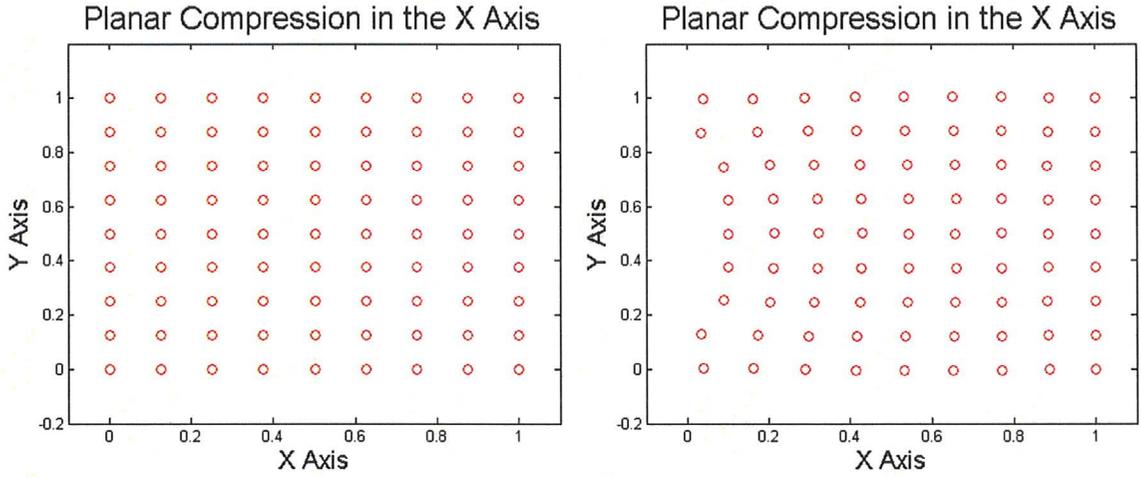


Figure 5.10: Compression in the X Axis by displacing the essential boundary by 0%, 10% respectively for Gaussian Quadrature method.

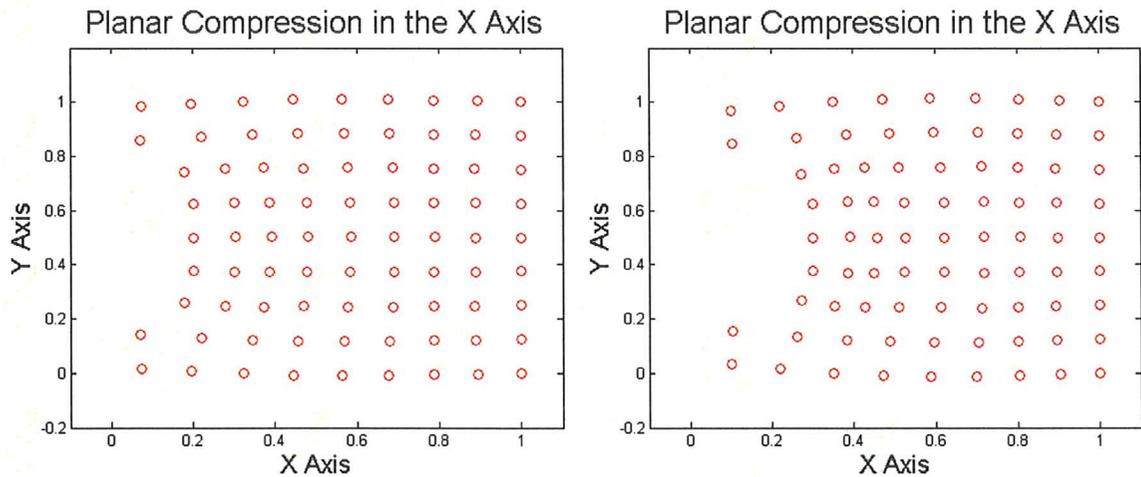


Figure 5.11: Compression in the X Axis by displacing the essential boundary by 20%, 30% respectively for Gaussian Quadrature method.

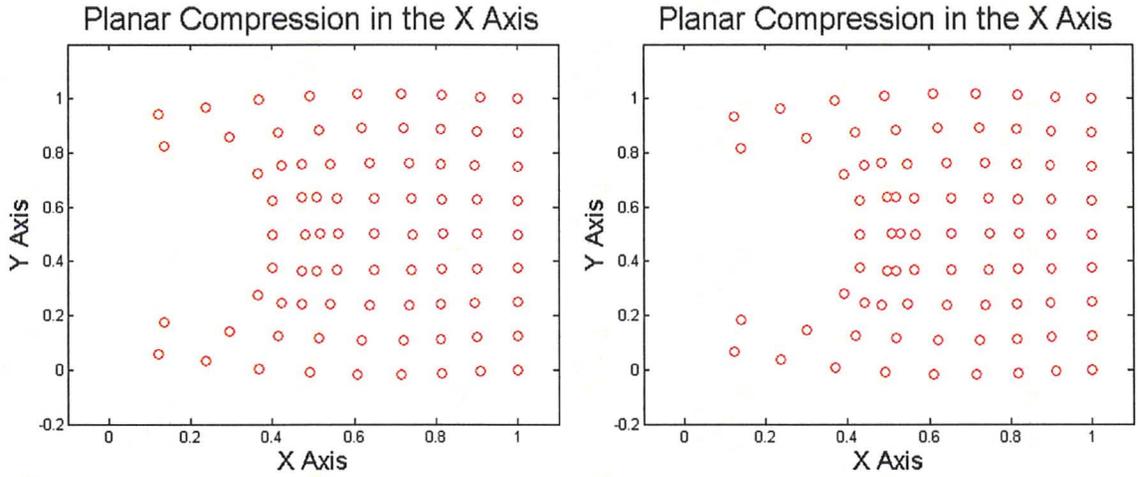


Figure 5.12: Compression in the X Axis by displacing the essential boundary by 40%, 43% respectively for Gaussian Quadrature method.

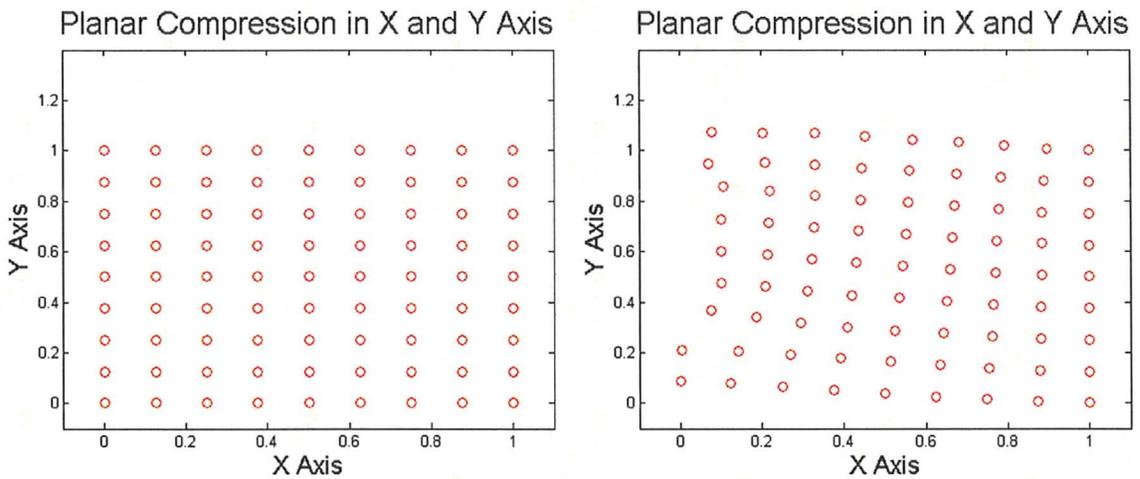


Figure 5.13: Compression in the X and Y Axis by displacing the essential boundary by 0%, 10% respectively for Gaussian Quadrature method.

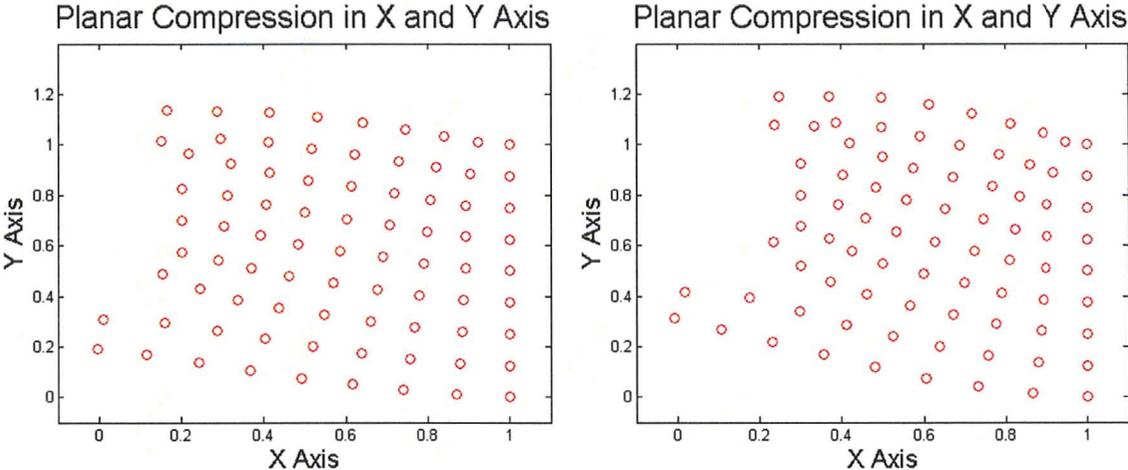


Figure 5.14: Compression in the X and Y Axis by displacing the essential boundary by 20%, 30% respectively for Gaussian Quadrature method.

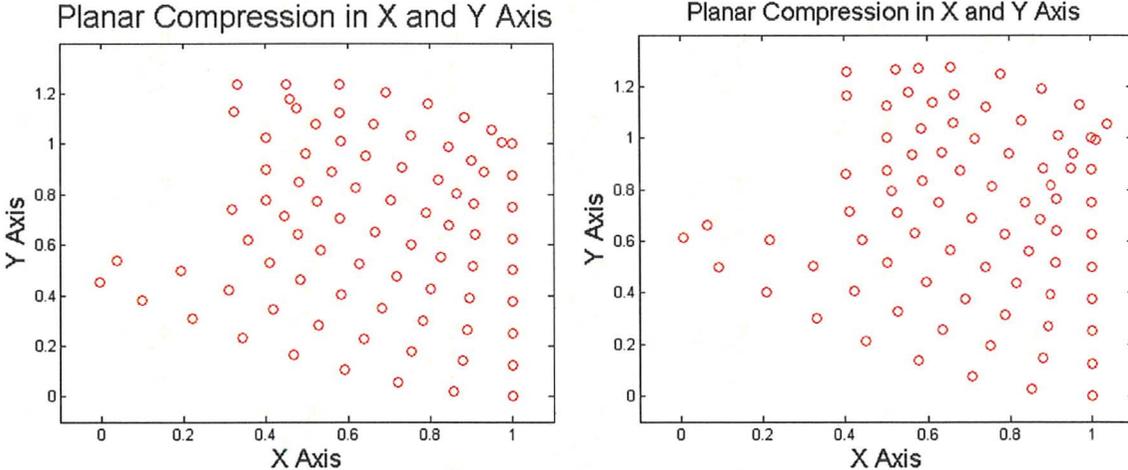


Figure 5.15: Compression in the X and Y Axis by displacing the essential boundary by 40%, 50% respectively for Gaussian Quadrature method.

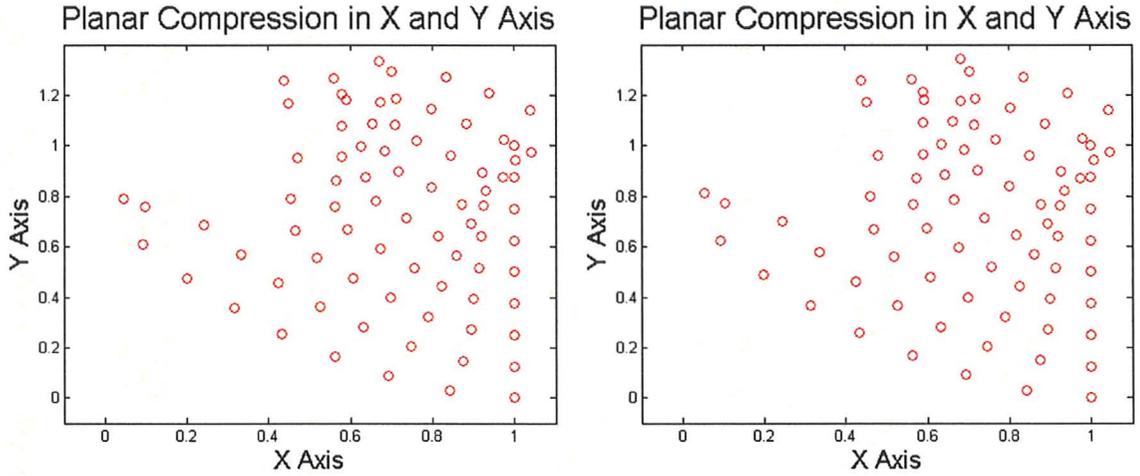


Figure 5.16: Compression in the X and Y Axis by displacing the essential boundary by 58%, 59% respectively for Gaussian Quadrature method.

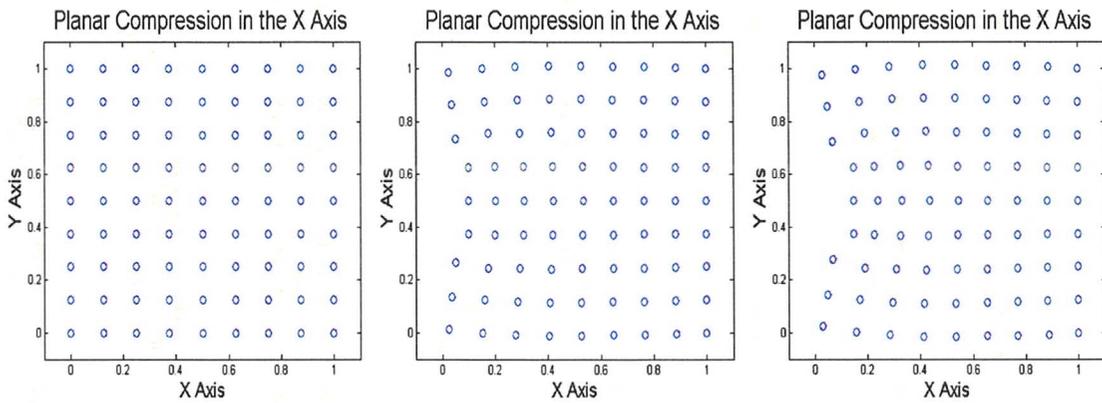


Figure 5.17: Compression in the X Axis by displacing the essential boundary by 0%, 10%, and 15% respectively for Collocated method.

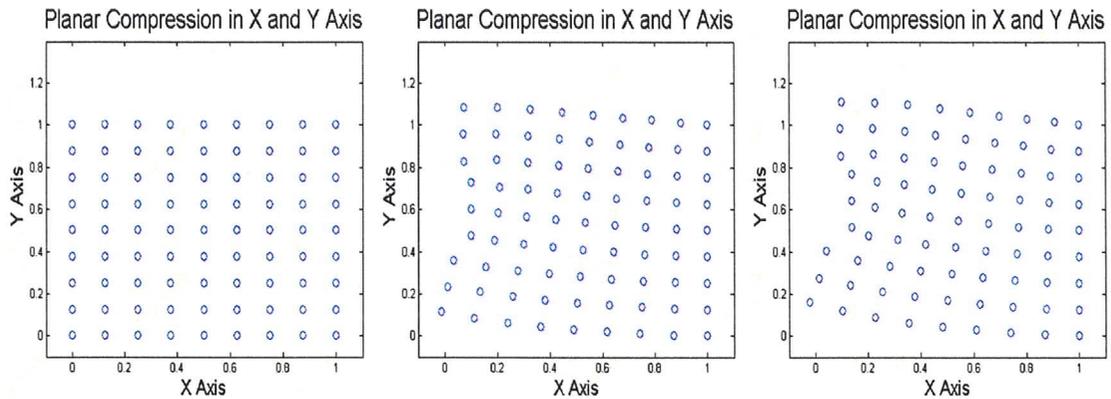


Figure 5.18: Compression in the X and Y Axis by displacing the essential boundary by 0%, 10%, and 14% respectively for Collocated method.

Set Point Regulation

Next, to prove the models can settle within an allowable error of the desired position, a set point regulation was simulated for the control point to move from $(0.00, 0.00)$ to $(0.05, 0.05)$. Figure 5.19 displays the settling of the control point using the Gaussian Quadrature method, while 5.20 depicts the settling using the Collocated method. In each of these figures, both the manipulation and control point are illustrated, and their relative displacements from their original positions are tracked, until a result within 0.0001 of the desired position is reached. In both cases the manipulation point maximum allowable increment was set to 1%, as to limit the large abrupt disturbances on the object. An interesting topic to look into further could be the response of the object as the this parameter is adjusted. The gain constant K in the manipulation displacement calculation was set to 0.05. It should be noted that for all of the remaining simulated object results the K and manipulation point maximum allowable increment values were kept as above. In each case the accuracy is equivalent, although the elapsed time to reach the target varied. To gauge the time lapse of the each solution method, the time was taken from the initial movement of the control point until it settled at its final position. The Collocated model time was approximately 6

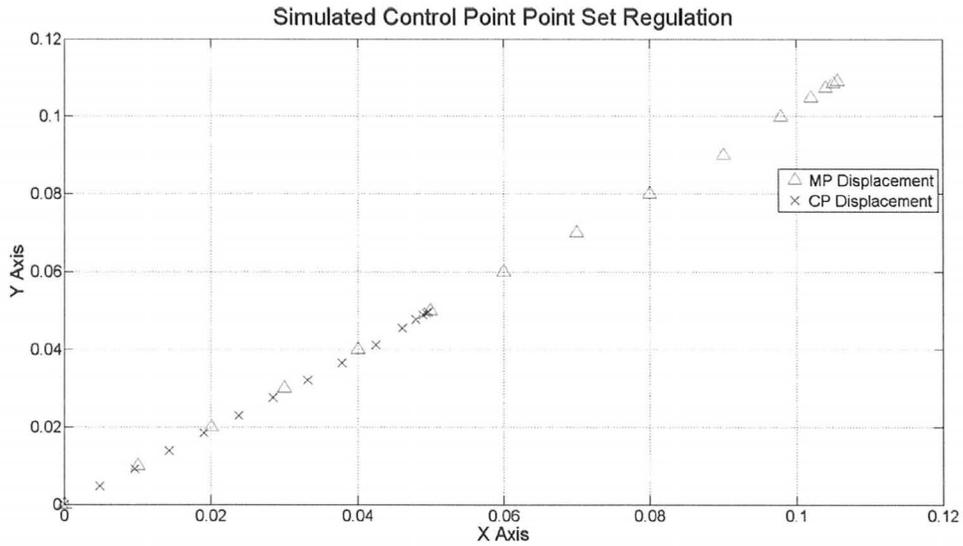


Figure 5.19: Set point regulation of control point via manipulation point control using the Gaussian Quadrature Method.

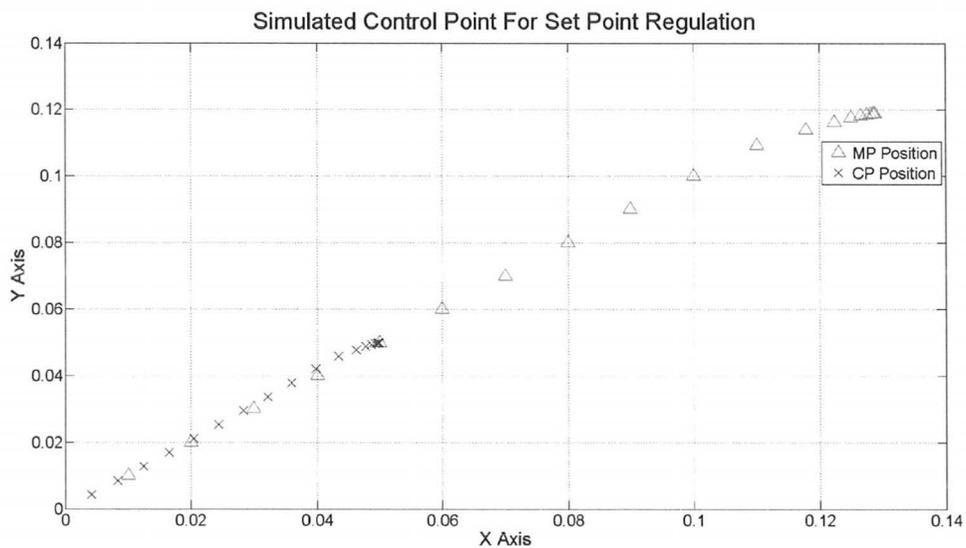


Figure 5.20: Set point regulation of control point via manipulation point control using the Collocated Method.

minutes while the Gaussian Quadrature model took 75 minutes.

Planned Path Simulations

The remaining simulations were performed on 9×9 planar objects, performing a control point trace of three different control point trajectories outlined in Tables 5.3, 5.4 and 5.5.

<i>Step</i>	<i>X</i>	<i>Y</i>	<i>Step</i>	<i>X</i>	<i>Y</i>
1	0.00	0.00	10	0.04	0.03
2	0.00	0.01	11	0.04	0.02
3	0.00	0.02	12	0.04	0.01
4	0.00	0.03	13	0.04	0.00
5	0.00	0.04	14	0.03	0.00
6	0.01	0.04	15	0.02	0.00
7	0.02	0.04	16	0.01	0.00
8	0.03	0.04	17	0.00	0.00
9	0.04	0.04			

Table 5.3: Desired path for control point following a box shape trajectory.

<i>Step</i>	<i>X</i>	<i>Y</i>	<i>Step</i>	<i>X</i>	<i>Y</i>
1	0.00	0.00	7	0.00	0.00
2	0.01	0.01	8	0.01	-0.01
3	0.01	0.02	9	0.01	-0.02
4	0.00	0.03	10	0.00	-0.03
5	-0.01	0.02	11	-0.01	-0.02
6	-0.01	0.01	12	-0.01	-0.01

Table 5.4: Desired path for control point following a figure 8 shape trajectory.

To illustrate the results from the Gaussian Quadrature method, Figures 5.21 and 5.22 show a box shape trajectory being traced followed by the control point error plot through out the simulation. Next, the Figure 8 trajectory was completed, shown in Figures 5.23 and 5.24. Similarly, Figures 5.25 and 5.26 show the control point following a straight line trajectory. In comparison, the Collocated results are shown below. The tracking

<i>Step</i>	<i>X</i>	<i>Y</i>	<i>Step</i>	<i>X</i>	<i>Y</i>
1	0.0	0.00	9	0.08	0.00
2	0.01	0.00	10	0.09	0.00
3	0.02	0.00	11	0.10	0.00
4	0.03	0.00	12	0.11	0.00
5	0.04	0.00	13	0.12	0.00
6	0.05	0.00	14	0.13	0.00
7	0.06	0.00	15	0.14	0.00
8	0.07	0.00	16	0.15	0.00

Table 5.5: Desired path for control point following a straight line.

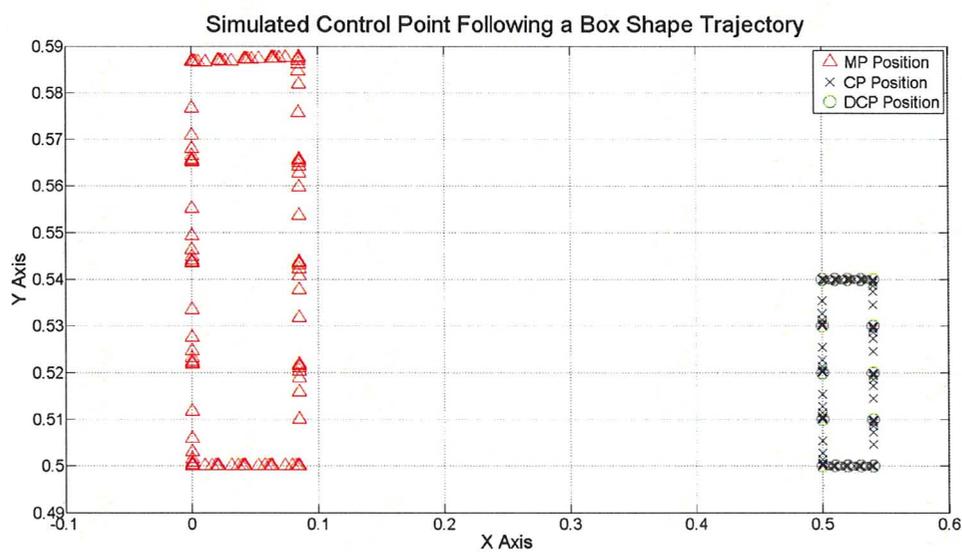


Figure 5.21: Simulating the tracking of a desired control point trajectory in Table 5.3 via object manipulation using the Gaussian Quadrature Method.

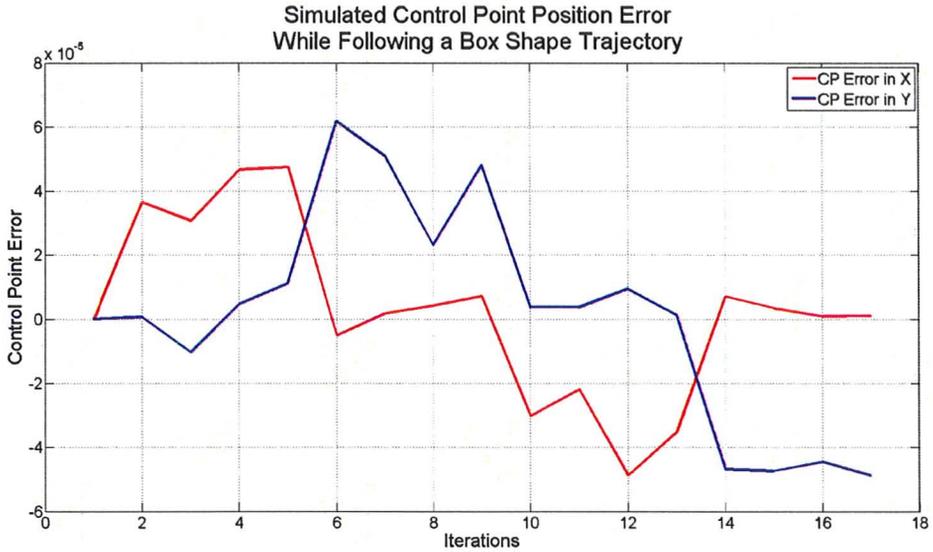


Figure 5.22: Simulated control point tracking error of trajectory in Table 5.3 using the Gaussian Quadrature Method.

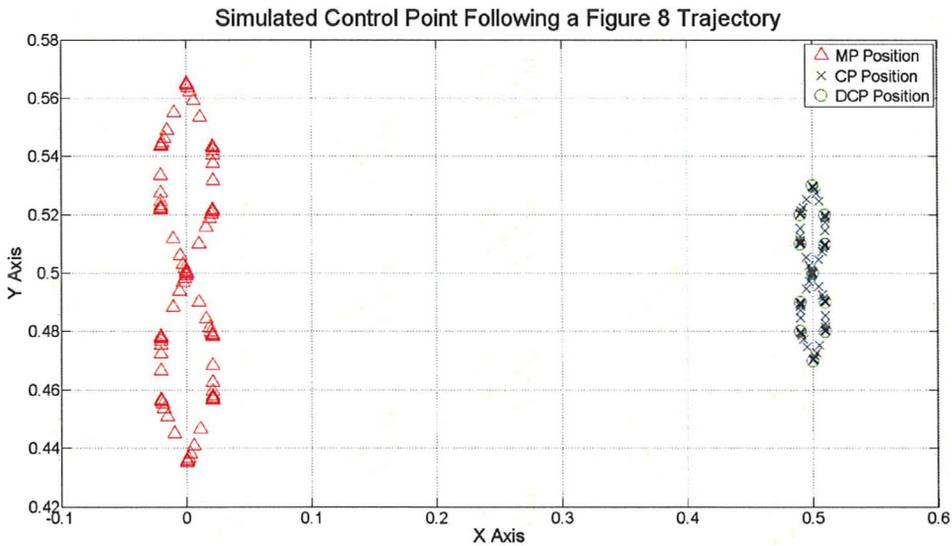


Figure 5.23: Simulating the tracking of a desired control point trajectory in Table 5.4 via object manipulation using the Gaussian Quadrature Method.

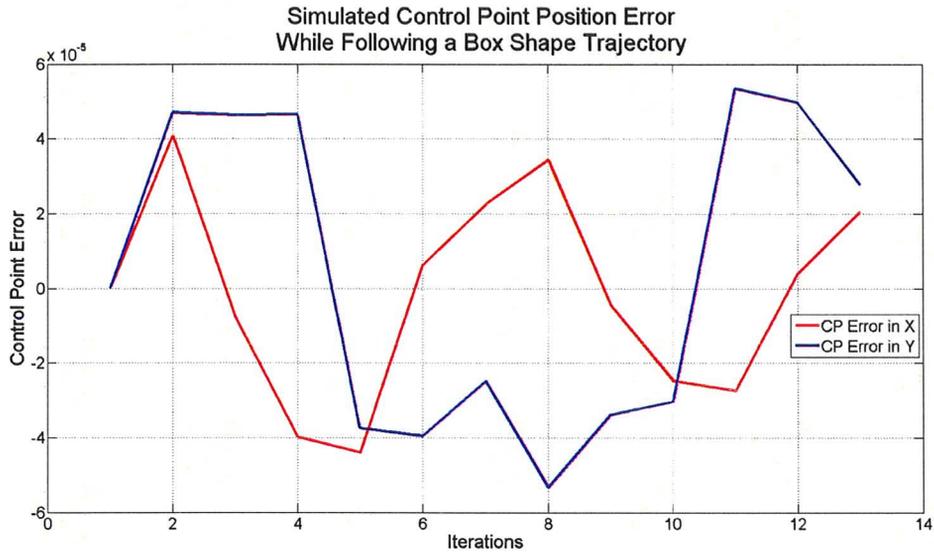


Figure 5.24: Simulated control point tracking error of trajectory in Table 5.4 using the Gaussian Quadrature Method.

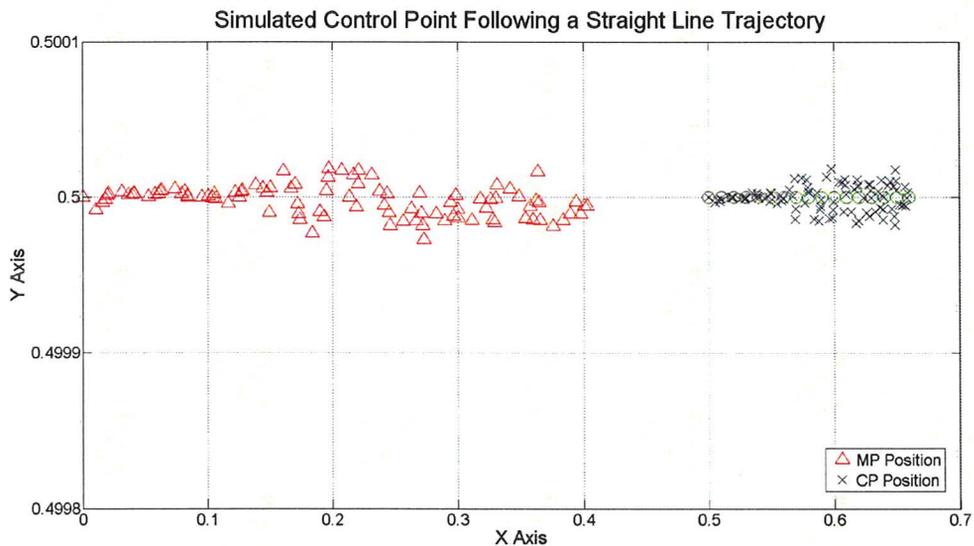


Figure 5.25: Simulating the tracking of a desired control point trajectory in Table 5.5 via object manipulation using the Gaussian Quadrature Method.

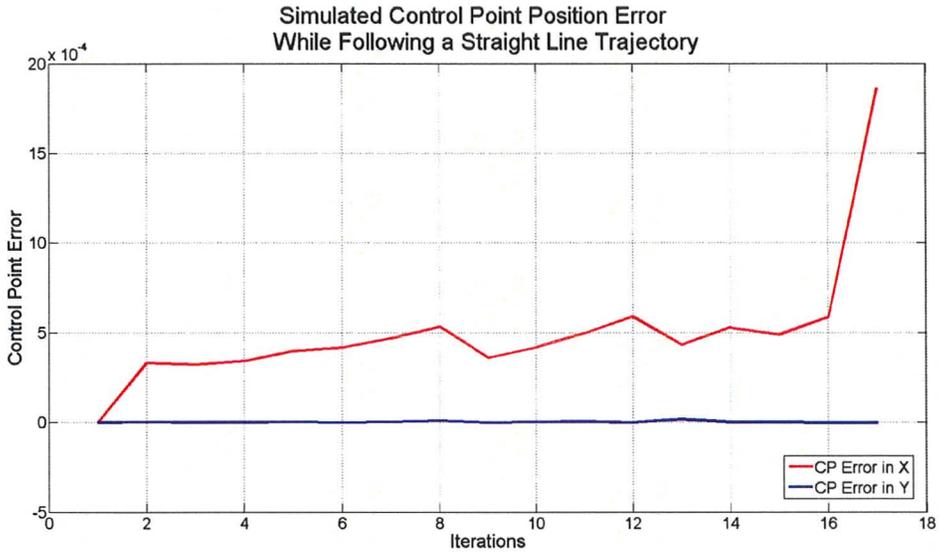


Figure 5.26: Simulated control point tracking error of trajectory in Table 5.5 using the Gaussian Quadrature Method.

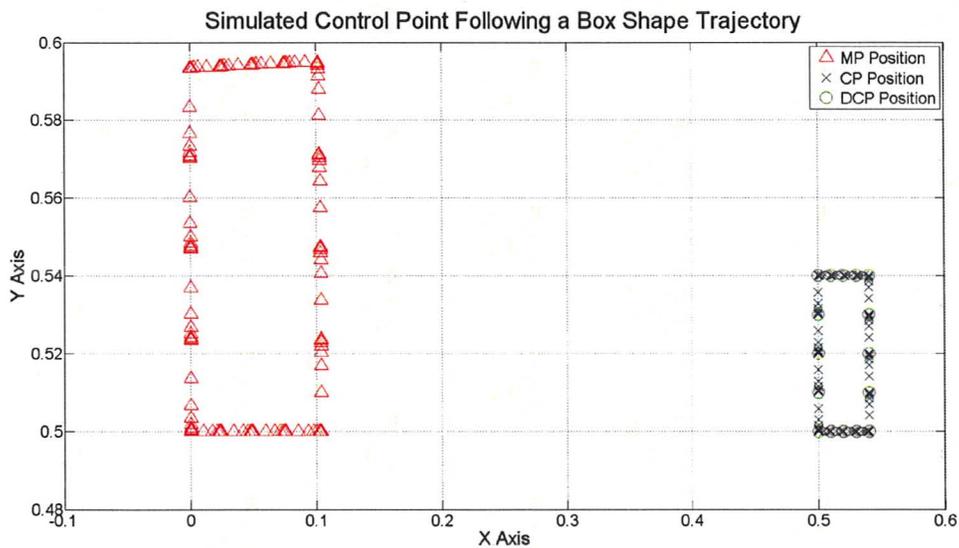


Figure 5.27: Simulating the tracking of a desired control point trajectory in Table 5.3 via object manipulation using the Collocated Method.

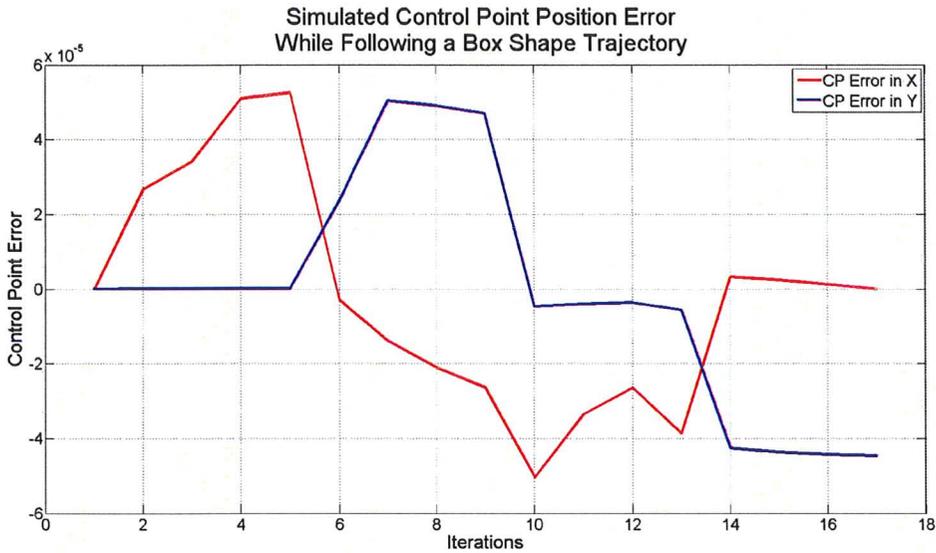


Figure 5.28: Simulated control point tracking error of trajectory in Table 5.3 using the Collocated Method.

of the control point in the box trajectory is shown in Figures 5.27 and 5.28. Further, the Figure 8 trajectory is shown in Figures 5.29 and 5.30. The straight line trajectory is shown in Figures 5.31 and 5.32. It should be noted that although the error deviations seem insignificant in the model simulation, they will be important for the physical model as a form of comparison. Also it is important to point out, since the time is a very crucial determining factor in whether the approach is physically feasible and realistic for practical usage, the Collocated method was chosen to be integrated into the physical system as its simulation times were significantly shorter. To show this, the Collocated model took 38 and 25 minutes to complete the box and figure 8 simulations. While the Gaussian Quadrature model took 625 and 412 minutes to complete the same shapes respectively. The format of all the previous results shows the manipulation points starting at the original position (0.00, 0.05) and the desired and tracking control points starting at (0.05, 0.05).

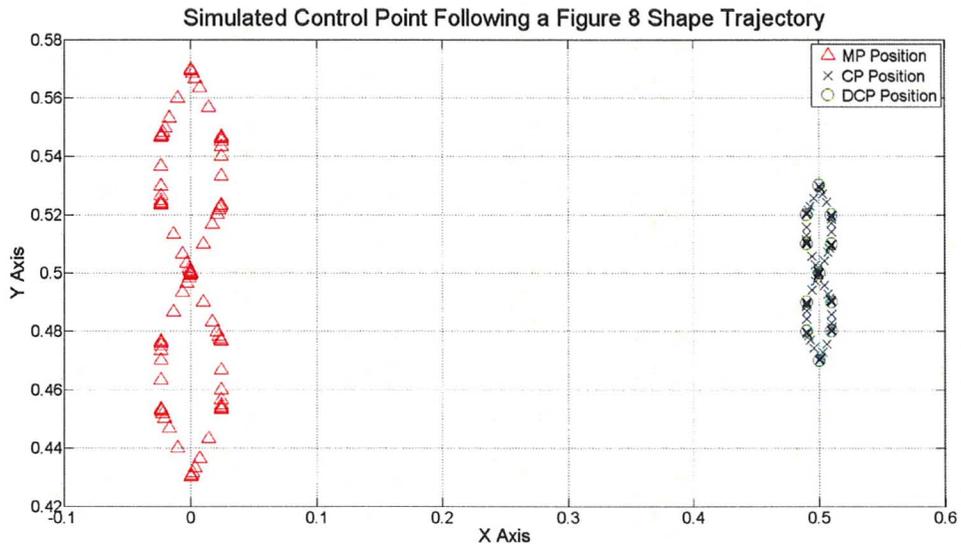


Figure 5.29: Simulating the tracking of a desired control point trajectory in Table 5.4 via object manipulation using the Collocated Method.

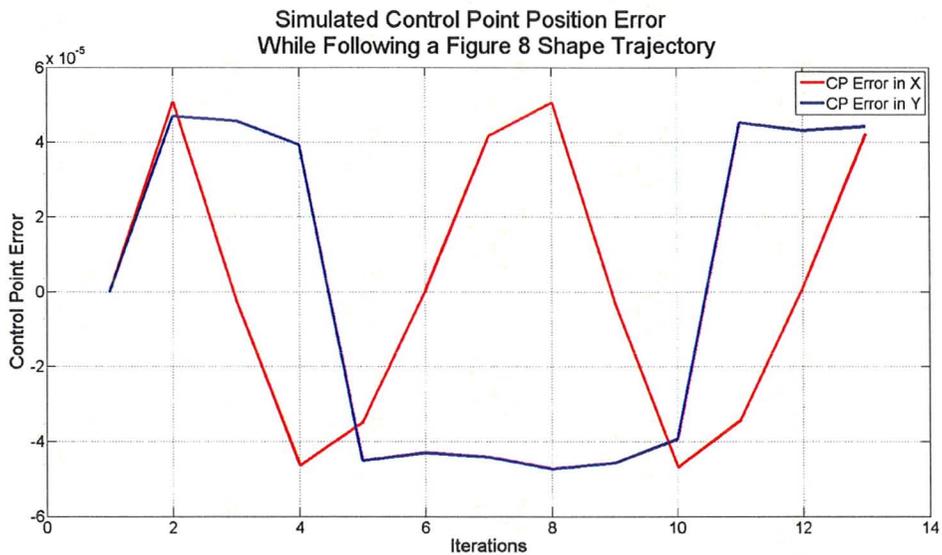


Figure 5.30: Simulated control point tracking error of trajectory in Table 5.4 using the Collocated Method.

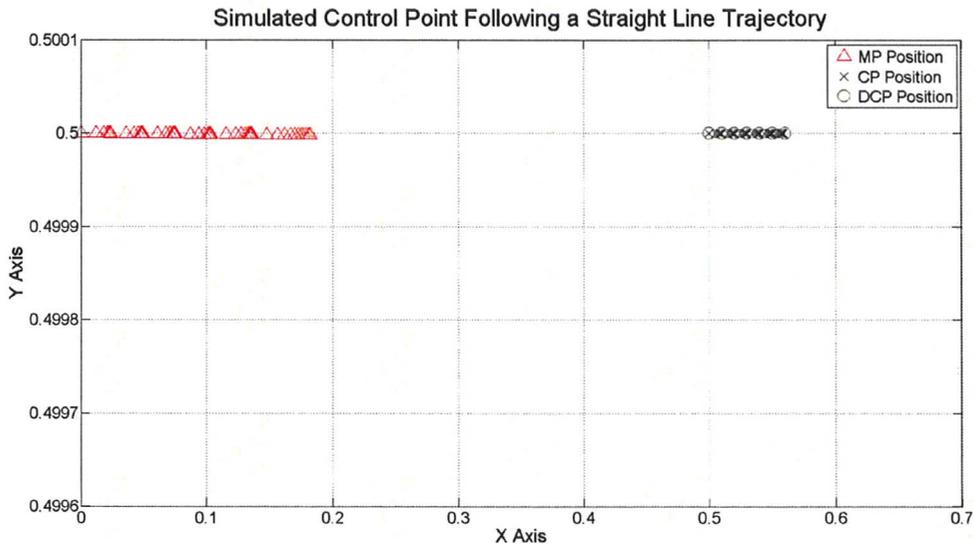


Figure 5.31: Simulating the tracking of a desired control point trajectory in Table 5.5 via object manipulation using the Collocated Method.

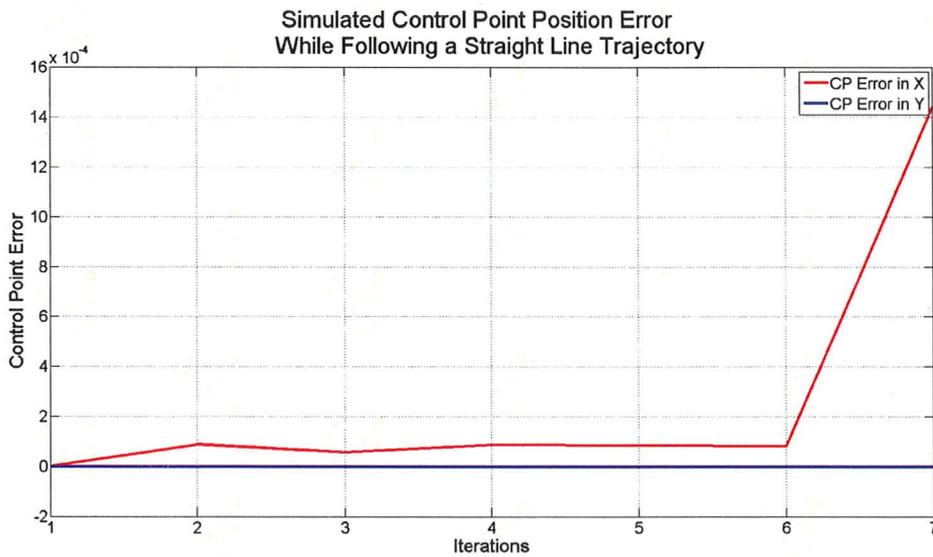


Figure 5.32: Simulated control point tracking error of trajectory in Table 5.5 using the Collocated Method.

Experimental Results

A physical system was developed to validate the deformable object model, and develop a reliable control for the interaction of the vision, communication, object and robotic systems. The system follows that described in Chapter 4, and was tested using the three above trajectories to show successful preliminary results. Using the straight line trajectory, the model linear parameter β were adjusted. The β value with the best tracking of the control points was 0.02. The adjustment of the nonlinear constant α was unnecessary since the physical model did not reach deformation values during the experiment for its effect to be significant. This was due to the fact that the physical object buckled in the z axis for larger deformations.

Set Point Regulation

The first important result to consider is that from a set point regulation experiment, to show that the control system is in fact working and moving toward the desired location. In Figures 5.33 and 5.35, the physical control point is shown to settle within a error margin of 1 millimeter of the target as a consequence of the actual robot maneuvering. The above does show that the marker is coming a settled position, but also give a clear example of the limitation of the camera accuracy. The last section of the recorded control point is blown up in Figure 5.34 to show that upon smaller increments of the manipulation points, the camera may read the control point incorrectly and cause the system to move in the opposite direction intended.

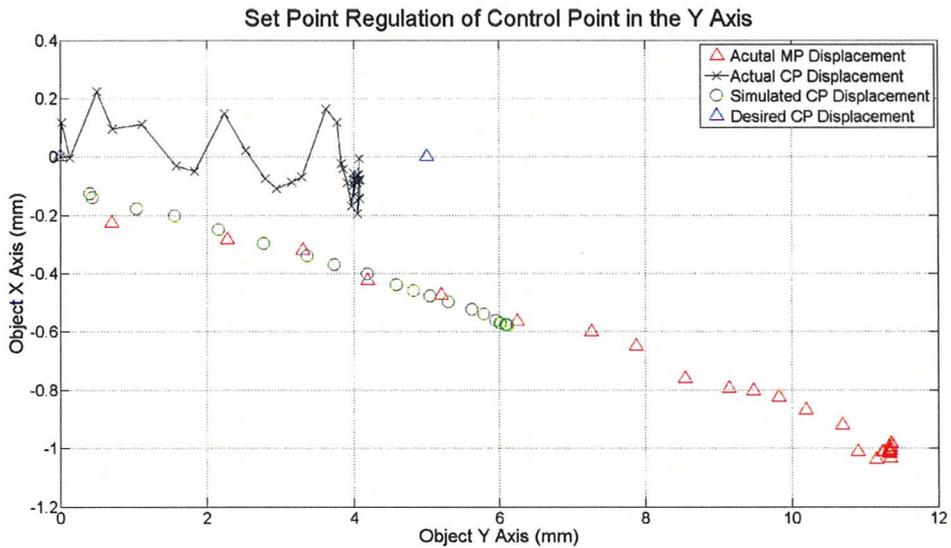


Figure 5.33: Set point regulation of control point via manipulation point control.

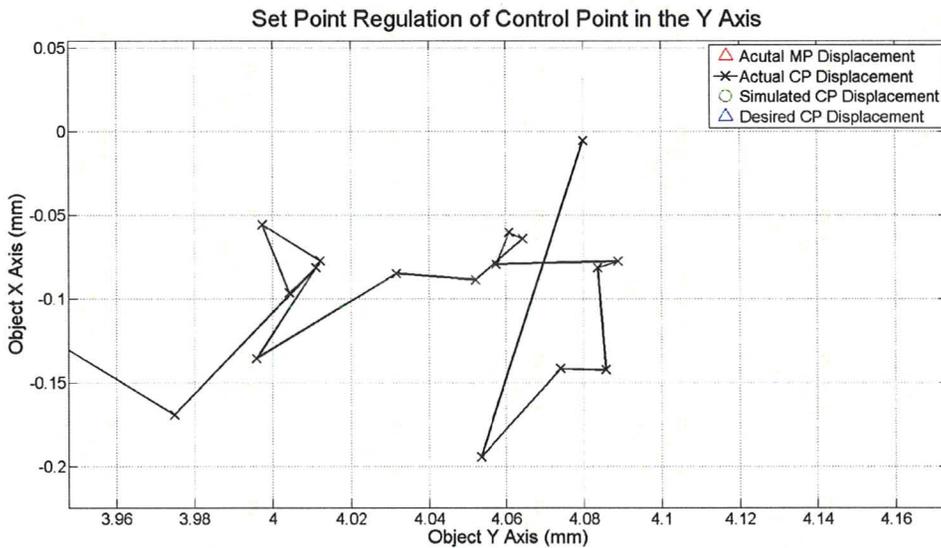


Figure 5.34: Indication that there is a potential problem when small manipulator movements are involved.

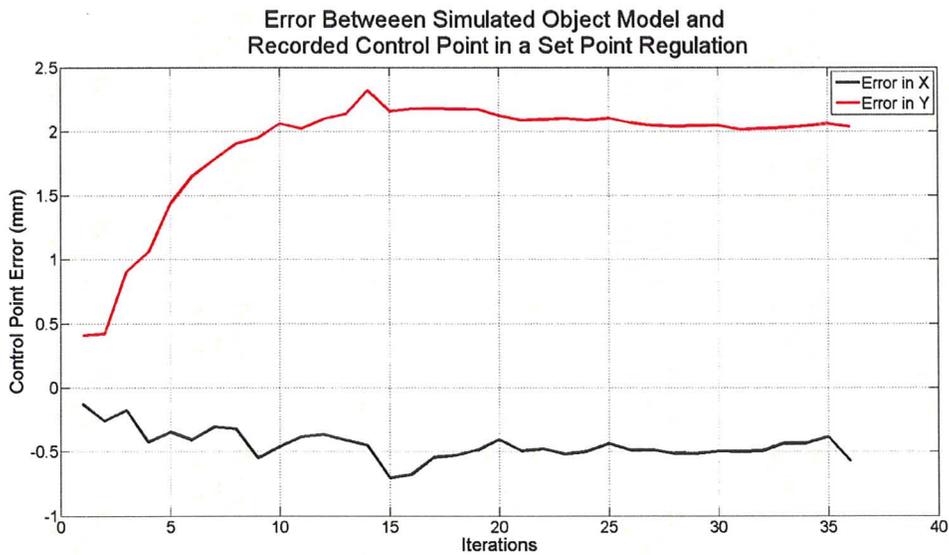


Figure 5.35: Path comparison between the simulated object and recorded control points in set point regulation problem.

Planned Path Experiments

Starting with the box trajectory, Figure 5.36 shows the movement of the recorded and simulated control point as a consequence of the robotic manipulator actuating on the physical object. The control point results are compared to the desired paths in the Figures 5.37 and 5.38, and the integrity of the object simulation control point tracking is compared to the recorded control point tracking in Figure 5.39. Next, a force analysis is conducted on each manipulation point and control points. The physical force data is plotted in Figure 5.40, which depicts the forces from the object acting on the tool tip. These correspond to the calculated the forces retrieved from the model results showing the force on the control point for each iteration step in Figure 5.41, and the set of uniformly moving manipulation points in Figure 5.42. The force on the individual points in the simulated object model can be calculated using the Cauchy Stress outlined in chapter 3. The equations are reproduced

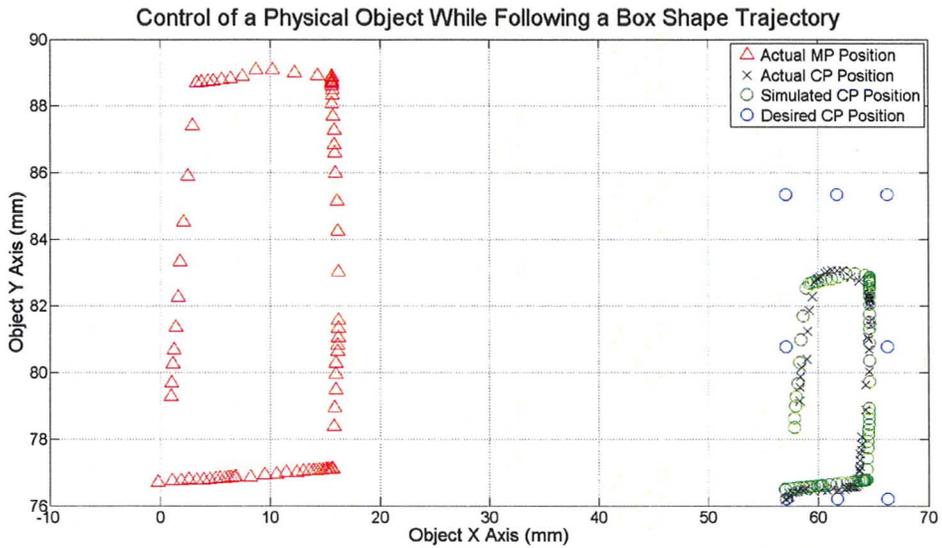


Figure 5.36: Control of physical object in tracking a box shape trajectory.

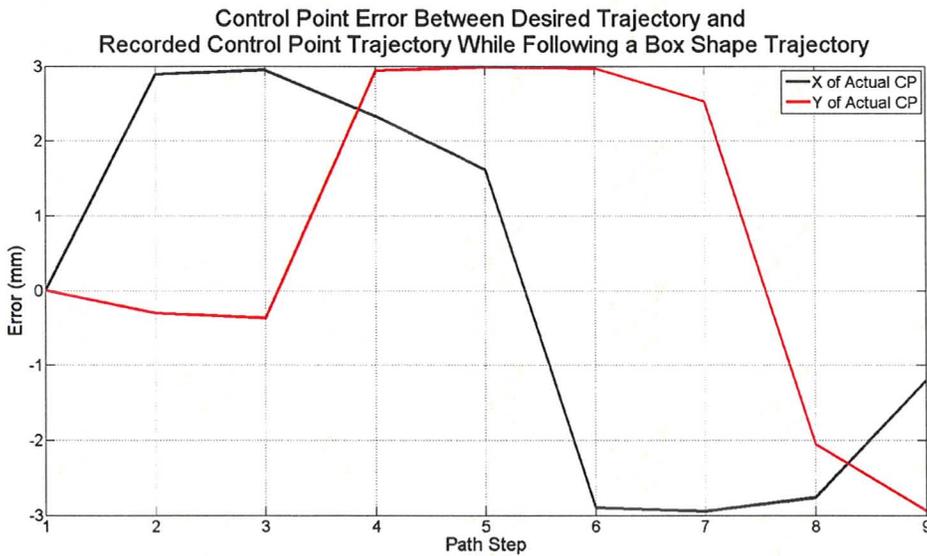


Figure 5.37: Path comparison between desired control point and recorded control point in tracking a box shape trajectory.

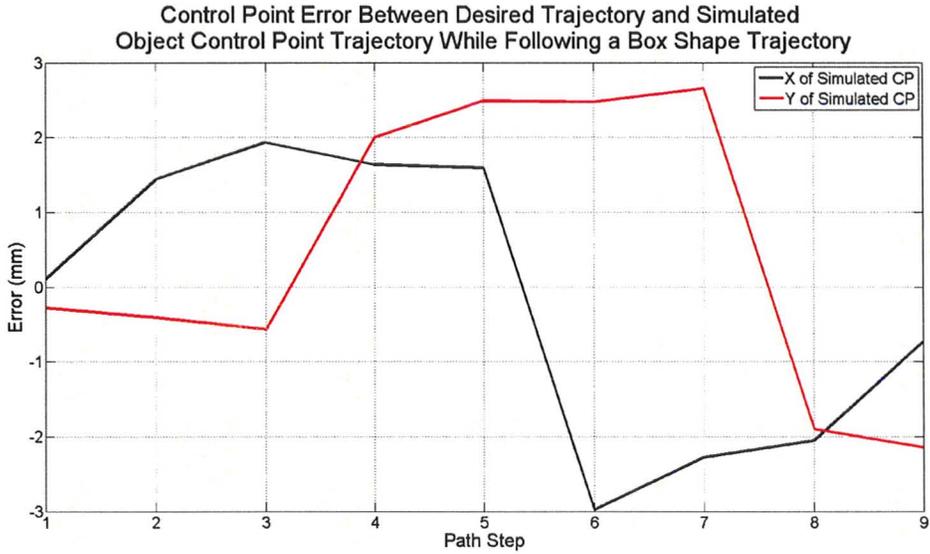


Figure 5.38: Path comparison between desired control point and simulated object control point in tracking a box shape trajectory.

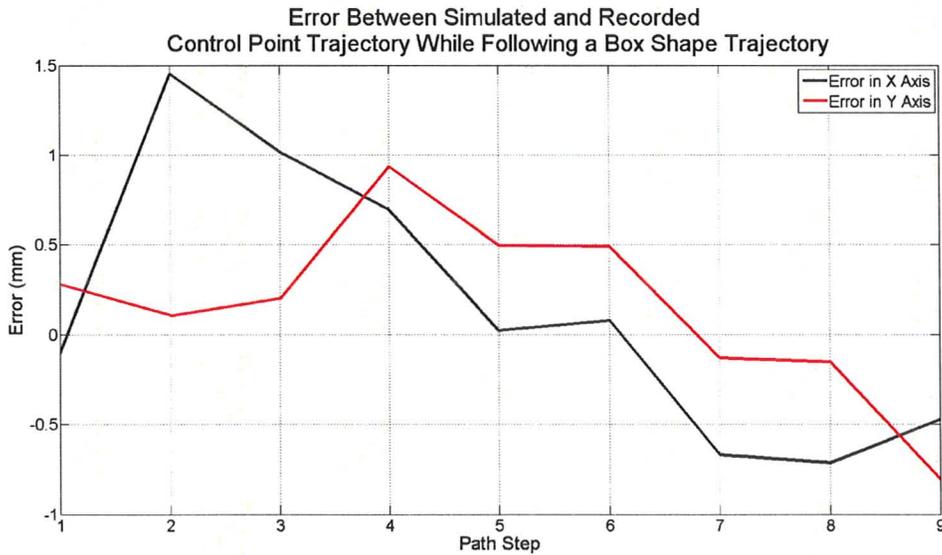


Figure 5.39: Path comparison between the simulated object and recorded control points in tracking a box shape trajectory.

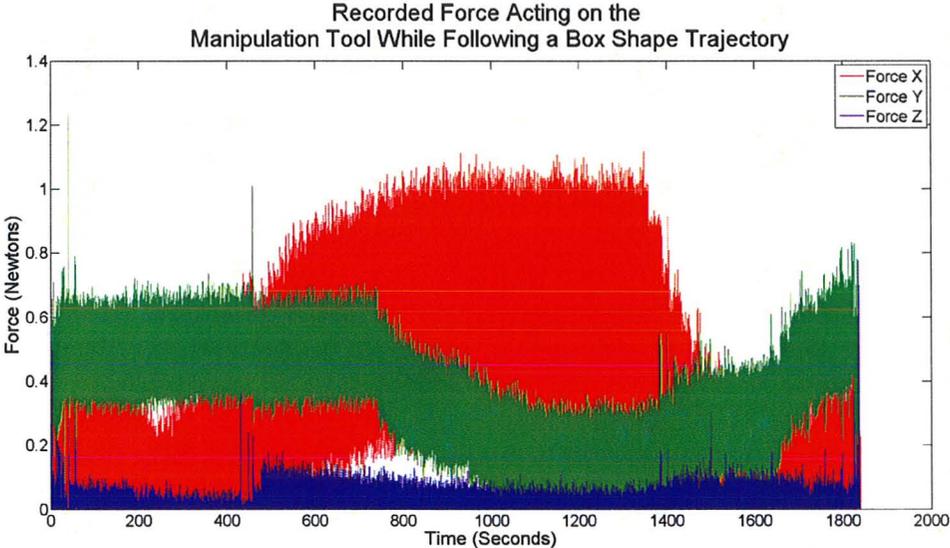


Figure 5.40: Recorded force data in space acting on the tool tip.

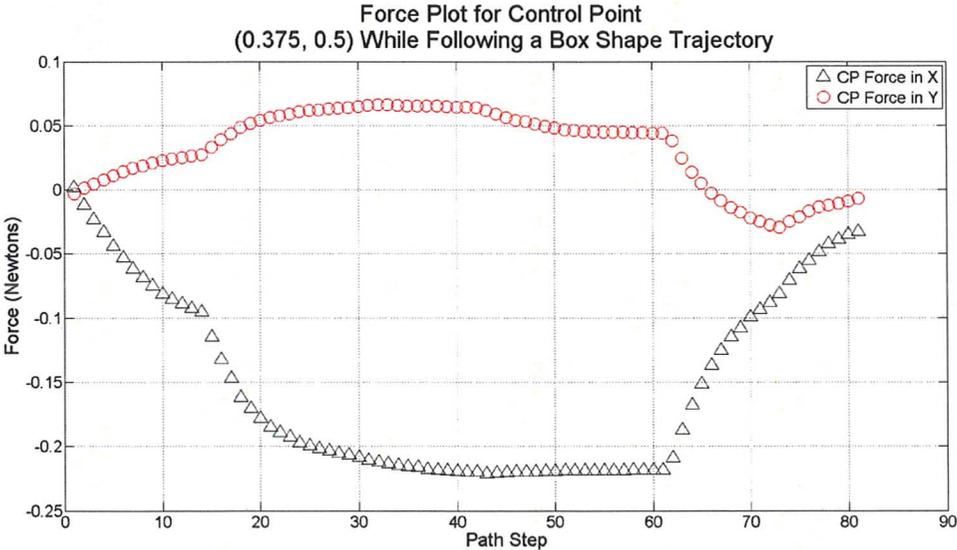


Figure 5.41: Calculated force on the object model control point for each successive path step.

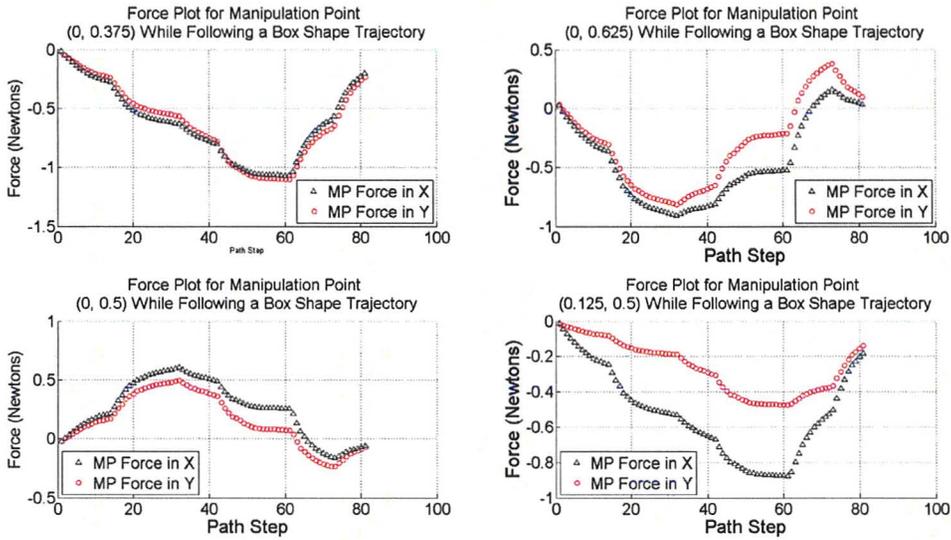


Figure 5.42: Calculated force on the object model manipulation points for each successive path step.

here followed by a sample force calculation:

$$\tau_{ij}n_j = h_i \tag{5.3}$$

where n_j is the normal vector on the object boundary, and τ is the Cauchy Stress

$$\tau_{ij} = \frac{1}{J} F_{im} S_{ij} F_{jn} \tag{5.4}$$

where $J = \det(F)$. An example calculation for the τ is shown below:

$$\tau_{00} = \frac{1}{\det(F)} F_{0m} S_{mn} F_{0n} \quad (5.5)$$

$$\tau_{10} = \frac{1}{\det(F)} F_{1m} S_{mn} F_{0n} \quad (5.6)$$

$$\tau_{01} = \frac{1}{\det(F)} F_{0m} S_{mn} F_{1n} \quad (5.7)$$

$$\tau_{11} = \frac{1}{\det(F)} F_{1m} S_{mn} F_{1n} \quad (5.8)$$

The step size between particles is 0.125 object units and the physical object with is 6×6 inches or 0.1524×0.1524 meters in size. The n_j is set as $[(2 * 0.125) * 0.1524; (2 * 0.125) * 0.1524]$ in the X and Y direction respectively. These values indicate the forces acting on the individual manipulation points. To obtain a force resembling the measure force, the resultant force between the individual points must be calculated to gather the collective force in each direction.

5.4 Discussion

In both model cases, the control point tracking for a single point was simulated successfully within desirable error margins. It was evident that larger deformations were better handled when using the Gaussian Quadrature integration rules, where successful boundary deformations of approximately 50% could be accomplished. In contrast to the Collocated model which broke down at around 15% deformation. With this in mind, neither method showed to possess an advantage in accuracy, yet the number of iterations was lower for the Gaussian Quadrature than the Collocated. Also, the drastic solution time difference between the two methods proved to only make one method feasible, the Collocated, unless a more rapid

computation technique could be used to obtain the results using Gaussian Quadrature.

The integrated system proved to work well within the limits of the camera's capabilities. Based on the camera's manufacture stereo accuracy calculations the distance, as the distance of the object from the camera increased the stereo accuracy diminishes rapidly. For the marker distances that were used on the object during the experiments, which were between 0.40 to 0.50 meters, the stereo accuracy was 0.4 to 0.5 millimeters.

Chapter 6

Target Stabilization in Needle Insertions

6.1 Summary

This chapter outlines a proposal for a novel target stabilization technique in a soft tissue model during a needle insertion procedure. The technique involves compensating for target movement by using object boundary manipulations in a configuration shown in Figure 6.1. During a needle insertion procedure, the object may deform resulting in target deflection away from the line trajectory of the inserted needle. As outlined below, this problem is approached in two ways. First, a robotic manipulator is used to immobilize a section of the object boundary, to minimize target movement. Therefore, if a manipulator is strategically fixed against the object boundary and restricts boundary movement, the target is shown to deviate less from the insertion line of the needle. An optimal paddle placement can be located experimentally by testing the target movement to the corresponding paddle placement. Alternatively, a dynamic manipulator is used. Emulating the procedure for the fixed manipulator, the optimal boundary location to give the user the highest target point controllability can be located. Homogeneous and non-homogeneous object model are also

incorporated here.

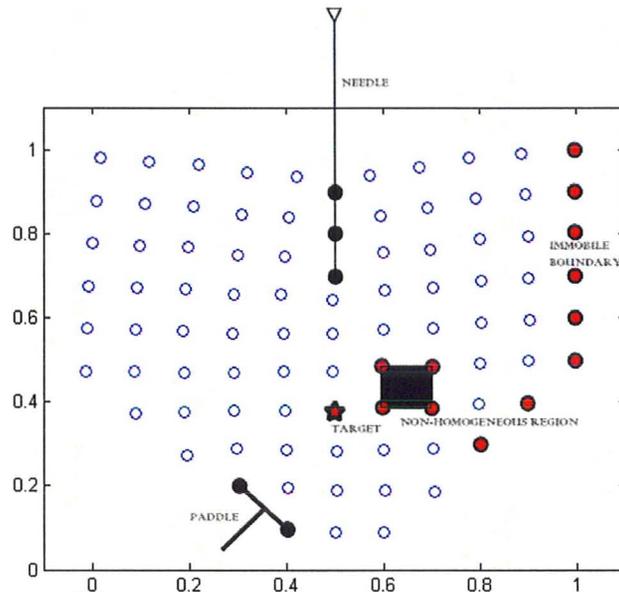


Figure 6.1: Deformed object model indicating immobile boundary, needle insertion, stabilizing paddle and a non-homogeneous region.

6.2 Procedure

The purpose of the experimental setup was to develop a model prescribed by anatomy, such that only a small portion of the object was accessible. Since various anatomical constraints could make certain manipulation procedures unfeasible, an approach with limited boundary access should produce more realistic results. It should be noted that previous soft object model approaches such as Mallapragada (2007), allowed for object interactions with any point on the object boundary. Full accessibility to an object would be difficult to assume if procedures were carried out on internal organs or glands such as a prostate. Further, as mentioned in Chapter 2, the focus of the study in Mallapragada (2007) was the constant adjustment of the boundaries to move the target point in line with the inserted needle. Later

in Mallapragada (2008), two new trials were performed. In the first, they demonstrated the movement of the inclusion during needle insertion. The phantom used for the procedure was inhomogeneous as to mimic the deflection of the inclusion during external disruptions. Secondly, the inclusion was placed at a desired location. In this case the needle was inserted into the tissue creating a disruption, and the external manipulators were used to compensate and move the inclusion to the needle in the 2D plane. Alternatively, complex algorithms for needle steering have been proposed in Alterovitz (2003), DiMaio (2003), Torabi (2009) and Glozman (2007).

The tested object was defined as having an immobile surface section depicted in Figure 6.1 and a limited accessible boundary for external paddle interaction. The object was chosen to roughly represent breast tissue and the object could also include non-homogeneities placed in any location. It is noted here that the same strain energy function and parameters were used for the following simulations as in chapter 5.

$$W = \alpha e^{E_{ij}C_{ijkl}E_{kl}} + \beta E_{ij}A_{ijkl}E_{kl} \quad (6.1)$$

was employed help define the object characteristics, where α and β are positive constants, E is the strain tensor represented as a vector and A and C are symmetric second-rank tensors. The values employed for A and C are listed in Table 6.1 determined through tissue measurement in Cowin (2006).

6.3 Simulation Results

During the target stabilizing procedure it was assumed that there was no boundary slip-page against the inserted needle shaft, and the needle insertion was modeled as a uniform

A_{ijkl}	Value	C_{ijkl}	Value
$A(0,0,0,0)$	1020	$C(0,0,0,0)$	3.5
$A(1,1,1,1)$	1020	$C(1,1,1,1)$	3.5
$A(0,1,0,1)$	254	$C(0,1,0,1)$	0.5
$A(1,0,1,0)$	254	$C(1,0,1,0)$	0.5
$A(0,0,1,1)$	383	$C(0,0,1,1)$	1.5
$A(1,1,0,0)$	383	$C(1,1,0,0)$	1.5
$A(0,1,1,0)$	383	$C(0,1,1,0)$	1.5
$A(1,0,0,1)$	383	$C(1,0,0,1)$	1.5

Table 6.1: Values for A and C tensors. All values not indicated are zero. Obtained from Chapter 15 Tissue Mechanics Cowin (2006).

compression of the three indicate nodal points in a straight line throughout the simulation.

Also, the paddle positions were chosen to overlap successive boundary nodal points on the object. The paddle position index is listed in Table 6.2.

Paddle Position Index	<i>Coordinates</i>
1	(0.4, 0.1), (0.3, 0.2)
2	(0.3, 0.2), (0.2, 0.3)
3	(0.2, 0.3), (0.1, 0.4)
4	(0.1, 0.4), (0.0, 0.5)
5	(0.0, 0.5), (0.0, 0.6)

Table 6.2: Index relating to paddle positions.

6.3.1 Target Point Stabilization with Static Paddle

Using a homogeneous object model, a needle initially in line with the desired target was inserted into the object and continued to follow a straight trajectory. During each insertion, a static robotic manipulator was kept in constant contact with the boundary while the object deformed to accommodate the needle. The fixture positions were incrementally changed to cover successive boundary nodal points on the object, beginning at the position shown in Figure 6.1. The algorithm below outlines the procedure.

```

procedure OPTIMAL STATIC PADDLE POSITION
  for  $i = 1 \dots N$  do
    Fix object boundary with paddle at paddle position index  $i$ 
    Compress object by set increment to simulate needle insertion
     $X_i = \text{target lateral deviation}$ 
  end for
  Set optimal paddle position to index corresponding to  $\min|X_i|$ 
end procedure

```

- $N \rightarrow$ Number of paddle positions given in Table 6.2
- $X \rightarrow$ Target lateral deviation

It must be noted that there is a limited boundary perimeter to which the robotic fixtures can be applied to. The resulting control point deviation from its original position is depicted in Figure 6.2 showing movement perpendicular to the line of needle insertion, as a

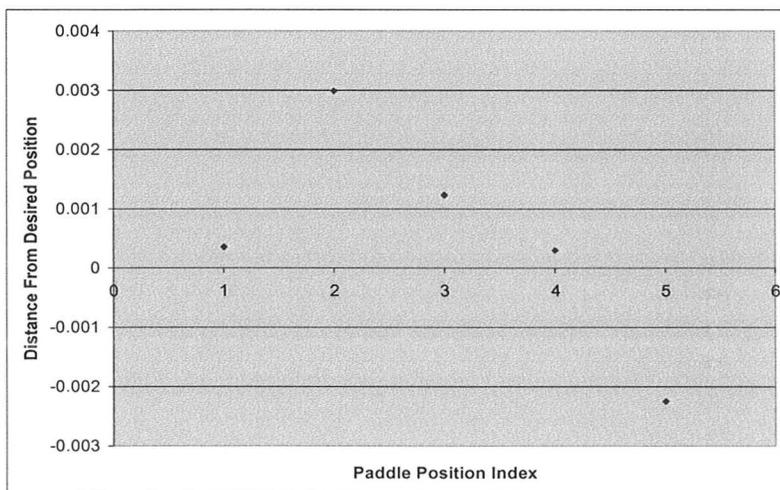


Figure 6.2: Control point error in the X-axis of a homogeneous object as a function of paddle position.

function of the paddle position along the object boundary. To gauge the success of using

these stabilizing static paddles, a control test was simulated to compare the deviation of the control target without any support. The consequence of the simulated needle compression the object is shown in Figure 6.3, where the target point deviation perpendicular to the

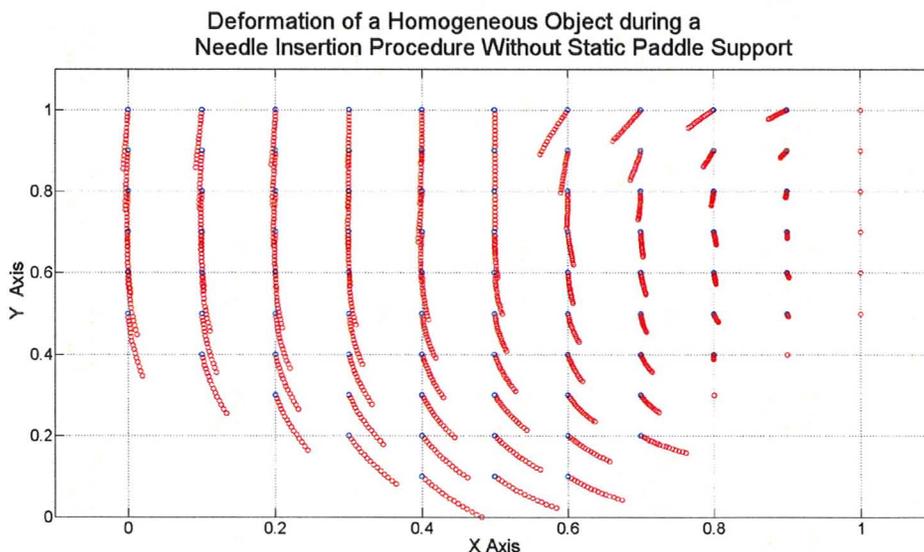


Figure 6.3: Simulated test object after a needle compression of 15% without the use of a static paddle support. Note the control point begins at position (0.5, 0.4)

insertion axis was 0.0307454. Therefore, the use of the static paddle decrease the control point deviation by a range of approximately $10\times$ in the worst case to $100\times$ in the best case at position 4. The end effect can be seen in Figure 6.4, where the optimal static paddle position shows the effects of object stabilization.

Next, a non-homogeneous square region was implemented into the object as shown in Figure 6.1, with a stiffness coefficient $100\times$ that of the surrounding object, at indices (0.6, 0.4), (0.7, 0.4), (0.7, 0.3), (0.6, 0.3). Emulating the previous procedure, the results are shown below in Figure 6.5. This study justifies the need to use a manipulatable fixture as it quite unpredictable as object becomes non-homogeneous.

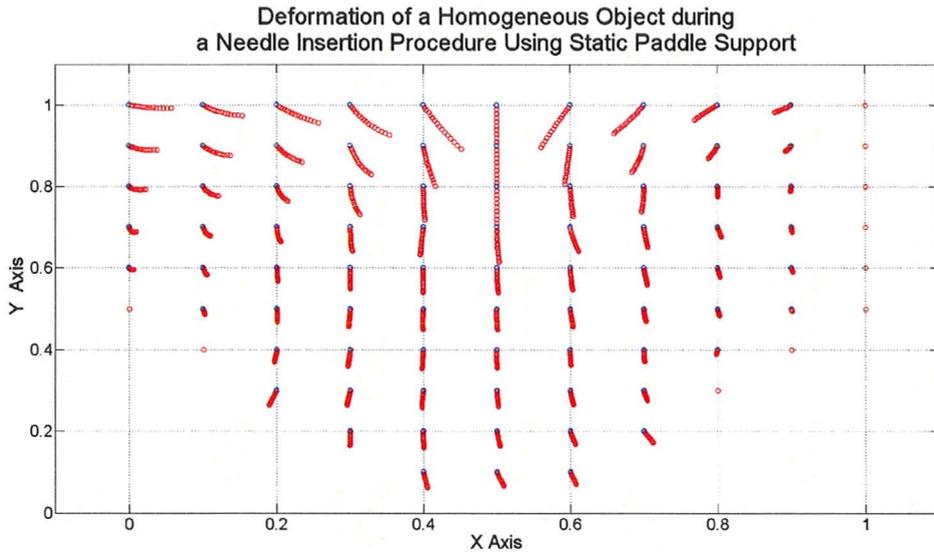


Figure 6.4: Simulated test object after a needle compression of 15% using static paddle support in position index 4. Note the control point begins at position (0.5,0.4)

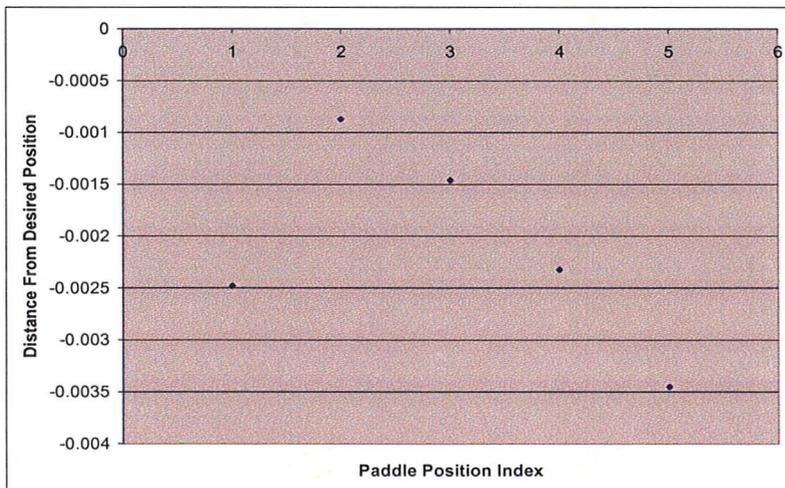


Figure 6.5: Control point error in the X-axis of an object with one non-homogeneous region located at indices (0.6,0.4), (0.7,0.4), (0.7,0.3), (0.6,0.3) as a function of paddle position.

Another non-homogeneous region at indices $(0.3, 0.4)$, $(0.4, 0.4)$, $(0.4, 0.5)$, $(0.3, 0.5)$ with the same stiffness as above gives different results are shown in Figure 6.6.

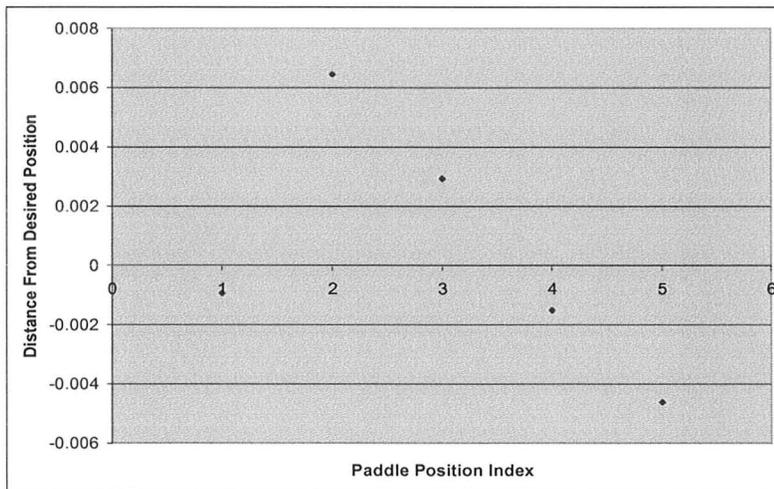


Figure 6.6: Control point error in the X-axis of an object with non-homogeneous region at indices $(0.3, 0.4)$, $(0.4, 0.4)$, $(0.4, 0.5)$, $(0.3, 0.5)$ as a function of paddle position.

6.3.2 Stabilization of Targeted Point with Dynamic Paddle

A direct application of the deformation planning results obtained in Chapter 5 are used to aid in the dynamic stabilization of a targeted area. In addition to the static paddle approach, it is important to constantly stabilize the desired target in one location throughout the simulation. In this next experiment, the stabilizing paddle was maneuvered to maintain the target point in line with the needle axis. A simple control law was derived using a Jacobian estimate of the transformation between a boundary displacement and control point displacement, shown in algorithm below.

- $\varepsilon \rightarrow$ Unit increment set for the robot manipulator

Algorithm 5 Jacobian Estimation

```

procedure COMPUTE J
  Save current object configuration
  for  $i = 1 \dots N_D$  do
     $MPA = 0_{N_D \times 1}$ 
     $MPA_i = \epsilon_i$ 
    Solve Equation 3.40 with MPA as displacements and retrieve CPA
     $J(:, i) = CPA / \epsilon_i$ 
  end for
  Restore object configuration
end procedure

```

- $N_D \rightarrow$ Dimension of the object

The estimation was achieved through perturbations of the control point in the relevant space dimensions, as shown in the path planning application. Given the Jacobian matrices, the optimal dynamic paddle position could be calculated using a similar algorithm to that of the static paddle experiment.

```

procedure OPTIMAL DYNAMIC PADDLE POSITION
  for  $i = 1 \dots N$  do
    Insert needle into object by set increment
    Compute  $J_i$ 
     $X_i = ||J_i^{-1}[10]^T||$ 
  end for
  Set optimal paddle position to index corresponding to  $\min|X_i|$ 
end procedure

```

- $N \rightarrow$ Number of paddle positions given in Table 6.2
- $X \rightarrow$ Target lateral deviation

Given the Jacobian matrix, represented below

$$J = \begin{pmatrix} dx_x & dx_y \\ dy_x & dy_y \end{pmatrix} \quad (6.2)$$

the controllability measure for a certain paddle position is defined as the norm of the required paddle displacement, in order to achieve a given displacement of the target in the direction orthogonal to the needle axis. In particular, $x_{min} = \|J^{-1}[10]^T\|$ determines which paddle position gives the best controllability in the x dimension. In other words, the optimal paddle position is the paddle position in which the smallest amount of boundary adjustment is required to move the target points back in line with the needle. Based on Table 6.3, it is clear paddle 3 has the greatest effect on the control point.

Paddle Index	Homogeneous Object	Non-Homogeneous Object
1	4.7822	6.3556
2	4.1544	5.3583
3	3.2715	4.8749
4	3.3748	5.4552
5	3.3999	5.9523

Table 6.3: δx_{min} as a result of calculated Jacobian Matrices.

6.4 Discussion

Gathering the results from the optimizing paddle placement simulations, placing a paddle at position 4 gives the best target point stabilizing effect for the homogeneous object, as these represent the smallest control point displacement in the lateral axis. Optimal paddle position becomes a function of the non-homogeneous region location as Figure 6.5 shows paddle 2 and Figure 6.6 shows paddle 1 as optimal for reducing target deviations. The

introduction of an impurity creates a obvious shift in the target point movement.

From these results it is clearly shown that a stabilizing paddle position can be computed for a given targeted region, although it is quite limited when the object become non-homogeneous. Predicting the target point deviation when more non-homogeneous areas are added becomes increasingly difficult. Due to this limitation, utilizing dynamic paddle manipulations of the object is justified.

The calculations gathered from stabilizing the target position with a mobile robotic fixture in Table 6.3, show that paddle 3 is optimal for both the homogeneous and non-homogeneous object in this case. This is based on the fact that the paddle provides the greatest target point controllability in the lateral axis. Although this fixture is case specific, the controllability of the target point is successful, and can be achieved for more complicated configurations and objects with non-homogeneous regions. The computed Jacobian algorithm provided accurate tracking of the control point position allowing for appropriate paddle manipulation to bring the control point back to its original location.

It is considered for further study to implement a physical object model that will be constructed from a silicon based material, to test our model.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The goal of this thesis was to develop a physical system, based on position feedback via stereo camera, to validate a deformable object model for various applications. The deformable object model was previously developed and implemented using the Reproducing Kernel Particle Method Chen (1996). Initially, validation tests were performed to explore the object's capabilities and characteristics. Next, a control scheme was developed to virtually control the object movements based on external manipulations and reference points arbitrarily chosen on the object. The control scheme involved the calculation of a Jacobian between these manipulation point sets and the control point sets, by simulating the object reaction to perturbations of size ε in all principle direction involved.

It was this same control scheme which was transferred to and utilized in the physical system. Reference and control points on the deformable model were realized on a physical model via IR LEDs used as tracking markers located by a stereoscopic camera. The positions of these points were successfully relayed back to the deformable model object that

could update the model during a simulation. The object model, made from a soft plastic, and physical model were calibrated through force simulations in the principle directions. This was accomplished as best as possible using the object model's Fung's strain energy density function parameters.

The results first present a comparison between two different integration methods which were used to solve the deformable object model, the Gaussian Quadrature and Collocated rules. Using the Gaussian Quadrature rules, it was found that the object had significantly larger deformation capabilities with the down side of extremely lengthy computation times as compared to the Collocated rules. In both case the accuracy values of the results were comparable. Although the amount of maximum deformation obtained by the collocated method was approximately $\frac{1}{3}$ to $\frac{1}{4}$ that of the Gaussian Quadrature, the computation times was the deciding factor for which method would be used for the physical system. A preliminary study was successfully completed using a coarse and refined object model to show a virtual concept of the physical system, using three dimensional objects under external manipulations. The simulation proved that using an object with a finite number of particle points could control objects which are composed of a significantly larger amount particles, which was used to imitate a physical object. Further, both methods were successful in simulating a tracing experiment for several different planned paths, using a single control point and manipulation point.

The physical system involved the setup of a robotic manipulator using a pivoting point calibration method, and an image registration software implemented to locate the accurate positions of the markers set on the physical object. The results from the physical system model showed a successful set point regulation and a box shape trajectory simulation. The

significant position error which was detected is believed to have resulted from an accumulation of camera error and object model parameter matching.

Next, using a same control method and model, a new assistive needle insertion model was developed which successfully reduced the movement of a target with respect to the insertion direction of the needle in the tissue. Two methods, including the static and dynamic paddle, were used to simulate the positive effects the procedures and an application of the deformable object model. For the static paddle approach, the optimal paddle was chosen based on the highest reduction in target lateral movement. This method was successful, although case specific, due to the fact that as the object characteristics were altered due to non-homogeneities, the optimal paddle position was shifted. As a result, a dynamic paddle approach was formulated to reposition the target back to its original position. This method was proved to be more robust, oblivious to whether the object characteristics changed.

7.2 Future Work

The future of this work can be branched into several areas. Regarding the object model, a quicker computation of the Gaussian Quadrature method would be beneficial as significantly larger deformations could be achieved. Also, more complicated object models should be implemented to include holes and non-homogeneities. This may require more complex maneuvering of the robotic manipulator. Next, the inclusion of multiple control and manipulation points could open a vast range of new applications for the model, include object manipulation in free space. It is likely that an optimization method to choose from a possibility of multiple paths, and a planned robotic manipulator trajectory will need to be implemented to accomplish these goals. Lastly, using the models strain and stress characteristics, an accurate model of tissue fracture and cracking could be accomplished as to

simulate surgical incision problems, or the behavior of an organ undergoing a cut out surgical procedure. Summing up all of these applications, the object model is very well suited to be implemented into a virtual surgery system for surgical trainers.

Appendix A

Appendix

A.1 Vision System

The stereo camera used was the Bumblebee 2 model from Point Grey Research Inc. This was accompanied by overlaying IR filters over the camera lens. The IR filters were obtained from Newport Corporation. With product number FSR-RG780 Color glass filters, 25.4 diameter.

A.2 Physical Object

The physical object was created using products purchased from:

M-F Manufacturing Co., INC.

P.O. Box 820442

Fort Worth, Texas 76182 – 0442

817 – 281 – 9488

4424 McLean Road

A.3 Matlab Robot Control

The overall system control is shown in Figure A.1, which incorporates the internet communications, robot control and tool calibration subsystems. Figures A.2, A.3 and A.4 show the individual components of the communication subsystem and coordinate transformations between tool and object space respectively.

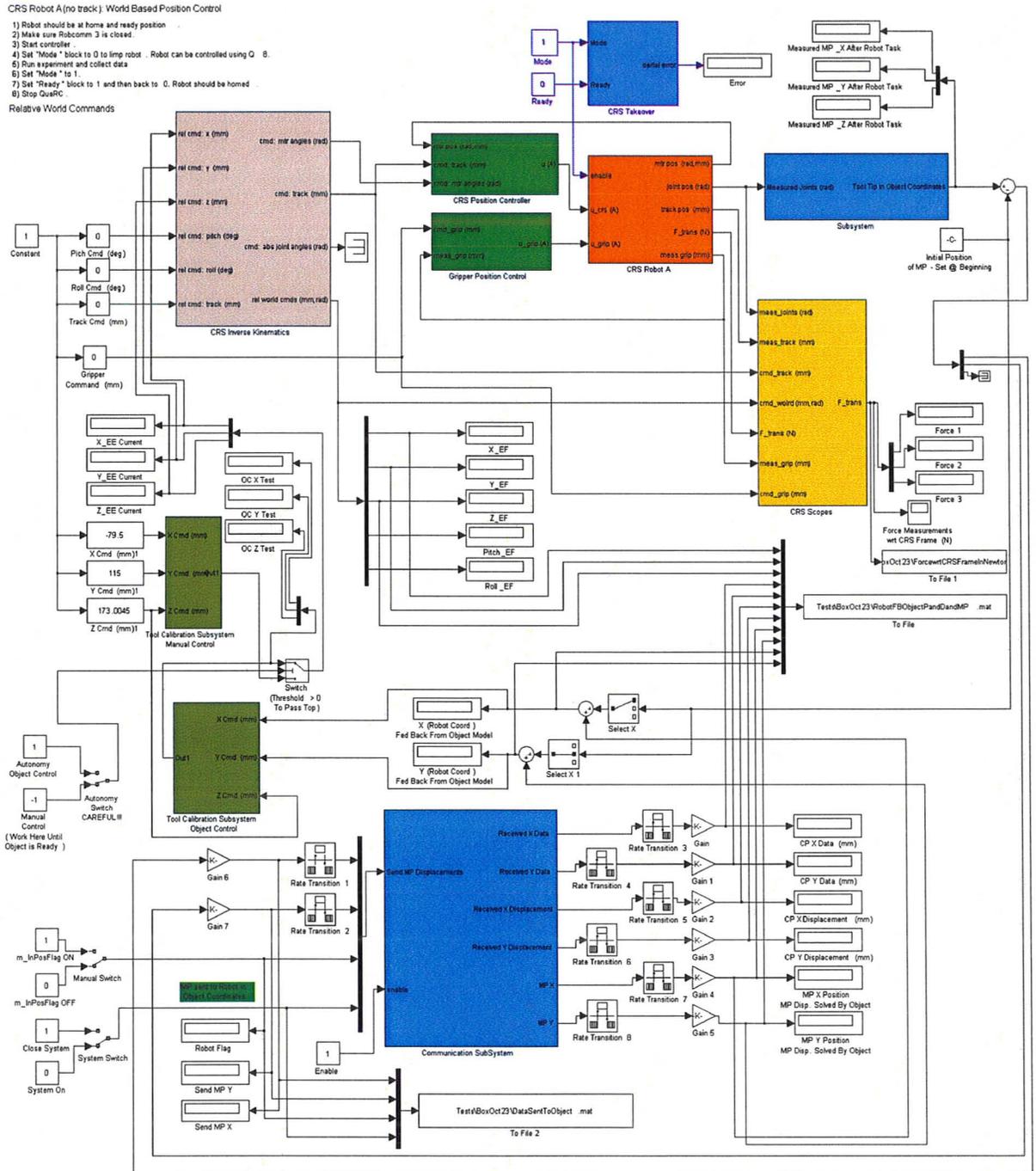


Figure A.1: Central control for communication between the robotic manipulator, stereo camera and object information.

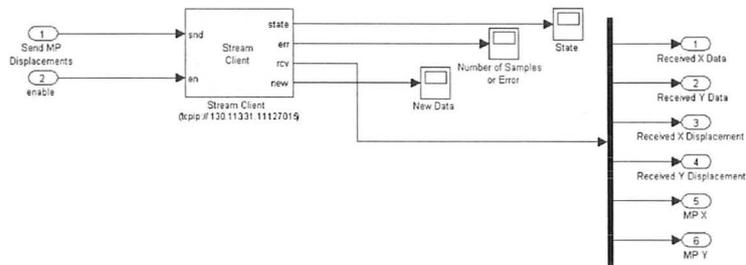


Figure A.2: TCP/IP communication block between robot and object model.

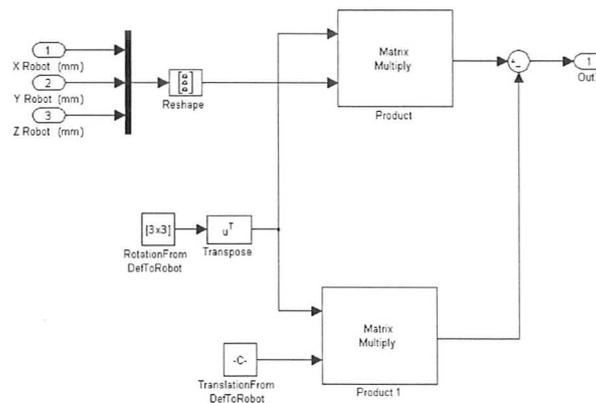


Figure A.3: Tool transformation from robot output coordinates to object input coordinates.

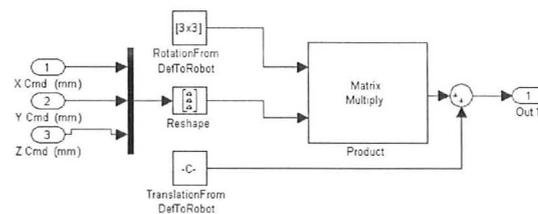


Figure A.4: Tool transformation from object output coordinates to robot input coordinates.

Bibliography

- Alterovitz, R.; Goldberg, K. (2003). Sensorless Planning for Medical Needle Insertion Procedures. *IEEE International Conference on Intelligent Robots and Systems*, pages 3337–3343.
- Alterovitz, R.; Goldberg, K. O. A. (2005). Planning for Steerable Bevel-tip Needle Insertion Through 2D Soft Tissue with Obstacles. *IEEE International Conference on Robotics and Automation*, pages 1652–1657.
- Arun, K.S.; Huang, T. S. B. S. D. (1987). Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on pattern analysis and machine intelligence*, **49**, 698–700.
- Belytschko, T.; Gu, L. L. Y. (1994). Fracture and crack growth by element free galerkin methods. *Modelling Simul. Mater. Sci. Eng.*, **2**, 519–534.
- Bicchi, A.; Kumar, V. (2000). Robotic Grasping and Contact: A Review. *IEEE International Conference on Robotics and Automation*, **1**, 348–353.
- Chanthasopeephan, T.; Desai, J. L. A. (2007). Modeling Soft-Tissue Deformation Prior to Cutting for Surgical Simulation: Finite Element Analysis and Study of Cutting Parameters. *IEEE Transactions on Biomedical Engineering*, **54**, 349–359.

- Chen, J.S.; Pan, C. W. C. L. W. (1996). Reproducing Kernel Particle Methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering*, **139**, 195–227.
- Chen, J.-S.; Pan, C. W. C.-T. (1997). Large deformation analysis of rubber based on a reproducing kernel particle method. *Computational Mechanics*, **19**, 211–227.
- Chen, J.S.; Yoon, S. W. C. (2002). Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, **58**, 2587–2615.
- Chen, C.;Zheng, Y. (1991). Deformation Identification and Estimation of One-Dimensional Objects by Using Vision Sensors. *IEEE International Conference on Robotics and Automations*, pages 2306–2311.
- Chen, M.Z.; Zheng, Y. (1993). Vibration-Free Movement of Deformable Beams by Robot Manipulator. *IEEE International Conference on Robotics and Automations*, **2**, 456–461.
- Cotin, S.; Delingette, H. A. N. (1999). Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions On Visualization and Computer Graphics*, pages 62–73.
- Cowin, S.C.; Doty, S. (2006). *Tissue Mechanics*. Springer.
- Craig, J.J; Raibert, M. (1981). Hybrid Position/Force Control of Manipulators. *J. Dyn. Systems, Meas. and Control, ASME Trans*, **102**, 126–133.
- DiMaio, S.P.; Salcudean, S. (2002a). Needle insertion modeling and simulation. *IEEE Transactions on Robotics and Automation*, **19**, 864–875.

- DiMaio, S.P.; Salcudean, S. (2002b). Simulated Interactive Needle Insertion. *Symp. On Haptic Interfaces For Virtual Envir. and Teleoperator Sysys.*, pages 344–351.
- DiMaio, S.P.; Salcudean, S. (2003). Needle Insertion Modeling and Simulation. *IEEE International Conference on Robotics and Automation*, **19**, 864–875.
- DiMaio, S.P.; Salcudean, S. (2005). Interactive Simulation of Needle Insertion Models. *IEEE Transactions on Biomedical Engineering*, **52**, 1167–1179.
- Doran, C.F.; McCormack, B. M. A. (2004). A Simplified Model to Determine the Contribution of Strain Energy in the Failure Process of Thin Biological Membranes during Cutting. *Strain*, **40**, 173–179.
- Gibson, S.F.F.; Mirtich, B. (1997). A Survey of Deformable Modeling in Computer Graphics.
- Glozman, D.; Shoham, M. (2007). Image-Guided Robotic Flexible Needle Steering. *IEEE Transactions on Robotics*, **23**, 459–467.
- Greminger, M.A.; Nelson, B. (2003). Deformable object tracking using the boundary element method. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, 289–294.
- Henrich, D.; Worn, H. E. (2000). Robot Manipulation of Deformable Objects.
- Hogan, H. (1987). Stable Execution of Contact Tasks Using Impedance Control. *IEEE International Conference on Robotics and Automation*, **4**, 1047–1054.
- Howard, A.M.; Bekey, G. (1997). Recursive learning for deformable object manipulation. *IEEE International Conference on Advanced Robotics*, pages 939–944.

- Howard, A.M.; Bekey, G. (1999). Intelligent learning for deformable object manipulation. *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 15–20.
- Khalil, F.F; Payeur, P. (2007). Robotic Interaction with Deformable Objects under Vision and Tactile Guidance - a Review. *IEEE International Workshop on Robotic and Sensors Environments*.
- Li, S.; Liu, W. (2002). Meshfree particle methods and their applications. *Applied Mechanics Review*, **55**, 1–34.
- Li, S.; Liu, W. (2004). *Meshfree Particle Methods*. Springer Berlin / Heidelberg.
- Liu, W.K.; Jun, S. Z. Y. (1995). Reproducing kernel particle methods. *Int. J. Numer. Methods Engrg.*, **20**, 1081–1106.
- Liu, G.R.; Gu, Y. (2005). *An Introduction to Meshfree Methods and Their Programming*. Springer.
- Liu, Y.-H. (2000). Computing n-Finger Form Closure Grasps on Polygon Objects. *The International Journal of Robotics Research*, **19**, 17–29.
- Lloyd, b.; Szekeley, G. H. M. (2007). Identification of spring parameters for deformable object simulation. *IEEE Transactions of Visualization and Computer Graphics*, **13**, 1081–1094.
- Mahvash, M.; Hayward, V. (2001). Haptic Rendering of Cutting: A Fracture Mechanics Approach. *The Electronic Journal of Haptics Research*, **2**, 1–12.

- Mallapragada, V.G.; Sarkar, N. P. V. (2007). A Robotic System for Real-time Tumor Manipulation During Image guided Breast Biopsy. *IEEE International Conference on Bioinformatics and Bioengineering*, pages 204–210.
- Mallapragada, V.G.; Sarkar, N. P. V. (2008). Robot Assisted Real-time Tumor Manipulation for Breast Biopsy. *IEEE International Conference on Robotics and Automation*, pages 2515–2520.
- Markenscoff, X.; Papadimitriou, C. (1989). Optimum grip of a polygon. *The International Journal of Robotics Research*, **8**, 17–29.
- Markenscoff, X.; Ni, L. P. C. (1990). The geometry of grasping. *The International Journal of Robotics Research*, **9**, 61–74.
- Maruyama, Y. (2000). Robotic applications for hot-line maintenance. *Industrial Robot: An International Journal*, **27**, 357–365.
- Meer, D.W.; Rock, S. (1994). Experiments in object impedance control for flexible objects. *IEEE International Conference on Robotics and Automation*, **2**, 1222–1227.
- Nakagaki, H.; Kitagaki, K. O. T. T.-H. (1997). Study of Deformation and Insertion Tasks of a Flexible Wire. *IEEE International Conference on Robotics and Automation*, **3**, 2397–2402.
- Nguyen, V. (1988). Constructing force-closure grasps. *The International Journal of Robotics Research*, **7**, 3–16.
- Ogden, R. (2007). Incremental Statics and Dynamics of Pre-Stressed Elastic Materials. In *Waves in Nonlinear Pre-Stressed Materials*, pages 1–26. Springer Vienna.

- Okamura, A.M.; Smaby, N. C. M. (2000). An Overview of Dexterous Manipulation. *IEEE International Conference on Robotics and Automation*, pages 255–262.
- Paloc, C.; Bello, F. K. R. D.-A. (2002). Online Multiresolution Volumetric Mass Spring Model for Real Time Soft Tissue Deformation. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2002*, pages 219–226. Springer Berlin / Heidelberg.
- Pincibono, G.; Delingette, H. A. N. (2001). Non-linear and anisotropic elastic soft tissue models for medical simulation. *IEEE International Conference on Robotics and Automation*, pages 1370–1375.
- Platt, J.C.; Barr, A. (1988). Constraint Methods for Flexible Models. *Computer Graphics*, **22**, 279–288.
- Saadat, M.; Nan, P. (2002). Industrial Applications of Automatic Manipulation for Flexible Materials. *Industrial Robot: An International Journal*, **29**, 434–442.
- Sundaraj, K.; Mendoza, C. L. C. (2002). A Fast Method to Simulate Virtual Deformable Objects with Force Feedback. *Seventh International Conference on Control, Automation, Robotics and Vision*, pages 413–418.
- Tanner, H.G.; Kyriakopoulos, K. (1999). Analysis of Deformable Object Handling. *IEEE International Conference on Robotics and Automation*, **4**, 2674–2679.
- Terzopoulos, D.; Witkin, A. (1988). Physically Based Models with Rigid and Deformable Components. *IEEE Computer Graphics and Applications*, **8**, 41–51.
- Terzopoulos, D.; Platt, J. B. A. F.-K. (1987). Elastically Deformable Models. *Computer Graphics*, **21**, 205–214.

- Tokumoto, S.; Fujita, Y. H. S. (1999). Deformation Modeling of Viscoelastic Objects for Their Shape Control. *IEEE International Conference on Robotics and Automation*, **1**, 767–772.
- Torabi, M.; Hauser, K. A. R. D.-V. G. K. (2009). Guiding Medical Needles Using Single-Point Tissue Manipulation. *IEEE International Conference on Robotics and Automation*, pages 2705–2710.
- Wada, T.; Hirai, S. K. S. (1998). Indirect Simultaneous Positioning Operations of Extensionally Deformable Objects. *IEEE International Conference on Intelligent Robots and Systems*, **2**, 1333–1338.
- Wada, T.; Hirai, S. K. S. (1999). Planning and control of indirect simultaneous positioning operation for deformable objects. *IEEE International Conference on Robotics and Automation*, **4**, 2572–2577.
- Wada, T.; Hirai, S. K. S. K.-N. (2001). Robust manipulation of deformable objects by a simple PID feedback. *IEEE International Conference on Robotics and Automation*, **1**, 85–90.
- Wakamatsu, H.; Hirai, S. (2004). Static Modeling of Linear Object Deformation Based on Differential Geometry. *The International Journal of Robotics Research*, pages 293–311.
- Yue, S.; Henrich, D. (2002). Manipulating deformable linear objects: sensor-based fast manipulation during vibration. *IEEE International Conference on Robotics and Automation*, pages 2467–2472.
- Zheng, Y.F.; Chen, M. (1993). Trajectory planning for two manipulators to deform flexible beams. *IEEE International Conference on Robotics and Automation*, **1**, 1019–1024.

- Zheng, Y.F.; Pei, R. C. C. (1991). Strategies for automatic assembly of deformable objects. *IEEE International Conference on Robotics and Automation*, **3**, 2598–2603.
- Zhong, Y.; Shirinzadeh, B. A. G. S.-J. (2005). A new methodology for deformable object simulation. *IEEE International Conference on Robotics and Automation*, pages 1914–1919.
- Zhong, Y.; Shirinzadeh, B. A. G. S.-J. (2006). A cellular neural network methodology for deformable object simulation. *IEEE Transactions on Information Technology in Biomedicine*, **10**, 749–762.