Distributed Control of a Segmented Telescope Mirror

by

Dan Kerley
B.Eng., University of Victoria, 2004

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Mechanical Engineering

Distributed Control of a Segmented Telescope Mirror

by

Dan Kerley

B.Eng., University of Victoria, 2004

Supervisory Committee

Dr. Edward Park (Department of Mechanical Engineering)

Supervisor

Dr. Afzal Suleman (Department of Mechanical Engineering)

Department Member

Dr. Panajotis Agathoklis (Department of Electrical & Computer Engineering)

Outside Member

Ms. Jennifer Dunn (Herzberg Institute of Astrophysics)

Additional Member

SUPERVISORY COMMITTEE

Dr. Edward Park (Department of Mechanical Engineering)

Supervisor

Dr. Afzal Suleman (Department of Mechanical Engineering)

Department Member

Dr. Panajotis Agathoklis (Department of Electrical & Computer Engineering)

Non-Department Member

Ms. Jennifer Dunn (Herzberg Institute of Astrophysics)

Additional Member

ABSTRACT

As astronomers continue to examine fainter objects and farther back in time, they require increasingly large telescopes due to the fundamental diffraction of optical elements. Therefore several of the next generation optical telescopes will employ extremely large primary mirrors. However to realistically construct mirrors of these magnitudes they will need to be assembled as a collection of many smaller mirrors. This mirror segmentation leads to the additional challenge of aligning the smaller mirror elements with respect to one another, and maintain that alignment in the presence of disturbances on the optical surface and its supporting structure. To achieve this alignment and disturbance rejection, a complex active control system will be required.

There are several possible solutions to the control problem ranging from fully decentralized control to a global control scheme. However since many of these segmented mirrors will be comprised of hundreds of mirror elements a global control scheme quickly becomes an intractable solution. On the other extreme, a highly scalable decentralized scheme is realizable, however, would lack any global sense of the system. Therefore an appealing solution is a scalable distributed network of controllers, where individual controllers ‘act locally’ yet ‘think globally’. This is achieved by coupling adjacent controller to one another, forming a lattice across the spatial extents of the system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

3 DOF	Three degrees-of-freedom, referring to displacement in a Cartesian coordinate system
6 DOF	Six degrees-of-freedom, referring to displacement and orientation in a Cartesian coordinate system
FEM	Finite element model/modeling, used to refer to both the process and the resulting model
LMI	Linear Matrix Inequality
RMS	Root Mean Square
SMCT	Segmented Mirror Control Testbed
TMT	Thirty Meter Telescope

ACKNOWLEDGEMENT

I would like to thank Dr. Edward Park for giving me the opportunity to be a part of his research team and for sharing his academic expertise and experience with me. I would like to show special appreciation to Jennifer Dunn for her continued support, understanding and encouragement. Also a special thanks to my lab-mates Kelly Sakaki, Kerem Karakoc, Kerem Gurses, Jung Keun Lee, William Liu, Vishalini Bundhoo and Kelly Stegman. And a heartfelt thank you to my parents for all their love and support throughout my life. To all my friends and family that have seen me through this, thank you.

DEDICATION

to my mom for all her help…

CHAPTER 1

INTRODUCTION

1.1 Motivation

The next generation of optical telescopes will consist of increasing large primary reflector mirrors, in the range of 20 to 100 meters in diameter [1] [2] [3]. The sheer size of the optical surface makes it infeasible to produce them as a single reflective surface. Therefore these primary mirrors will be comprised of an array of many smaller mirrors. For example, the highly anticipated Thirty Meter Telescope (TMT), shown in Figure 1, will consist of a 30 meter primary mirror made up of 492 segments [4]. Sophisticated active control will be required to stabilize these large and highly segmented optical surfaces from a variety of external disturbances, including: wind shake, gravity loading,

thermal effects, seismic and telescope tracking induced vibrations, and adaptive optics and instrument correction offloading.

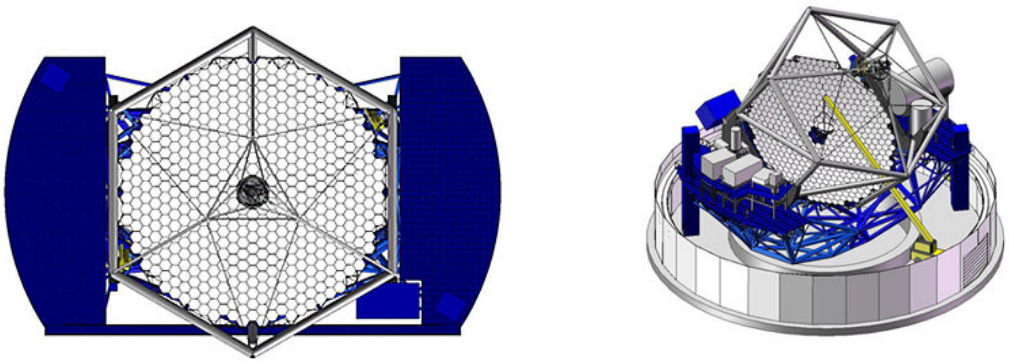

Figure 1. Renderings of the Thirty Meter Telescope (TMT) [1]

Due to the enormity of these next generation segmented mirrors, the number of actuators and sensors may run from several hundred to several thousand, depending on the number of segments. This creates the challenging problem of developing a tractable active alignment control system. Due to the shear dimensions of the system, a single monolithic global control scheme would be infeasible; instead a highly scalable control solution is desired.

A traditional solution to this control problem, which has been used on existing segmented telescopes is to have each segment controlled independently based on position error estimations from edge sensors [5][6]. The alternative extreme would be to implement a centralized control scheme where all segments are commanded by a single controller, which would improve global controllability and overall performance. However the use of

a centralized control scheme quickly becomes intractable as the number of segments increase due to the overwhelmingly large number of inputs and outputs of the resulting control system.

A viable and highly scalable solution is the development of a decentralized control scheme, where each segment is controlled by an independent controller. However since each segment controller acts independently we have sacrificed global performance for scalability.

An alternative solution is a compromise between the above two extremes where the resulting controller provides global control performance, while maintaining scalability and tractability. This alternative solution is based on the concept of modeling the highly segmented primary mirror as an interconnected network of spatially invariant subsystems, or spatially discrete units, where each unit represents an individual mirror segment. The spatially invariant condition means that a single discrete unit can be used to model any segment regardless of its position in the overall mirror surface. Each segment unit is dynamically coupled to each of its neighboring segment units. Such a system can be represented in a distributed manner based on the work by [7]. The resulting control scheme is then one that consists of a scalable network of distributed controllers, one controller unit for each segment unit, working cooperatively to achieve the required global performance.

1.2 Project Background

Existing control systems for segmented mirrors on telescopes are typically comprised of localized, actuator based control loops [8][9][10]. This approach allows for the simultaneous control of the mirror position actuators without the need for a large centralized control system, since each actuator is driven somewhat independently. The term ‘somewhat independently’ is used since the position error estimation is often achieved through relative edge sensors, therefore the localized control loops are coupled through the sensor readings. As telescope segmented mirrors increase in overall size the requirement for a scalable control system become even more important. However it is also expected that these extremely large next generation mirrors will need to bear increasingly large disturbances including those from wind buffeted on the larger surface area of the mirror [11][12]. The envisioned control system for several of these next generation segmented telescopes is an extension of the localized control loops deployed on their smaller predecessors [13][14][15].

However by implementing a more coupled control system, yet retaining a scalable solution, the overall disturbance rejection of the control system may be improved. The novel approach of distributed control of spatially invariant systems [7] is well suited to the control of a highly segmented mirror, since it provided a scalable control solution with global performance. One implementation of this scheme is such that each segment is controlled by a single distributed controller. These controllers are, however, not isolated, since they share state information with their neighboring controllers. This sharing of

information, or interconnection, allows for the control system to act globally since information is rippled throughout the entire system through the interconnected network. A caveat to this solution is the spatially invariant requirement, since for any realistic large scale segmented mirror, the dynamic response of the individual segments will vary over the extent of the mirror surface. This means the distributed controllers must be robust enough to support the variation in system dynamics yet still perform effectively for all segments.

This type of distributed control system has been applied to telescope control system design in the past [16], where it was used to design a possible control system for an adaptive secondary for the Atacama Telescope. The current work intends to extend upon this by the application of the distributed control techniques to a segmented primary. A major difference between this work and the previous work, is that the earlier work is controlling an adaptive mirror in which the reflective surface is a continuous structure. Therefore the individual actuators are coupled directly through the spatially continuous reflective surface. In this work the reflective surface is comprised of spatially discrete surfaces which are coupled through a common support structure. However the primary difference between this work and the previous attempts is the modeling approach used to develop the distributed model of the system, which is subsequently used to synthesize the distributed controller. With the adaptive secondary and other distributed control systems the distributed model is formulated from a small set of differential equations, however due to the complexity of the dynamics of a segmented mirror support structure this approach is infeasible. Therefore a distributed model technique was developed based on

the finite element method. This allows for the distributed modeling of very complex systems, including but not limited segment mirror support structures, where a finite element model (FEM) can be created using a commercial finite element software package and then post processed into the distributed modeling form.

In order to experimentally test control systems on a segmented mirror, the Segmented Mirror Control Testbed (SMCT) assembly was designed. Although there are several segmented mirror testbeds in use [17][18], the SMCT was designed for disturbance rejection control based experiments, in particular for distributed control. This project is being expanded into the Control System Integrated Modeling (CSIM) testbed project which will have a broader scope for control based experimentation on a segmented mirror.

Due to time constraints the SMCT was not fabricated and the distributed controller that was developed and analyzed as a simulation. The simulation software developed for the SMCT is based heavily off the Integrated Modeling Toolset, or IM, developed at HIA [19][20], which combined ANSYS, MATLAB and Zemax for end-to-end simulations of opto-mechanical systems. The SMCT simulation software is not intended as a replacement to the Integrated Modeling Toolset, but as a supplement to it, such that the distributed control scheme developed could be ported to the IM.

1.3 Research Objectives

In order to understand the complexities and interactions of controlling a segmented mirror the SMCT assembly, consisting of seven segments shown in Figure 2, was designed for control based experiments. This testbed project is being expanded and is continuing under the name of CSIM, as joint venture between the University of Victoria (UVic) and the Herzberg Institute of Astrophysics (HIA). The work presented here, however, is limited to the initial designs and simulations for the SMCT project.

In the SMCT assembly each segment is supported by three actuators and ringed with edge sensors to measure relative displacements between segments. The ultimate aim of this project is to advance the development of active optical control systems in the next-generation of very large optical telescopes.

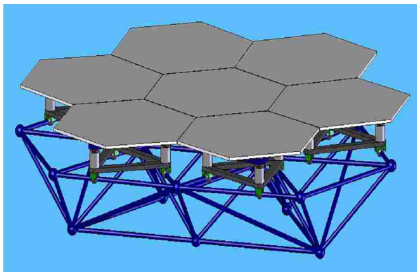

Figure 2. Initial Concept of the Segmented Mirror Control Testbed

The primary objective of this work it to apply distributed modeling of spatially invariant interconnected system to the SCMT assembly, and then use the resulting model to synthesis a scalable distributed control system. To this end, the SMCT was spatially

discretized, such that each segment in the system is modeled by a single distributed unit model that is directly coupled to neighboring distributed units. In addition to the distributed control system a global and decentralized control scheme was also developed to highlight the advantages and disadvantages of the distributed control system.

Due to the complexity of the SMCT structure, it was modeled using finite element methods. To develop the distributed unit model from the FEM a new technique was required. This technique is based on the concept of using information relating to bounded nodes, nodes that reside on neighbouring distributed units, to form the required interconnected linear model and is a simplification to the method developed in [21].

In order to synthesize a distributed controller based on the FEM, the model must first undergo temporal and spatial state reduction. Although similar in purpose to the conventional temporal state reduction methods, spatial state reduction required an alternative approach since spatial states cannot be treated the same as the standard temporal states in a state-space model. The spatial state reduction method developed for this research is based on the observation that neighbouring units are coupled directly and exclusively through the temporal states. Therefore the minimum order and sensitivity of spatial states is governed by the singular values of the linear transform from one unit's temporal states to another's.

As an initial step in the validation of the distributed control scheme applied to a segmented mirror, a closed loop control based simulation of the SMCT was developed.

This simulation analysis follows the work done for the development of the Integrated Modeling Toolset, and is intended to be convolved with the IM to add distributed control functionality.

1.4 Scope of Thesis

In Chapter 2 the parametric model of the SMCT assembly is introduced. This model defines the location of key nodes within the SMCT assembly based upon a collection of fundamental parameters. Though this model is explicitly discussed in the context of the single ring, the seven segment SMCT assembly, the general process can easily be expanded to accommodate any number of segment rings.

Chapter 3 presents the modeling techniques used to model the SMCT, including a standard global finite element model, and decentralized finite element model, and a distributed model. The global and decentralized FEMs are based on standard finite element techniques. However the distributed model is based on an augmentation to the standard finite element method, where bounded node information is used to couple distributed unit models together. Additionally reduction techniques are discussed for each of the three models, including the introduction of a spatial states reduction method for the distributed model.

In Chapter 4 the controller synthesis techniques for a global, decentralized and distributed control scheme are presented. Details are given as to various numeral issues related to the

synthesis procedure, however detailed proofs of the controllers' formulation are left as references to previous work.

Chapter 5 presents the closed loop simulation software for the SMCT and summarizes the preliminary simulation results. The SMCT simulation was developed in MATLAB/Simulink and supports a variety of controller configurations. However the results presented as a part of the current work only relate to simulations of continuous-time global, decentralized, and distributed control.

CHAPTER 2

SEGMENTED MIRROR GEOMETRIC LAYOUT

2.1 Chapter Overview

Since extremely large monolithic mirrors are not practical to manufacture, a viable alternative is to produce the same optical surface out of an array of small mirror segments. Commonly these mirror segments are roughly hexagonal in shape and arranged in a honeycomb like pattern, with a nominal separation between them [22][23][24], as is the case for the SMCT. The geometric layout of the support structure for these hexagonal segments is based on a series of projections of planes, where key nodes are defined at the intersection of these planes. The nodes are connected together by element member to form the structure as shown in Figure 3.

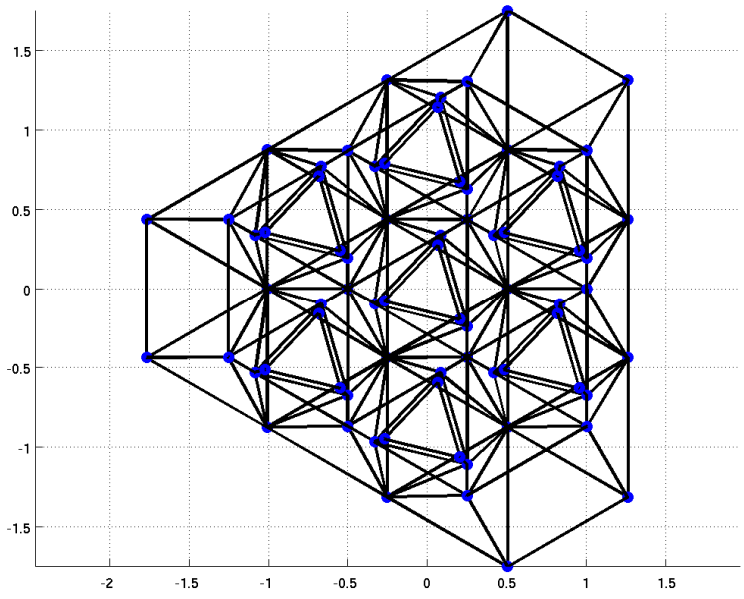

Figure 3. SMCT Nodes and Elements

2.2 Segment Layout

The method used to segment the primary mirror surface for this work is such that the segments and the inter-segment gaps are equal and regular over the entire primary surface when viewed for the perspective of the star light. That is to say the mirror segments are described in such way that when projected onto the plane perpendicular to the primary mirror optical axis, known as the optical axis plane, the segments appear as an array of regular and equally spaced hexagons, as depicted in Figure 4. This is a commonly used segmentation method, and was the method adopted by the KECK telescopes [25].

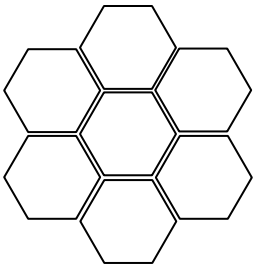

Figure 4. Segmented Primary Mirror Surface Projection

In the optical axis plane, two parameters govern the geometry of the projected segments: the segment side length and the segment gap length. The segment side length is defined as the distance between two adjacent corners of a projected segment, while the segment gap length is defined as the distance between two sides of adjacent projected segments, as shown in Figure 5.

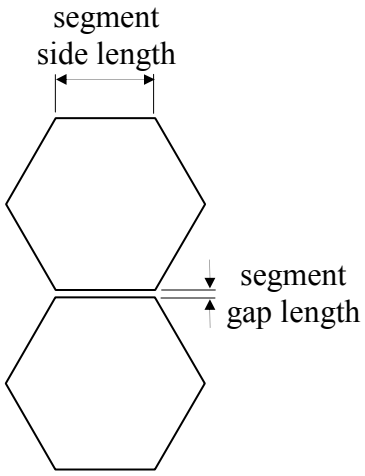

Figure 5. Segment Side and Gap Length

A true primary mirror surface fits an aspheric curve, defined by the radius of curvature and conical constant. With this segmentation method, the curvature of the primary surface leads to the individual mirror segments and their gap spacing being distorted as a function of its radius from the optical axis, as demonstrated by Figure 6. In addition to the distortion of the hexagonal shape of the segments, each segment will have an aspherical curve matched to its local curvature for its position in the segmented surface.

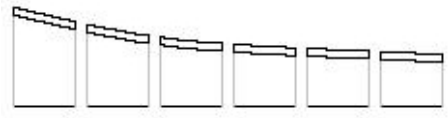

Figure 6. Segmented Primary Mirror Cross Section Projection

However producing distorted aspheric optical surfaces is cost prohibitive for the SMCT project and therefore identical flat non-optical segment blanks will be used in the SMCT testbed. Thus, for this work, it is assumed that all segments are identical flat true hexagons, positioned on a best fit location to the overall aspherical surface. A description of the best fit method used when positioning the segment blanks is given in the following sections.

2.3 Support Structure Layout

To define the support structure truss layout, first the projected segment geometry must be defined as depicted in Figure 4. Then a single point is defined in the centre of each projected segment in the optical axis plane. These points are referred to as the central

projection nodes. The central projection nodes for each segment are projected along a line parallel to the optical axis onto the aspherically curved primary mirror surface, defining a new set of points called the segment centre nodes. The planes tangent to the aspherically curved surface at the segment centre nodes are defined as the segment planes.

A second plane is defined for each of the segment planes, parallel to the segment plane, however, offset by a fixed distance know as the actuator length. These offset planes are known as the top truss planes and are depicted by the yellow shaded regions in Figure 7. The intersection of three adjacent top truss planes defines a top truss node; these top truss nodes, shown in red in Figure 7, define the geometry of the top surface of the segment support truss.

A line is defined for each top truss node, which passes through that node and whose dot product with respect to its three adjacent top truss planes is equal. These lines are known, for reasons that will be made clear shortly, as pyramid truss vectors. For each pyramid truss vector a plane is defined, known as a base truss plane, such that it is perpendicular to the pyramid truss vector and offset from the corresponding top truss node by distance known as the pyramid truss height. The intersection of three adjacent base truss planes defines a base truss node; these base truss nodes define the geometry of the base surface of the segment support truss. The base truss planes are shown as the grey shaded regions in Figure 8, while the base truss nodes are shown in green and the top truss nodes are shown in red. For each base truss plane, the three base truss nodes on that plane are tied

to the single top truss node along the corresponding pyramid truss vector, forming a triangular pyramid truss, as shown in Figure 8.

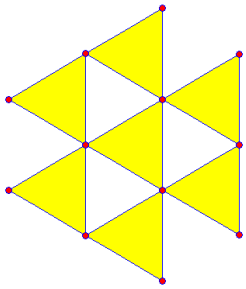

Figure 7. Top Truss Planes

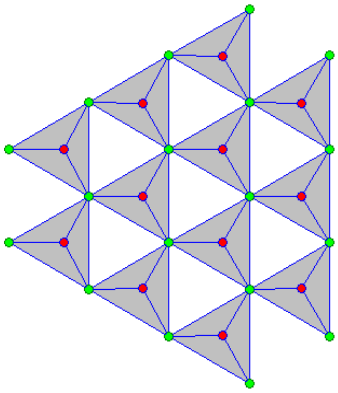

Figure 8. Base Truss Planes

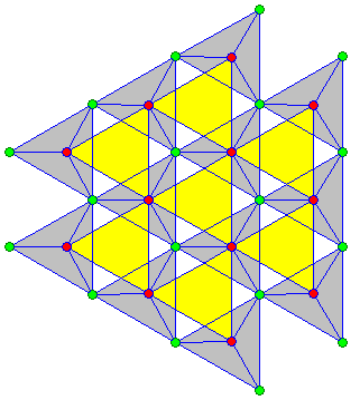

Figure 9. Support Structure Truss

Combining the top and base truss nodes define the total support truss, as depicted in Figure 9. This geometric pattern can be extended to the limits of the segmented curved surface.

2.4 Actuator Layout

For each top truss plane, the three top truss nodes on that plane are connected to form the triangular top truss in that plane. Since each segment is supported by three actuators, the base of one actuator will be connected to each of the three truss members of the triangular top truss. The actuator base connection nodes together are known as the actuator triad, and form a triangle inscribed within the triangular top truss. However due to the aspherical curvature rippled down through the truss structure layout from the primary surface curvature, the triangular top trusses are not equilateral triangles, with exception to the triangular top truss that the optical axis passes through. Yet it is desirable to have the actuator triad form an equilateral triangle with a side length specified as the actuator triad side length. Therefore the nominal actuator triad is centred at the geometric centre of the triangular top truss and rotated until a best fit location is found, then the actuator triad is distorted such that it is inscribed within the triangular top truss. The dimension and orientation, relative to a global reference frame, of each actuator triad is optimized to minimize the deviation from the central equilateral triangle which is defined as the segment which the optical axis passes through. This is done to minimize the variations in the structural dynamics and interconnections of each segment across the entire segmented surface; since spatial invariance is a base assumption in the distributed modeling technique discussed in Section 3.5.

Three points are defined on the surface of each segment forming an equilateral triangle which is referred to as the segment triad and are used to define the segment's position and

orientation plane. The side length of this triangle is specified as the segment triad side length. The segment triad also defines the connection points for the actuators tips, corresponding directly to the actuator triad, which defines the connection points for the base of the actuators.

2.4 Global Coordinate System

The global coordinate system is defined such that the z-axis is collinear with the optical axis of the segmented mirror and in the direction of the reflected light off the mirror's surface. The x and y-axes are defined as shown in Figure 10.

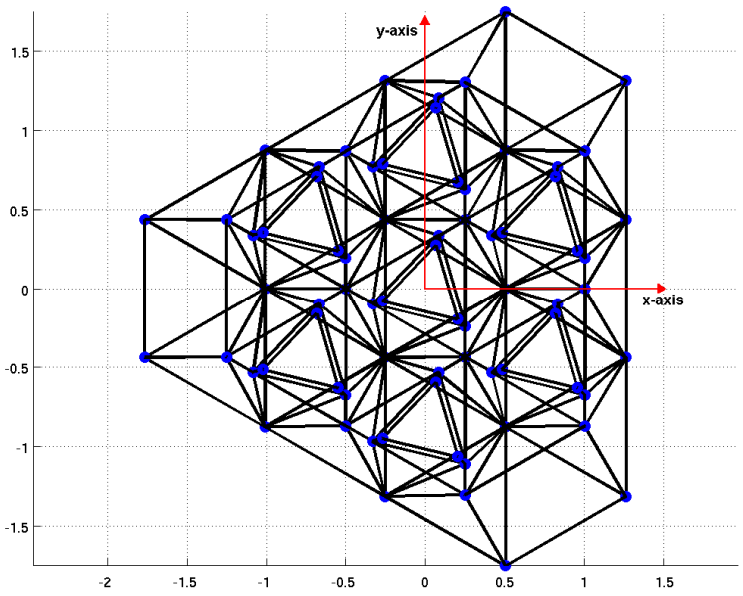

Figure 10. Global Coordinate System

2.5 Sensor Layout

In order to actively control the segmented surface, accurate information on the position and orientation of each segment is required. Position and orientation error estimations can be achieved by several different schemes.

Accurate alignment errors of individual segments and the surface as a whole can be achieved via optical means by use of wavefront sensing. This method is often used for the quasi-static phasing and alignment calibration of a segmented telescope's primary mirror [26].

The commonly used method for segment position error estimation is the use of relative edge sensing between segments, employed on the KECK telescopes [4] and is the baseline segment sensing scheme for TMT project [27]. Due to its usage on existing segmented telescopes this is the scheme selected for the SMCT project. In this scheme, relative displacement is measured between two points on adjacent segment edges by a linear displacement sensor. By mounting several sensors around the perimeter of each segment this allows for calculation of the segments tip, tilt and piston error relative to its neighboring segments. Each segment is effectively equipped with twelve edge sensors, two per segment side, as shown in Figure 11. Adjacent segments share the two edge sensors that are common between the segments' bordering sides.

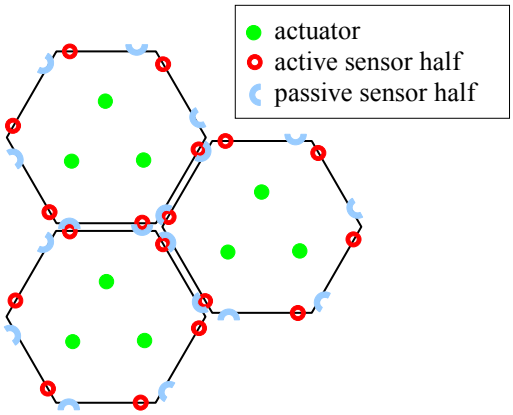

Figure 11. Edge Sensor and Actuator Layout

As shown in Figure 11, two nodes are defined per edge sensor. These nodes represent the points on the segment on which one half of the sensor is mounted, such that one half of the sensor is mounted on one segment and the other half is mounted on an adjacent segment. In terms of the SMCT the edge sensor locations are governed by a parameter called the sensor corner offset, which is the distance from the sensor node to the nearest corner of the segment.

This geometry gives redundant information as to the relative alignment of the segment and therefore can be used to estimate the required actuator displacement to bring the segment into relative alignment with its neighboring segments. The shortcoming of the relative edge sensing scheme is that it is completely blind to the 6 DOF global motion of the segmented surface and therefore aligning the segments based on the relative errors may result in a global tip, tilt, and piston error that are unobserved, or poorly observed, by the edge sensors. These issues are discussed in more detail in Section 3.6.

CHAPTER 3

SEGMENTED MIRROR MODELING

3.1 Chapter Overview

There are three modeling methods utilized to model the SMCT. The first was a standard FE state-space modeling technique of the entire system. This model is referred to as the global model, and was used as the system plant in all simulations. The second decentralized model uses the same method as the global model; however only represents a single isolated segment and its support structure. The third modeling technique was based on state-space representation for spatially invariant interconnected distributed modeling [7]. The modeling technique developed to produce this state-space representation is an augmentation of the FE state-space modeling technique used in the other two models. The decentralized and distributed models are developed for the sole purpose of synthesizing controllers, while the global model is used for both controller synthesis and as the system plant in simulations.

3.2 Finite Element Modeling

A finite element model (FEM) was developed assuming each connection between nodes was a single cylindrical rod element and was modeled as lumped mass grid elements. Proportion damping was used as a simple approximation for the damping matrix.

The mass and stiffness element matrices shown below are defined for a local coordinate system where the x-axis is collinear with the element member, which is to say that the x-axis points from one element node to the other. The total mass of the element, m_e, is defined in (1), where r_e is the element's cross-sectional radius, l_e is the length of the element, ρ_e is the material density of the element. The principal mass moments of inertia, I_{xx}, I_{yy}, and I_{zz}, about the x, y, and z-axes respectively for a solid cylinder are defined in (2) and (3) respectively.

$$m_e = \pi r_e^2 \rho_e l_e \qquad \textbf{(1)}$$

$$I_{xx} = \frac{m_e r_e^2}{2} \qquad \textbf{(2)}$$

$$I_{yy} = I_{zz} = \frac{m_e \left(3r_e^2 + l_e^2\right)}{12} \qquad \textbf{(3)}$$

I_{yy} is equal to I_{zz} since the element is assumed to symmetric in the cross-sectional plane of the element, and therefore the y and z-axes are arbitrarily define in this plane. The resulting lumped mass element equation is shown below, where f_k and x_k are the 6 DOF

force and displacement, respectively, at node k. k_a and k_b are the node numbers that define the two ends of the element.

$$M_{e_{aa}} = M_{e_{bb}} = \begin{bmatrix} \frac{m_e}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{m_e}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{m_e}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix}_{6\times 6} \quad \textbf{(4)}$$

$$M_e = \begin{bmatrix} M_{e_{aa}} & 0 \\ 0 & M_{e_{bb}} \end{bmatrix}_{12\times 12} \quad \textbf{(5)}$$

$$f_k = \begin{bmatrix} f_{kx} \\ f_{ky} \\ f_{kz} \\ \tau_{kx} \\ \tau_{ky} \\ \tau_{kz} \end{bmatrix}_{6\times 1} \qquad x_k = \begin{bmatrix} d_{kx} \\ d_{ky} \\ d_{kz} \\ \phi_{kx} \\ \phi_{ky} \\ \phi_{kz} \end{bmatrix}_{6\times 1} \quad \textbf{(6)}$$

$$\begin{bmatrix} f_{k_a} \\ f_{k_b} \end{bmatrix} = M_e \begin{bmatrix} \ddot{x}_{k_a} \\ \ddot{x}_{k_b} \end{bmatrix} \quad \textbf{(7)}$$

The polar moment of inertia, J_e, and the area moment of inertia about the x-axis, I_x, for a solid cylinder is defined in (8) and (9) respectively.

$$J_e = \frac{\pi r_e^4}{2} \quad \textbf{(8)}$$

$$I_x = \frac{\pi r_e^4}{4} \quad \textbf{(9)}$$

The grid element equations, as shown below, are used in modeling the element stiffness matrix, where E_e is the material young's modulus of the element, G_e, is the material shear modulus of the element.

$$K_{e_{aa}} = \begin{bmatrix} \frac{\pi r_e^2 E_e}{l_e^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12 I_x E_e}{l_e^3} & 0 & 0 & 0 & \frac{6 I_x E_e}{l_e^2} \\ 0 & 0 & \frac{12 I_x E_e}{l_e^3} & 0 & \frac{6 I_x E_e}{l_e^2} & 0 \\ 0 & 0 & 0 & \frac{G_e J_e}{l_e} & 0 & 0 \\ 0 & 0 & \frac{6 I_x E_e}{l_e^2} & 0 & \frac{4 I_x E_e}{l_e} & 0 \\ 0 & \frac{6 I_e E_e}{l_e^2} & 0 & 0 & 0 & \frac{4 I_x E_e}{l_e} \end{bmatrix}_{6\times 6} \quad \textbf{(10)}$$

$$K_{e_{ab}} = \begin{bmatrix} -\frac{\pi r_e^2 E_e}{l_e^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12 I_x E_e}{l_e^3} & 0 & 0 & 0 & \frac{6 I_x E_e}{l_e^2} \\ 0 & 0 & -\frac{12 I_x E_e}{l_e^3} & 0 & \frac{6 I_x E_e}{l_e^2} & 0 \\ 0 & 0 & 0 & -\frac{G_e J_e}{l_e} & 0 & 0 \\ 0 & 0 & -\frac{6 I_x E_e}{l_e^2} & 0 & \frac{2 I_x E_e}{l_e} & 0 \\ 0 & -\frac{6 I_x E_e}{l_e^2} & 0 & 0 & 0 & \frac{2 I_x E_e}{l_e} \end{bmatrix}_{6\times 6} \quad \textbf{(11)}$$

$$K_{e_{ba}} = \begin{bmatrix} -\frac{\pi r_e^2 E_e}{l_e^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12 I_x E_e}{l_e^3} & 0 & 0 & 0 & -\frac{6 I_x E_e}{l_e^2} \\ 0 & 0 & -\frac{12 I_x E_e}{l_e^3} & 0 & -\frac{6 I_x E_e}{l_e^2} & 0 \\ 0 & 0 & 0 & -\frac{G_e J_e}{l_e} & 0 & 0 \\ 0 & 0 & \frac{6 I_x E_e}{l_e^2} & 0 & \frac{2 I_x E_e}{l_e} & 0 \\ 0 & \frac{6 I_x E_e}{l_e^2} & 0 & 0 & 0 & \frac{2 I_x E_e}{l_e} \end{bmatrix}_{6\times 6} \quad \textbf{(12)}$$

$$K_{e_{bb}} = \begin{bmatrix} \frac{\pi r_e^2 E_e}{l_e^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12 I_x E_e}{l_e^3} & 0 & 0 & 0 & -\frac{6 I_x E_e}{l_e^2} \\ 0 & 0 & \frac{12 I_x E_e}{l_e^3} & 0 & -\frac{6 I_x E_e}{l_e^2} & 0 \\ 0 & 0 & 0 & \frac{G_e J_e}{l_e} & 0 & 0 \\ 0 & 0 & -\frac{6 I_x E_e}{l_e^2} & 0 & \frac{4 I_x E_e}{l_e} & 0 \\ 0 & -\frac{6 I_x E_e}{l_e^2} & 0 & 0 & 0 & \frac{4 I_x E_e}{l_e} \end{bmatrix}_{6\times 6} \quad \textbf{(13)}$$

$$K_e = \begin{bmatrix} K_{e_{aa}} & K_{e_{ab}} \\ K_{e_{ba}} & K_{e_{bb}} \end{bmatrix}_{12\times 12} \quad \textbf{(14)}$$

$$\begin{bmatrix} f_{k_a} \\ f_{k_b} \end{bmatrix} = K_e \begin{bmatrix} x_{k_a} \\ x_{k_b} \end{bmatrix} \quad \textbf{(15)}$$

Since proportional damping was used, the element damping matrix is defined as shown in (16), with the additional constraint in (18) that bounds the damping matrix.

$$D_e = \alpha_e M_e + \beta_e K_e \quad \textbf{(16)}$$

$$\begin{bmatrix} f_{k_a} \\ f_{k_b} \end{bmatrix} = D_e \begin{bmatrix} \dot{x}_{k_a} \\ \dot{x}_{k_b} \end{bmatrix} \quad \textbf{(17)}$$

$$1 = \alpha_e + \beta_e \quad \textbf{(18)}$$

As stated above the element mass, stiffness and damping matrices are defined in a local coordinated system, such that the local x-axis is collinear with the beam element. However, to construct the system mass, stiffness and damping matrices, these individual element matrices must be rotated from their local coordinate system to the global coordinate system. This is achieved by first defining a unit vector v_k, shown in (19), that is aligned to the local x-axis of element in the global coordinate system, where p_{k_a} and p_{k_b} are the nominal positions of the two ends of the element in the global coordinate system.

$$v_k = \begin{bmatrix} v_{kx} \\ v_{ky} \\ v_{kz} \end{bmatrix} = \frac{p_{k_b} - p_{k_a}}{\left| p_{k_b} - p_{k_a} \right|} \quad \textbf{(19)}$$

A rotation matrix is then defined by rotating about the global z-axis by φ_k, such that the intermediary x-axis is aligned with the projection of v_k onto the global xy-plane; where φ_k is the angle between the global x-axis and the projection of v_k onto the global xy-plane. Then the resulting intermediary coordinate system is then rotated about its y-axis by θ_k to align its x-axis with v_k; where θ_k is the angle between v_k and the projection of v_k onto the global xy-plane.

$$\varphi_k = \arctan\frac{v_{ky}}{v_{kx}} \tag{20}$$

$$\theta_k = -\arctan\frac{v_{kz}}{\sqrt{\left(v_{kx}\right)^2 + \left(v_{ky}\right)^2}} \tag{21}$$

$$\begin{aligned} R_k &= \begin{bmatrix} \cos\theta_k & 0 & -\sin\theta_k \\ 0 & 1 & 0 \\ \sin\theta_k & 0 & \cos\theta_k \end{bmatrix}_{3\times3} \begin{bmatrix} \cos\varphi_k & \sin\varphi_k & 0 \\ -\sin\varphi_k & \cos\varphi_k & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3\times3} \\ &= \begin{bmatrix} \cos\theta_k\cos\varphi_k & \cos\theta_k\sin\varphi_k & -\sin\theta_k \\ -\sin\varphi_k & \cos\varphi_k & 0 \\ \sin\theta_k\cos\varphi_k & \sin\theta_k\sin\varphi_k & \cos\theta_k \end{bmatrix}_{3\times3} \end{aligned} \tag{22}$$

The following transformation matrix converts from the global coordinate system to the element's local coordinate, as denoted by the superscript G and subscript E respectively.

$$R_G^E = \begin{bmatrix} R_k & 0 & 0 & 0 & 0 & 0 \\ 0 & R_k & 0 & 0 & 0 & 0 \\ 0 & 0 & R_k & 0 & 0 & 0 \\ 0 & 0 & 0 & R_k & 0 & 0 \\ 0 & 0 & 0 & 0 & R_k & 0 \\ 0 & 0 & 0 & 0 & 0 & R_k \end{bmatrix}_{12\times12} \tag{23}$$

$$R_E^G = \left(R_G^E\right)^* \tag{24}$$

Therefore element mass, stiffness and damping matrices in the global coordinate systems, M_{e^G}, K_{e^G}, and D_{e^G} respectively, are:

$$M_{e^G} = R_E^G M_e R_G^E \tag{25}$$

$$K_{e^G} = R_E^G D_e R_G^E \tag{26}$$

$$D_{e^G} = R_E^G D_e R_G^E \tag{27}$$

All the element mass, stiffness and damping matrices in the global coordinate systems are assembled together, by superposition of the individual element matrices

[28], to form the systems mass, stiffness and damping matrices, M_g, K_g and D_g respectively, such that:

$$M_g\ddot{x}_g + D_g\dot{x}_g + K_g x_g = f_g \qquad \textbf{(28)}$$

where

$$x_g = \begin{bmatrix} x_1 \\ \vdots \\ x_{n_g} \end{bmatrix}_{6n_g \times 1} \qquad f_g = \begin{bmatrix} f_1 \\ \vdots \\ f_{n_g} \end{bmatrix}_{6n_g \times 1} \qquad \textbf{(29)}$$

where n_g is the total number of nodes.

3.2.1 Support Structure Modeling

The support structure is the truss assembly that supports and interconnects the individual segment-actuator assemblies, as shown in Figure 3. A segment-actuator assembly consists of a single segment propped up by three actuators. Each segment-actuator assembly interfaces to the support structure at three points where the three actuators mount to the support structure. Each truss element is modeled as a solid cylindrical beam element, as described in detail above.

3.2.2 Segment Modeling

A segment represents the reflective surface and the associated mounting brackets and supports for a single optical element in the segmented surface. For the SMCT the segments are represented as a solid, flat, non-optical hexagonal plates. The primary reason for this simplification is cost associated with producing large optical surfaces. In many segmented mirror designs the interface between the actuators and the segment would require elaborate mounting and pre-tensioning and to reduce actuator print-through on the segment surface and statically position the segment [29]. As an example, shown in Figure 12 is a segment support structure designed for TMT, which interfaces the actuators with the optical surface.

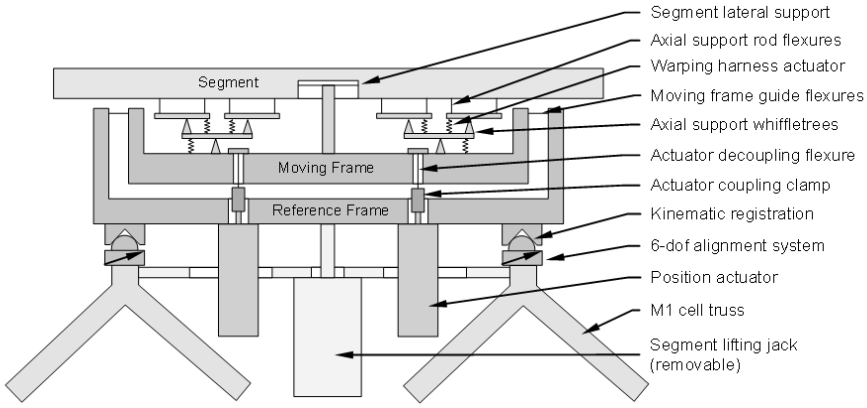

Figure 12. TMT Segment Support Structure [29]

However since actuator print-through is out of the scope of the SMCT project the actuators will be connected directly to the segment plate with a ball or pin joint. This joint may also require a flexure to allow for tip and tilt motion of the segment, as is

described below in Section 3.2.3, however, this is left as a future design detail to be decided upon as the SMCT mechanical design matures, and is not currently modeled.

A simplistic approach to defining the finite element model for a segment was taken as a first step to producing the mathematical models appropriate for the control synthesis processes described in Chapter 4. Each segment was modeled as an assembly of three solid cylindrical beam elements that form an equilateral triangle, whose vertexes are at segment triad points discussed in Section 2.3. Figure 13 depicts the segment beam elements superimposed on a segment plate.

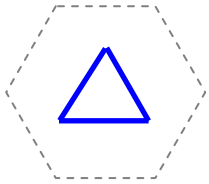

Figure 13. Segment Beam Model

The diameter of the beam elements was set such that the total volume of the three beam elements is equal to the total volume of the solid, flat, hexagonal plates it is intended to represent. This was done to maintain the same mass distribution, and act as a rough approximation of the stiffness of the segment plate and the associated actuator mounting components, since there are several undefined mechanical details with respect to the segment-actuator interface. The segment beam elements themselves are handled in the same manner as the support structure elements.

3.2.3 Actuator Modeling

Similar to the other elements in the system, the actuators are modeled as stiff solid cylindrical beam elements. To extend or contract the actuator an equal and opposite force is applied at either of the beam element, along the line of the element. This approach is used to model the reactionary effects of the actuator on both the segment surface and the support structure [19].

A complication with modeling the actuator as a rigid beam is the bending constraint that this added. If the three actuators for a given segment are constrained in all degrees of freedom, at both the base where they connect to the support structure, and the tip where they connect to the segment, the resulting model with be fully constrained. That is if a single actuator is extended or contracted this will require either or both of the support structure and segment to deform. To avoid this over constrained problem, the actuators are allowed to freely rotate in the plane which both the vector along the actuator and a vector from the actuator to the centre of the segment lay in. In a physical sense, this is equivalent to a parallel pin joint at the base of the actuator and a ball joint at the tip of the actuator. These joints allow for one, or more, actuator to extend or contract, causing a tilt on the segment, without causing a static deformation of either the segment or the support structure. This is pictorially represented is Figure 14, which is a cross-sectional view of a segment in its nominal position is superimposed with the segments displaced by the extension of one actuator.

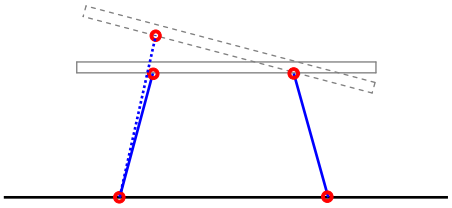

Figure 14. Actuator Pin Joint Extension

The axis of rotation of the pin joint is assumed to be parallel with the y-axis of the element based coordinate system. Therefore the pin joint is modeled by removing the bending terms about the y-axis in the element stiffness matrix, resulting in matrices shown in (30) to (33). The ball joint is modeled similarly but the bending terms about the x and z axes are also removed.

$$K_{e_{aa}} = \begin{bmatrix} \frac{\pi r_e^2 E_e}{l_e^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12 I_x E_e}{l_e^3} & 0 & 0 & 0 & \frac{6 I_x E_e}{l_e^2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{G_e J_e}{l_e} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{6 I_e E_e}{l_e^2} & 0 & 0 & 0 & \frac{4 I_x E_e}{l_e} \end{bmatrix}_{6\times 6} \quad \textbf{(30)}$$

$$K_{e_{ab}} = \begin{bmatrix} -\frac{\pi r_e^2 E_e}{l_e^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12 I_x E_e}{l_e^3} & 0 & 0 & 0 & \frac{6 I_x E_e}{l_e^2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{G_e J_e}{l_e} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{6 I_x E_e}{l_e^2} & 0 & 0 & 0 & \frac{2 I_x E_e}{l_e} \end{bmatrix}_{6\times 6} \quad \textbf{(31)}$$

$$K_{e_{ba}} = \begin{bmatrix} -\frac{\pi r_e^2 E_e}{l_e^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12 I_x E_e}{l_e^3} & 0 & 0 & 0 & -\frac{6 I_x E_e}{l_e^2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{G_e J_e}{l_e} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{6 I_x E_e}{l_e^2} & 0 & 0 & 0 & \frac{2 I_x E_e}{l_e} \end{bmatrix}_{6\times 6} \quad \textbf{(32)}$$

$$K_{e_{bb}} = \begin{bmatrix} \frac{\pi r_e^2 E_e}{l_e^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12 I_x E_e}{l_e^3} & 0 & 0 & 0 & -\frac{6 I_x E_e}{l_e^2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{G_e J_e}{l_e} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{6 I_x E_e}{l_e^2} & 0 & 0 & 0 & \frac{4 I_x E_e}{l_e} \end{bmatrix}_{6\times 6} \quad \textbf{(33)}$$

The element mass and damping matrices are formed the same way as presented above. However the global to element rotation matrix R_k in (22) must be modified since the z-y plane of the element is no longer arbitrary. The axis of rotation of the pin joint in the global coordinate system is defined by (34), where p_{k_c} is the geometric centre of the segment.

$$j_k^G = \frac{\left(p_{k_b} - p_{k_a}\right) \times \left(p_{k_b} - p_{k_c}\right)}{\left|\left(p_{k_b} - p_{k_a}\right) \times \left(p_{k_b} - p_{k_c}\right)\right|} = \frac{v_k \times \left(p_{k_b} - p_{k_c}\right)}{\left|v_k \times \left(p_{k_b} - p_{k_c}\right)\right|} \quad \textbf{(34)}$$

This axis is rotated into the local coordinate system defined by the transform R_k as shown in (35). Note that there is no local x-axis component since the pin axis of rotation is perpendicular to the local x-axis along the line of the actuator element.

$$j_k^L = \begin{bmatrix} 0 \\ j_{ky}^L \\ j_{kz}^L \end{bmatrix} = R_K j_k^G \qquad \textbf{(35)}$$

The angle between pin axis of rotation and the local y-axis is ϕ_k, shown in (36). Therefore the rotation matrix R_k must be further rotated by ϕ_k about the local x-axis to align the local y-axis and pin axis of rotation. The resulting modified rotation matrix $\tilde{R}_k$, shown in (37), is used in place of R_k when transforming actuator element system sub-matrices form the local coordinate system the global coordinate system.

$$\phi_k = \arctan \frac{j_{kz}^L}{j_{ky}^L} \qquad \textbf{(36)}$$

$$\tilde{R}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_k & \sin\phi_k \\ 0 & -\sin\phi_k & \cos\phi_k \end{bmatrix}_{3\times 3} R_k \qquad \textbf{(37)}$$

3.3 System Plant Modeling

The FEM system matrices, discussed in Section 3.2, are generated based on the geometric layout of the entire system described in Section 2. The system matrices are bound by the exterior base nodes of the support truss, as shown in red in Figure 15. The primary reason for bounding these nodes is that they are the anticipated mounting points of the physical testbed to the fixed frame of reference.

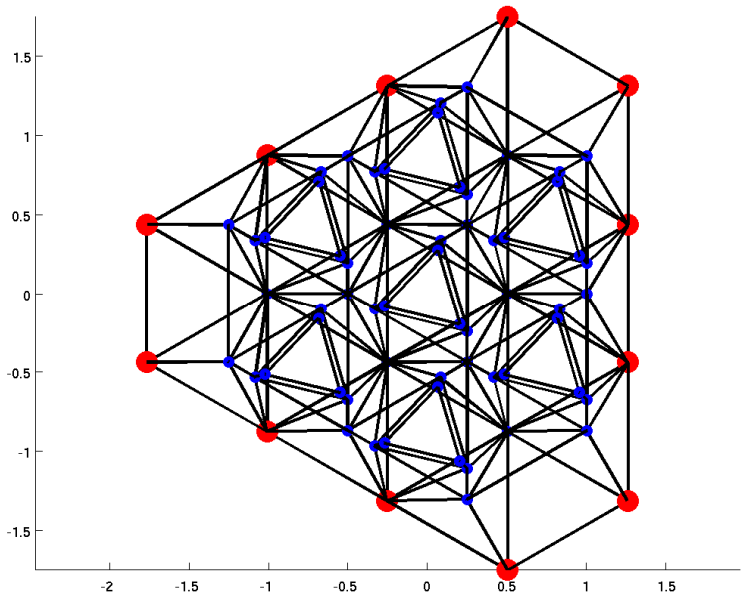

Figure 15. Plant Model Bounded Nodes

These bounded nodes are removed from the system matrices by the permutation matrix P_g to form the bound system matrices M_p, K_p, and D_p, as shown in equations (39) to (41), such that the position and orientation of these nodes are fixed in space.

$$x_p = P_g x_g \quad \textbf{(38)}$$

$$M_p = P_g M_g \left(P_g\right)^* \quad \textbf{(39)}$$

$$D_p = P_g D_g \left(P_g\right)^* \quad \textbf{(40)}$$

$$K_p = P_g K_g \left(P_g\right)^* \quad \textbf{(41)}$$

These matrices are then converted to a state-space form in the standard way as shown in (42), where the superscript P denotes the open-loop plant system model.

$$\begin{bmatrix}\begin{bmatrix}\dot{x}_p\\ \ddot{x}_p\end{bmatrix}\\ z^P\end{bmatrix}=\begin{bmatrix}\begin{bmatrix}0 & I\\ -M_p^{-1}K_p & -M_p^{-1}D_p\end{bmatrix} & \begin{bmatrix}0\\ M_p^{-1}\end{bmatrix}P_p\\ P_p^{-1}\begin{bmatrix}I & 0\end{bmatrix} & 0\end{bmatrix}\begin{bmatrix}\begin{bmatrix}x_p\\ \dot{x}_p\end{bmatrix}\\ d^P\end{bmatrix} \quad \textbf{(42)}$$

$$x_t^P=\begin{bmatrix}x_p\\ \dot{x}_p\end{bmatrix}_{12n_p\times 1} \quad \textbf{(43)}$$

$$\begin{bmatrix}A^P & B_d^P\\ C_z^P & D_{zd}^P\end{bmatrix}=\begin{bmatrix}\begin{bmatrix}0 & I\\ -M_p^{-1}K_p & -M_p^{-1}D_p\end{bmatrix}_{12n_p\times 12n_p} & \begin{bmatrix}0\\ M_p^{-1}P_p\end{bmatrix}_{12n_p\times 3n_p}\\ \begin{bmatrix}\left(P_p\right)^* & 0\end{bmatrix}_{3n_p\times 12n_p} & 0_{3n_p\times 3n_p}\end{bmatrix} \quad \textbf{(44)}$$

$$\therefore\begin{bmatrix}\dot{x}_t^P\\ z^P\end{bmatrix}=\begin{bmatrix}A^P & B_d^P\\ C_z^P & D_{zd}^P\end{bmatrix}\begin{bmatrix}x_t^P\\ d^P\end{bmatrix} \quad \textbf{(45)}$$

The matrix P_p is a permutation matrix, defined below, so the system includes only the 3 DOF force inputs d^P and displacement outputs z^P of all nodes in the global coordinate system, excluding the moment inputs and the angular outputs of the system.

$$P_p=\begin{bmatrix}\tilde{I} & & 0\\ & \ddots & \\ 0 & & \tilde{I}\end{bmatrix}_{6n_p\times 3n_p} \quad \textbf{(46)}$$

$$\tilde{I}=\begin{bmatrix}I_{3\times 3}\\ 0\end{bmatrix}_{6\times 3} \quad \textbf{(47)}$$

The system model is then modified to include control signals, as shown in (48).

$$\begin{bmatrix}\dot{x}_t^P\\ z^P\end{bmatrix}=\begin{bmatrix}A^P & B_d^P & B_u^P\\ C_z^P & D_{zd}^P & D_{zu}^P\end{bmatrix}\begin{bmatrix}x_t^P\\ d^P\\ u^P\end{bmatrix} \quad \textbf{(48)}$$

The vector u^P , show in equation (49), is the actuator control forces. The matrix B_u^P maps the control forces to a change in $\dot{x}_t^P$. Since the actuators have no direct feed through effect on the nodal displacements, the matrix D_{zu}^P is zero.

$$u^P = \begin{bmatrix} u_1 \\ \vdots \\ u_k \\ \vdots \\ u_{n_a} \end{bmatrix}_{n_a \times 1} \quad \textbf{(49)}$$

where n_a is the total number of actuators , and u_k is the magnitude and sign of the control force on the k^{th} actuator, with the convention that a positive sign specifies an extension force and a negative sign specifies a contraction force on the actuator. The control force u_k applies an equal and opposite force to either end of the k^{th} actuator along the line of action of the actuator, which results in extending or contracting motion of the actuator. In other words, based on the element unit vector v_k defined in equation (19), the control force u_k is applied to node k_a in the positive v_k direction and the to k_b in the negative v_k direction, where k_a and k_b are the movable tip of the actuator and relative fixed base of the actuator respectively. Therefore the control force is defined along the x-axis of the local coordinate system, which is collinear with the line of action v_k of the actuator element. The control force is transformed into the global coordinate system by the rotation matrix R_k define in equations (20) to (22), such that:

$$T_k^u = \left(R_k\right)^* \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta_k \cos\varphi_k \\ \cos\theta_k \sin\varphi_k \\ -\sin\theta_k \end{bmatrix}_{3\times 1} \quad \textbf{(50)}$$

$$f_{k_a} = T_k^u u_k \tag{51}$$

$$f_{k_b} = -T_k^u u_k \tag{52}$$

where f_{k_a} and f_{k_a} are the net actuation force in the global coordinate system, applied at nodes k_a and k_b respectively. Therefore the total resulting force on the system f_u due to the control forces u^P is:

$$T_a^u = \begin{bmatrix} T_1^u & & & & \\ & \ddots & & & \\ & & T_k^u & & \\ & & & \ddots & \\ & & & & T_{n_a}^u \end{bmatrix}_{3n_a \times n_a} \tag{53}$$

$$T_b^u = -T_a^u \tag{54}$$

$$T_u = \begin{bmatrix} T_a^u \\ T_b^u \end{bmatrix} \tag{55}$$

$$f_u = \begin{bmatrix} f_{1_a} \\ \vdots \\ f_{k_a} \\ \vdots \\ f_{(n_a)_a} \\ f_{1_b} \\ \vdots \\ f_{k_b} \\ \vdots \\ f_{(n_a)_b} \end{bmatrix}_{6n_a \times 1} = \begin{bmatrix} T_1^u & & & & & \\ & \ddots & & & & \\ & & T_k^u & & & \\ & & & \ddots & & \\ & & & & T_{n_a}^u & \\ -T_1^u & & & & & \\ & \ddots & & & & \\ & & -T_k^u & & & \\ & & & \ddots & & \\ & & & & & -T_{n_a}^u \end{bmatrix}_{6n_a \times n_a} \begin{bmatrix} u_1 \\ \vdots \\ u_k \\ \vdots \\ u_{n_a} \end{bmatrix}_{n_a \times 1} = T_u u^P \tag{56}$$

A $3n_p \times 6n_a$ permutation matrix P_u is then defined that maps total actuation force vector f_u in the global coordinate system to align with ordering of the nodal input disturbance forces d^P, such that:

$$d_u^P = P_u f_u = P_u T_u u^P \quad \textbf{(57)}$$

The resulting reordered force vector d_u^P can then simply be treated as a disturbance force to the system and can be applied directly to the input disturbance force matrix B_d^P. This gives us a mapping between the control forces u^P and change in $\dot{x}_t^P$ that defines B_u^P as:

$$B_u^P = B_d^P P_u T_u \quad \textbf{(58)}$$

The system model is then modified to include sensors signals y^P, as shown in (59). The matrix C_y^G maps the state vector x_t^P to the displacement of the actuator tip node k_a in the direction of v_k. Since the disturbance and actuator control force have no direct feed through to the sensor output, the matrices D_{yd}^P and D_{yu}^P are zero:

$$\begin{bmatrix} \dot{x}_t^P \\ z^P \\ y^P \end{bmatrix} = \begin{bmatrix} A^P & B_d^P & B_u^P \\ C_z^P & D_{zd}^P & D_{zu}^P \\ C_y^P & D_{yd}^P & D_{yu}^P \end{bmatrix} \begin{bmatrix} x_t^P \\ d^P \\ u^P \end{bmatrix} \quad \textbf{(59)}$$

The sensor signals y^P are the displacement of the actuator tip node k_a in the direction of the actuation motion v_k. The vector y^P is an approximation to the actual edge sensing error estimation and is only used for controller synthesis purposes. In the simulation of the system the node displacement vector z^P will be used to calculate the sensor reading

which will then be used to estimate the actuator tip displacement that y^P represents, as discussed in Section 4.6:

$$y^P = \begin{bmatrix} y_1 \\ \vdots \\ y_k \\ \vdots \\ y_{n_a} \end{bmatrix}_{n_a \times 1} \quad \textbf{(60)}$$

The displacement of the actuator tip nodes δ_a, in the global coordinate system, is shown in (61):

$$\delta_a = \begin{bmatrix} \delta_{1_a} \\ \vdots \\ \delta_{k_a} \\ \vdots \\ \delta_{(n_a)_a} \end{bmatrix}_{3n_a \times 1} \quad \textbf{(61)}$$

Where δ_{k_a} is the displacement of actuator tip node k_a. Since the vector δ_{k_a} is the 3 DOF displacement of the actuator tip nodes in the global coordinate system, the transformation matrix T_a^u defined in (53) can be used to transform the actuator tip node displacement into the displacements along line of action of the actuator, such that:

$$y^P = \left(T_a^u\right)^* \delta_{k_a} \quad \textbf{(62)}$$

A $3n_a \times 3n_p$ permutation matrix P_y is then defined that maps node displacement vector z^P to the actuator tip displacement vector δ_{k_a}, since z^P is a vector of displacements for all nodes, including the actuator tip nodes, such that:

$$\delta_{k_a} = P_y z^P = P_y C_z^P x_t^P \quad \textbf{(63)}$$

Therefore from (62) and (63) the sensor output matrix C_y^P is defined as in (65):

$$y^P = \left(T_a^u\right)^* P_y C_z^P x_t^P \quad \textbf{(64)}$$

$$C_y^P = \left(T_a^u\right)^* P_y C_z^P \quad \textbf{(65)}$$

3.3.1 State Space Model Reduction

The computational load of the synthesis process for a H_∞ controller is a function of the number of system inputs, outputs and states in the model. Model reduction techniques are used to reduce the number of states, which lightens the computational load, and make the synthesis problem tractable. Additionally by reducing the number of states in the controller synthesis problem, it allows for the controller to target only the most controllable and observable state in the system. This often leads to improved controller performance since control effort is not wasted on poorly controlled or observed modes, since the controller is designed to only control specific retained modes. The reduced model is only used for controller synthesis purposes, and is not used as the plant model for any simulations. In context of the control system the only relevant outputs of the system are the actuators displacements, represented by the signal y^P, therefore in the reduced model, the system output z^P become $\tilde{z}^P$, where $\tilde{z}^P = y^P$. Consequently the state reduction process is based on all disturbances and control force inputs but only the actuator displacement outputs.

Several different state reduction techniques were investigated, however the Balanced Model Truncation via Square Root Method [30][31][35] was selected for reasons that are discussed in Section 3.5.2. The Balanced Model Truncation via Square Root Method is realized by computing the controllability and observability grammians as shown in equations (66) and (67) respectively.

$$W_c = \int_0^\infty e^{\left(A^P\right)\tau} \begin{bmatrix} B_d^P & B_u^P \end{bmatrix} \begin{bmatrix} \left(B_d^P\right)^* \\ \left(B_u^P\right)^* \end{bmatrix} e^{\left(A^P\right)^*\tau} d\tau \qquad \textbf{(66)}$$

$$W_o = \int_0^\infty e^{\left(A^P\right)^*\tau} \begin{bmatrix} \left(C_z^P\right)^* & \left(C_y^P\right)^* \end{bmatrix} \begin{bmatrix} C_z^P \\ C_y^P \end{bmatrix} e^{\left(A^P\right)\tau} d\tau \qquad \textbf{(67)}$$

A singular value decomposition is then preformed on both the controllability and observability grammians as shown in equations (68) and (69) respectively:

$$W_c = U_{W_c} \Sigma_{W_c} \left(V_{W_c}\right)^* \qquad \textbf{(68)}$$

$$W_o = U_{W_o} \Sigma_{W_o} \left(V_{W_o}\right)^* \qquad \textbf{(69)}$$

The square root of the grammians computed by equations (70) and (71):

$$L_{W_c} = U_{W_c} \sqrt{\Sigma_{W_c}} \qquad \textbf{(70)}$$

$$L_{W_o} = U_{W_o} \sqrt{\Sigma_{W_o}} \qquad \textbf{(71)}$$

where $\sqrt{\Sigma_{W_c}}$ and $\sqrt{\Sigma_{W_o}}$ are the element-wise square root of the diagonal singular value matrices Σ_{W_c} and Σ_{W_o}. The Hankel singular values of the system are then computed from the singular value decomposition of $\left(L_{W_o}\right)^* L_{W_c}$, as shown in (72):

$$\left(L_{W_o}\right)^* L_{W_c} = U_W \Sigma_W \left(V_W\right)^* \qquad \textbf{(72)}$$

where Σ_W are the Hankel singular values, shown in (73), where $\Sigma_{W_k} \geq \Sigma_{W_{k+1}}$:

$$\Sigma_W = \begin{bmatrix} \Sigma_{W_1} & & & & \\ & \ddots & & & \\ & & \Sigma_{W_k} & & \\ & & & \ddots & \\ & & & & \Sigma_{W_{12n_p}} \end{bmatrix}_{12n_p \times 12n_p} \quad \textbf{(73)}$$

The number of states in the system is reduced by retaining only the largest Hankel singular values $\tilde{\Sigma}_W$ as shown in (74), where n_r is the number of retained states:

$$\tilde{\Sigma}_W = \begin{bmatrix} I_{n_r \times n_r} & 0 \end{bmatrix}_{n_r \times 12n_p} \Sigma_W \begin{bmatrix} I_{n_r \times n_r} \\ 0 \end{bmatrix}_{12n_p \times n_r} \quad \textbf{(74)}$$

The left and right reduction matrices, T_L and T_R, shown in (75) and (76) respectively, are used to reduce the number of states in the original system as shown in (77):

$$T_L = L_{W_o} \left(U_W \begin{bmatrix} I_{n_r \times n_r} \\ 0 \end{bmatrix}_{12n_p \times n_r} \right) \sqrt{\tilde{\Sigma}_W} \quad \textbf{(75)}$$

$$T_R = L_{W_c} \left(V_W \begin{bmatrix} I_{n_r \times n_r} \\ 0 \end{bmatrix}_{12n_p \times n_r} \right) \sqrt{\tilde{\Sigma}_W} \quad \textbf{(76)}$$

$$\begin{bmatrix} \tilde{A}^P & \begin{bmatrix} \tilde{B}_d^P & \tilde{B}_u^P \end{bmatrix} \\ \begin{bmatrix} \tilde{C}_z^P \\ \tilde{C}_y^P \end{bmatrix} & \begin{bmatrix} \tilde{D}_{zd}^P & \tilde{D}_{zu}^P \\ \tilde{D}_{yd}^P & \tilde{D}_{yu}^P \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \left(T_L\right)^* A^P T_R & \left(T_L\right)^* \begin{bmatrix} B_d^P & B_u^P \end{bmatrix} \\ \begin{bmatrix} C_z^P \\ C_y^P \end{bmatrix} T_R & \begin{bmatrix} D_{zd}^P & D_{zu}^P \\ D_{yd}^P & D_{yu}^P \end{bmatrix} \end{bmatrix} \quad \textbf{(77)}$$

3.4 Decentralized System Modeling

The decentralized system model is formulated very similarly to the plant state-space model, yet only represents a single, isolated segment. The decentralized system mass, stiffness and damping matrices, M_d, K_d, and D_d are generated in the same manner as in Section 3.2: however, only contain information on nodes and elements directly connected to the central segment of the SMCT model, as shown in Figure 16 to Figure 19. Though the decentralized model is generated from the central segments, it is used as a general model for all segments, where any variations between segments' dynamics are treated as unmodeled dynamics or uncertainty in the system:

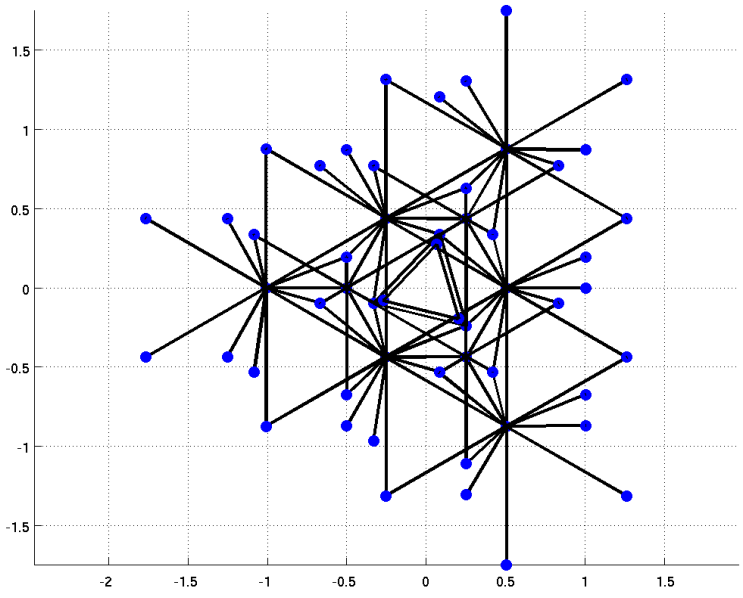

Figure 16. Decentralized Nodes and Elements – X-Y Plane View

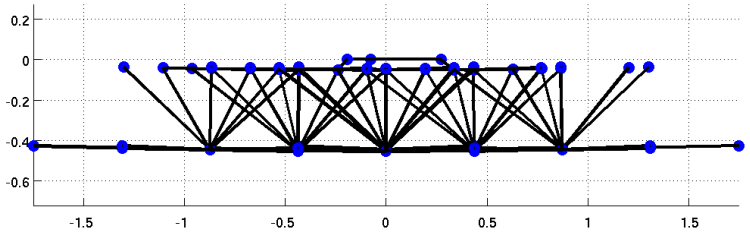

Figure 17. Decentralized Nodes and Elements – Y-Z Plane View

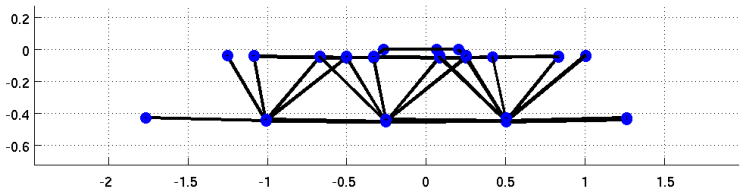

Figure 18. Decentralized Nodes and Elements – X-Z Plane View

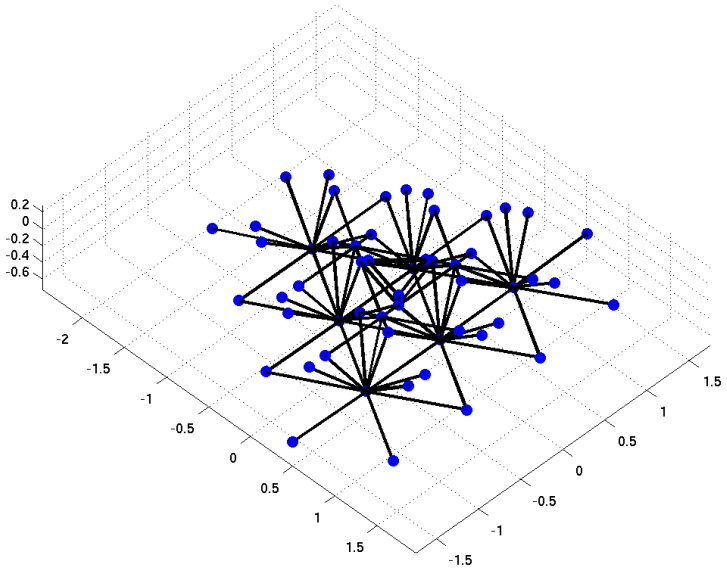

Figure 19. Decentralized Nodes and Elements – Perspective View

The decentralized system matrices are bound by the nodes shown in red in Figure 20, resulting in the bound decentralized system matrices M_c, K_c, and D_c, as shown in equations (79) to (81).

$$x_c = P_d x_d \quad \textbf{(78)}$$

$$M_c = P_d M_d \left(P_d\right)^* \quad \textbf{(79)}$$

$$D_c = P_d D_d \left(P_d\right)^* \quad \textbf{(80)}$$

$$K_c = P_d K_d \left(P_d\right)^* \quad \textbf{(81)}$$

The bound nodes are those nodes which lie on adjacent segments exclusively, such that a decentralized segment is modeled as if it is rigidly bound by all its neighboring segments.

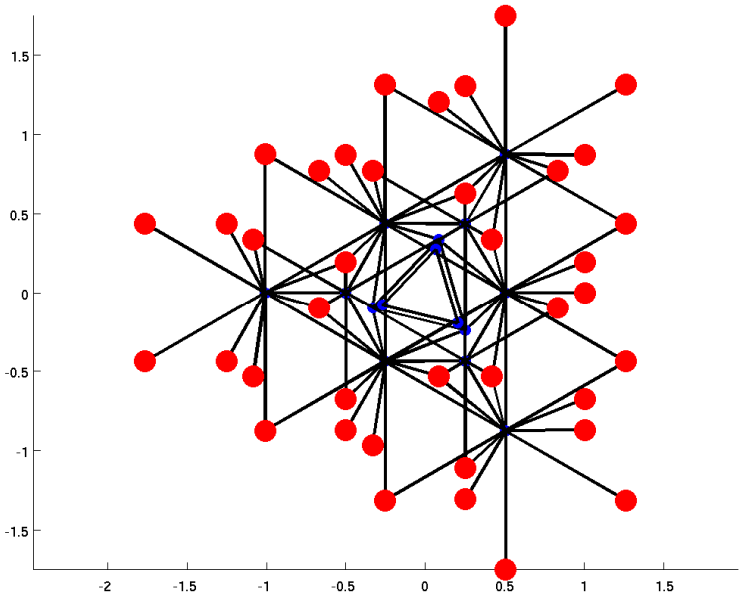

Figure 20. Decentralized Model Bounded Nodes

The bound system matrices are then converted to a state-space as shown in (82), similar to equation (42):

$$\begin{bmatrix} \begin{bmatrix} \dot{x}_c \\ \ddot{x}_c \end{bmatrix} \\ z^C \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & I \\ -M_c^{-1}K_c & -M_c^{-1}D_c \end{bmatrix} & \begin{bmatrix} 0 \\ M_c^{-1} \end{bmatrix} P_c \\ P_c^{-1}\begin{bmatrix} I & 0 \end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} x_c \\ \dot{x}_c \end{bmatrix} \\ d^C \end{bmatrix} \quad \textbf{(82)}$$

The matrix P_c is a permutation matrix, defined below, similar to permutation P_p in equation (83), so the system includes only the 3 DOF force inputs d^C and displacement outputs z^C of unbound nodes of the decentralized system:

$$P_c = \begin{bmatrix} \tilde{I} & & 0 \\ & \ddots & \\ 0 & & \tilde{I} \end{bmatrix}_{6n_c \times 3n_c} \quad \textbf{(83)}$$

The modeling process continues along equivalent to the plant modeling procedures in Section 3.3, including the model reduction step. The final decentralized model is a reduced state-space system as shown in (84):

$$\begin{bmatrix} \dot{\tilde{x}}_c \\ \begin{bmatrix} \tilde{z}^C \\ y^C \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \tilde{A}^C & \begin{bmatrix} \tilde{B}_d^C & \tilde{B}_u^C \end{bmatrix} \\ \begin{bmatrix} \tilde{C}_z^C \\ \tilde{C}_y^C \end{bmatrix} & \begin{bmatrix} \tilde{D}_{zd}^C & \tilde{D}_{zu}^C \\ \tilde{D}_{yd}^C & \tilde{D}_{yu}^C \end{bmatrix} \end{bmatrix} \begin{bmatrix} \tilde{x}_c \\ \begin{bmatrix} d^C \\ u^C \end{bmatrix} \end{bmatrix} \quad \textbf{(84)}$$

3.5 Distributed System Modeling

A distributed modeling technique was developed using the modified state-space framework for spatially interconnected systems, presented in [7]. This model technique is based on the concept of spatially discretizing the structure into an array of spatially invariant interconnected systems. Each spatially discrete system unit is directly coupled

to its neighboring units by interconnection signals. These interconnections act as spatial transfer functions; disturbances in one section of the interconnected system are transferred though the network of interconnection signals and affect all other parts of the system. However, the effects may attenuate rapidly after only a few transfer processes. This illustrates the global nature of the distributed system and is one of the key motivators for developing the distributed model.

Interconnection signals are specified along a finite number of fixed spatial dimensions. Each spatial dimension has an associated positive and negative direction. This allows for the construction on an infinite interconnected system along each direction of spatial dimensions. Show in Figure 21 is an example of the system interconnection with three spatial dimensions, know as triangular lattice [32].

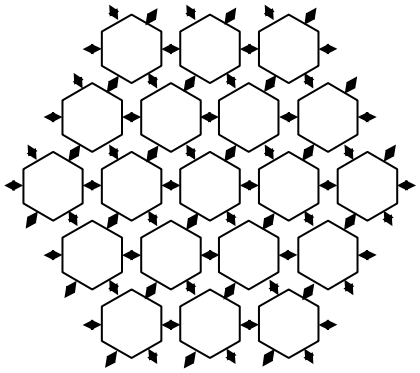

Figure 21. Interconnected System with three Spatial Directions

The system can be scaled by simply adding or removing distributed units from the interconnected system. This allows for the construction of a potentially very large system which is based only on a single simpler model. To create a finite interconnected system,

exterior signals, the interconnection signals that have no neighboring unit to connect to, can be treated as a special case and are discussed in more detail in [33]. However in the case of SMCT the exterior interconnection signals are simply zeroed, such that the distributed system network acts as if it was rings by a set of distributed units that are fixed in space. The three spatial dimensions along which the system is discretized are shown superimposed upon the plant node layout in Figure 22.

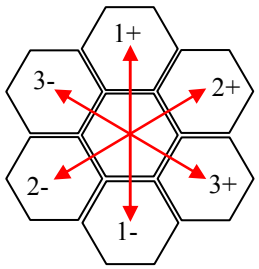

Figure 22. Three Spatial Dimensions for the Distributed Model

3.5.1 Distributed Unit Modeling of a Mirror Segment

In the case of a segmented mirror an obvious choice for the distributed system unit is a single segment and its associated support structure. This distributed system unit would encompass the same nodes and elements as in decentralized model, discussed in Section 3.4, but with the addition of the interconnection signals that couple the system unit to its neighboring system units. The spatially invariant condition is only approximately satisfied since the structural dynamics vary as a function of the radius from the optical axis. However in the case of the SMCT these variations are kept to a minimum, and merely add to the uncertainty of the distributed unit model.

Since each segment is surrounded by six other segments, excluding segments on the outer or inner edge of the segmented surface as they are handled as a special case, three spatial dimensions are defined, each with positive and negative direction, as shown in Figure 21 above. Figure 23 below is a representation of single system unit.

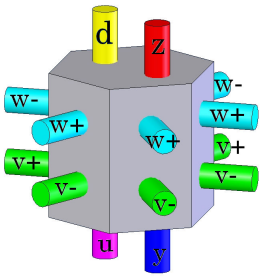

Figure 23. Distributed Modeling Unit with Three Spatial Dimensions

The signals z, y, d, and u represent the standard state-space I/O signals: system output, sensor output, disturbance input, and control input, respectively, as discussed in the above sections. Additionally w_+ are the interconnection output signals in the positive directions, while w_- are the interconnection output signals in the negative directions. Similarly v_+ and v_- are interconnection input signals in the positive and negative directions, respectively. This results in six interconnected inputs and six interconnected outputs, such that one input/output pair is coupled to each of the six surrounding segment models, as shown in Figure 24.

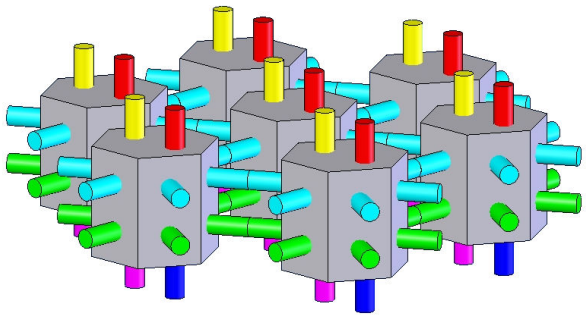

Figure 24. Interconnected System of Distributed Units

The system is spatially discretized such that each distributed unit consists of a segment triad that is supported by three triangular truss pyramids. However this discretization leads to the complication that the same nodes are being modeled multiple times in neighboring distributed units. Figure 25 shows an exploded view of four neighboring distributed unit models. When these distributed units are collapsed into a single interconnected system, nodes that share a common colour, excluding blue nodes, are coincident, and in fact represent the same structure node.

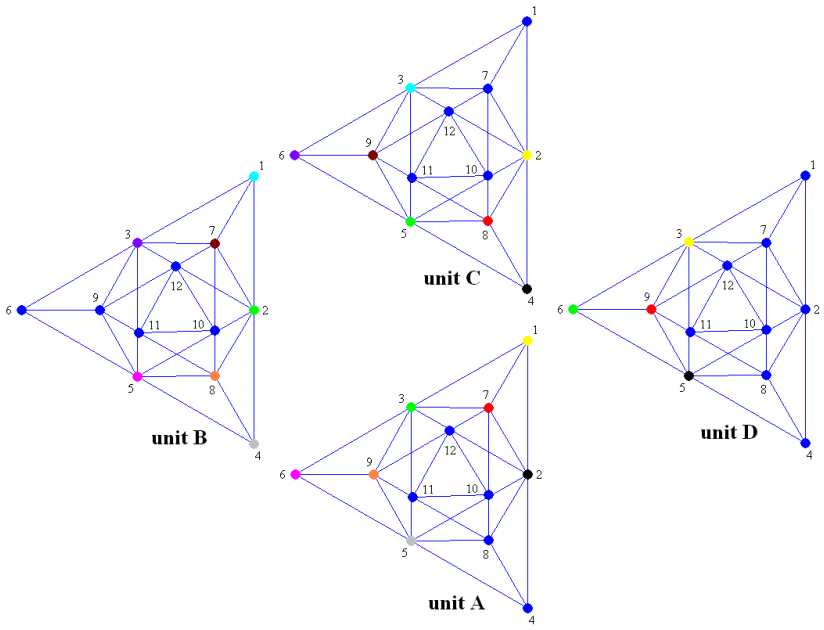

Figure 25. Exploded View of Distributed Unit Truss Models

The physical interpretation of interconnected signals in this case are displacements and velocities of neighboring nodes that are directly connected to the unit, yet not present in the unit itself. As an example node 1 of unit A is directly connected nodes 1, 2, 7, 11 and 12 of unit D and nodes 1, 3, 7 , 10, and 12 of unit C, yet none of these nodes are present in the model of unit A. As a result, the interconnected signals simply become a subset of the neighboring unit's state variables.

The formulaic representation of a distributed unit, as a modified state-space form [7], is shown below in (85):

$$\begin{bmatrix} \dot{x}_t^D \\ w^D \\ z^D \end{bmatrix} = \begin{bmatrix} A_{tt}^D & A_{ts}^D & B_{td}^D \\ A_{st}^D & A_{ss}^D & B_{sd}^D \\ C_{zt}^D & C_{zs}^D & D_{zd}^D \end{bmatrix} \begin{bmatrix} x_t^D \\ v^D \\ d^D \end{bmatrix} \qquad \textbf{(85)}$$

The matrices A_{tt}^{D}, $B_{t\bullet}^{D}$, $C_{\bullet t}^{D}$, and $D_{\bullet\bullet}^{D}$ are the standard temporal based state-space matrices, equivalent to the decentralized state-space matrices described in Section 3.4. While the remaining matrices, denoted with an s subscript, are associated with the newly introduced interconnected signals. The method developed for generating the distributed model matrices from the distributed FEM model is outlined below.

The distributed model's system mass, stiffness and damping matrices are the same as the unbounded system mass, stiffness and damping matrices, M_d, K_d and D_d respectively, produced for the decentralized model in Section 3.4. These matrices are used to form an unbound, and therefore unstable, state-space model as shown in (86):

$$\begin{bmatrix} \begin{bmatrix} \dot{x}_d \\ \ddot{x}_d \end{bmatrix} \\ z^d \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & I \\ -M_d^{-1}K_d & -M_d^{-1}D_d \end{bmatrix}_{12n_d \times 12n_d} & \begin{bmatrix} 0 \\ M_d^{-1} \end{bmatrix}_{12n_d \times 6n_d} \\ I_{12n_d \times 12n_d} & 0_{12n_d \times 6n_d} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} x_d \\ \dot{x}_d \end{bmatrix} \\ d^d \end{bmatrix} \tag{86}$$

$$x^d = \begin{bmatrix} x_d \\ \dot{x}_d \end{bmatrix}_{12n_d \times 1} \tag{87}$$

$$\begin{bmatrix} A^d & B_d^d \\ C_z^d & D_{zd}^d \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & I \\ -M_d^{-1}K_d & -M_d^{-1}D_d \end{bmatrix}_{12n_d \times 12n_d} & \begin{bmatrix} 0 \\ M_d^{-1} \end{bmatrix}_{12n_d \times 6n_d} \\ I_{12n_d \times 12n_d} & 0_{12n_d \times 6n_d} \end{bmatrix} \tag{88}$$

$$\therefore \begin{bmatrix} \dot{x}^d \\ z^d \end{bmatrix} = \begin{bmatrix} A^d & B_d^d \\ C_z^d & D_{zd}^d \end{bmatrix} \begin{bmatrix} x^d \\ d^d \end{bmatrix} \tag{89}$$

Where n_d is the number of nodes in distributed unit model, shown in Figure 19. The states that are associated with the unbound nodes of the decentralized model define in (78), are the same states that represent the temporal states x_t^D of the distributed unit, shown in (90). This is result of the observation that the decentralized model is merely a

distributed unit in which the interconnection nodes, nodes strictly associated with neighboring distributed units, are bounded:

$$x_t^D = x_t^C = \begin{bmatrix} P_d & P_d \end{bmatrix}_{12n_c \times 12n_d} \begin{bmatrix} x_d \\ \dot{x}_d \end{bmatrix}_{12n_d \times 1} = P_t x^d \qquad \textbf{(90)}$$

The outgoing interconnect signals w^D are a subset of the temporal state vector x_t^D as shown in (91), where n_s is the number of nodes associated with the interconnection signals in each spatial directions.

$$w^D = \begin{bmatrix} w_{1+} \\ w_{1-} \\ w_{2+} \\ w_{2-} \\ w_{3+} \\ w_{3-} \end{bmatrix} = \begin{bmatrix} P_{w_{1+}} & & & & & \\ & P_{w_{1-}} & & & & \\ & & P_{w_{2+}} & & & \\ & & & P_{w_{2-}} & & \\ & & & & P_{w_{3+}} & \\ & & & & & P_{w_{3-}} \end{bmatrix}_{6(12n_s) \times 12n_c} \qquad x_t^D = P_w x_t^D = P_w P_t x^d \qquad \textbf{(91)}$$

In other words the spatial states in a give direction are the nodal displacements and velocities of a subset of the unit's nodes. These interconnection nodes are identified as the nodes which are unbounded in the decentralized model and at the same time are the bounded nodes of a neighboring unit, connecting to that unit directly by a single element. The nodes corresponding to the outgoing interconnection signals in the six directions are shown in red in Figure 26 to Figure 31. Note that the nodes in $1+$ and $2+$ direction are the same. Despite this, the interconnection effects along the $1+$ and $2+$ directions are different since these nodes connect to different units by a different truss element configuration as shown in the definition of the incoming interconnection nodes. The same is true for the $1-$ and $3+$ directions, as well as the $2-$ and $3-$ directions.

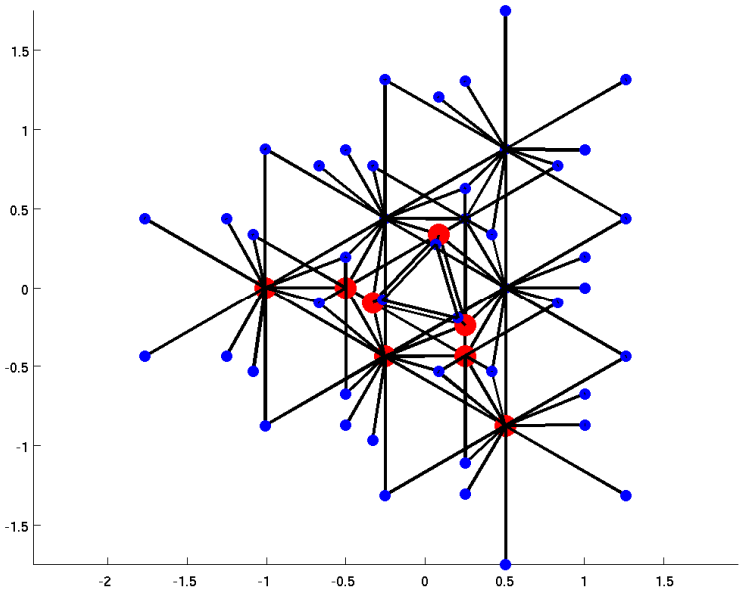

Figure 26. Nodes corresponding to w_{1+}

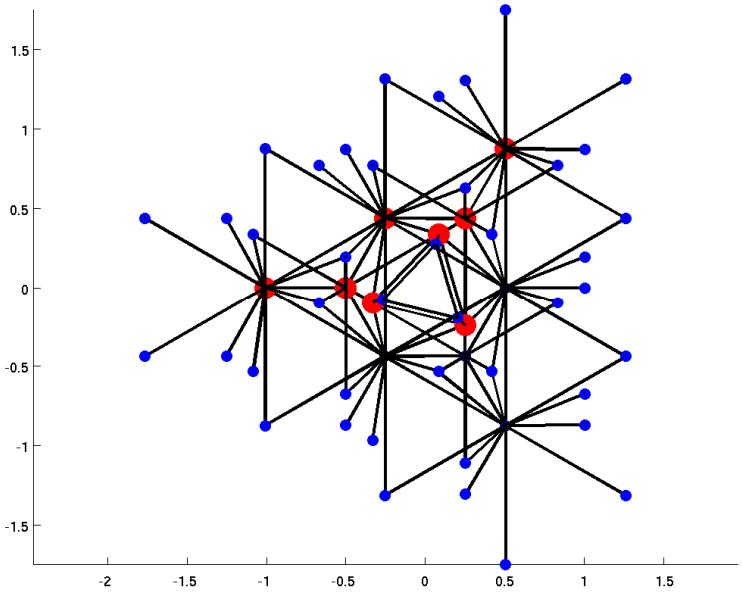

Figure 27. Nodes corresponding to w_{1-}

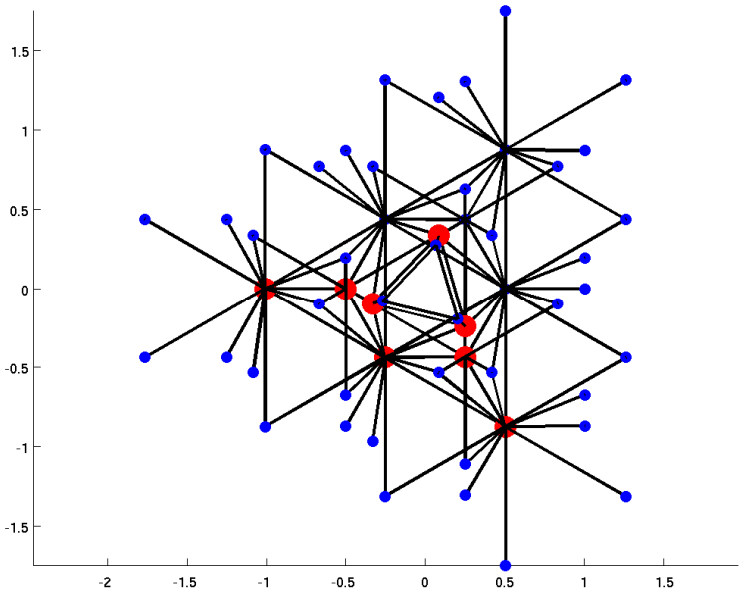

Figure 28. Nodes corresponding to w_{2+}

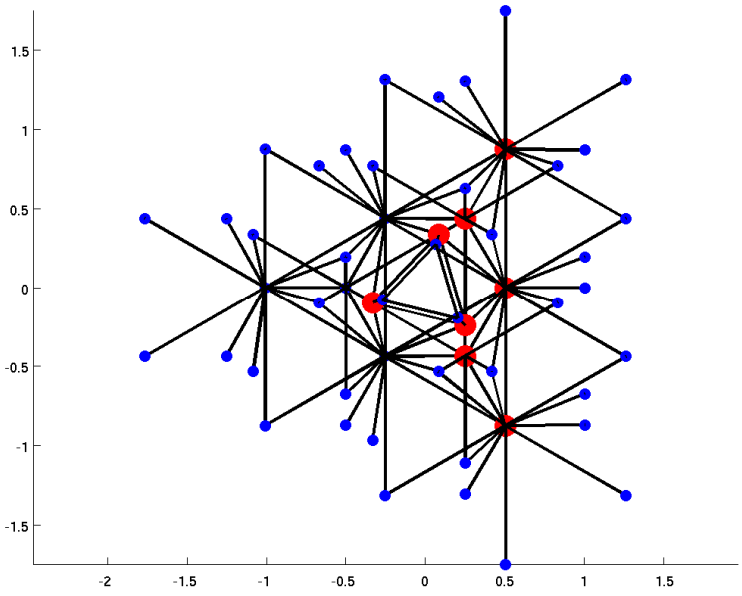

Figure 29. Nodes corresponding to w_{2-}

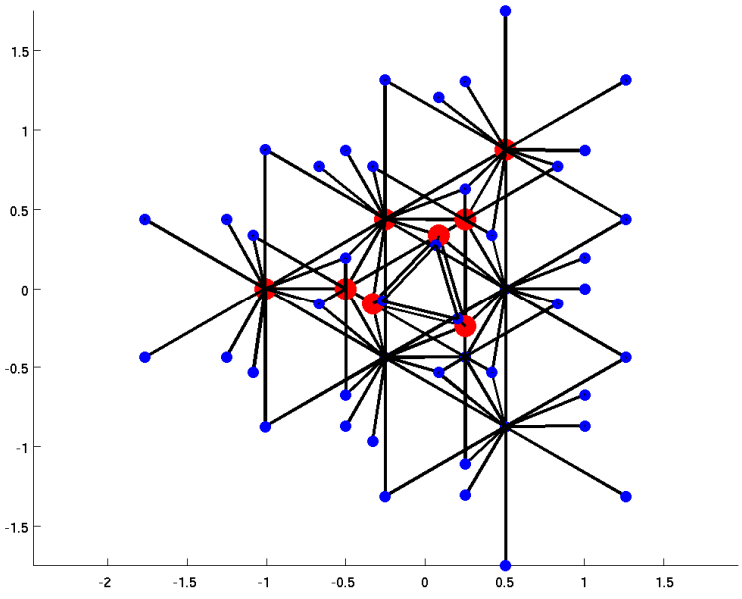

Figure 30. Nodes corresponding to w_{3+}

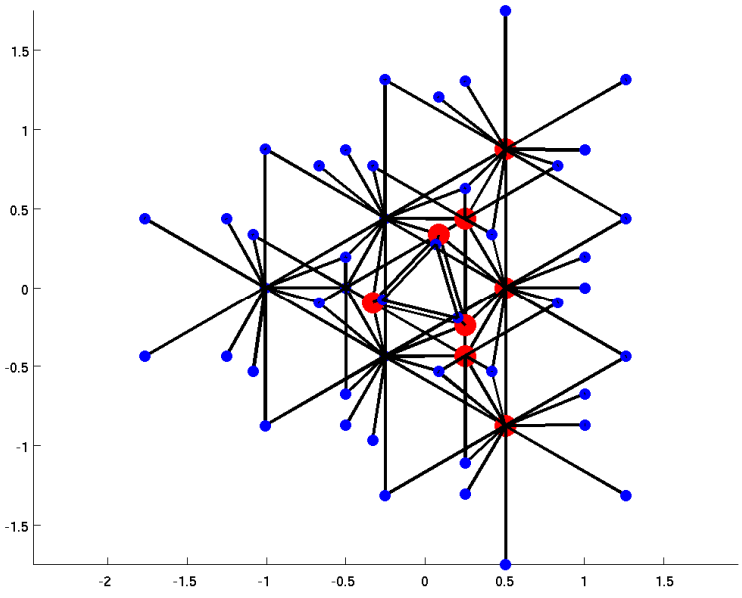

Figure 31. Nodes corresponding to w_{3-}

As stated above, the interconnect signals are a subset of the neighboring unit's temporal state variables, the vector of neighboring unit's states x_s^D are referred to as the spatial states and are defined in (92). The incoming interconnection signals v^D are therefore depended only on the spatial states, as shown in (94), which are equivalent to the states of the bounded nodes presented in the decentralized model:

$$x_s^D = P_s x^d \tag{92}$$

$$\left\{ x_s^D \;:\; x_s^D \subset x^d \;\wedge\; x_s^D \cap x_t^D = \varnothing \;\wedge\; x_s^D \cup x_t^D = x^d \right\} \tag{93}$$

$$v^D = \begin{bmatrix} v_{1+} \\ v_{1-} \\ v_{2+} \\ v_{2-} \\ v_{3+} \\ v_{3-} \end{bmatrix} = \begin{bmatrix} P_{v_{1+}} & & & & & \\ & P_{v_{1-}} & & & & \\ & & P_{v_{2+}} & & & \\ & & & P_{v_{2-}} & & \\ & & & & P_{v_{3+}} & \\ & & & & & P_{v_{3-}} \end{bmatrix}_{6(12n_s)\times 12(n_d - n_c)} \qquad x_s^D = P_v x_s^D = P_v P_s x^d \tag{94}$$

The incoming interconnection nodes that represent the corresponding outgoing interconnection nodes are shown in red in Figure 32 to Figure 37, such that w_{1+} corresponds to v_{1+}, and that w_{1-} corresponds to v_{1-}, and so on.

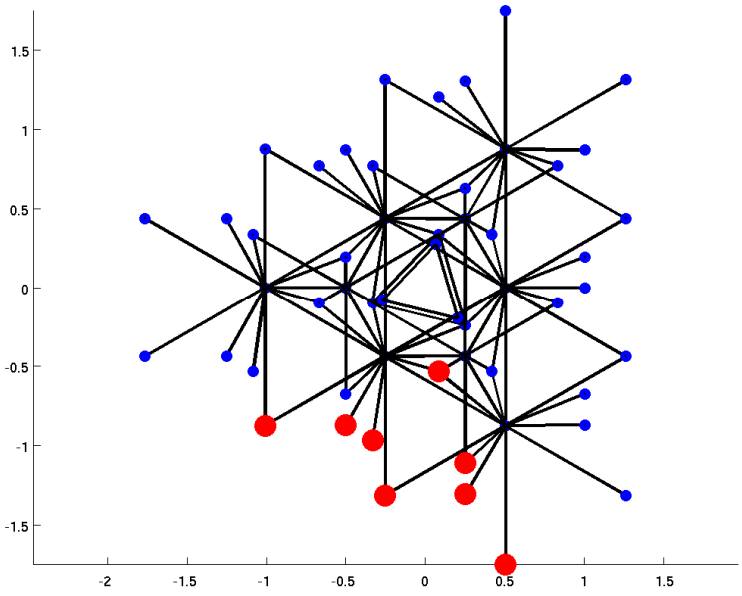

Figure 32. Nodes corresponding to v_{1+}

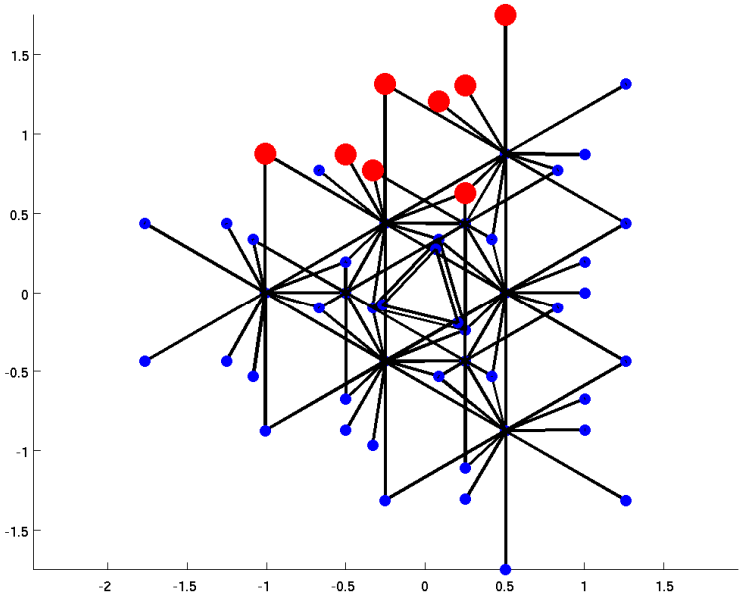

Figure 33. Nodes corresponding to v_{1-}

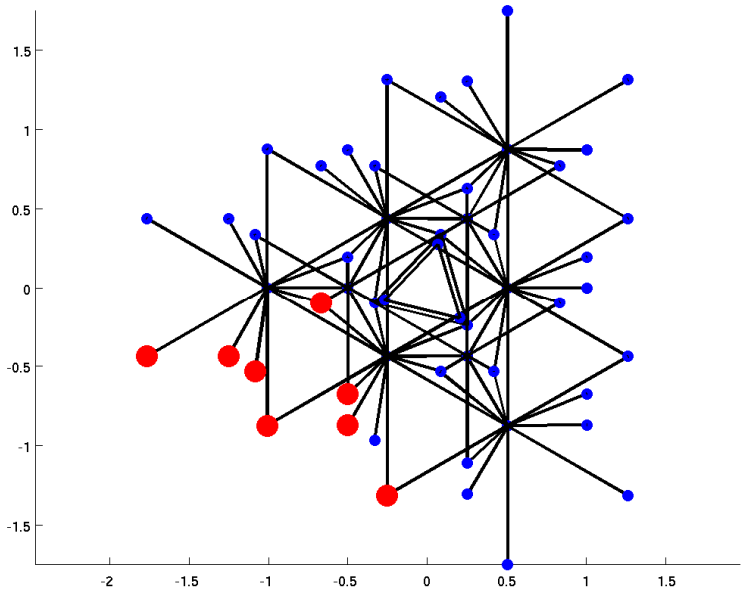

Figure 34. Nodes corresponding to v_{2+}

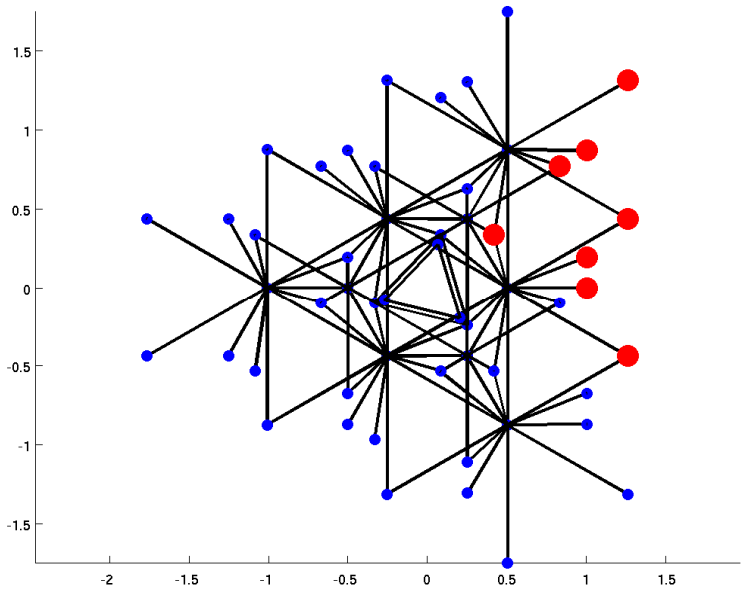

Figure 35. Nodes corresponding to v_{2-}

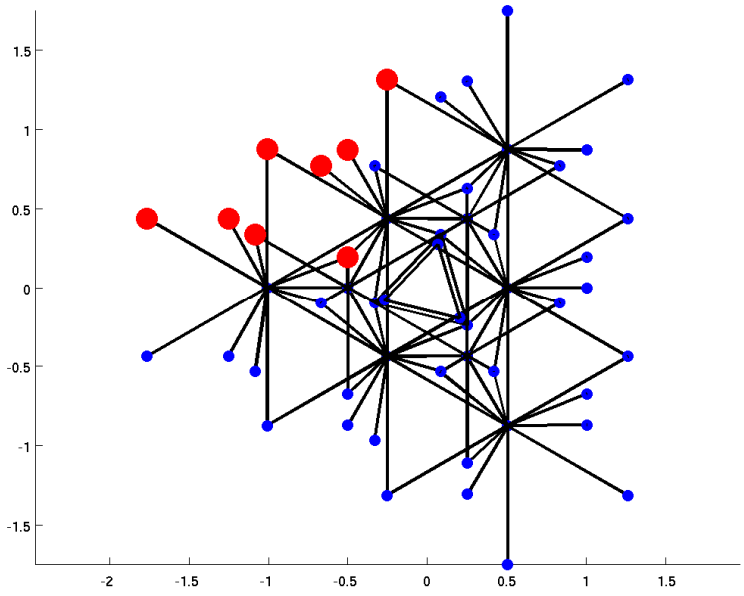

Figure 36. Nodes corresponding to v_{3+}

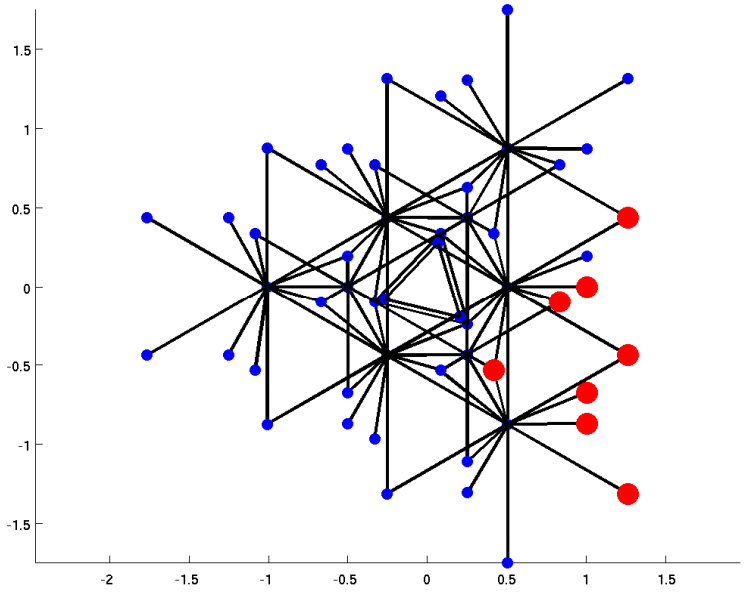

Figure 37. Nodes corresponding to v_{3-}

Due to this configuration some interconnection nodes are present on two different distributed units. Despite the fact that they represent the same physical node, there may be slight discrepancies in the state of the node between the two units. Additionally the effect of these nodes' state is applied twice since it is an input to the distributed unit from two different neighboring units. Therefore the inputs that corresponded to the interconnection nodes that are present on two neighboring units are divided by two. In other words the input state of these nodes is the average between to the two states of the neighboring units.

Base on equations (90) and (92) the state-space model in (89) can be rewritten in the form of (95):

$$\begin{bmatrix}\begin{bmatrix}\dot{x}_t^D\\ \dot{x}_s^D\end{bmatrix}\\ \begin{bmatrix}z_t^D\\ z_s^D\end{bmatrix}\end{bmatrix}=\begin{bmatrix}\begin{bmatrix}P_tA^d\left(P_t\right)^* & P_tA^d\left(P_s\right)^*\\ P_sA^d\left(P_t\right)^* & P_sA^d\left(P_s\right)^*\end{bmatrix} & \begin{bmatrix}P_tB_d^d\left(P_t\right)^* & P_tB_d^d\left(P_s\right)^*\\ P_sB_d^d\left(P_t\right)^* & P_sB_d^d\left(P_s\right)^*\end{bmatrix}\\ \begin{bmatrix}P_tC_z^d\left(P_t\right)^* & P_tC_z^d\left(P_s\right)^*\\ P_sC_z^d\left(P_t\right)^* & P_sC_z^d\left(P_s\right)^*\end{bmatrix} & \begin{bmatrix}P_tD_{zd}^d\left(P_t\right)^* & P_tD_{zd}^d\left(P_s\right)^*\\ P_sD_{zd}^d\left(P_t\right)^* & P_sD_{zd}^d\left(P_s\right)^*\end{bmatrix}\end{bmatrix}\begin{bmatrix}\begin{bmatrix}x_t^D\\ x_s^D\end{bmatrix}\\ \begin{bmatrix}d_t^D\\ d_s^D\end{bmatrix}\end{bmatrix} \quad \textbf{(95)}$$

Where d_t^D and z_t^D correspond to the 6 DOF force inputs and 6 DOF displacement and velocity outputs, respectively, associated with the temporal states, and d_s^D and z_s^D correspond to the 6 DOF force inputs and 6 DOF displacement and velocity outputs, respectively, associated with the spatial states. Since the spatial states are simply temporal states of the neighboring distributed units, the force inputs, displacement and velocity outputs and derivatives of the spatial states, d_s^D, z_s^D and $\dot{x}_s^D$, respectively, can be ignored when considering a single disturbed unit, as they are handled by the neighboring

units' model and only the interconnections are important in this case. Therefore equation (95) simplifies to (96):

$$\begin{bmatrix} \dot{x}_t^D \\ z_t^D \end{bmatrix} = \begin{bmatrix} P_t A^d \left(P_t\right)^* & P_t A^d \left(P_s\right)^* & P_t B_d^d \left(P_t\right)^* \\ P_t C_z^d \left(P_t\right)^* & P_t C_z^d \left(P_s\right)^* & P_t D_{zd}^d \left(P_t\right)^* \end{bmatrix} \begin{bmatrix} x_t^D \\ x_s^D \\ d_t^D \end{bmatrix} \qquad \textbf{(96)}$$

Then by introducing the interconnection signals w^D and v^D from equations (91) and (94), respectively, into (96) one can form (97):

$$\begin{bmatrix} \dot{x}_t^D \\ w^D \\ z_t^D \end{bmatrix} = \begin{bmatrix} P_t A^d \left(P_t\right)^* & P_t A^d \left(P_s\right)^* \left(P_v\right)^* & P_t B_d^d \left(P_t\right)^* \\ P_w & 0 & 0 \\ P_t C_z^d \left(P_t\right)^* & P_t C_z^d \left(P_s\right)^* \left(P_v\right)^* & P_t D_{zd}^d \left(P_t\right)^* \end{bmatrix} \begin{bmatrix} x_t^D \\ v^D \\ d_t^D \end{bmatrix} \qquad \textbf{(97)}$$

The system input and output signals, d_t^D and z_t^D respectively, in (97) are reduced to the 3 DOF force inputs and displacement outputs excluding the moment inputs and rotation velocity outputs, d^D and z^D respectively, by application of the permutation matrix P_c defined in (83). The resulting system, shown in (98) when compared with (85) defines the distributed unit matrices as shown in (99) to (102):

$$\begin{bmatrix} \dot{x}_t^D \\ w^D \\ z^D \end{bmatrix} = \begin{bmatrix} P_t A^d \left(P_t\right)^* & P_t A^d \left(P_s\right)^* \left(P_v\right)^* & P_t B_d^d \left(P_t\right)^* P_c \\ P_w & 0 & 0 \\ \left(P_c\right)^* \begin{bmatrix} I & 0 \end{bmatrix} P_t C_z^d \left(P_t\right)^* & \left(P_c\right)^* \begin{bmatrix} I & 0 \end{bmatrix} P_t C_z^d \left(P_s\right)^* \left(P_v\right)^* & \left(P_c\right)^* \begin{bmatrix} I & 0 \end{bmatrix} P_t D_{zd}^d \left(P_t\right)^* P_c \end{bmatrix} \begin{bmatrix} x_t^D \\ v^D \\ d^D \end{bmatrix} \qquad \textbf{(98)}$$

$$\begin{bmatrix} A_{tt}^D & A_{ts}^D \\ A_{st}^D & A_{ss}^D \end{bmatrix} = \begin{bmatrix} P_t A^d \left(P_t\right)^* & P_t A^d \left(P_s\right)^* \left(P_v\right)^* \\ P_w & 0 \end{bmatrix} \qquad \textbf{(99)}$$

$$\begin{bmatrix} B_{td}^D \\ B_{sd}^D \end{bmatrix} = \begin{bmatrix} P_t B_d^d \left(P_t\right)^* P_c \\ 0 \end{bmatrix} \qquad \textbf{(100)}$$

$$\begin{bmatrix} C_{zt}^D & C_{zs}^D \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \left(P_c\right)^* & 0 \end{bmatrix} P_t C_z^d \left(P_t\right)^* & \begin{bmatrix} \left(P_c\right)^* & 0 \end{bmatrix} P_t C_z^d \left(P_s\right)^* \left(P_v\right)^* \end{bmatrix} \qquad \textbf{(101)}$$

$$\begin{bmatrix} D_{zd}^D \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \left(P_c\right)^* & 0 \end{bmatrix} P_t D_{zd}^d \left(P_t\right)^* P_c \end{bmatrix} \qquad \textbf{(102)}$$

Note that by construction in (86), D_{zd}^{d} is a zero matrix and therefore so is D_{zd}^{D}. Additionally the permutation matrices P_t and P_s select an exclusive subset of the x^d state vector such that:

$$\begin{aligned} P_t\left(P_s\right)^* &= 0 \\ P_s\left(P_t\right)^* &= 0 \end{aligned} \qquad \textbf{(103)}$$

Since C_z^d is an identity matrix, therefore the matrix the C_{zS}^{D} reduces to a zero matrix as shown in (104):

$$C_{zs}^{D} = \begin{bmatrix} \left(P_c\right)^* & 0 \end{bmatrix} P_t C_z^d \left(P_s\right)^* \left(P_v\right)^* = \begin{bmatrix} \left(P_c\right)^* & 0 \end{bmatrix} P_t \left(P_s\right)^* \left(P_v\right)^* = 0 \qquad \textbf{(104)}$$

This results in the distributed unit state-space model as shown in (105):

$$\begin{bmatrix} A_{tt}^{D} & A_{ts}^{D} & B_{td}^{D} \\ A_{st}^{D} & A_{ss}^{D} & B_{sd}^{D} \\ C_{zt}^{D} & C_{zs}^{D} & D_{zd}^{D} \end{bmatrix} = \begin{bmatrix} P_t A^d \left(P_t\right)^* & P_t A^d \left(P_s\right)^* \left(P_v\right)^* & P_t B_d^d \left(P_t\right)^* P_c \\ P_w & 0 & 0 \\ \begin{bmatrix} \left(P_c\right)^* & 0 \end{bmatrix} P_t C_z^d \left(P_t\right)^* & 0 & 0 \end{bmatrix} \qquad \textbf{(105)}$$

The construction of the distributed unit model parallels that of the decentralized model, with the important distinction that the so-called bounded nodes actually represent nodes on neighboring distributed units. Therefore instead of discarding dynamic information to bound the system, this dynamic information is used to construct the interconnection dynamics of the system. A single unit in the overall system is therefore bound by its neighbors who are in turn bound by their neighbors and so on, which leads directly into the global transfer of information in the overall system.

The distributed unit is then modified, as done in Section 3.3, to include sensor and actuator control signals, where the interconnection signals are temporarily treats as

additional inputs and outputs of the system, therefore temporarily reforming the distributed model as a standard state-space model. The resulting distributed unit model is shown in (106):

$$\begin{bmatrix} \dot{x}_t^D \\ w^D \\ z^D \\ y^D \end{bmatrix} = \begin{bmatrix} A_{tt}^D & A_{ts}^D & B_{td}^D & B_{tu}^D \\ A_{st}^D & A_{ss}^D & B_{sd}^D & B_{su}^D \\ C_{zt}^D & C_{zs}^D & D_{zd}^D & D_{zu}^D \\ C_{yt}^D & C_{ys}^D & D_{yd}^D & D_{yu}^D \end{bmatrix} \begin{bmatrix} x_t^D \\ v^D \\ d^D \\ u^D \end{bmatrix} \quad \textbf{(106)}$$

3.5.2 Distributed Model Reduction

For the same reasons as discussed in Section 3.3.1, the number of temporal states and system outputs in the distributed model is reduced. Again the reduced model's system output z^D become $\tilde{z}^D$, where $\tilde{z}^D = y^D$. However an added complication is that not only are there temporal states, as present in plant and decentralized models, but there are also spatial states in the form of the interconnected signals w^D and v^D. For simplicity the reduction of these two types of states are handled separately by first reducing the temporal states, as done with the two other models, and then reducing the spatial states based on the remaining temporal states.

The Balanced Model Truncation via Square Root Method as applied above can be applied to the distributed model with only some slight alterations required, as a result of the interconnection signals. To reiterate, the interconnected signals w^D and v^D are 6 DOF displacements and velocities of nodes on the distributed unit and its neighboring units. Therefore the units of measurement of the signals w^D and v^D are a combination of

meters, meters per second, radians and radians per second. The input disturbance forces d^D and control forces u^D are both in newtons. The output displacements $\tilde{z}^D$ and sensor estimated displacements y^D are in meters. This means that the inputs to the distributed unit, v^D, d^D, and u^D are comprised of different units of measurement. Similarly the outputs from the distributed unit, w^D, $\tilde{z}^D$, and y^D also comprise different units of measurement. As a result the formation of the controllability and observability grammians will be skewed if the interconnection signals are simply treated as additional inputs and outputs of the system. As an example, a unit step signal in one of the displacement entries of the v^D vector would result in very large displacements at the output of the system, since it would imply that a node on a neighboring unit has been displaced by one meter. However a unit step signal in one of the entries of the d^D vector would result in relatively small displacements at the output of the system, since the structure is stiff and one newton of force will not have a large effect on the system. Therefore the magnitude of transfer functions from the v^D inputs to the system outputs overwhelm the transfer function relating to the disturbance and control forces. For this reason the interconnection signals are temporarily ignored in order to perform the temporal state reduction. Therefore the matrices associated with the interconnection signals, those detonated with s, are temporarily removed from the unit model.

Accurately modeling the unit system response, baring the interconnection signals, is of primary importance for the controller synthesis, for the unit is what the controller will have direct influence over. The interconnection signals act as additional information used

to supplement the directly observed information. By not including the interconnection signals in the temporal reduction process, we are sacrificing the fidelity of the interconnection signals to preserve the modeling of the distributed unit as a single element. In other words we are sacrificing global performance for local performance. There may be schemes to weight the interconnection signals in conjunction with the unit's standard inputs and outputs, to balance the global and local performance; however such a scheme is left as a future detail to be resolved.

The Balanced Model Truncation via Square Root Method is carried out in the same manner as presented in Section 3.3.1, with the alteration that equation (77) becomes (107) which reintroduces the interconnection matrices:

$$\begin{bmatrix} \tilde{A}^D_{tt} & \tilde{A}^D_{ts} & \begin{bmatrix} \tilde{B}^D_{td} & \tilde{B}^D_{tu} \end{bmatrix} \\ \tilde{A}^D_{st} & \tilde{A}^D_{ss} & \begin{bmatrix} \tilde{B}^D_{sd} & \tilde{B}^D_{su} \end{bmatrix} \\ \begin{bmatrix} \tilde{C}^D_{zt} \\ \tilde{C}^D_{yt} \end{bmatrix} & \begin{bmatrix} \tilde{C}^D_{zs} \\ \tilde{C}^D_{ys} \end{bmatrix} & \begin{bmatrix} \tilde{D}^D_{zd} & \tilde{D}^D_{zu} \\ \tilde{D}^D_{yd} & \tilde{D}^D_{yu} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \left(T_L\right)^* A^D_{tt} T_R & \left(T_L\right)^* A^D_{ts} & \left(T_L\right)^* \begin{bmatrix} B^D_{td} & B^D_{tu} \end{bmatrix} \\ A^D_{st} T_R & A^D_{ss} & \begin{bmatrix} B^D_{sd} & B^D_{su} \end{bmatrix} \\ \begin{bmatrix} C^D_{zt} \\ C^D_{yt} \end{bmatrix} T_R & \begin{bmatrix} C^D_{zs} \\ C^D_{ys} \end{bmatrix} & \begin{bmatrix} D^D_{zd} & D^D_{zu} \\ D^D_{yd} & D^D_{yu} \end{bmatrix} \end{bmatrix} \quad \textbf{(107)}$$

The subsequent spatial state reduction process developed for this distributed model is based on the observation that the matrices $\tilde{A}^D_{ss}$, $\tilde{B}^D_{s\bullet}$, and $\tilde{C}^D_{\bullet s}$ are zero, and therefore neighboring units are only coupled through their $\tilde{A}^D_{ts}$ and $\tilde{A}^D_{st}$ matrices. This observation is key to spatial state reduction method introduced in this section, and was a motivator for selecting the Balanced Model Truncation via Square Root Method as the temporal state reduction method, as it does not populate the $\tilde{A}^D_{ss}$, $\tilde{B}^D_{s\bullet}$, and $\tilde{C}^D_{\bullet s}$ matrices. By construction the interconnection output signal w_{1+} on one unit is equal to the interconnection input

signal v_{1+} of the neighboring unit in the $1+$ direction. Similar is true for all other spatial interconnection directions. Assuming only one spatial interconnection direction at the moment, the corresponding sub-matrices of $\tilde{A}_{ts}^{D}$ and $\tilde{A}_{st}^{D}$ are $\tilde{A}_{ts}^{\circ}$ and $\tilde{A}_{st}^{\circ}$ respectively, where $\circ$ designates the spatial interconnection direction $1\pm$, $2\pm$, or $3\pm$. Two neighboring units, unit A and unit B, are coupled by the following equations:

$$\begin{aligned}\dot{x}_t^{\mathrm{A}} &= \tilde{A}_{tt}^{D} x_t^{\mathrm{A}} + \tilde{A}_{ts}^{\circ} v_{\circ}^{\mathrm{A}} + \tilde{B}_{td}^{D} d^{\mathrm{A}} + \tilde{B}_{tu}^{D} u^{\mathrm{A}} \\ &= \left(\tilde{A}_{tt}^{D} x_t^{\mathrm{A}} + \tilde{B}_{td}^{D} d^{\mathrm{A}} + \tilde{B}_{tu}^{D} u^{\mathrm{A}} \right) + \tilde{A}_{ts}^{\circ} w_{\circ}^{\mathrm{B}}\end{aligned} \qquad \textbf{(108)}$$

$$w_{\circ}^{\mathrm{B}} = \tilde{A}_{st}^{\circ} x_t^{\mathrm{B}} \qquad \textbf{(109)}$$

$$\therefore \dot{x}_t^{\mathrm{A}} = \left(\tilde{A}_{tt}^{D} x_t^{\mathrm{A}} + \tilde{B}_{td}^{D} d^{\mathrm{A}} + \tilde{B}_{tu}^{D} u^{\mathrm{A}} \right) + \tilde{A}_{ts}^{\circ} \tilde{A}_{st}^{\circ} x_t^{\mathrm{B}} \qquad \textbf{(110)}$$

From equation (110) we can see that the change in temporal states of unit A are directly coupled to the temporal states of unit B by the matrix $\tilde{A}_{ts}^{\circ} \tilde{A}_{st}^{\circ}$. Therefore the minimum required dimension of the interconnection signal vector in the single spatial interconnection direction is equal to the rank of $\tilde{A}_{ts}^{\circ} \tilde{A}_{st}^{\circ}$. Performing a singular value decomposition on $\tilde{A}_{ts}^{\circ} \tilde{A}_{st}^{\circ}$, as shown in (111), leads to a convenient method to reduce the interconnection vector lengths.

$$\tilde{A}_{ts}^{\circ} \tilde{A}_{st}^{\circ} = U_s \Sigma_s \left(V_s \right)^{*} \qquad \textbf{(111)}$$

$$\Sigma_s = \begin{bmatrix} \Sigma_1 & & \\ & \ddots & \\ & & \Sigma_{n_r} \end{bmatrix}_{n_r \times n_r} \qquad \textbf{(112)}$$

Where n_r is the number of temporal states retained after the temporal state reduction. The number of spatial states can then be reduce to n_i states, where $n_i \le n_r$, by the permutation matrix P_r, defined in (113):

$$P_r = \begin{bmatrix} I_{n_i \times n_i} \\ 0 \end{bmatrix}_{n_r \times n_i} \quad \textbf{(113)}$$

Finally the matrices $\tilde{A}_{ts}^{\circ}$ and $\tilde{A}_{st}^{\circ}$ are reduced and become (114) and (115) respectively:

$$\tilde{A}_{ts}^{\circ} \Rightarrow U_s \Sigma_s P_r \quad \textbf{(114)}$$

$$\tilde{A}_{st}^{\circ} \Rightarrow \left(V_s P_r\right)^* \quad \textbf{(115)}$$

This process can be applied to all the spatial interconnection direction, resulting in the total length of the interconnection vectors w^D and v^D to equal $6n_i$. The temporally and spatially reduced system, as shown in (116), is then used as the distributed system unit in the distributed controller synthesis procedures discussed in Section 4.5:

$$\begin{bmatrix} \dot{\tilde{x}}_t^D \\ \tilde{w}^D \\ \tilde{z}^D \\ y^D \end{bmatrix} = \begin{bmatrix} \tilde{A}_{tt}^D & \tilde{A}_{ts}^D & \tilde{B}_{td}^D & \tilde{B}_{tu}^D \\ \tilde{A}_{st}^D & \tilde{A}_{ss}^D & \tilde{B}_{sd}^D & \tilde{B}_{su}^D \\ \tilde{C}_{zt}^D & \tilde{C}_{zs}^D & \tilde{D}_{zd}^D & \tilde{D}_{zu}^D \\ \tilde{C}_{yt}^D & \tilde{C}_{ys}^D & \tilde{D}_{yd}^D & \tilde{D}_{yu}^D \end{bmatrix} \begin{bmatrix} \tilde{x}_t^D \\ \tilde{v}^D \\ d^D \\ u^D \end{bmatrix} \quad \textbf{(116)}$$

3.6 Edge Sensor System Modeling

The baseline design of the edge sensors for the SMCT is based on the envisioned design for the TMT project. The sensor is a parallel plate capacitive sensor shown in Figure 38 and described in detail in [27].

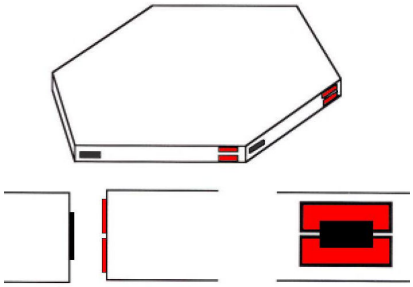

Figure 38. Capacitive Parallel Plate Edge Sensor

The modeling of the parallel plate capacitive sensor is simplified such that the sensor model output is equal to the relative vertical displacement in local a coordinate system between the two halves of the sensor. This assumes that the change in capacitance and electronics required to convert this change into a usable signal is directly proportional to the relative displacement. As well the sensor model is blind to changes in the dihedral angle between the two edges of the segments. This is a desirable effect as sensitivity to the dihedral angle would translate to a sensing error when estimated the actuator displacement. Insensitivity to the dihedral angle can be achieved by augmenting the capacitive sensor shown in Figure 38, with an additional capacitive sensor used to measure the dihedral angle, who's output can then be subtracted from the primary sensor reading.

One half of an edge sensor is assumed to be the active sensor node which defines the reference coordinate system for the relative displacement. The other half of the sensor is known as the passive sensor node. Since there are twelve sensor connection points per

segment, each connection point represents one half of a relative displacement edge, there are therefore six active sensor nodes and six passive sensor nodes per segment.

The vertical displacement of the passive sensor node in the active sensor node's coordinate system is the sensor model's output. Since sensors are shared between adjacent segments, sensor output from the same sensor but from the perspective of either conjoined segment has an equal magnitude, but an opposite sign, as a result of the relative nature of sensing scheme.

The active sensor node coordinate system is defined based on the position and orientation of the segment in the global coordinate system. This assumes that the segment is a rigid body, as discussed in Section 3.2.2.

As stated above a segment is modeled as a rigid body defined by three points, known as the segment triad, that form an equilateral triangle. The nominal location of these three points t_1, t_2, and t_3 in the global coordinated system are used to describe a rotation matrix R_T^G and offset c_t from the nominal local segment coordinate system to global coordinate system, such that:

$$R_T^G = \left[\hat{i}_t \quad \hat{j}_t \quad \hat{k}_t \right]_{3\times 3} \tag{117}$$

$$c_t = \frac{t_1 + t_2 + t_3}{3} \tag{118}$$

$$\hat{i}_t = \frac{t_1 - t_2}{\left| t_1 - t_2 \right|} \tag{119}$$

$$\hat{j}_t = \frac{t_3 - \left(t_1 + t_2\right)/2}{\left|t_3 - \left(t_1 + t_2\right)/2\right|} \quad \textbf{(120)}$$

$$\hat{k}_t = \hat{i}_t \times \hat{j}_t \quad \textbf{(121)}$$

As the segment moves in the global coordinate system, so do the segment triad points, such that that displaced segment triad node positions $\overline{t_1}$, $\overline{t_2}$ and $\overline{t_3}$ are:

$$\overline{t_1} = t_1 + \delta_1 \quad \textbf{(122)}$$

$$\overline{t_2} = t_2 + \delta_2 \quad \textbf{(123)}$$

$$\overline{t_3} = t_3 + \delta_3 \quad \textbf{(124)}$$

where δ_1 , δ_2 and δ_3 are the displacement of the respective segment triad nodes in the global coordinate system. The displaced triad points are used to define the rotation matrix $R_{\overline{T}}^{G}$ and offset $c_{\overline{T}}$ from the displaced local segment coordinate system to the global coordinate system, in the same manner as nominal local segment coordinate system is equations (117) and (118). These two rotation matrices are used to define a transformation matrix $T_T^{\overline{T}}$, that transforms the segment triad nodes from their nominal positions to their displaced positions, in the nominal local segment coordinate system.

$$T_T^{\overline{T}} = R_{\overline{T}}^{T} = R_G^T R_{\overline{T}}^{G} \quad \textbf{(125)}$$

Since the segments are modeled as rigid bodies, the sensor nodes are fixed in space relative to the segment triad nodes. Therefore we can use rotation and transformation matrices defined above to transform the nominal sensor nodes to their displaced position in the global coordinate system, by the following transformation:

$$\overline{s}_n = T_G^{\overline{G}} \left(s_n - c_t \right) + \overline{c}_t \qquad \textbf{(126)}$$

$$T_G^{\overline{G}} = R_T^G T_T^{\overline{T}} R_G^T = R_T^G R_G^T R_{\overline{T}}^G R_G^T = R_{\overline{T}}^G R_G^T \qquad \textbf{(127)}$$

Where s_n and $\overline{s}_n$ are the nominal and displaced node position respectively for each of the twelve edge sensor nodes, where n goes from 1 to 12, on a given segment. Since this transformation is relative to a single segment, each segment will have an independently calculated transformation matrix $T_G^{\overline{G}}$ and offsets c_t , and $\overline{c}_t$.

The sensor model output is the vertical displacement of the passive sensor node in the active sensor node's coordinate system, where the vertical direction is the $\hat{k}_{\overline{T}}$ unit vector, that is to say it is in the direction of the vector normal to displaced segment's surface, aligned in the path of the reflected light off the surface. To calculate this, all sensors are displaced from their nominal position in the global coordinate system, based on the actuator triad displacements. The relative displacement is then calculated between all edge sensor node pairs. These displacements are then rotated into their respective active sensor node coordinate systems via the associated $R_{\overline{T}}^G$ rotation matrix, and the $\hat{k}_{\overline{T}}$ direction is retained as the sensor reading. As an example, assume $\overline{s}_a$ and $\overline{s}_p$ are the displaced node positions of the active and passive sensor node pair respectively, in the global coordinate system. The sensor output reading s_o for a single sensor node pair is:

$$
\begin{aligned}
s_o &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}_{1\times 3} R_G^{\bar{T}} \left(\bar{s}_p - \bar{s}_a \right) \\
&= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}_{1\times 3} \left(R_{\bar{T}}^{G} \right)^{*} \left(\bar{s}_p - \bar{s}_a \right) \\
&= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}_{1\times 3} \left(\begin{bmatrix} \hat{i}_{\bar{t}} & \hat{j}_{\bar{t}} & \hat{k}_{\bar{t}} \end{bmatrix} \right)^{*} \left(\bar{s}_p - \bar{s}_a \right) \\
&= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}_{1\times 3} \begin{bmatrix} \left(\hat{i}_{\bar{t}} \right)^{*} \\ \left(\hat{j}_{\bar{t}} \right)^{*} \\ \left(\hat{k}_{\bar{t}} \right)^{*} \end{bmatrix}_{3\times 3} \left(\bar{s}_p - \bar{s}_a \right) \\
&= \left(\hat{k}_{\bar{t}} \right)^{*} \left(\bar{s}_p - \bar{s}_a \right)
\end{aligned}
\qquad \textbf{(128)}
$$

where $R_G^{\bar{T}}$, and subsequently $\hat{k}_{\bar{t}}$, is based on displaced segment triad associated with the active sensor node $\bar{s}_a$.

CHAPTER 4

SEGMENTED MIRROR CONTROLLER DESIGN AND SYNTHESIS

4.1 Chapter Overview

H_∞ control is based on finding a feasible state-space feedback controller that satisfies the following conditions:

- the closed-loop system is internally stable and the
- the closed-loop system performance satisfies

$$\left\| \langle G, K \rangle \right\|_\infty < \gamma \qquad \textbf{(129)}$$

Global control based on the H_∞ norm provides a tempting solution to the segment mirror control problem. However as the number of segments increase to the order of hundreds or thousands of segments, the problem becomes intractable due to the excessive

computation complexity and capacity required. Even with tens of segments the computation burden pushes the limits of current desktop computing.

One potential solution to the scalability problem is to develop a decentralized segment based H_∞ controller. Each segment would then be controlled by a discrete and isolated controller. Scaling up the number of segments only increases the number of controllers required, not the complexity of synthesizing the controller. By gaining scalability we have sacrificed global control, since each controller works independently. An alternative solution is a network of distributed controllers, where discrete distributed controller units are interconnected to share information with neighboring controller units. This solution satisfies both the scalability requirement and introduces a global control aspect into the control system. Both the decentralized and distributed controllers should be robust enough such that that same controller can be used for all segments, regardless of its position in the segment surface or the number of segments in the surface, yet the distributed controller should provide superior overall performance.

4.2 H_∞ Controller Synthesis Overview

The closed loop system is shown in the standard state-space form in (130), where the superscripts C, G, and K denote the closed loop system, the open loop system and the controller respectively. It is assumed that $D_{yu}^{G} = 0$, without loss of generality by loop shifting [34]:

$$\begin{bmatrix} A^C & B^C \\ C^C & D^C \end{bmatrix} = \begin{bmatrix} A_o & B_o \\ C_o & D_o \end{bmatrix} + \begin{bmatrix} \beta \\ D_{12} \end{bmatrix} \begin{bmatrix} A^K & B^K \\ C^K & D^K \end{bmatrix} \begin{bmatrix} \varsigma & D_{21} \end{bmatrix} \quad \textbf{(130)}$$

$$A_o = \begin{bmatrix} A^G & 0 \\ 0 & 0 \end{bmatrix} \quad \textbf{(131)}$$

$$B_o = \begin{bmatrix} B_d^G \\ 0 \end{bmatrix} \quad \textbf{(132)}$$

$$C_o = \begin{bmatrix} C_z^G & 0 \end{bmatrix} \quad \textbf{(133)}$$

$$D_o = D_{zd}^G \quad \textbf{(134)}$$

$$\beta = \begin{bmatrix} 0 & B_u^G \\ I & 0 \end{bmatrix} \quad \textbf{(135)}$$

$$\varsigma = \begin{bmatrix} 0 & I \\ C_y^G & 0 \end{bmatrix} \quad \textbf{(136)}$$

$$D_{12} = \begin{bmatrix} 0 & D_{zu}^G \end{bmatrix} \quad \textbf{(137)}$$

$$D_{21} = \begin{bmatrix} 0 \\ D_{yd}^G \end{bmatrix} \quad \textbf{(138)}$$

The standard H_∞ sub-optimal problem is based on the closed loop response of the system satisfying the condition in (139) and that A^C is stable, such that the real portion of all eigenvalues are less than 0.

$$\left\| D^C + C^C \left(sI - A^C \right)^{-1} B^C \right\|_\infty < \gamma \quad \textbf{(139)}$$

A linear matrix inequality (LMI) approach, as discussed in [38], is used to synthesize the H_∞ controller. The equivalent LMI closed loop response condition is (140):

$$\begin{bmatrix} \left(A^C\right)^* X + XA^C & XB^C & \left(C^C\right)^* \\ \left(B^C\right)^* X & -\gamma I & \left(D^C\right)^* \\ C^C & D^C & -\gamma I \end{bmatrix} < 0 \qquad \textbf{(140)}$$

The above condition can be rewritten in the following LMI form shown in (141) to (146). This is done to remove the controller matrices, A^K, B^K, C^K and D^K from the LMI, so that it can be solved for a feasible solution of X by first solving for X^G and Y^G:

$$\begin{bmatrix} N_Y & 0 \\ 0 & I \end{bmatrix}^* \begin{bmatrix} \begin{bmatrix} A^G Y^G + Y^G (A^G)^* & Y^G (C_z^G)^* \\ C_z^G Y^G & -\gamma I \end{bmatrix} & \begin{bmatrix} B_d^G \\ D_{zd}^G \end{bmatrix} \\ \begin{bmatrix} (B_d^G)^* & (D_{zd}^G)^* \end{bmatrix} & -\gamma I \end{bmatrix} \begin{bmatrix} N_Y & 0 \\ 0 & I \end{bmatrix} < 0 \qquad \textbf{(141)}$$

$$\begin{bmatrix} N_X & 0 \\ 0 & I \end{bmatrix}^* \begin{bmatrix} \begin{bmatrix} (A^G)^* X^G + X^G A^G & X^G B_d^G \\ (B_d^G)^* X^G & -\gamma I \end{bmatrix} & \begin{bmatrix} (C_z^G)^* \\ (D_{zd}^G)^* \end{bmatrix} \\ \begin{bmatrix} C_z^G & D_{zd}^G \end{bmatrix} & -\gamma I \end{bmatrix} \begin{bmatrix} N_X & 0 \\ 0 & I \end{bmatrix} < 0 \qquad \textbf{(142)}$$

$$\begin{bmatrix} X^G & I \\ I & Y^G \end{bmatrix} \geq 0 \qquad \textbf{(143)}$$

$$N_Y = \begin{bmatrix} B_u & D_{zu} \end{bmatrix} \qquad \textbf{(144)}$$

$$N_X = \begin{bmatrix} C_y & D_{yd} \end{bmatrix} \qquad \textbf{(145)}$$

$$X = \begin{bmatrix} X^G & X^{GK} \\ \left(X^{GK}\right)^* & X^K \end{bmatrix} = Y^{-1} = \begin{bmatrix} Y^G & Y^{GK} \\ \left(Y^{GK}\right)^* & Y^K \end{bmatrix}^{-1} \qquad \textbf{(146)}$$

By simultaneously solving the LMIs (141), (142) and (143), we get a feasible solution for X^G and Y^G, which coupled by (146). Equation (146) can be rewritten in the form of (147):

$$XY = I = \begin{bmatrix} X^G & X^{GK} \\ \left(X^{GK}\right)^* & X^K \end{bmatrix} \begin{bmatrix} Y^G & Y^{GK} \\ \left(Y^{GK}\right)^* & Y^K \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \quad \textbf{(147)}$$

$$X^G Y^G + X^{GK}\left(Y^{GK}\right)^* = I \quad \textbf{(148)}$$

$$\left(X^{GK}\right)^* Y^G + X^K \left(Y^{GK}\right)^* = 0 \quad \textbf{(149)}$$

$$X^G Y^{GK} + X^{GK} Y^K = 0 \quad \textbf{(150)}$$

$$\left(X^{GK}\right)^* Y^{GK} + X^K Y^K = I \quad \textbf{(151)}$$

Rewriting equations (148) to (151) allows us to compute the two full column rank matrices X^{GK} and Y^{GK} via the signal value decomposition, as shown below:

$$X^{GK}\left(Y^{GK}\right)^* = I - X^G Y^G \quad \textbf{(152)}$$

$$\left(I - X^G Y^G\right) = U_{GK} \Sigma_{GK} \left(V_{GK}\right)^* \quad \textbf{(153)}$$

$$X^{GK} = U_{GK} \Sigma_{GK} \quad \textbf{(154)}$$

$$Y^{GK} = V_{GK} \quad \textbf{(155)}$$

Then from equation (149) X^K can be computed, which together with the above results yields an adequate solution for X. The feasible solution for X is then use to solve the LMI in (140) for feasible solution for controller matrices, A^K, B^K, C^K and D^K.

The bounding limit γ is minimized subject to the feasibility of the solution to the above equations, to provide a sub-optimal controller. Minimizing γ does not strictly improve the controller's performance, as it is only an upper bound on the magnitude of closed

loops system's transfer functions. Therefore care must be taken when selecting γ, as the open loop system is typically a reduced system to ease the computation burden of solving for the LMI constraints, meaning the resulting controller may excite unmodeled modes that degrade the closed loop performance.

4.3 Global H_∞ Control

Global H_∞ control is based on the concept of tying all the sensor information into one centralized controller which determines an appropriate course of action for all the actuators in the system. Potentially this control solution may yield the best overall performance, however as discussed above, may quickly become an intractable problem as the system grows in size and complexity. For the seven segments SMCT model, considered in this work, global control is a viable solution, and is used as a baseline comparator for more scalable solutions.

The synthesis of the global H_∞ controller follows the H_∞ synthesis procedure as presented above, where the open loop system model is the reduced plant state-space model presented in Section 3.3.1. The resulting controller acts with knowledge of the entire system and subsequently attempts to control the system as a single entity.

4.4 Decentralized H_∞ Control

Decentralized H_∞ control is a highly scalable solution to the segmented mirror control problem, in which each segment is controlled independently with no knowledge of one another. This solution sacrifices global performance for scalability, since the addition of further segments in the system only requires the addition of an equal number of isolated controllers. However since individual segment controllers are isolated and act independently, they may act with contrary objectives and therefore interfere with one another since the segments themselves are not isolated systems.

The synthesis of the decentralized H_∞ controller follows the same procedure as the global H_∞ controller does, however the open loop system model is replaced by the reduced decentralized state-space model presented in Section 3.4. Therefore a single segment based decentralized controller is produced which can be replicated to control each of the segments in the system independently. The resulting control system is an assembly of individual controllers acting on localized information only with no regard for the other controllers in the system.

Note that the same decentralized controller model is used for all segments in the SMCT system. This controller therefore must be robust enough to handle any variation in segment dynamics, which can be thought of as unmodeled dynamics in the system that adds to the model's uncertainty.

4.5 Distributed Control

Distributed control attempts to combine the scalability of the decentralized control with the overall performance of global control. This is achieved by generating an array of interconnected distributed controller units, each dedicated to the control of a single segment, however directly sharing information with surrounding controllers. This control system is constructed very similarly to the decentralized control system, with the important alteration that individual controllers are directly coupled to adjacent controllers, such that information can ripple from one extent of the system to the other. The formulation of a distributed controller unit, shown in (156), is similar to the distributed system unit, described is Section 3.5, which it is based on:

$$\begin{bmatrix} \dot{x}_t^K \\ w^K \\ u^K \end{bmatrix} = \begin{bmatrix} A_{tt}^K & A_{ts}^K & B_t^K \\ A_{st}^K & A_{ss}^K & B_s^K \\ C_t^K & C_s^K & D^K \end{bmatrix} \begin{bmatrix} x_t^K \\ v^K \\ y^K \end{bmatrix} \qquad \textbf{(156)}$$

Similarly to the decentralized controller the same distributed controller unit model is used for all segments in the SMCT system. The synthesis process for a distributed controller unit is a modified form of the LMI H_∞ synthesis process described in Section 4.2. This modified procedure is summarized below, while a detailed proof of the distributed controller synthesis process is given in [7].

The distributed controller synthesis process is based on the notion of modifying the distributed system unit, presented in (85), such that it can be treated as a standard state-

space system in the context of the H_∞ LMI approach discussed in Section 4.2. The resulting controller, in a standard state-space form, is then modified to be in the form of (156), such that the interconnected control system acts within certain performance bounds.

The modification required to the distributed system unit is shown in equation (158) to (161), and can be considered comparable to a bilinear transform of the discrete spatial states into equivalent continuous spatial states:

$$H = \begin{bmatrix} I_{n_i} & & & & 0 \\ & -I_{n_i} & & & \\ & & \ddots & & \\ & & & I_{n_i} & \\ 0 & & & & -I_{n_i} \end{bmatrix}_{6n_i \times 6n_i} \qquad \textbf{(157)}$$

$$\bar{A}_{ss}^{G} = -\left(I + \tilde{A}_{ss}^{G}\right)\left(I + \tilde{A}_{ss}^{G}\right)^{-1} H \qquad \textbf{(158)}$$

$$\begin{bmatrix} \bar{A}_{st}^{G} & \bar{B}_{s\bullet}^{G} \end{bmatrix} = \sqrt{2}\left(I - \tilde{A}_{ss}^{G}\right)^{-1}\begin{bmatrix} \tilde{A}_{st}^{G} & \tilde{B}_{s\bullet}^{G} \end{bmatrix} \qquad \textbf{(159)}$$

$$\begin{bmatrix} \bar{A}_{ts}^{G} \\ \bar{C}_{\bullet s}^{G} \end{bmatrix} = -\sqrt{2}\begin{bmatrix} \tilde{A}_{ts}^{G} \\ \tilde{C}_{\bullet s}^{G} \end{bmatrix}\left(I - \tilde{A}_{ss}^{G}\right)^{-1} H \qquad \textbf{(160)}$$

$$\begin{bmatrix} \bar{A}_{tt}^{G} & \bar{B}_{t\bullet}^{G} \\ \bar{C}_{\bullet t}^{G} & \bar{D}_{\bullet\bullet}^{G} \end{bmatrix} = \begin{bmatrix} \tilde{A}_{tt}^{G} & \tilde{B}_{t\bullet}^{G} \\ \tilde{C}_{\bullet t}^{G} & \tilde{D}_{\bullet\bullet}^{G} \end{bmatrix} + \begin{bmatrix} \tilde{A}_{ts}^{G} \\ \tilde{C}_{\bullet s}^{G} \end{bmatrix}\left(I - \tilde{A}_{ss}^{G}\right)^{-1}\begin{bmatrix} \tilde{A}_{st}^{G} & \tilde{B}_{s\bullet}^{G} \end{bmatrix} \qquad \textbf{(161)}$$

This modified system can then be used as the open-loop system in the modified set of LMIs, shown below, that are analogous to the LMI presented in (141) to (143):

$$\begin{bmatrix} \bar{N}_Y & 0 \\ 0 & I \end{bmatrix}^* \begin{bmatrix} \begin{bmatrix} \bar{A}_{\bullet\bullet}^G \bar{Y}^G + \bar{Y}^G (\bar{A}_{\bullet\bullet}^G)^* & \bar{Y}^G (\bar{C}_{z\bullet}^G)^* \\ \bar{C}_{z\bullet}^G \bar{Y}^G & -\gamma I \end{bmatrix} & \begin{bmatrix} \bar{B}_{\bullet d}^G \\ \bar{D}_{zd}^G \end{bmatrix} \\ \begin{bmatrix} (\bar{B}_{\bullet d}^G)^* & (\bar{D}_{zd}^G)^* \end{bmatrix} & -\gamma I \end{bmatrix} \begin{bmatrix} \bar{N}_Y & 0 \\ 0 & I \end{bmatrix} < 0 \quad \textbf{(162)}$$

$$\begin{bmatrix} \bar{N}_X & 0 \\ 0 & I \end{bmatrix}^* \begin{bmatrix} \begin{bmatrix} (\bar{A}_{\bullet\bullet}^G)^* \bar{X}^G + \bar{X}^G \bar{A}_{\bullet\bullet}^G & \bar{X}^G \bar{B}_{\bullet d}^G \\ (\bar{B}_{\bullet d}^G)^* \bar{X}^G & -\gamma I \end{bmatrix} & \begin{bmatrix} (\bar{C}_{z\bullet}^G)^* \\ (\bar{D}_{zd}^G)^* \end{bmatrix} \\ \begin{bmatrix} \bar{C}_{z\bullet}^G & \bar{D}_{zd}^G \end{bmatrix} & -\gamma I \end{bmatrix} \begin{bmatrix} \bar{N}_X & 0 \\ 0 & I \end{bmatrix} < 0 \quad \textbf{(163)}$$

$$\begin{bmatrix} \bar{X}_t^G & I \\ I & \bar{Y}_t^G \end{bmatrix} \geq 0 \quad \textbf{(164)}$$

$$\bar{N}_Y = \begin{bmatrix} \bar{B}_{\bullet u} & \bar{D}_{zu} \end{bmatrix} \quad \textbf{(165)}$$

$$\bar{N}_X = \begin{bmatrix} \bar{C}_{y\bullet} & \bar{D}_{yd} \end{bmatrix} \quad \textbf{(166)}$$

$$\bar{X} = \begin{bmatrix} \bar{X}^G & \bar{X}^{GK} \\ (\bar{X}^{GK})^* & \bar{X}^K \end{bmatrix} = \bar{Y}^{-1} = \begin{bmatrix} \bar{Y}^G & \bar{Y}^{GK} \\ (\bar{Y}^{GK})^* & \bar{Y}^K \end{bmatrix}^{-1} \quad \textbf{(167)}$$

$$\bar{X}^G = \begin{bmatrix} \bar{X}_t^G & & & 0 \\ & \bar{X}_{s1}^G & & \\ & & \bar{X}_{s2}^G & \\ 0 & & & \bar{X}_{s3}^G \end{bmatrix} \quad \textbf{(168)}$$

$$\bar{X}^{GK} = \begin{bmatrix} \bar{X}_t^{GK} & & & 0 \\ & \bar{X}_{s1}^{GK} & & \\ & & \bar{X}_{s2}^{GK} & \\ 0 & & & \bar{X}_{s3}^{GK} \end{bmatrix} \quad \textbf{(169)}$$

$$\bar{X}^K = \begin{bmatrix} \bar{X}_t^K & & & 0 \\ & \bar{X}_{s1}^K & & \\ & & \bar{X}_{s2}^K & \\ 0 & & & \bar{X}_{s3}^K \end{bmatrix} \quad \textbf{(170)}$$

An important difference in the distributed form of these LMIs is the structure of the feasible $\bar{X}$ and $\bar{Y}$ matrices; which are block diagonal matrices where $\bar{X}_t$ relates to the temporal states, and $\bar{X}_{s\bullet}$ relates to the spatial states in the positive and the negative

directions for each of the three spatial axes. This results in the inequality (164), where $\overline{X}^G$ and $\overline{Y}^G$ are coupled exclusively through $\overline{X}_t^G$ and $\overline{Y}_t^G$, since the system is spatially symmetric and the notion of positive and negative directions along a spatial axes are arbitrary. However the system is not temporally symmetric therefore this inequality can not be ignored.

As done in the standard H_∞ approach, $\overline{X}^G$ and $\overline{Y}^G$ are used to solve for $\overline{X}$. The feasible solution for $\overline{X}$ is then used to solve for a feasible solution of $\overline{A}_{\bullet\bullet}^K$, $\overline{B}_{\bullet}^K$, $\overline{C}_{\bullet}^K$ and $\overline{D}^K$, based on a equivalent LMI of (140). The resulting controller then needs to be modified in an opposite manner from that which is done in (158) to (161). This reverse transformation can be thought of as transforming the continuous spatial states of the controller in to discrete spatial states, such that the distributed controller units can be assembled as discrete spatially invariant elements. This is done by first solving the LMI in (171):

$$\left(\overline{A}_{ss}^K\right)^* X_s^K + X_s^K \overline{A}_{ss}^K < 0 \qquad \textbf{(171)}$$

$$X_s^K = \begin{bmatrix} X_{s1}^K & & 0 \\ & X_{s2}^K & \\ 0 & & X_{s3}^K \end{bmatrix} \qquad \textbf{(172)}$$

The sub-matrices X_{s1}^K, X_{s2}^K and X_{s3}^K are then factors as shown in (173):

$$X_{s\,i}^K = \left(T_i^K\right)^* \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} T_i^K \quad where \quad i = \{1,2,3\} \qquad \textbf{(173)}$$

The method developed to perform this factorization is based on performing an eigen-value decomposition on the sub-matrices $X_{s\,i}^{K}$, as shown in (174):

$$X_{s\,i}^{K} V_i = V_i D_i \tag{174}$$

Since by construction $X_{s\,i}^{K}$ is symmetric, V_i must be a unitary matrix, therefore we can rewrite (174) as (175):

$$X_{s\,i}^{K} = V_i D_i \left(V_i\right)^{*} \tag{175}$$

By assuming that half of the eigen-values in D_i are strictly greater than 0, and the other half are strictly less than 0, we can reorder V_i and D_i using the permutation matrix P_λ as shown in (176) and (177) where λ_+ are the positive eigen-values and λ_- are the negative eigen-values. This assumption is handled due to the formation of the H matrix in (157).

$$\overline{D}_i = \begin{bmatrix} \lambda_+ & 0 \\ 0 & \lambda_- \end{bmatrix} = P_\lambda D_i \left(P_\lambda\right)^{*} \tag{176}$$

$$\overline{V}_i = V_i \left(P_\lambda\right)^{*} \tag{177}$$

By taking the element wise square-root of the eigen-values as shown in (178) we can then rewrite $\overline{D}_i$ as done in (179).

$$\overline{S}_i = \begin{bmatrix} \sqrt{\lambda_{+,1}} & & & & & \\ & \ddots & & & 0 & \\ & & \sqrt{\lambda_{+,n}} & & & \\ & & & \sqrt{-\lambda_{-,1}} & & \\ & 0 & & & \ddots & \\ & & & & & \sqrt{-\lambda_{-,n}} \end{bmatrix} \tag{178}$$

$$\bar{D}_i = \bar{S}_i \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \bar{S}_i \qquad \textbf{(179)}$$

Therefore from (173) , (175) and (179) we can compute T_i^K as shown in (180) and then define a coordinate transformation matrix T^K shown in (181):

$$T_i^K = \bar{S}_i \left(\bar{V}_i \right)^* \qquad \textbf{(180)}$$

$$T^K = \begin{bmatrix} I & & & 0 \\ & T_1^K & & \\ & & T_2^K & \\ 0 & & & T_3^K \end{bmatrix} \qquad \textbf{(181)}$$

The transformation matrix T_i^K is then applied to the continuous spatial state controller, as shown in (182), to transform the spatial states into a coordinate system that can readily be converted from continuous to discrete states:

$$\begin{bmatrix} \tilde{A}_{\bullet\bullet}^K & \tilde{B}_{\bullet}^K \\ \tilde{C}_{\bullet}^K & \tilde{D}^K \end{bmatrix} = \begin{bmatrix} \left(T^K\right)^{-1} \bar{A}_{\bullet\bullet}^K T^K & \left(T^K\right)^{-1} \bar{B}_{\bullet}^K \\ \bar{C}_{\bullet}^K T^K & \bar{D}^K \end{bmatrix} \qquad \textbf{(182)}$$

The conversion of the controller's spatial states is finally achieved by the process shown in equations (183) to (186), where H^K is defined analogously to H :

$$A_{ss}^K = -\left(\tilde{A}_{ss}^K H^K - I \right)^{-1} \left(\tilde{A}_{ss}^K H^K + I \right) \qquad \textbf{(183)}$$

$$\begin{bmatrix} A_{st}^K & B_s^K \end{bmatrix} = -\sqrt{2} \left(\tilde{A}_{ss}^K H^K - I \right)^{-1} \begin{bmatrix} \tilde{A}_{st}^K & \tilde{B}_s^K \end{bmatrix} \qquad \textbf{(184)}$$

$$\begin{bmatrix} A_{ts}^K \\ C_s^K \end{bmatrix} = \sqrt{2} \begin{bmatrix} \tilde{A}_{ts}^K \\ \tilde{C}_s^K \end{bmatrix} H^K \left(\tilde{A}_{ss}^K H^K - I \right)^{-1} \qquad \textbf{(185)}$$

$$\begin{bmatrix} A_{tt}^K & B_t^K \\ C_t^K & D^K \end{bmatrix} = \begin{bmatrix} \tilde{A}_{tt}^K & \tilde{B}_t^K \\ \tilde{C}_t^K & \tilde{D}^K \end{bmatrix} - \begin{bmatrix} \tilde{A}_{ts}^K \\ \tilde{C}_s^K \end{bmatrix} H^K \left(\tilde{A}_{ss}^K H^K - I \right)^{-1} \begin{bmatrix} \tilde{A}_{st}^K & \tilde{B}_s^K \end{bmatrix} \qquad \textbf{(186)}$$

4.6 Position Error Estimation

Since there are twelve edge sensors per segment and each segment shares these sensors with each of its six neighbor segments, there are effectively six independent sensor signals for each segment. This however assumes that there are edge sensors at the boundaries of the segmented surface as a whole. However the SMCT does not have edge sensors at the boundaries of the segmented surface. There are several motivations to exclude edge sensors at the boundaries which include: cost of the addition sensors, the requirement of having a structure around the boundaries to measure relative to, and given a sufficient number of segments the non-boundary edge sensor providing enough information to estimate the actuator displacements.

The number of non-boundary edge sensors, assuming complete rings of segments and no interior segments are removed, can be calculated by the following simple formula:

$$n_s = 6n_r\left(3n_r + 1\right) \quad \textbf{(187)}$$

where n_s is the number of non-boundary edge sensors, and n_r is the number of rings in the segmented surface, not including the single central segment. In the case of the one ring seven segment SMCT testbed, there are 24 non-boundary edge sensors. This results in redundant information of 3 degrees of freedom of the seven segments relative to their neighbouring segments, which allows for an estimation of the displacement errors for the 21 actuators.

This redundant sensor information is utilized by means of an interaction matrix. The pseudo-inverse of the interaction matrix, referred to as the control matrix, is used to estimate the actuator tip displacement based on the edge sensor readings as discussed in [36]. However this sensing scheme is insensitive to certain displacement modes and sensor error is greatly amplified in these directions [37], as discussed below. The transformation from sensor output to actuator displacement error is achieved by the generation of the interaction matrix. The interaction matrix is a linear approximation of the transformation from actuator displacement to sensor output. To generate the interaction matrix, edge sensor outputs are stored with the actuators in their nominal position. This set of sensor readings is referred to as the base sensor output s_b. In the case of the SMCT, the segments are flat plates fitted to an aspherical surface, which leads to non-zero readings from the edge sensor model when the segments are nominally aligned. In a true segmented mirror the individual segments would be curved to fit the aspherical curve of the overall mirror surface, however non-zero sensor readings would still be expected as a result of any misalignment between the two sensor halves and any other DC effects when the segmented surface is properly phased. Next each actuator is displaced one at a time by a fixed amount. The base sensor output is subtracted from sensor output at each steady-state response to the actuator displacement s_k, and then divided by the magnitude of the actuator displacement $|\delta_k|$, as shown below:

$$\varepsilon_k = \frac{s_k - s_b}{|\delta_k|} \qquad \textbf{(188)}$$

The resulting array ε_k, is then inserted as a column in the interaction matrix, where the subscript k denotes the column number in the interaction matrix and corresponds to the

actuator that was displaced. This array gives a linear response of the sensor output to a unit displacement of an actuator and assembled together as the interaction matrix, E, which gives a linear approximation of the sensor output, $\overline{s}$, based on the set of actuator displacement, δ, as shown below:

$$\mathrm{E} = \begin{bmatrix} \varepsilon_1 & \cdots & \varepsilon_n \end{bmatrix}_{n_s \times n_a} \qquad \textbf{(189)}$$

$$\tilde{s} = \mathrm{E}\delta \qquad \textbf{(190)}$$

where n_a and n_s are the number of actuators and sensors respectively. However the approximation required is the inverse of the interaction matrix, which transforms sensor output to a linear approximation of actuator displacement error. To achieve this, a singular value decomposition, or SVD, is performed on the interaction matrix, as shown below:

$$\mathrm{E} = U_{n_s \times n_s} \Sigma_{n_s \times n_a} V^*_{n_a \times n_a} \qquad \textbf{(191)}$$

The singular value decomposition gives important information as to the sensitivity of the sensor geometry to particular displacement modes. The unitary matrix, V^*, can be viewed as a transform from actuator displacements to singular values modes [36], henceforth referred to sensor modes. The diagonal matrix Σ scales the sensor modes based on their sensitivity with respect to the sensor geometry. The unitary matrix U then transforms the scaled sensor modes into sensor output. As mention above in Section 2.5, the relative edge sensing scheme is completely blind to the global motion of the segmented surface. However since the segmented surface forms an aspheric curve and the three actuators per segment provide only tip, tilt, and piston control of an individual

segment, control of the pure global motion of the segmented surface is not possible, meaning the actuators cannot act in such a way to move the segmented surface as if it was a rigid body. Therefore any combination actuator displacements will be sensed by the edge sensor scheme. The caveat to this statement is that not all sensor modes are well sensed. These poorly sensed modes correspond to small singular values in the Σ matrix, that is to say large motion in these poorly sensed mode directions will result in small variations in the sensor output. If noise is present in the sensor output signals, then sensitivity to these motions could be completely lost.

The control matrix, Γ, which transforms sensor output to a linear approximation of actuator displacement error, is defined as the pseudo-inverse of the interaction matrix, which is calculated by the follow:

$$\Gamma = \mathrm{E}^{-1} = \left(U\Sigma V^{*}\right)^{-1} = V\Sigma^{-1}U^{*} \qquad \textbf{(192)}$$

$$\tilde{\delta} = \Gamma s \qquad \textbf{(193)}$$

where

$$\Sigma = \begin{bmatrix} \Sigma_1 & & 0 \\ & \ddots & \\ 0 & & \Sigma_{n_a} \\ 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{n_s \times n_a} \qquad \textbf{(194)}$$

This assumes that n_s, the number or sensors, is greater then n_a, the number of actuators, which is true in the case of SMCT. The $\left(n_s - n_a\right)$ by n_a block of zeros at the bottom of

the Σ matrix corresponds to the redundant sensor information as a result of having more sensors than actuators. Therefore the inverse of the matrix Σ is given below:

$$\Sigma^{-1} = \begin{bmatrix} 1/\Sigma_1 & & 0 & 0 & \cdots & 0 \\ & \ddots & & \vdots & & \vdots \\ 0 & & 1/\Sigma_{n_a} & 0 & \cdots & 0 \end{bmatrix}_{n_a \times n_s} \quad \textbf{(195)}$$

From this one can see that the poorly sensed modes, those associated with small singular values in the interaction matrix, are associated with large singular values in the control matrix. That means a small error in these poorly sensed modes will be greatly amplified by the control matrix, leading to large errors in the actuator position estimation. For this reason it is suggested that the poorly sensed modes are removed from the control matrix, making the control matrix completely insensitive to any sensor readings corresponding to these modes [37]. This insensitivity is achieved by zeroing the corresponding singular values in the Σ^{-1} matrix. A discussion of which modes are considered poorly sensed and subsequently removed from the control matrix is Section 5.2.3.

CHAPTER 5

SEGMENTED MIRROR CONTROL SIMULATION

5.1 Chapter Overview

The SMCT simulation is based heavily of the software design of the Integrated Model Toolset developed by HIA. It is a highly configurable closed loop simulation custom developed in MATLAB/Simulink, in which a variety of disturbance cases can be applied to an assortment of control scheme configuration including, global, decentralized and distributed controllers. As a case study the simulation parameters used are based on the initial design of the SMCT system. Table 1 summarizes the key values of this model.

Parameter Description	Value
segment side length	0.5 m
segment gap length	0.001 m
radius of curvature	60 m
conical constant	0.9
actuator length	0.1 m
pyramid truss height	0.4 m
actuator triad side length	0.5 m
segment triad side length	0.6 m
sensor corner offset	0.05 m
segment element material	aluminum
truss element material	steel
actuator element material	steel
base truss element diameter	0.04 m
pyramid truss element diameter	0.02 m
actuator to base truss element diameter	0.03 m
top truss element diameter	0.02 m
base actuator support truss element diameter	0.03 m
actuator element diameter	0.02 m

Table 1. Parametric Values Used for SMCT Simulation

5.2 MATLAB/Simulink Simulation

The simulation model developed in MATLAB/Simulink is based on the design of the SMCT. It implements closed loop simulation and consists of five major modules: disturbance force screen model, plant model, edge sensor model, control system model, and output data handling, as shown in Figure 39.

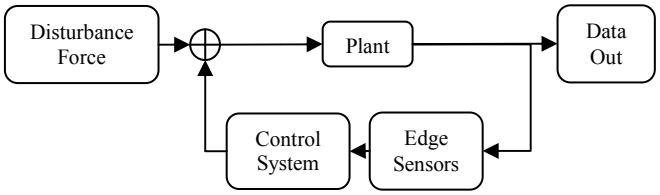

Figure 39. SMCT simulation overview

Force data from the disturbance model is fed into the plant, which results in the displacement of the structural model. These displacements are then fed to the edge sensor model, which calculates the edge sensors readings based on the current segment alignment configuration. The sensor readings are fed into the control system model, which calculates a control signal based on the controller selected for the simulation. The control signals are then fed back into the plant to adjust the actuator positions, which closes the loop. Also at each time step the node displacements and control signals are saved for further analysis.

5.2.1 Disturbance Force Module

There are several different disturbance cases as part of the SMCT simulation software including: random, frozen flow screen, step, impulse, sine wave, and gravity. The random disturbance case applied temporal and spatial random forces in all directions on all of SCMT nodes. These disturbance forces are temporal filtered by a bandpass filter. For preliminary wind buffeting analysis a frozen flow screen case is available, where a force screen is randomly generated and spatially filtered. This force screen is then moved across the structure to simulate wind buffeting on mirror. This model is intended only for

preliminary testing purposes and is not intended as representative wind disturbance force. The step and impulse disturbance cases applied as force step or impulse signal to one or more nodes in the system. The sine wave disturbance case applies an in-phase sinusoidal force to all nodes in the system. Finally the gravity disturbance case gradually ramps up force on all the nodes to equal the gravitational force of the system.

5.2.2 Plant Module

The plant model can take two forms, either the continuous distributed state-space model form or a standard continuous state-space model based on the entire system. This allows for the validation of the distributed modeling form. However for all simulation results presented as a part of the current work, the standard continuous state-space model, as described in Section 3.3, was used as the simulation plant.

5.2.3 Sensor Module

The edge sensor sub-system has three separate modes. In one mode the system output signal z, the actuator displacement along the line of the actuator, is fed directly into the controller model. This mode is used primarily for diagnostics and trouble-shooting of the feedback loop. The second mode is the full edge sensor calculation mode where, system node displacements are fed into the edge sensor model. The resulting sensor reading vector is multiplied by the control matrix which estimates the actuator displacement error. The final mode is a combination of the other two modes, where system output

signal z is multiplied by the interaction matrix to calculate an estimation of the edge sensor readings. The estimated edge sensor readings are multiplied by the control matrix to estimate the actuator displacement error. This is the default mode for the sensor module, since it is more computationally efficient than the full edge sensor mode, yet more realistic than the direct fed-through of the error signal, since the control matrix is blind to particular displacement modes of the system.

As discussed above in Section 4.6, the sensors are insensitive to certain displacement modes, in particular the global motion of the entire segmented surface. The singular values of the interaction matrix, which are directly proportional to the sensitivity of a given displacement mode, are shown in Figure 40, where the x-axis is the sensor mode number and the y-axis is the corresponding singular value.

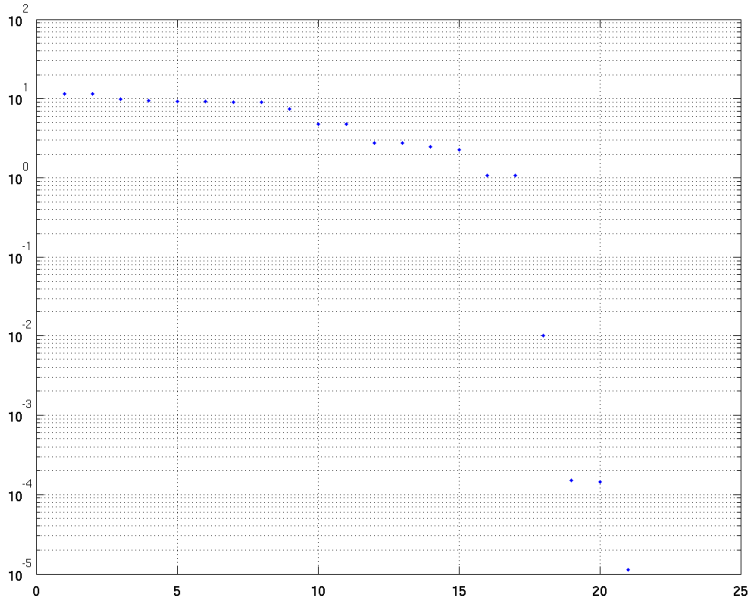

Figure 40. Interaction Matrix Singular Values

The three least sensitive modes, shown in Figure 41, roughly corresponding to the 3 DOF rigid body motion of the entire segmented surface, are removed from the control matrix to reduce error amplification along these directions.

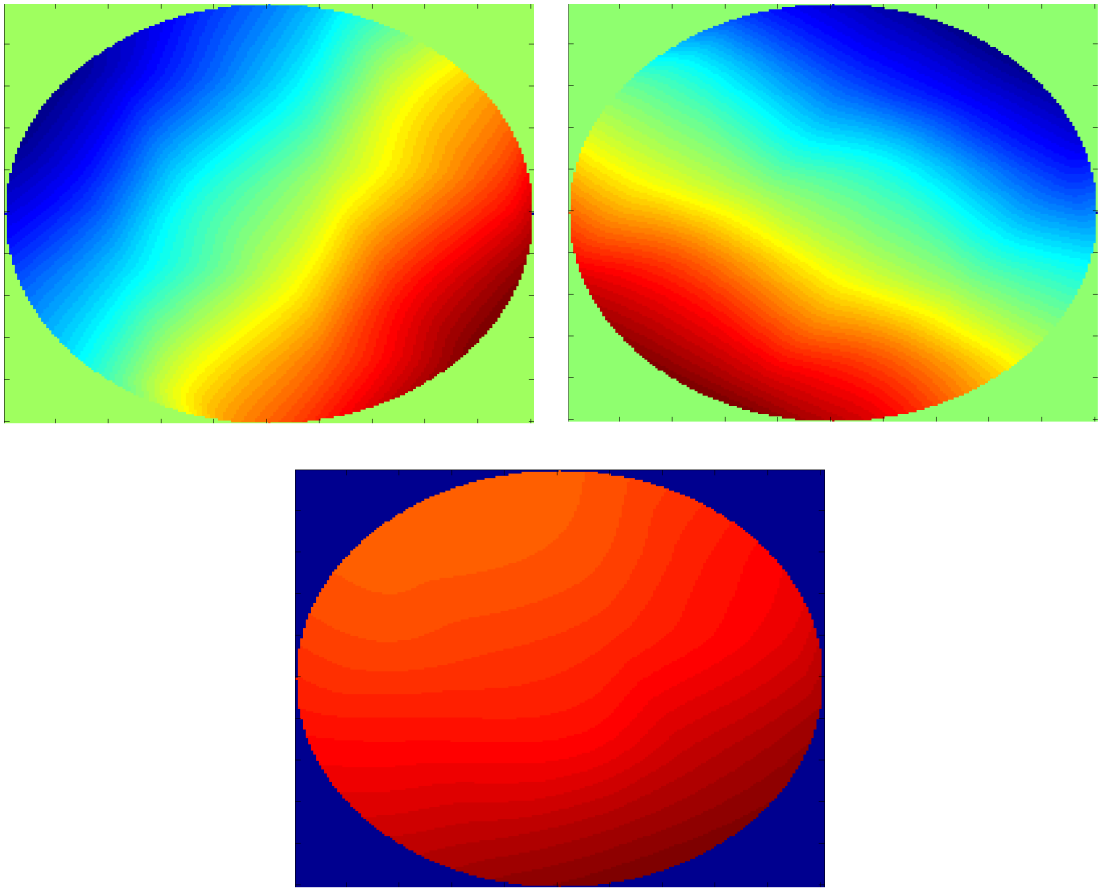

Figure 41. Interaction Matrix Insensitive Singular Value Modes

This results in the control matrix being blind to these modes and therefore the subsequent control system which it feeds is completely unaware of any directly sensed displacements in these directions. The remaining 18 adequately sensed modes are shown in Appendix A.

5.2.4 Control System Module

The control system model can take several forms including: open loop, global state-space, decentralized state-space, PID and distributed, as well each can be run in either continuous or discrete time modes. Open loop control means the control signals are held constant at zero, therefore no active control. The global state-space controller is a single controller for the entire system. The decentralized state-space controller consists of several spatially discrete controllers, each associated with a single segment. The distributed controller consists of several spatially interconnected controllers, each controlling one segment. The PID controller implements actuator based proportional-integral-derivative control where each actuator is controlled independently. However in the scope of this work, only the continuous-time global state-space, decentralized state-space and distributed controller modes are used.

5.2.5 Output Data Module

The output module simply stores all relevant data at each time step of the simulations. This data includes the plant's nodal displacements, the plant's actuator displacements, the edges sensor readings, the estimation of the actuator displacement errors, and the controller force outputs. This information is stored in MATLAB signal structures, inside of a defined output module data structure for further post-processing.

5.2.6 Integral Module

The integral module, which solves the continuous differential state-space equations, is built in as part of Simulink. The solver selected for all simulations was ODE15s. This solver was selected primarily for its efficiency with solving stiff systems [39], as is the case with the SCMT model presented above.

5.3 Controller Synthesis

As discussed in Sections 3.3, 3.4, and 3.5 only the most controllable and observable temporal states are retained in the system models for performance and computational reasons as a part of the controller synthesis process. For the purpose of the synthesizing the SMCT controllers discussed above, the temporal states of the system models were reduced to equal the number of actuators in the system. That is to say the system was reduced to match the controllable degrees of freedom.

After the system model was reduced the controllers were synthesized as described in Chapter 4. For the syntheses of the all the controllers, the decision variable γ was held constant at the values of 1×10^{-4}. This value was chosen as it resulted in adequately performing controllers in each of the controller cases. It should be noted that γ merely sets an upper bound on the performance closed loop system, with respect to the reduced system model, and does not directly govern the final performance when the controller is

applied to the unreduced plant or the real system. Additionally, the γ value was not heavily optimized to avoid any unrealistic bias in the controller's performance, since the inputs system models were not augmented, therefore loop-shaping design of the controllers was not performed.

The algorithm used to solve the LMI as part of the synthesis process is based on the Nesterov and Memiroski's Projective Method described in [40][41]. This algorithm is implemented as a part of the MATLAB Robust Control Toolbox.

5.3.1 Global Controller

In the case of the global plant model there are 720 states. Only the states corresponding to the 21 largest Hankel singular values, shown in Figure 42, were retained in order to synthesis the global controller. In the figure below the x-axis is the singular value index number, and the y-axis is the magnitude of the corresponding Hankel singular values.

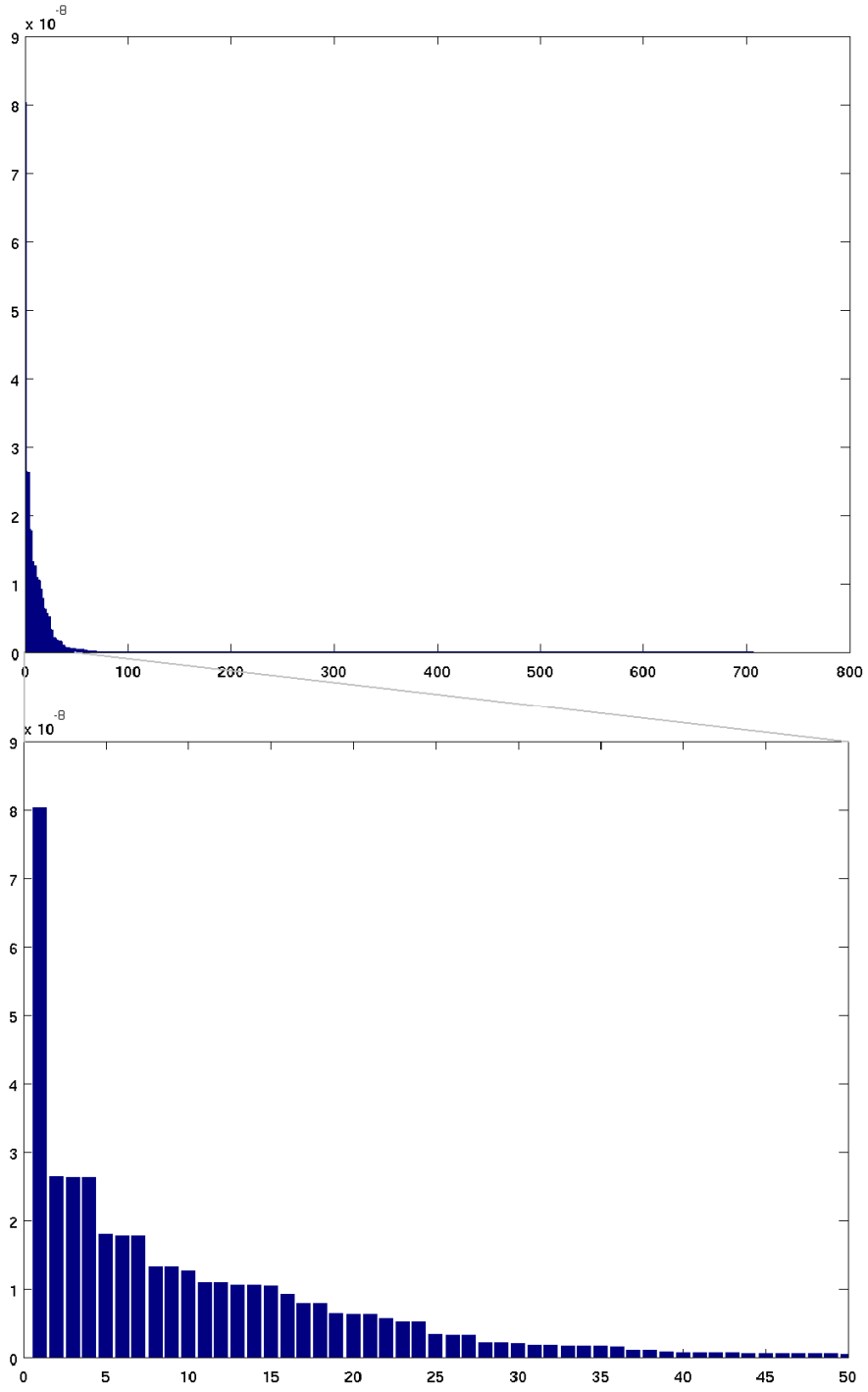

Figure 42. Hankel Singular Values of the Global Model

The reduced global model was then used to synthesis the global controller as discussed in Section 4.3.

5.3.2 Decentralized Controller

The decentralized model consists of 180 states, which was reduced to only 3 states, based on the system's Hankel singular values shown in Figure 43, where the x-axis is the singular value index number, and the y-axis is the magnitude of the corresponding Hankel singular values.

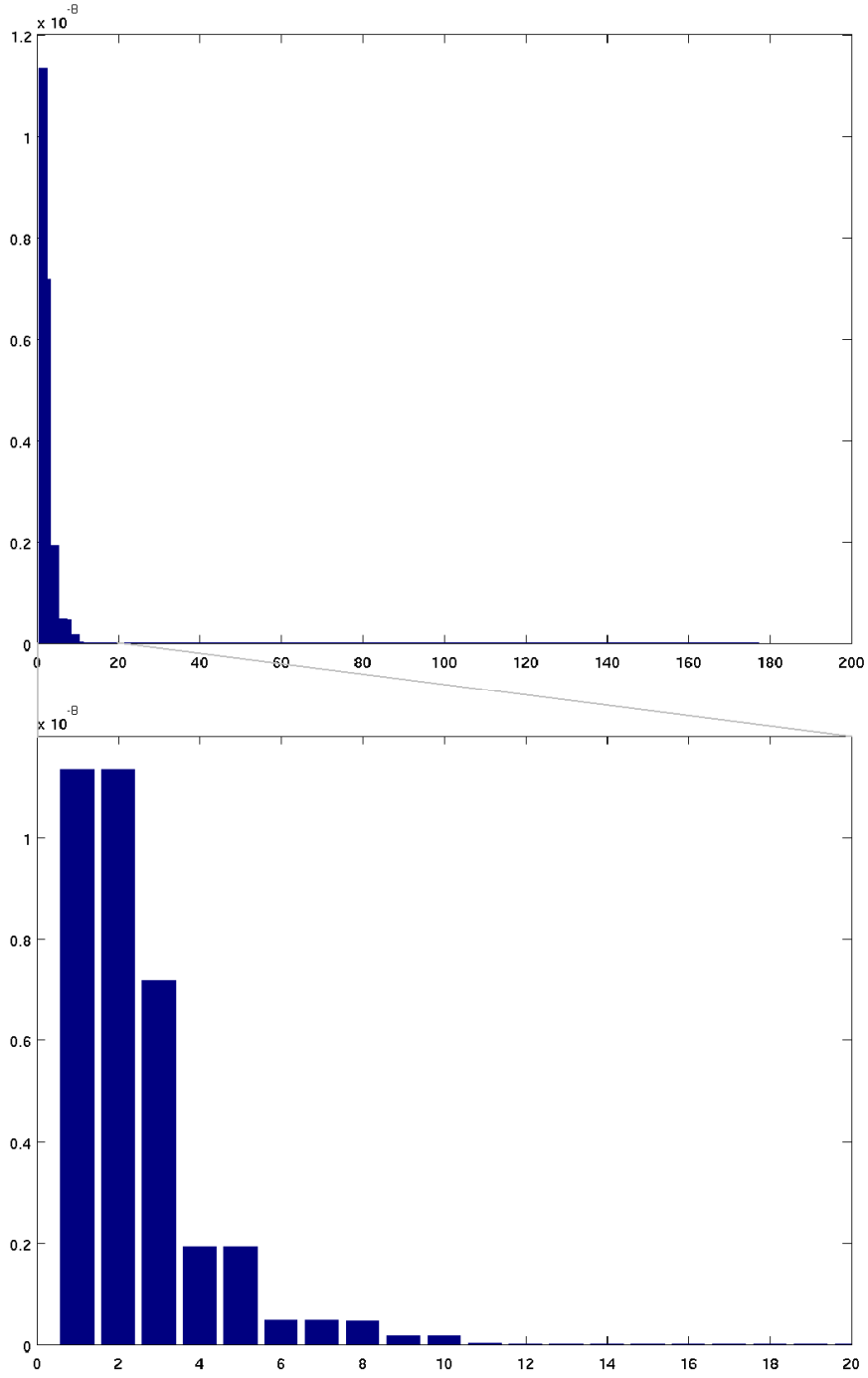

Figure 43. Hankel Singular Values of the Decentralized Model

The reduced decentralized model was then used to synthesis the decentralized controller, as discussed in Section 4.4.

5.3.1 Distributed Controller

In the case of the distributed model there are 180 temporal states, comparable to the decentralized model's states, however with an additional 96 spatial states in each of the 6 spatial directions, for a total of 576 spatial states. As outlined in Section 3.5.2, the disturbed model was reduced to 3 temporal states, based on the system's Hankel singular values shown in Figure 44, where the x-axis is the singular value index number, and the y-axis is the magnitude of the corresponding Hankel singular values.

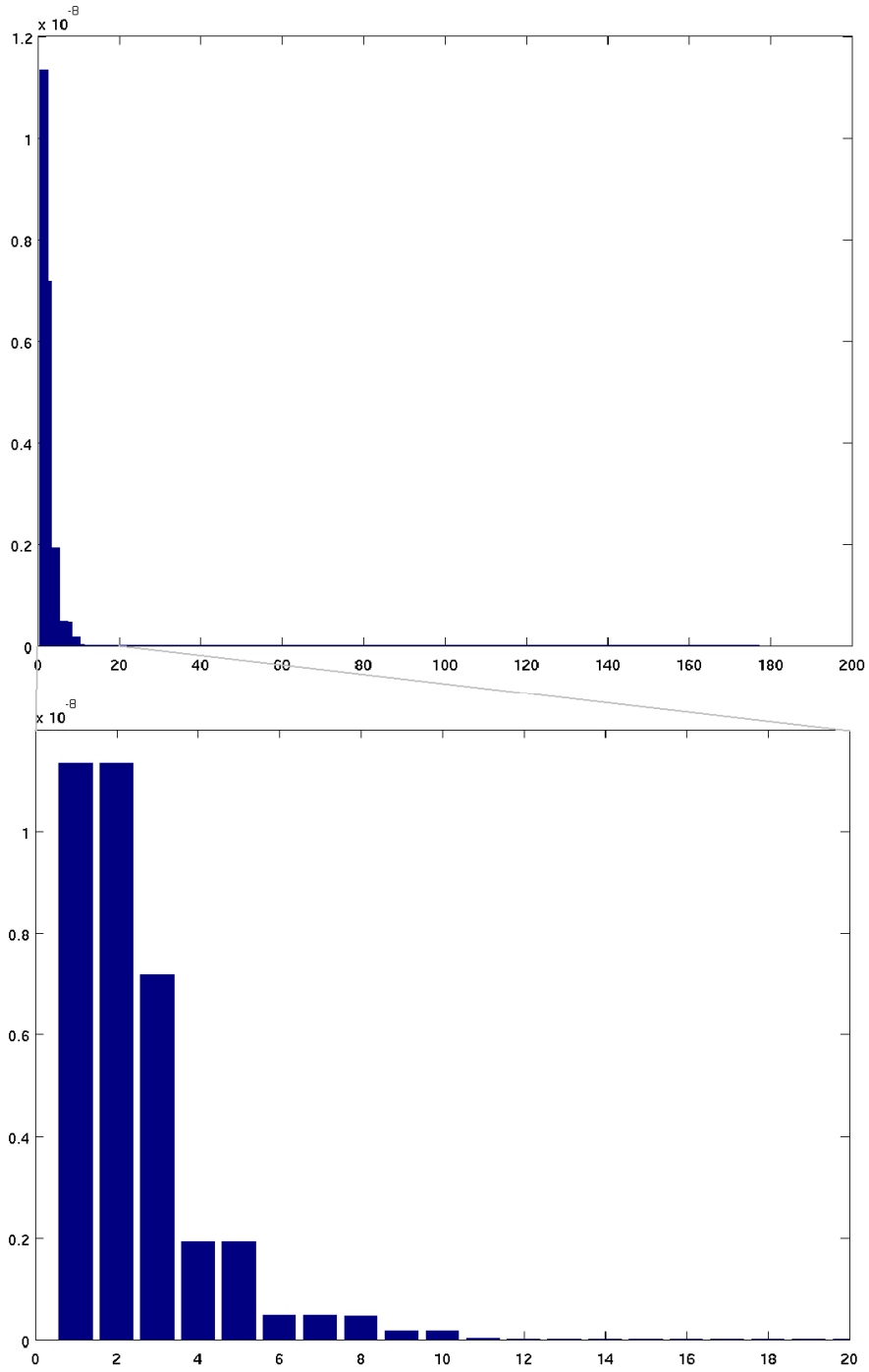

Figure 44. Hankel Singular Values of the Distributed Model

Since the distributed unit model was reduced to only 3 temporal states, this leads to the condition that only 3 spatial states are required in each of the 6 spatial directions, without

the loss of any addition interconnection information. The spatial states could be reduced even further based on the singular values, shown in Figure 45to Figure 50, of the matrix $\tilde{A}_{ts}^{\circ}\tilde{A}_{st}^{\circ}$ from equation (111). However to avoid losing additional spatial information, all 3 of the required states in each of the 6 spatial directions were retained, for at total of 18 spatial states. In the figures below the x-axis is the singular value index number, and the y-axis is the magnitude of the corresponding spatial singular values.

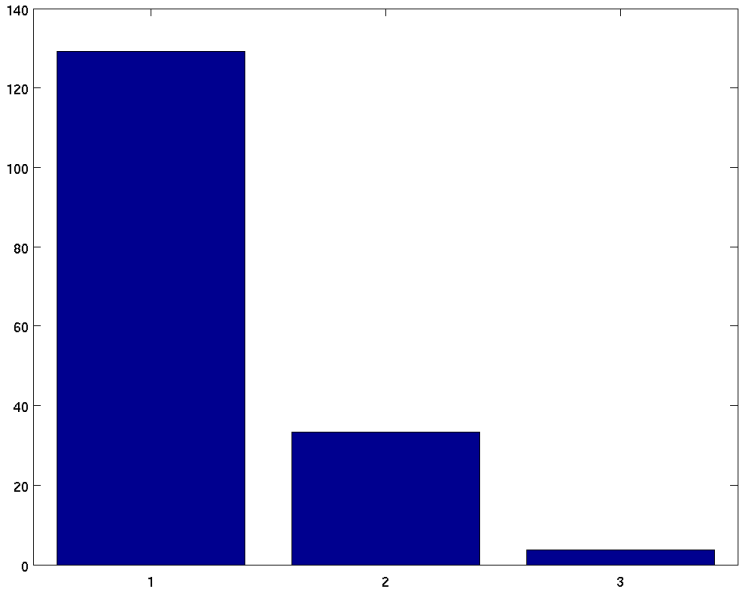

Figure 45. Spatial Singular Values in the 1+ direction

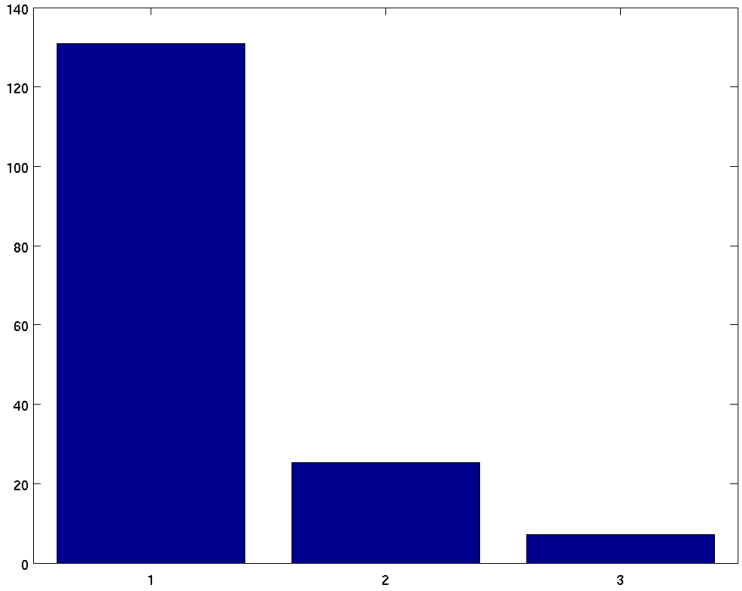

Figure 46. Spatial Singular Values in the 1- direction

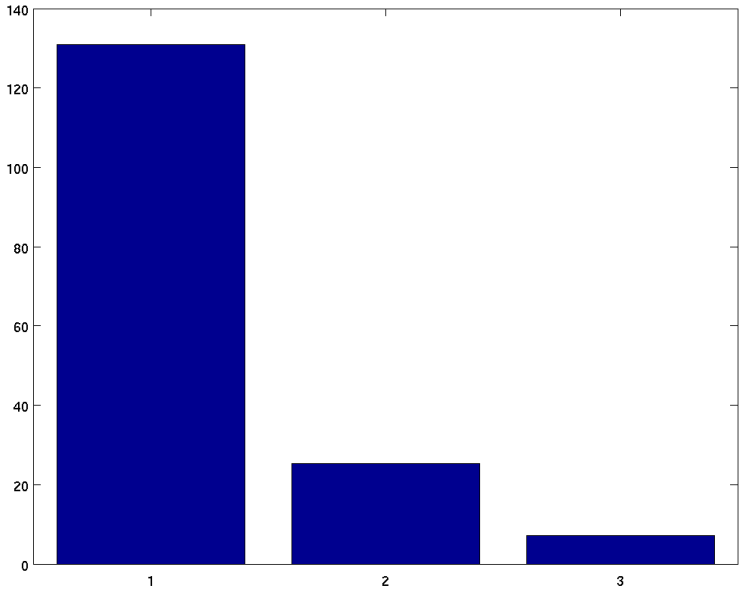

Figure 47. Spatial Singular Values in the 2+ direction

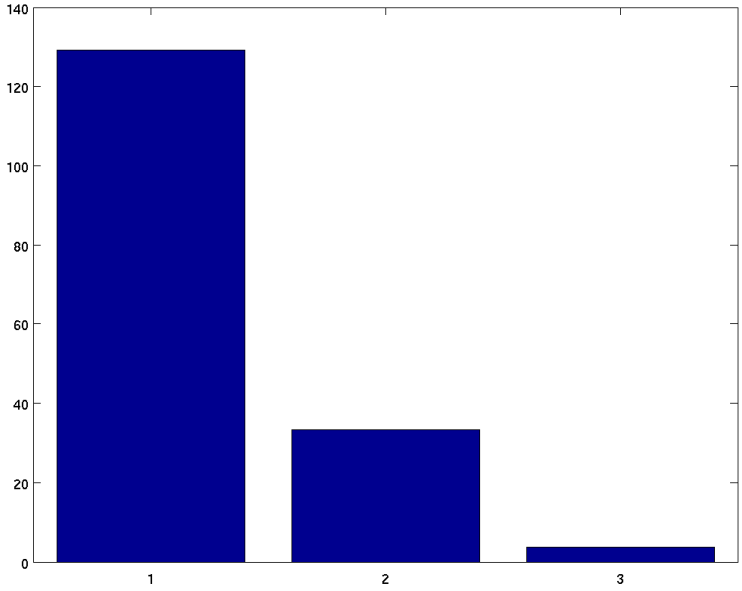

Figure 48. Spatial Singular Values in the 2- direction

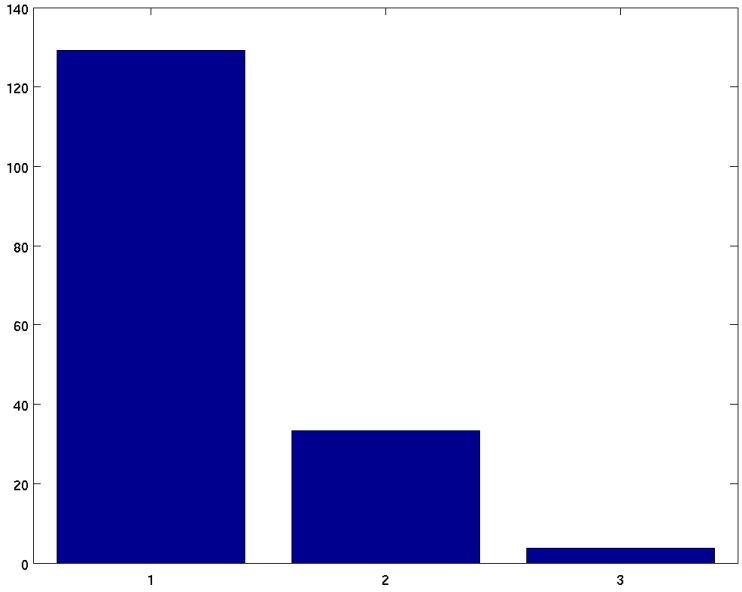

Figure 49. Spatial Singular Values in the 3+ direction

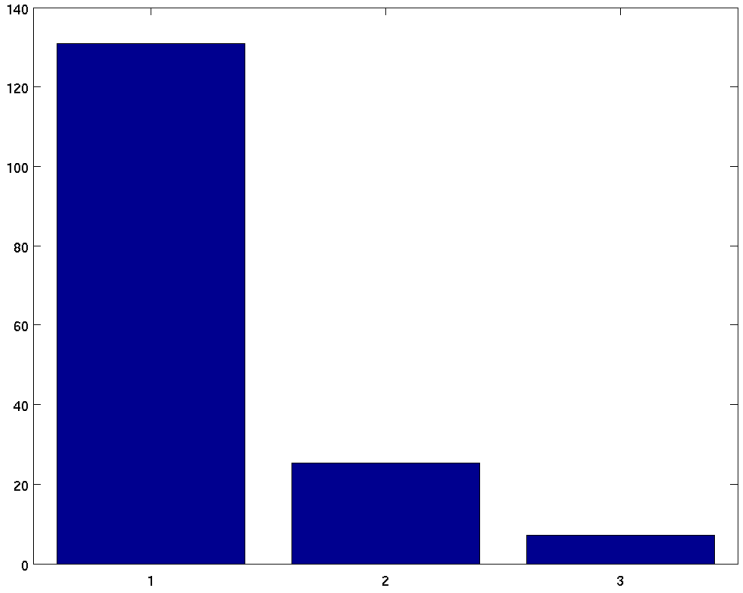

Figure 50. Spatial Singular Values in the 3- direction

The reduced distributed model was then used to synthesis the distributed controller, as discussed in Section 4.5. In addition to the distributed controller, two other forms of the distributed controller were generated. The first form is a disconnected distributed controller, which is to say that the synthesized distributed controller units are not interconnected with their neighboring units. This is achieved by simply removing the matrices corresponding to the spatial states from (156) to form the controller as standard decentralized state-space controller as shown in (196).

$$\begin{bmatrix} \dot{x}_t^K \\ u^K \end{bmatrix} = \begin{bmatrix} A_{tt}^K & B_t^K \\ C_t^K & D^K \end{bmatrix} \begin{bmatrix} x_t^K \\ y^K \end{bmatrix} \quad \textbf{(196)}$$

The purpose of this so called disconnected distributed controller is to contrast the effect that the interconnect signals play in the standard distributed controller network.

The other alternative form of the distributed controller is what is referred to as the assembled distributed controller. In this form the individual distributed units are assembled together to form a single global state space controller. The individual temporal states of the controller units are concatenated together to form an overall state vector. There is no loss of spatial information as a result of the assembly process since spatial states are directly linked to the temporal states of neighboring units. The purpose of the controller is to demonstrate the idealized performance of the distributed controller network, since there is no delay on the interconnection information that is passed between neighboring controller units.

Both of the additional forms of the distributed controller are not intended as a solution to the segmented mirror control problem, nevertheless are useful in the analysis of the distributed control's advantages and disadvantages.

5.4 System Analysis

The parameters of the SMCT model, in particular the element diameters were selected, to achieve a system natural frequency of approximately 60 Hz. The nodal displacement mode at the natural frequency of SMCT global model is shown in Figure 51 to Figure 54, where the x, y, and z axes are in meters and the colour coding represents nodal displacement, which occurs at 60.010 Hz. In addition the element strain at the natural

frequency is also shown in Figure 55 to Figure 58 , where the x, y, and z axes are in meters and the colour coding represents element strain,

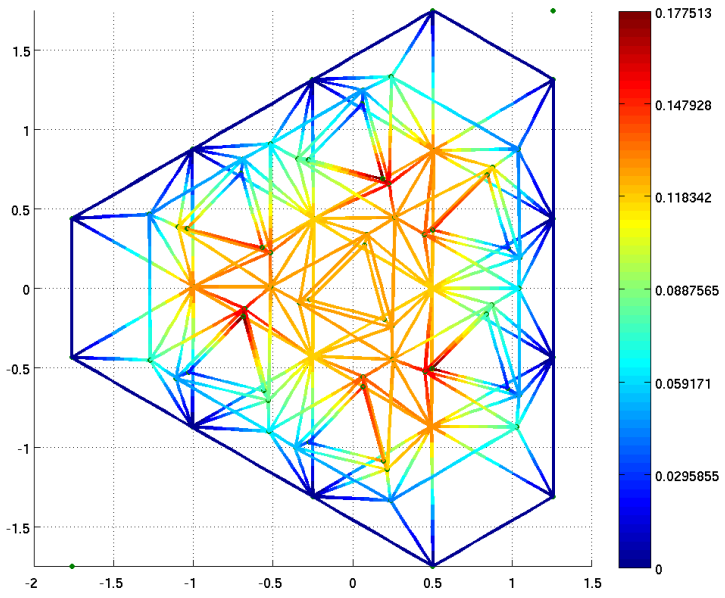

Figure 51. Natural Frequency Nodal Displacement – X-Y Plane View

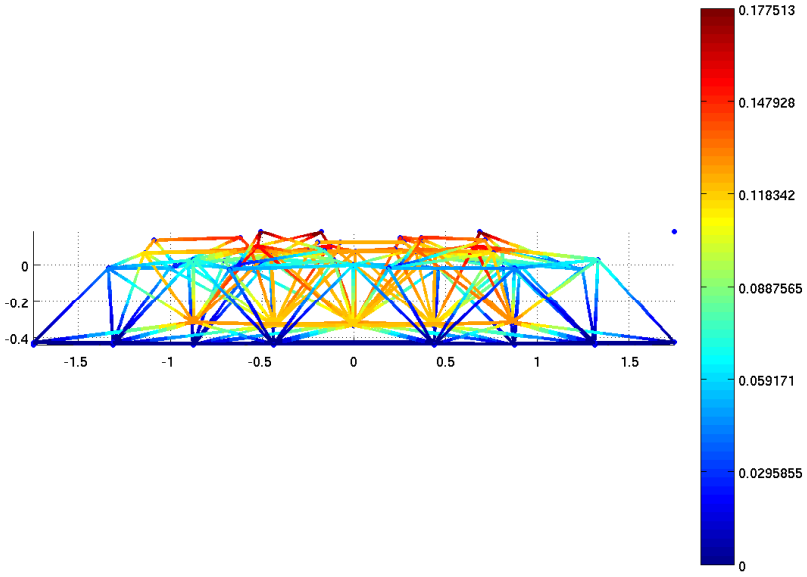

Figure 52. Natural Frequency Nodal Displacement – Y-Z Plane View

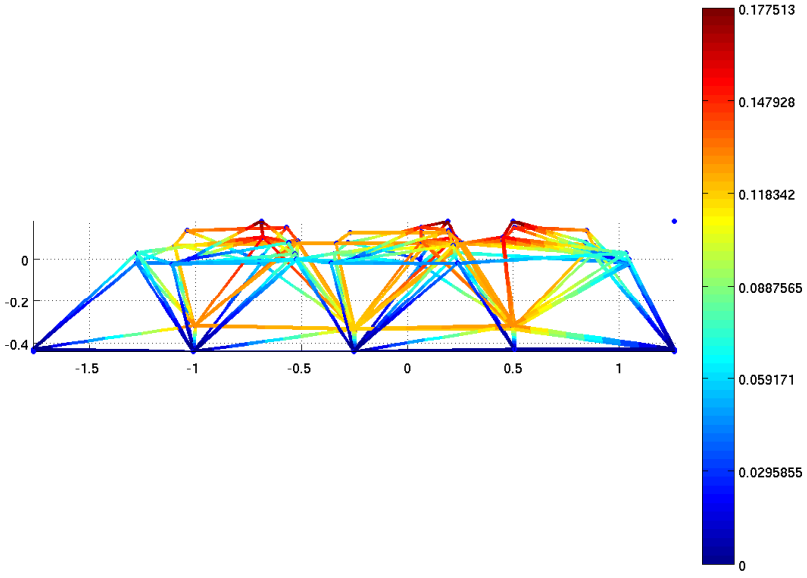

Figure 53. Natural Frequency Nodal Displacement – X-Z Plane View

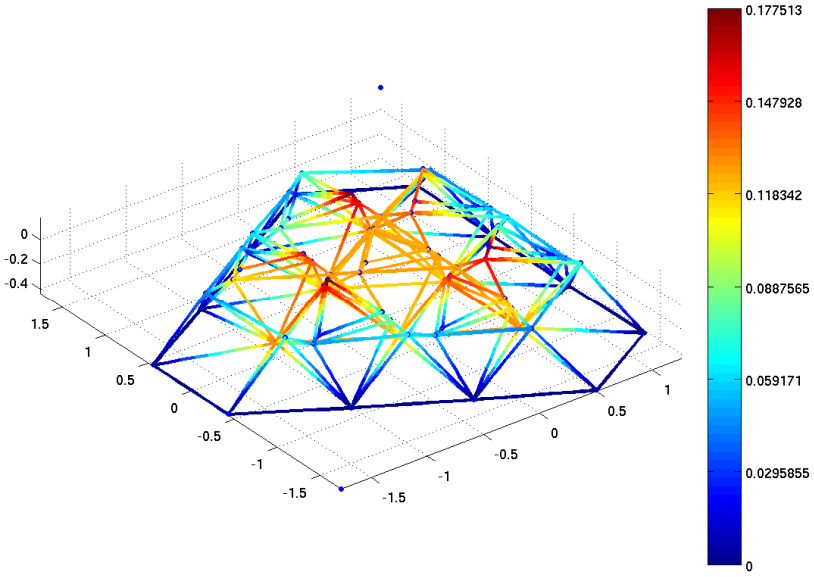

Figure 54. Natural Frequency Nodal Displacement – Perspective View

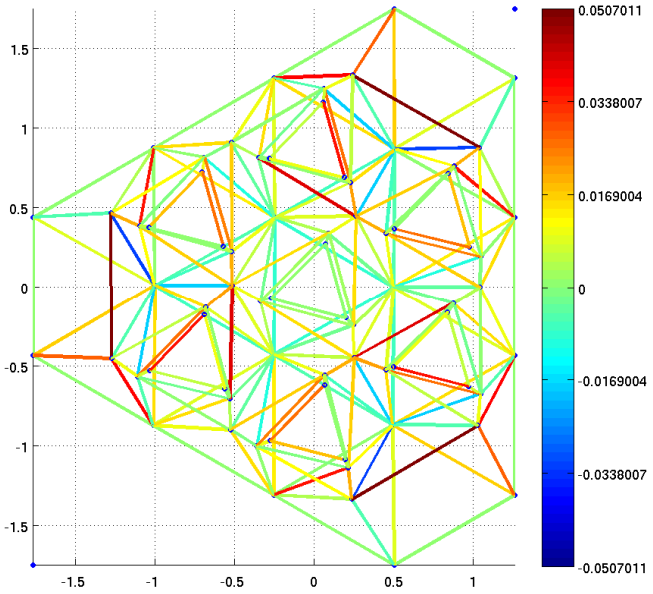

Figure 55. Natural Frequency Element Strain – X-Y Plane View

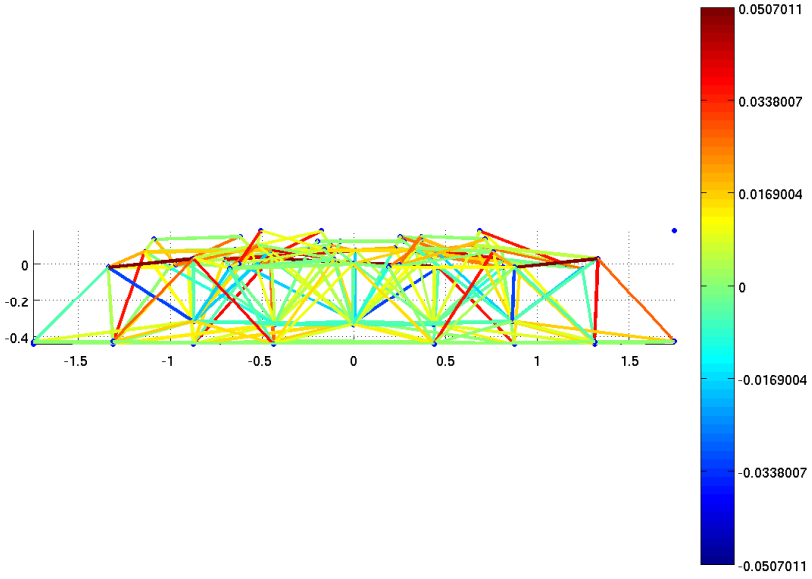

Figure 56. Natural Frequency Element Strain – Y-Z Plane View

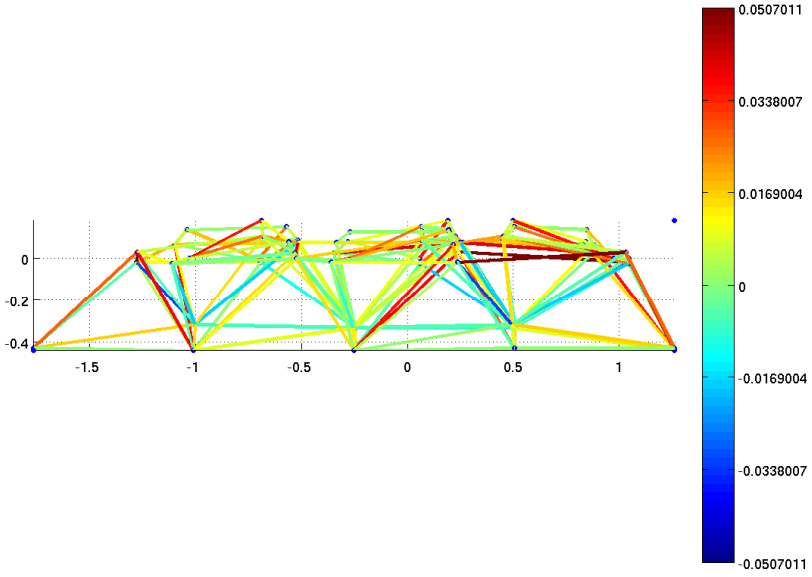

Figure 57. Natural Frequency Element Strain - X-Z Plane View

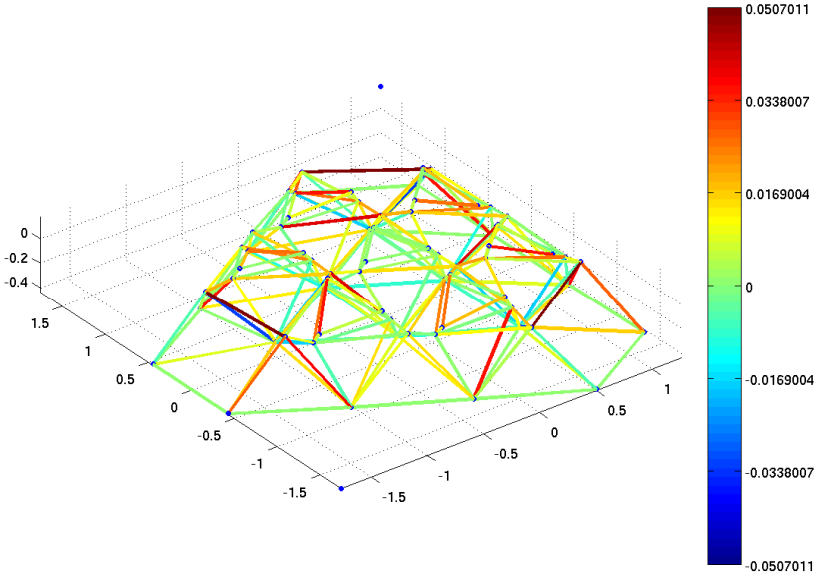

Figure 58. Natural Frequency Element Strain – Perspective View

The proportional damping weights α_e and β_e were set to 0.998 and 0.002 respectively for all three of the system models. These values were selected to achieve an under damped system such that the damping ratio was around 0.4. The damping ratio at the natural frequency of the SMCT global model was 0.378. The resonance frequencies, in Hertz, of the damped global system are shown in Figure 59, where the x-axis is the real axis, and the y-axis is the imaginary axis.

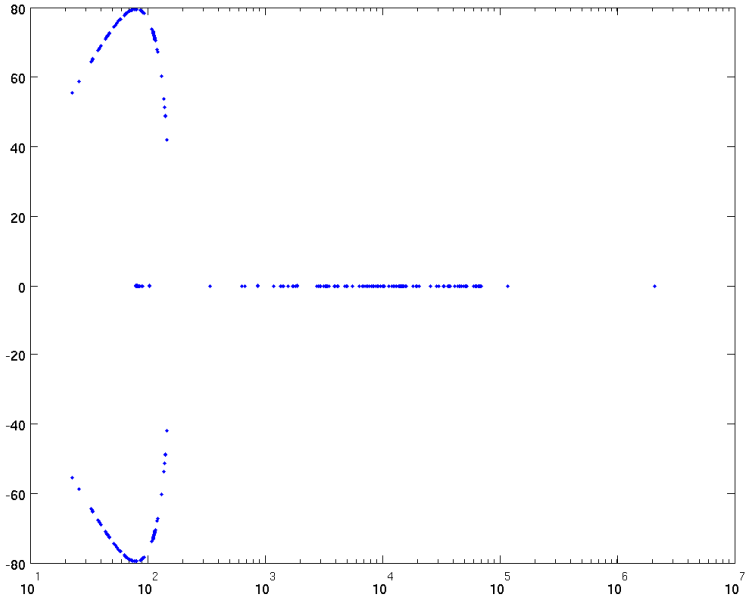

Figure 59. Resonance Frequencies

The static nodal displacement and element strain of the SMCT global model under gravity are shown in Figure 60 to Figure 63 and Figure 64 to Figure 67, respectively. In Figure 60 to Figure 63 the x, y, and z axes are in meters and colour coding represents the nodal displacement in meters, while in Figure 64 to Figure 67 the colour coding represents element strain in meters. This assumes that gravity acts in the negative z-direction of the global coordinate system.

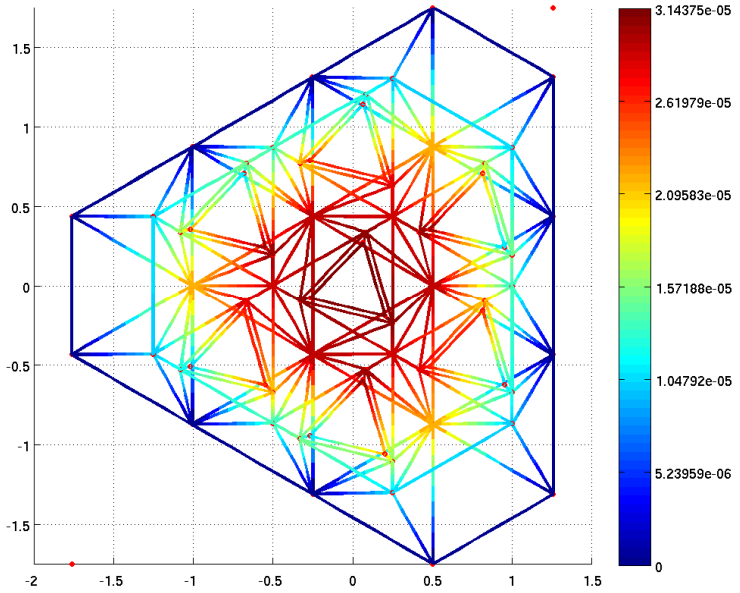

Figure 60. Nodal Displacement under Gravity

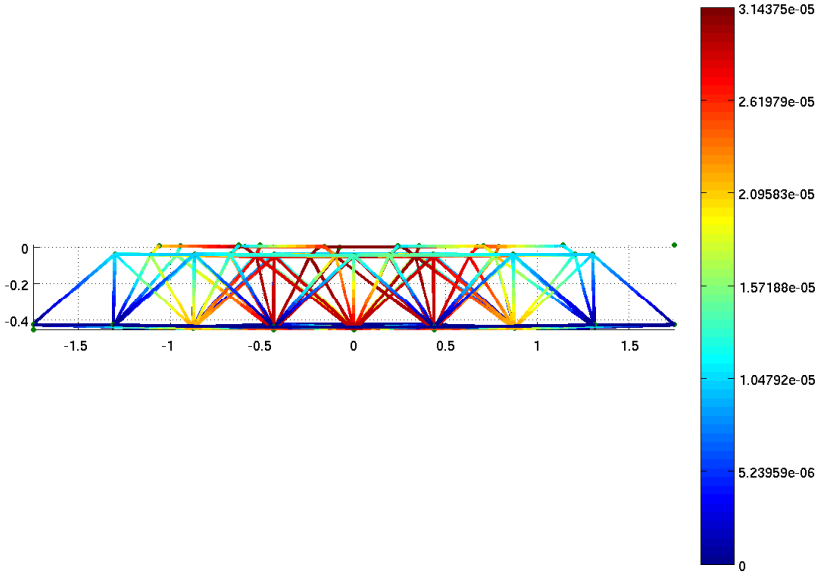

Figure 61. Nodal Displacement under Gravity

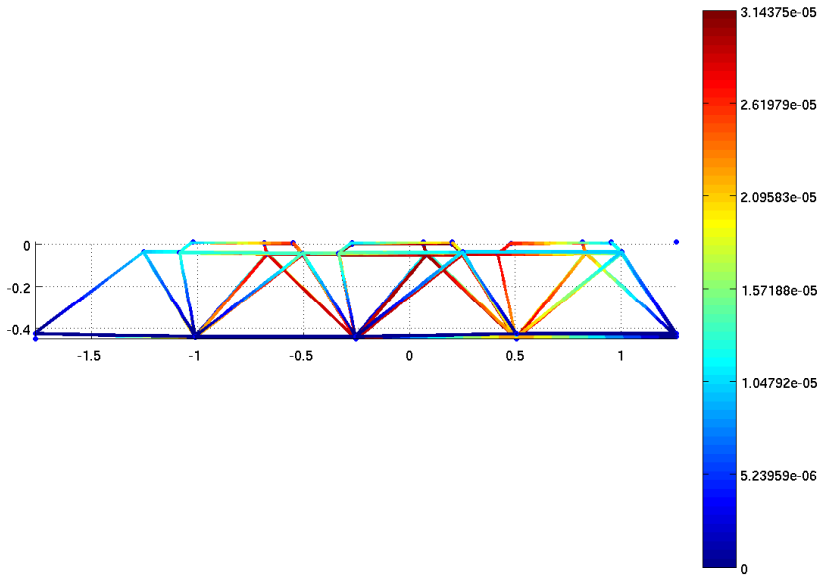

Figure 62. Nodal Displacement under Gravity

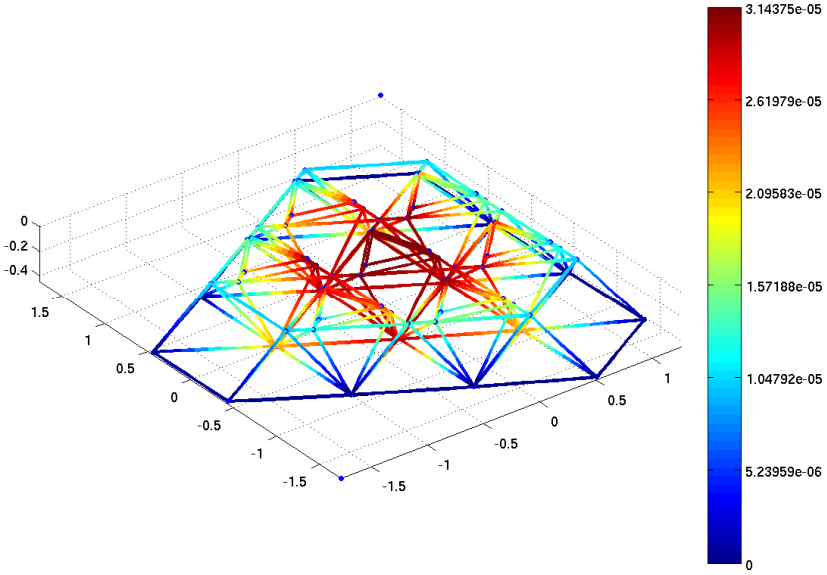

Figure 63. Nodal Displacement under Gravity

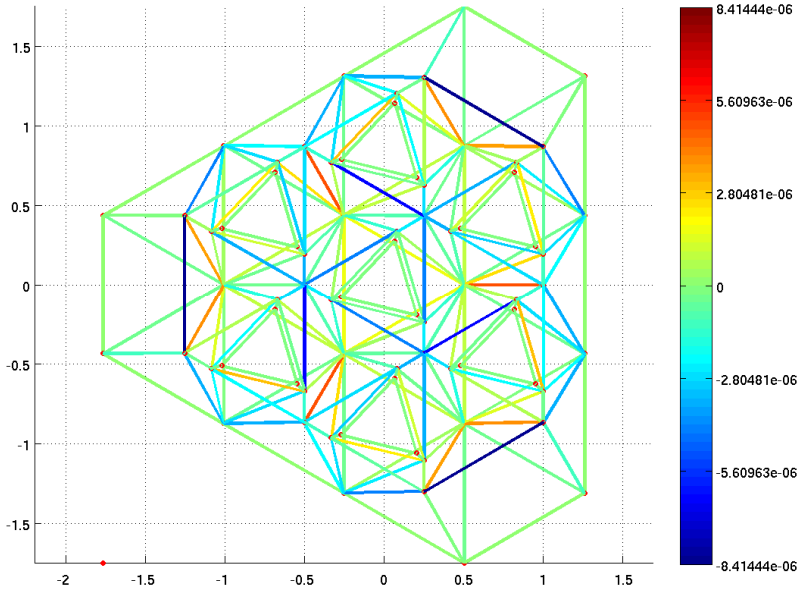

Figure 64. Element Strain under Gravity – X-Y Plane View

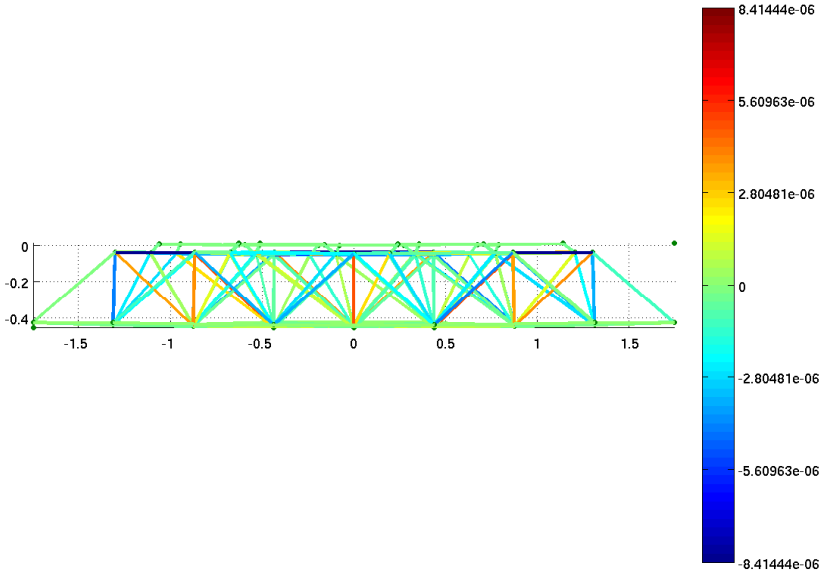

Figure 65. Element Strain under Gravity – Y-Z Plane View

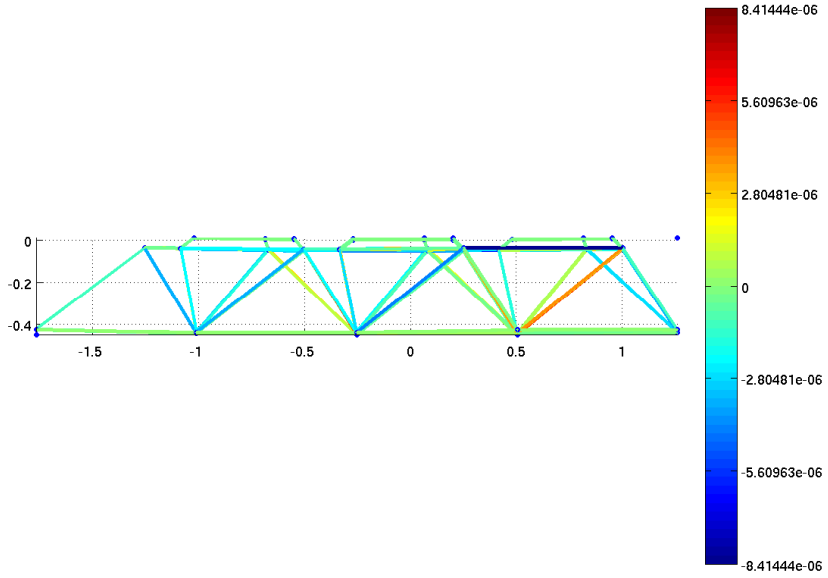

Figure 66. Element Strain under Gravity – X-Z Plane View

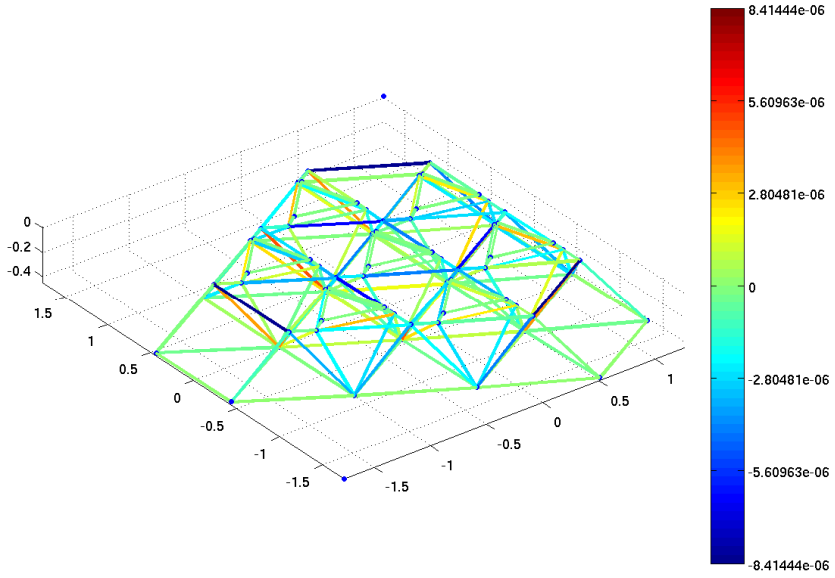

Figure 67. Element Strain under Gravity – Perspective View

Under it's own weight the SMCT model maximum nodal displacement is less than 32 μm and maximum element strain is less than 8.5 μm.

5.6 Closed Loop Simulation Results

For each disturbance load case the SCMT simulation was run with six different controller configurations. These controller configurations are: open loop, global control, decentralized control, distributed control, disconnected distributed control and assembled distributed control. The edge sensor mode used in all the simulations as part of the current work was the interaction matrix and control matrix approximation mode as discussed in Section 5.2.3. Therefore all actuator position error estimations that fed into the controllers are blind to the three least sensitive edge sensor modes. The output of all simulations was resampled, using linear interpolation, at 1 kHz. The spatial RMS, in meters, of the segment triad nodes' displacements along the line of actuation was then calculated at each time step. This time series of RMS values, henceforth referred to as the total triad displacement RMS. This RMS calculation is the primary figure of merit used for comparing the controller configuration on the SMCT. The legend which specifies the controller configuration that was applied given in Figure 68.

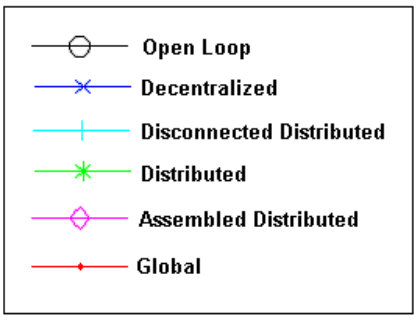

Figure 68. Simulation Results Legend

It should be observed that total triad displacement RMS is based on the full displacement of the nodes, that is to say, it contains displacement modes that are unobservable by the edge sensor system, as discussed in Section 4.6. For this reason, prior to the spatial RMS calculation, the segment triad nodes' displacements are decomposed into the sensed and unsensed displacement modes. The spatial RMS time series of the sensed modes, the displacement modes corresponding to the 18 retained sensor modes discussed in Section 5.2.3, is referred to as the sensed triad displacement RMS. The spatial RMS time series of the unsensed modes, the displacement modes corresponding to the 3 removed sensor modes also discussed in Section 5.2.3, is referred to as the unsensed triad displacement RMS. This allows us to compare the controllers' performance with respect to the modes which are directly sensed by the edge sensor with the modes that are not sensed at all.

The load case investigated is random disturbance case, where, both spatially and temporally, randomized force is applied to each node of the SMCT model. The disturbance force applied was between ±10 N, with a temporal pass band between 6.25 Hz and 12.5 Hz. The resulting triad displacement RMS for the six controller

configurations is shown in Figure 69, where the x-axis is time in seconds, and the y-axis is the spatial RMS of actuator position error in meters.

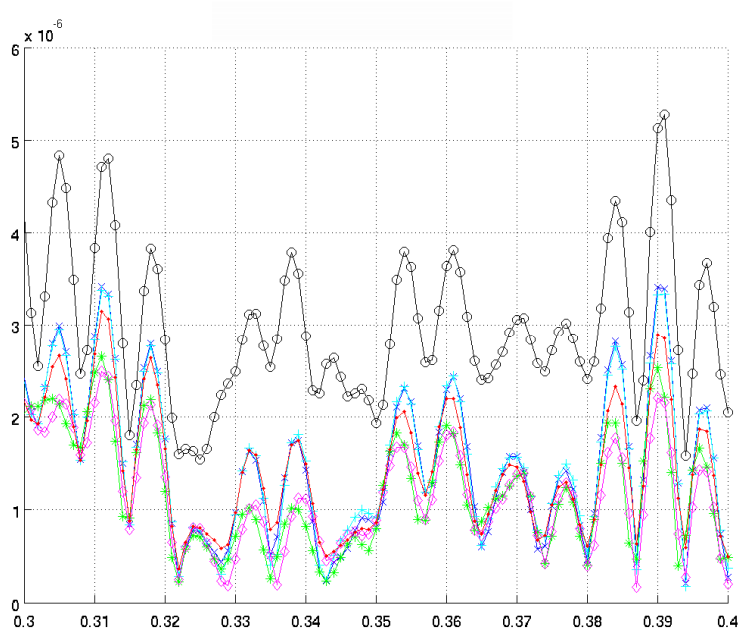

Figure 69. Total Triad Displacement RMS

From these results we can see that all the closed loop controllers help to stabilize the segmented surface, with the distributed and assembled distributed controller performing better overall. However there is still a large residual position error in each of the controller cases. This positioning error is a direct result of the edge sensor system being blinded to its three least sensitive modes. This can be observed by examining the sensed triad displacement RMS from the same set of simulation data, as shown in Figure 70 and Figure 71, where the x-axis is time in seconds, and the y-axis is the spatial RMS of sensed actuator position error in meters.

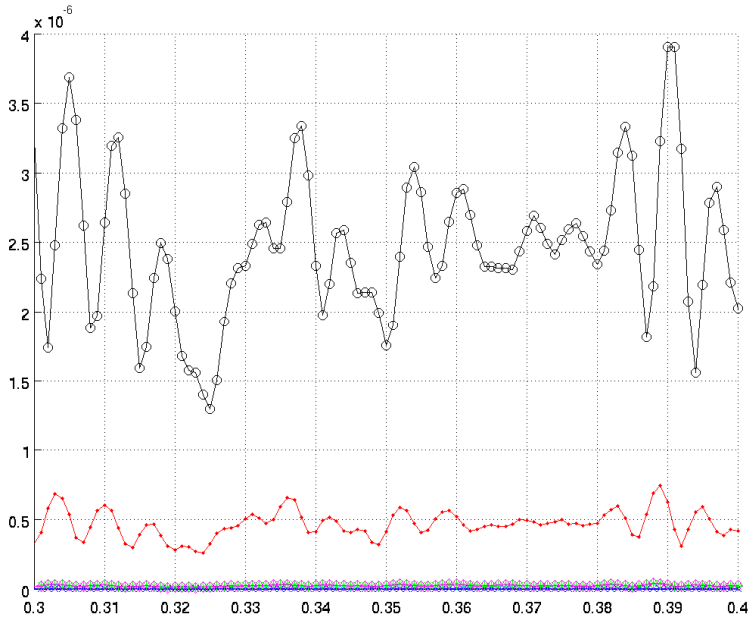

Figure 70. Sensed Triad Displacement RMS

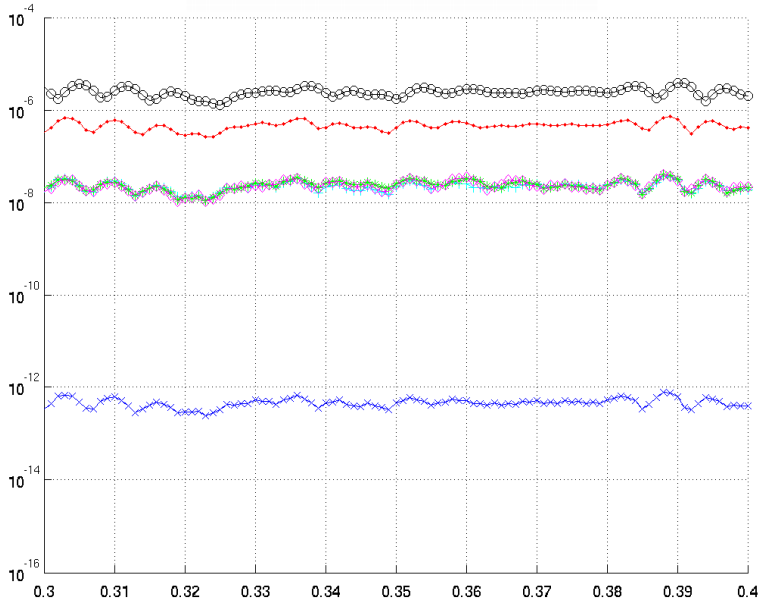

Figure 71. Sensed Triad Displacement RMS – Log Scale

Based only on sensed triad displacement RMS, the decentralized controller performs the best, rejecting the disturbance well below the practical sensor noise limit. Next the three forms of the distributed controller all perform comparably. Since the disconnected distributed controller performed comparably, and arguably slightly better, with the distributed and assembled distributed controllers, this suggests that the interconnection of the distributed controls has little or even a slight negative impact on the rejection of the sensed modes. This now leads to the interesting results of unsensed triad displacement RMS, shown in Figure 72, where the x-axis is time in seconds, and the y-axis is the spatial RMS of unsensd actuator position error in meters.

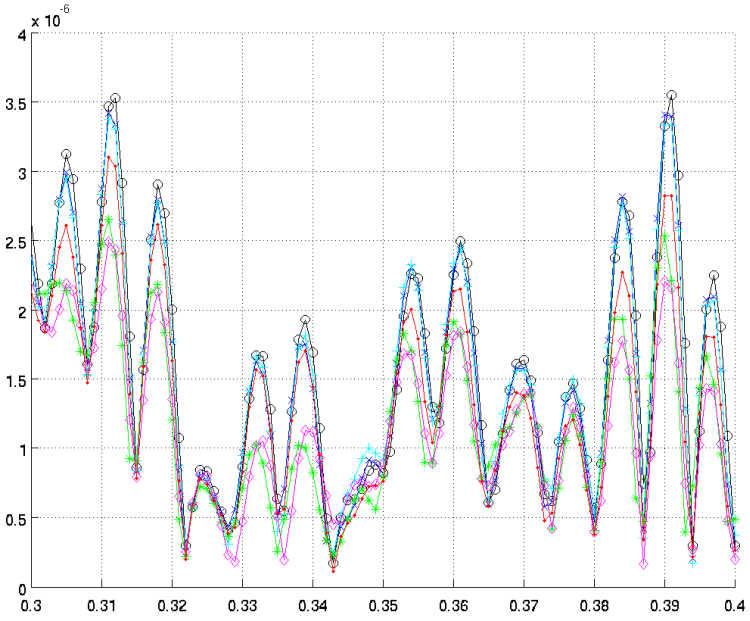

Figure 72. Unsensed Triad Displacement RMS

As expected the decentralized controller has little impact on the unsensed modes. The global controller however is able to somewhat compensate for some of these modes even though they are not being sensed by the edge sensor system. This ability comes directly

from the fact the global controller was synthesized based on the global model and therefore the control signal response is a function of not just localized actuator position errors, but based on all the actuator position errors. By taking all the actuator position errors into consideration and since the controller is designed with knowledge of the system as a whole, the global controller is able to internally compensate for the unsensed modes.

Curiously the distributed and assembled distributed controllers are also able to reject some of the displacement in the unsensed modes. Since the distributed and assembled distributed controllers performanced very similar overall it can be concluded that the small latencies in the interconnection signals have only minor impact on the overall distributed controller's perform in this case. The disconnected distributed controller however behaves similar to the decentralized controller, which is not unexpected since it essentially is a decentralized controller. These results imply that the interconnected signals within distributed and assembled distributed controllers allow them to effectively internally estimate position errors in the unsensed modes, since the disturbed unit controllers were synthesis with knowledge of the segment's support dynamics and how they are coupled throughout the system.

Similar results are also observed from all disturbance case simulations, as shown in Appendix B to I, suggesting strongly that the distributed control system may prove to be a viable and highly scalable solution to the control of a segmented mirror.

In addition to the time domain simulations, the open and closed loop transfer functions of the system were also examined in the frequency domain. The maximum transfer function gain over all disturbance inputs and system outputs combination were calculated at each frequency sample. This was done for four controller cases: open loop with no active control, Global H_∞ control, decentralized H_∞ control, and assembled distributed control, as shown in Figure 73, where the x-axis is frequency in Hertz and the y-axis is the maximum transfer function gain. The control and interaction matrices were used together to approximate the edge sensors in the closed loop system. Therefore the control system is blind to the three least sensitive sensor modes, the same as in the time domain simulations.

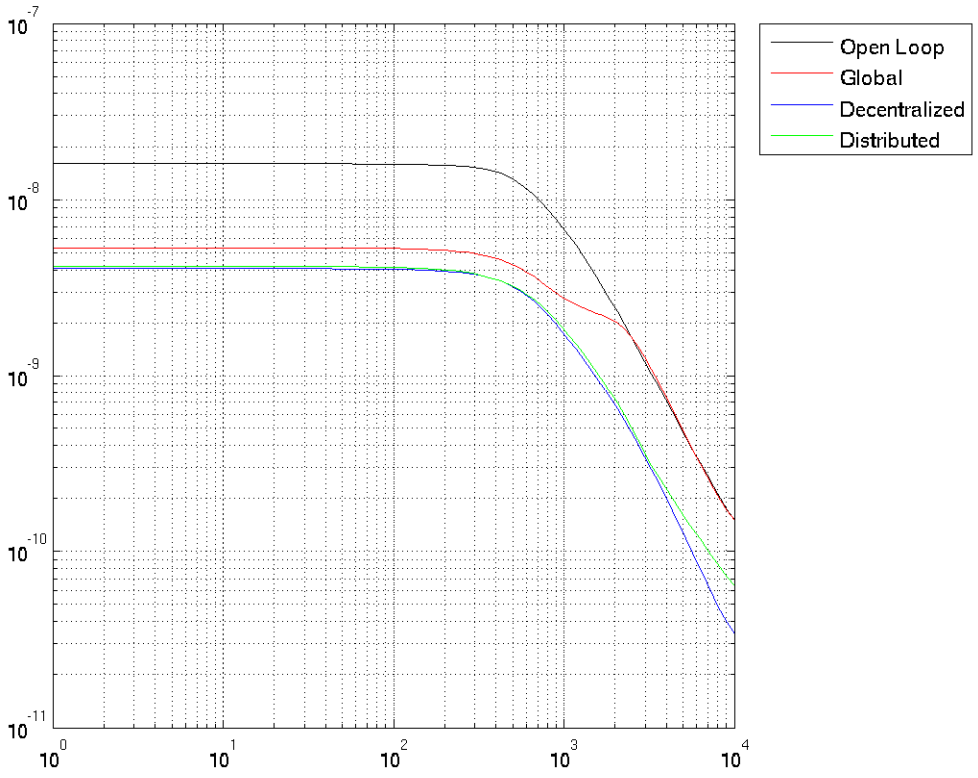

Figure 73. Maximum Transfer Function Gain

When the same transfer functions are used but only the sensed modes are considered on output of the system, the maximum transfer function gains become as shown in Figure 74, where the x-axis is frequency in Hertz and the y-axis is the maximum transfer function gain.

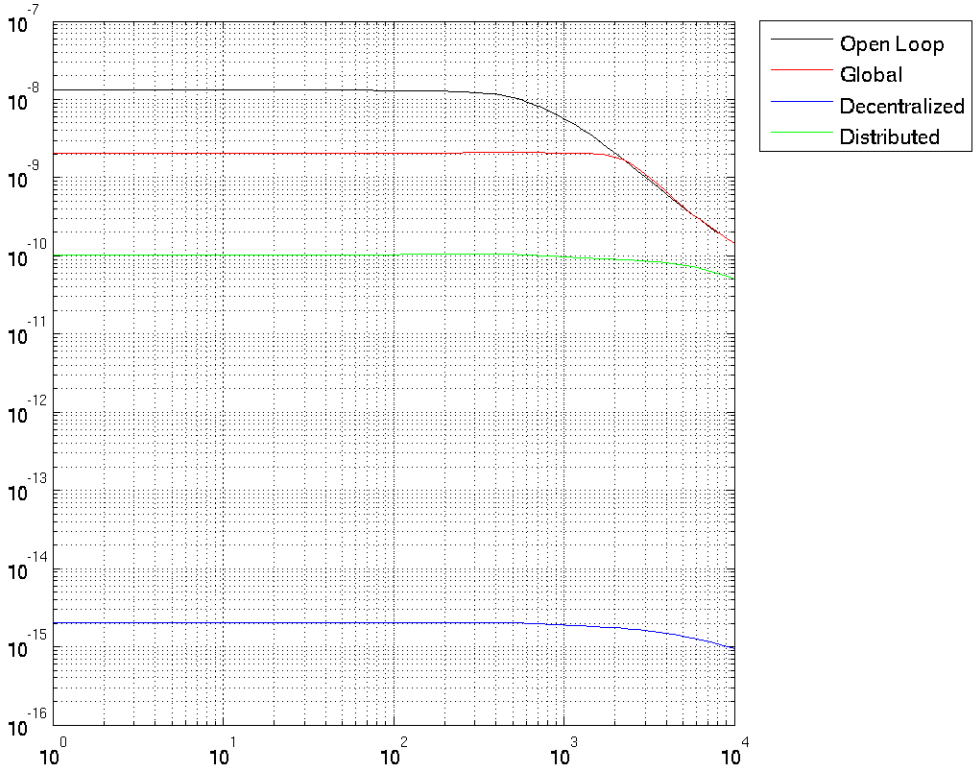

Figure 74. Maximum Transfer Function Gain, Sensed Modes

The transfer function gains presented in both Figure 73 and Figure 74 corroborate the results seen the time domain simulation results presented above.

It should be noted that since the controllers are suboptimal and are not augmented according to loop-shaping, that the individual controller performance cannot be directly compared. The level of disturbance rejection cannot be compared between controllers; however trends in controller performance can be analyzed.

CHAPTER 6

CONCLUSIONS

6.1 Contributions

As a part of this work a parametric geometric layout of the SMCT was presented. The SMCT design was used to generate a finite element model, based on custom finite element code that was developed explicitly for this project. In order to generate a distributed model using a finite element approach, a new technique was developed which exploits information that would typically be disregarded as a part of the bounding process of the FEM. This technique is not limited to a model generated from the SMCT finite element code developed for this work, nor is it limited to a segmented mirror application. In fact it can be expanded to general FEM, given the mass, stiffness and damping matrices.

In addition to the modeling technique developed an approach to reduce the number of required spatial state in a distributed system was also presented. The reduction method does rely on the fact that the distributed units are only coupled through their temporal states. However based on that, the spatial states in each spatial direction can be reduced to the order of the temporal states without the loss of fidelity.

A closed loop simulation of the SMCT was developed, including global, decentralized and distributed controller modes. This simulation allowed for the initial validation of the distributed control scheme applied to the segmented mirror disturbance rejection problem. It also revealed the ability of the distributed control system to compensate for displacement modes that are not directly sensed by the edge sensor scheme. In the decentralized controller case, since the controllers act independently they provide little in way of counteracting the unsensed modes.

6.2 Recommendations for Future Works

The SMCT project is being expanded to the two ring, nineteen segment testbed under the name of Control System and Integrated Modeling (CSIM) Testbed. As a result the overall segment support structure is being redesigned. This redesigned phase will include the investigation of alternative support structures, segment mounting schemes and cost benefit analyses of the feasible options.

When the CSIM design has achieved an appropriate level of maturity, a finite element model should be developed in a commercial structural analysis software package. It is the author's recommendation that the package be ANSYS to take advantage of the Integrated Modeling Toolset HIA has developed. The primary advantage of using the Integrated Modeling Toolset is that the system mass, stiffness and damping matrices can be exported directly from ANSYS to MATLAB. These matrices can then be used to generate the plant, decentralized and distributed models presented in Sections 3.3, 3.4, and 3.5, respectively. This effectively replaces the FEM that was presented in Section 3.2.

Additionally further investigation on temporal state reducution method for the distributed model is required. Presented as a part of this work, the temporal state reduced method for the distributed model ignores the distributed spatial states. This results in a potential loss of fidelity with respect to how the temporal states are coupled to the spatial states. Development of a reduction method that balances temporal and spatial states would lead to improved accuracy of the distributed model with respect to its global performance.

Observations from the simulation results show the benefit of the distributed control scheme on controlling the indirectly sensed global modes. Further analysis is required to determine the conditions under which these indirectly sensed modes are controllable, as well as the impact of model tolerances and model uncertainty on the controllability of these modes.

Currently the actuator displacement error estimation via the edge sensors and control matrix are centralized. This problem can be parallelized such that only localized sensor readings are required for local error estimations. This however results in a loss of any globally sensed information, even those that are not poorly sensed. Applying the distributed modeling techniques presented above may yield a scalable, yet global method for solving the control matrix problem. This concept can be expanded such that the edge sensor geometry and the control matrix are distributed and convolved with the distributed controller synthesis directly, to improve the overall performance of the distributed control system.

The controllers presented above are synthesized based an un-augmented system model. That is to say they are unbounded in terms of bandwidth and are sub-optimal over a practical frequency range. Through appropriate loop-shaping techniques the controllers can be optimized with regard to the decision variable γ, controller bandwidth and the required closed loop performance over a given frequency band. Once the continuous time controllers have been augmented and optimized, discrete time versions should be generated. The practical implementation details of producing a temporally discrete distributed control system also need to be investigated further.

As discussed in Section 5.3.1, in the simulation there is a necessity for a finite delay on the interconnection signals between interconnected units. Additionally in a physical implementation of the system of distributed units, most likely the interconnection signals would be discretely sampled. Therefore this finite delay or temporal sampling should be

incorporated into the synthesis of the distributed controller, by way of the addition of a transfer function applied to either each interconnected signal input or output.

Currently the interconnection signals on the exterior of the system are simply zeroed. Approaches have been developed to handle these types of distributed system over finite spatial extents, however they have not yet been applied to the SMCT simulation. Additionally the SMCT simulation should be expanded to support a larger number of segments, possible in the range of a TMT like primary mirror. This would help overcome any overly dominant boundary effect that may be present on the current SMCT design, as well as validate the extension of the distributed controller over a larger system.

Finally and arguably the most important future work required, with respect to the distributed control of segmented reflector, is the application and validation of the distributed control network on a physical system. This should be a key area of focus for the CSIM project as it is developed and manufactured.

APPENDIX A – Sensor Modes

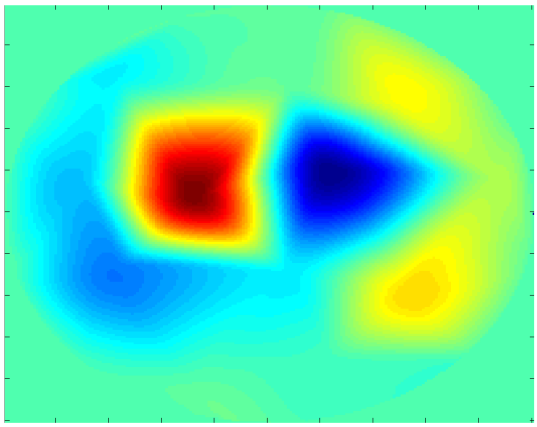

Figure 75. Sensor Mode 1

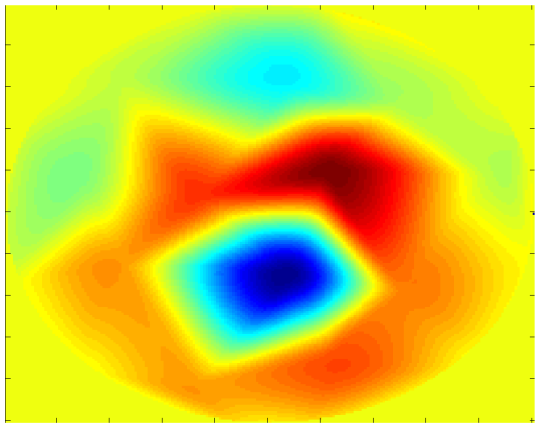

Figure 76. Sensor Mode 2

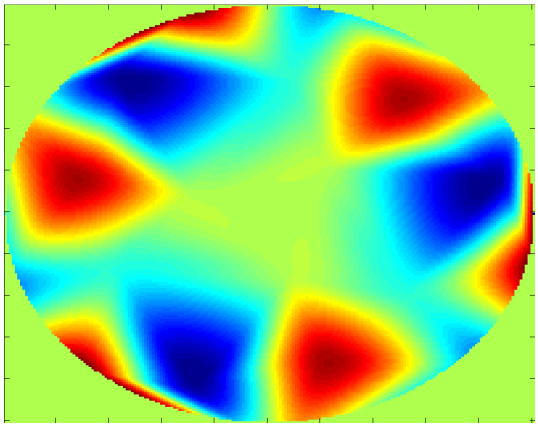

Figure 77. Sensor Mode 3

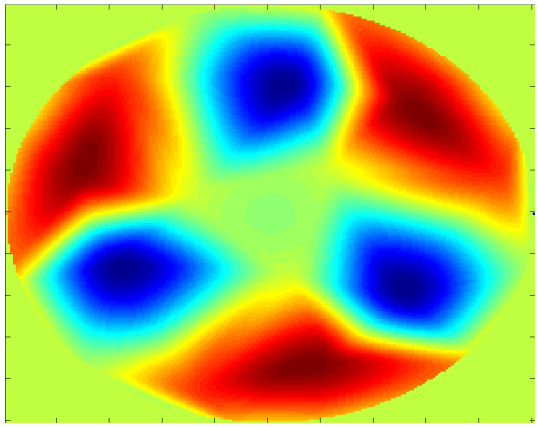

Figure 78. Sensor Mode 4

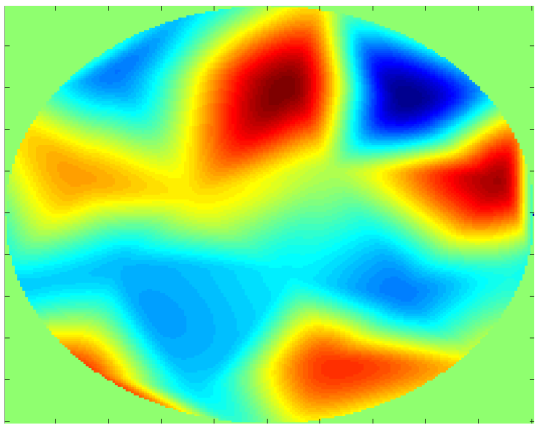

Figure 79. Sensor Mode 5

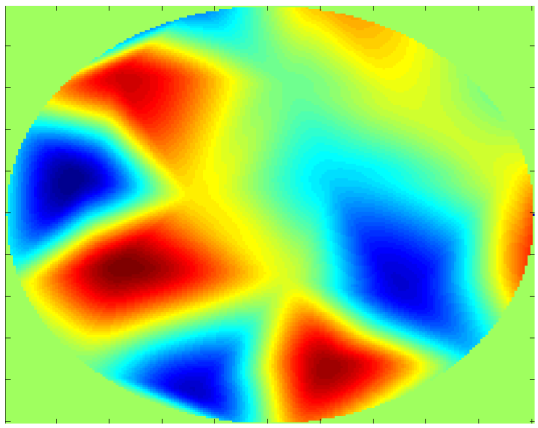

Figure 80. Sensor Mode 6

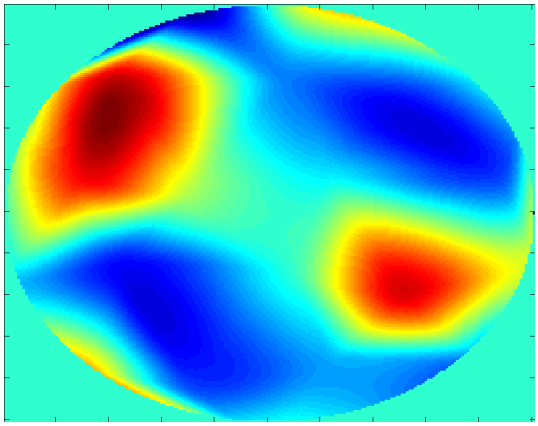

Figure 81. Sensor Mode 7

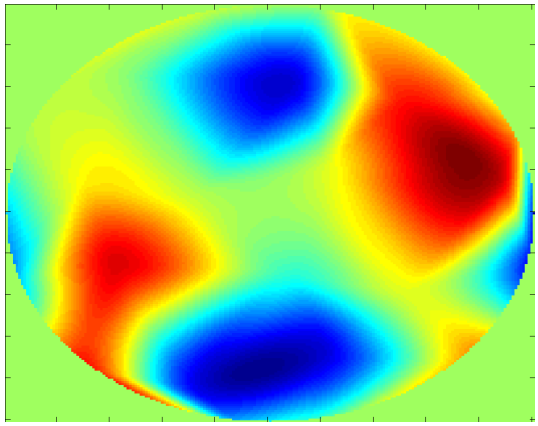

Figure 82. Sensor Mode 8

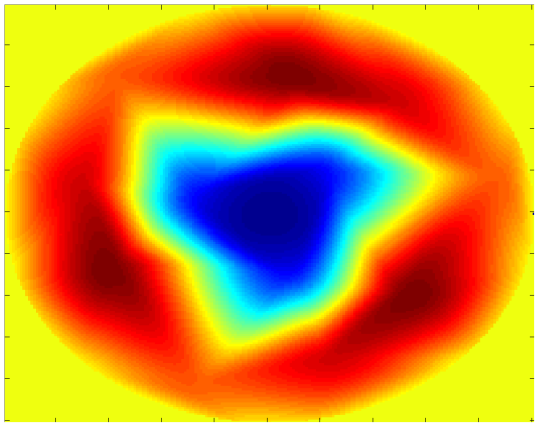

Figure 83. Sensor Mode 9

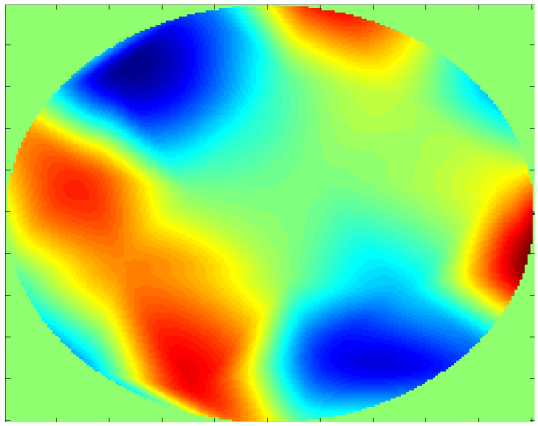

Figure 84. Sensor Mode 10

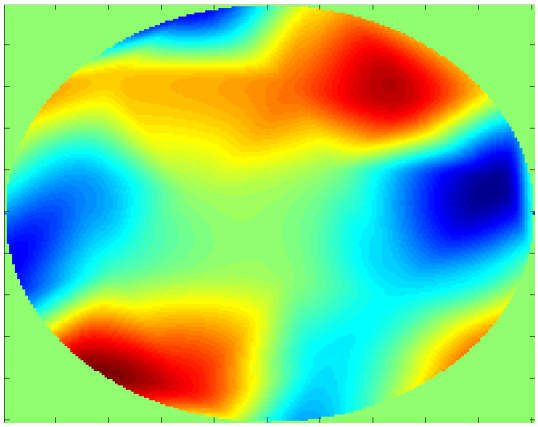

Figure 85. Sensor Mode 11

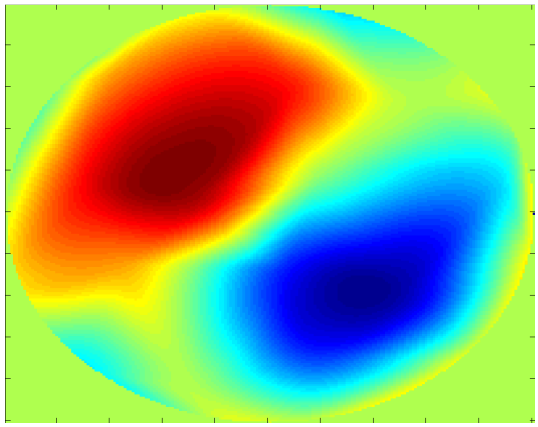

Figure 86. Sensor Mode 12

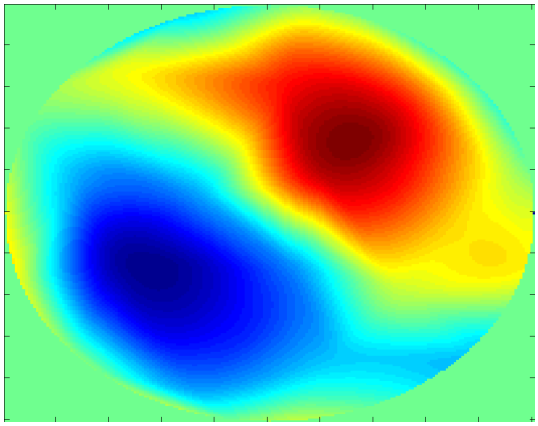

Figure 87. Sensor Mode 13

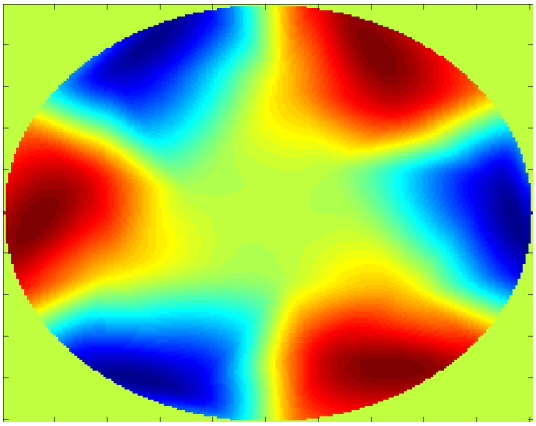

Figure 88. Sensor Mode 14

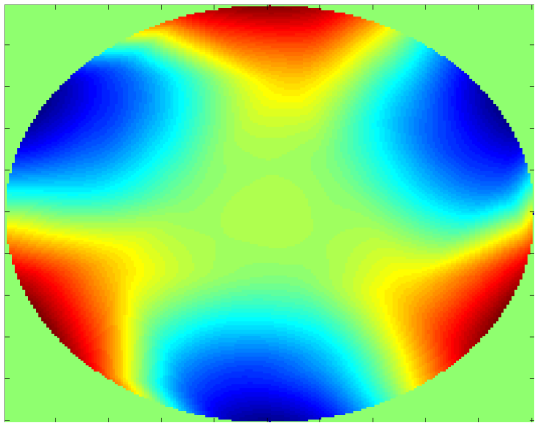

Figure 89. Sensor Mode 15

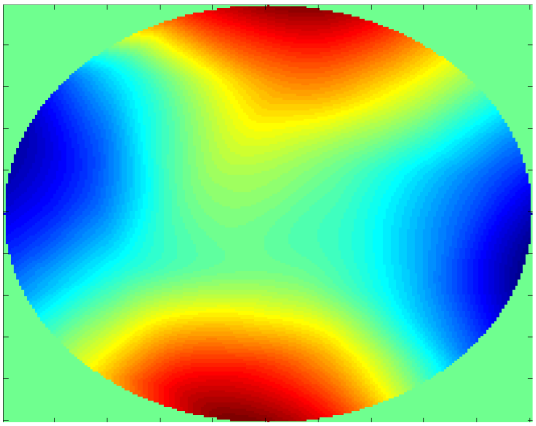

Figure 90. Sensor Mode 16

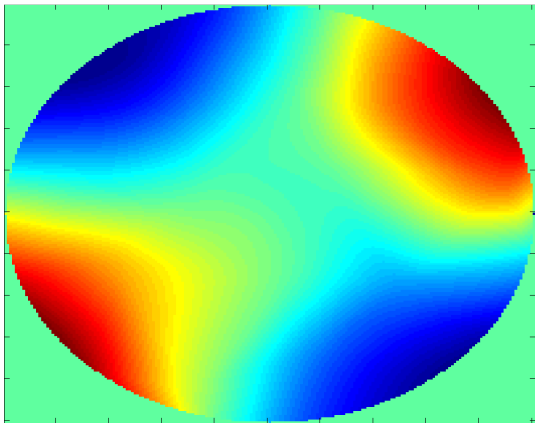

Figure 91. Sensor Mode 17

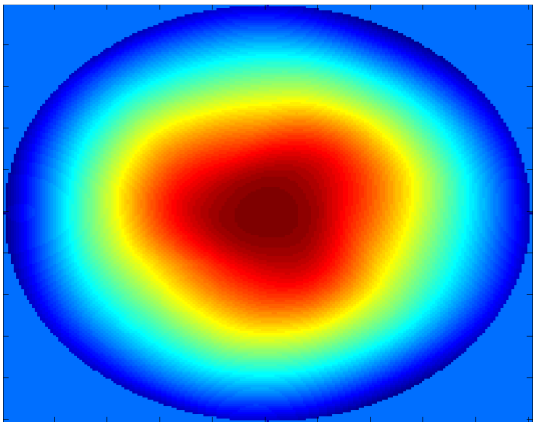

Figure 92. Sensor Mode 18

APPENDIX B – Simulation Results for Random Forces, 6.25 - 12.5 Hz

Random forces were generated for all nodes in all directions. The force on each node was temporally filtered by a second order bandpass butterworth filter with cutoff frequencies at 6.25 and 12.5 Hz. The following figures show the spatial RMS of the actuator position errors: overall, sensed modes, and un-sensed modes, respectively. Where the x-axes are time in seconds and the y-axes are spatial RMS of actuator position error in meters.

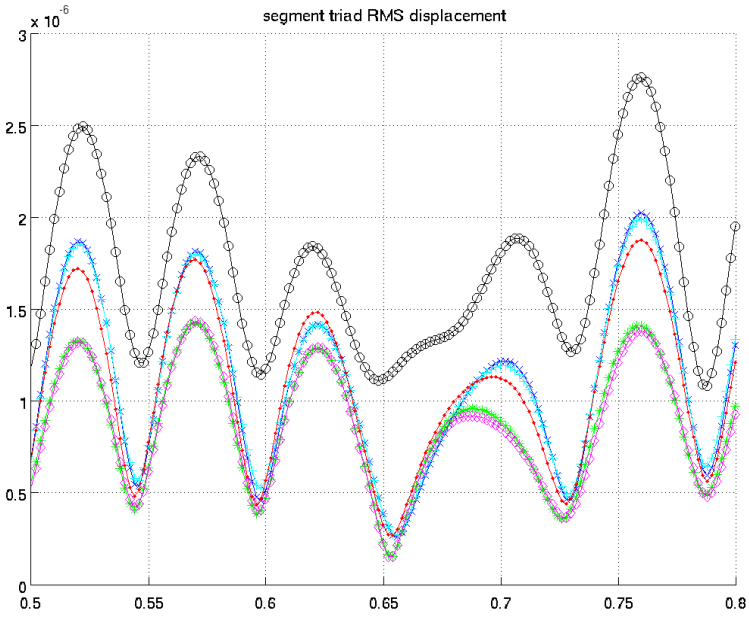

Figure 93. Random Force 6.25 to 12.5 Hz RMS Actuator Position Error

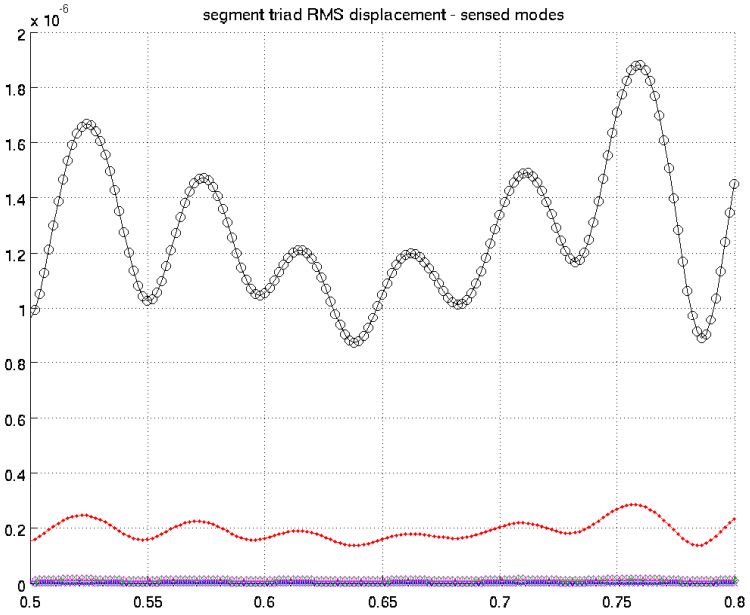

Figure 94. Random Force 6.25 to 12.5 Hz RMS Actuator Position Error Sensed Modes

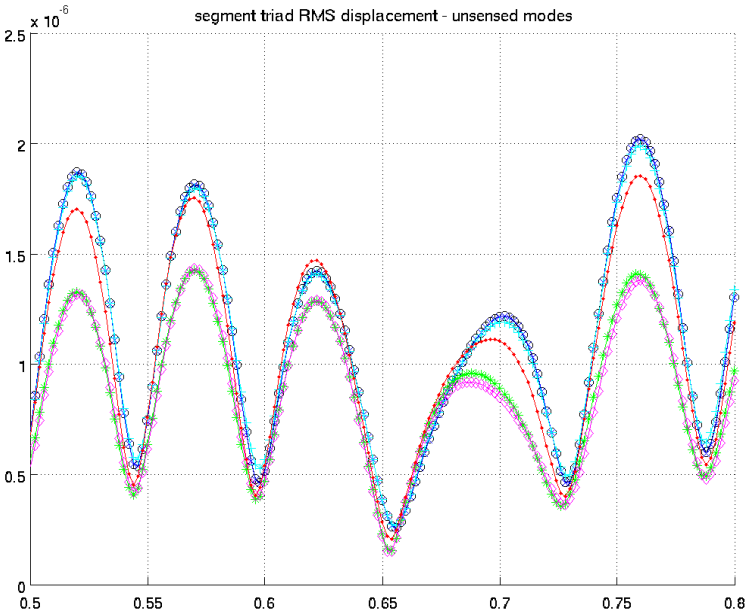

Figure 95. Random Force 6.25 to 12.5 Hz RMS Actuator Position Error Unsensed Modes

The following set of figures shows the actuator displacement errors for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator position error in meters.

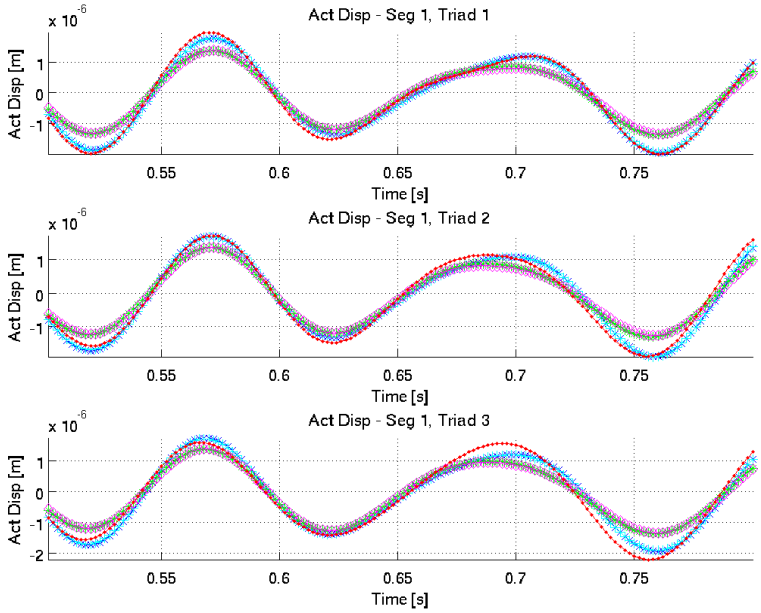

Figure 96. Random Force 6.25 to 12.5 Hz Actuator Position Error, Segment 1

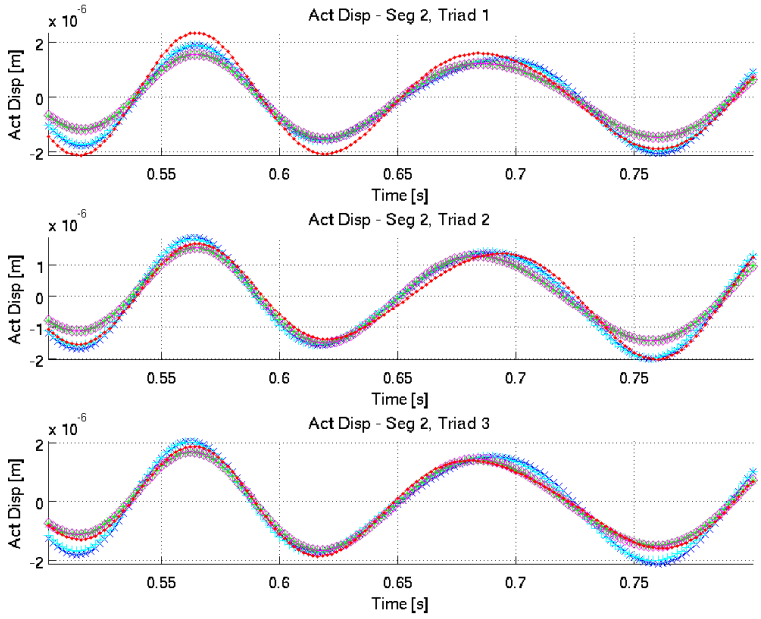

Figure 97. Random Force 6.25 to 12.5 Hz Actuator Position Error, Segment 2

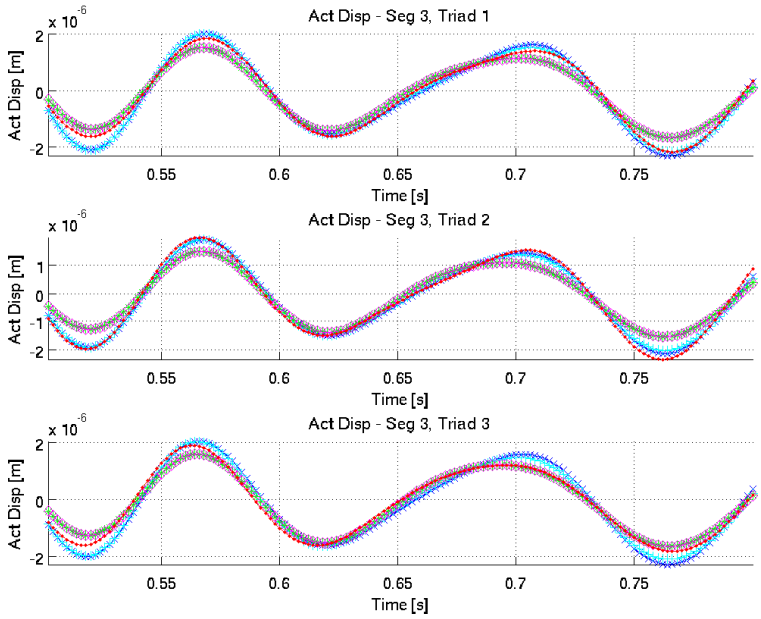

Figure 98. Random Force 6.25 to 12.5 Hz Actuator Position Error, Segment 3

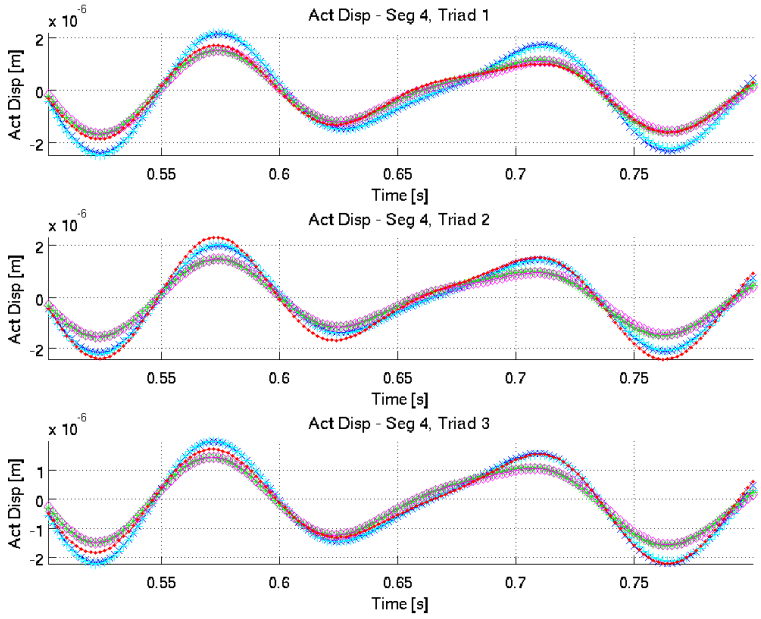

Figure 99. Random Force 6.25 to 12.5 Hz Actuator Position Error, Segment 4

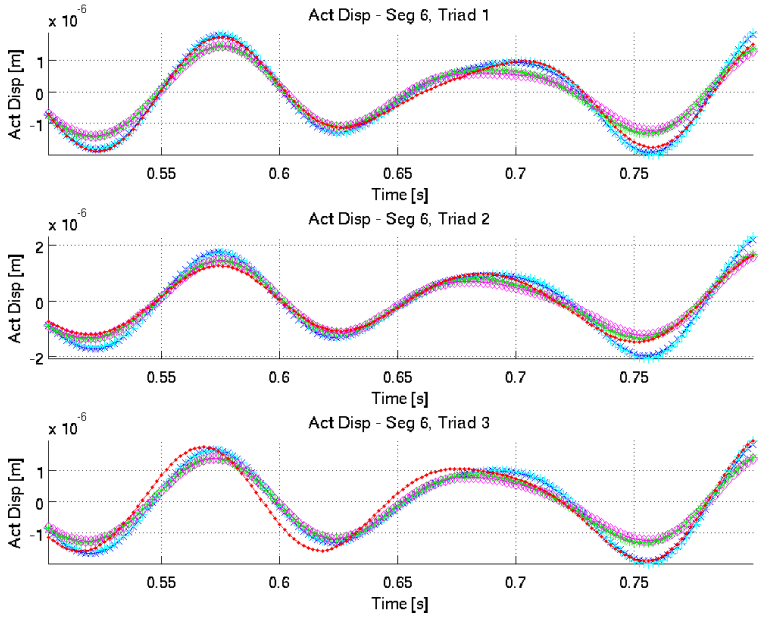

Figure 100. Random Force 6.25 to 12.5 Hz Actuator Position Error, Segment 5

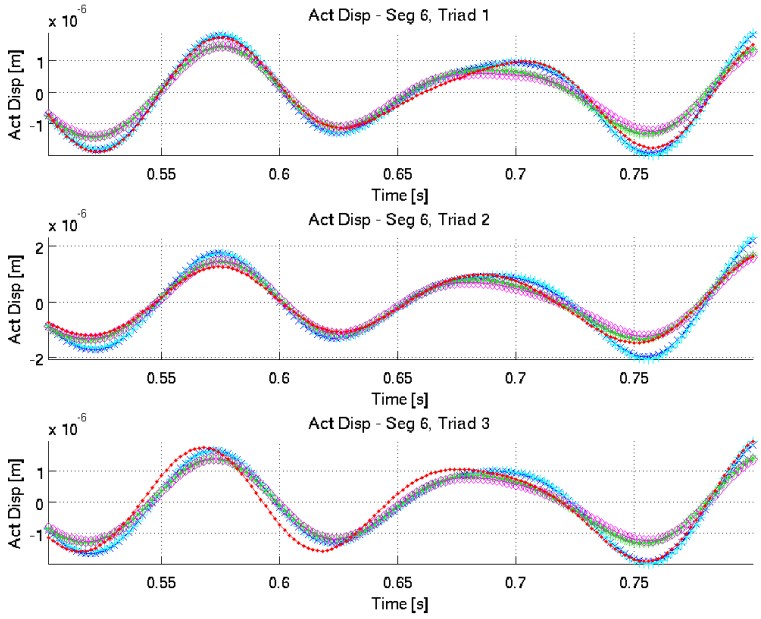

Figure 101. Random Force 6.25 to 12.5 Hz Actuator Position Error, Segment 6

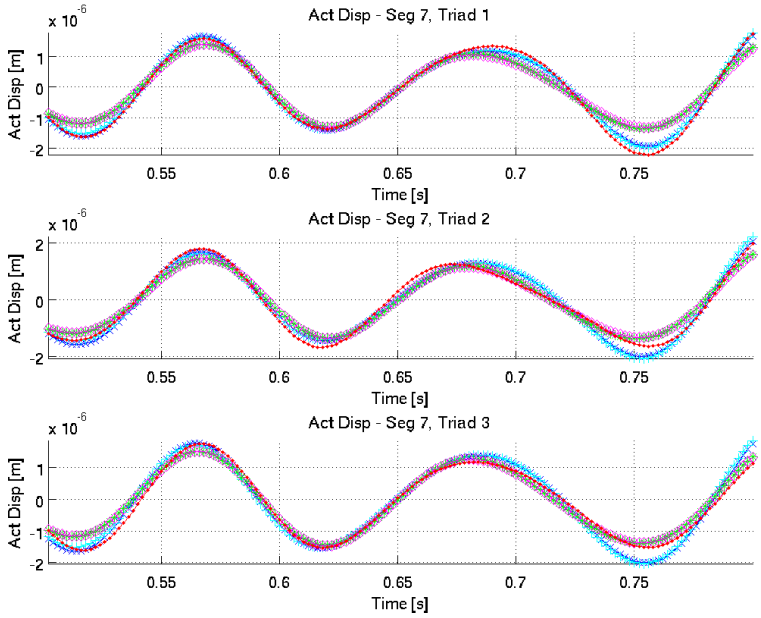

Figure 102. Random Force 6.25 to 12.5 Hz Actuator Position Error, Segment 7

The following set of figures shows the actuator control effort for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator control effort signals.

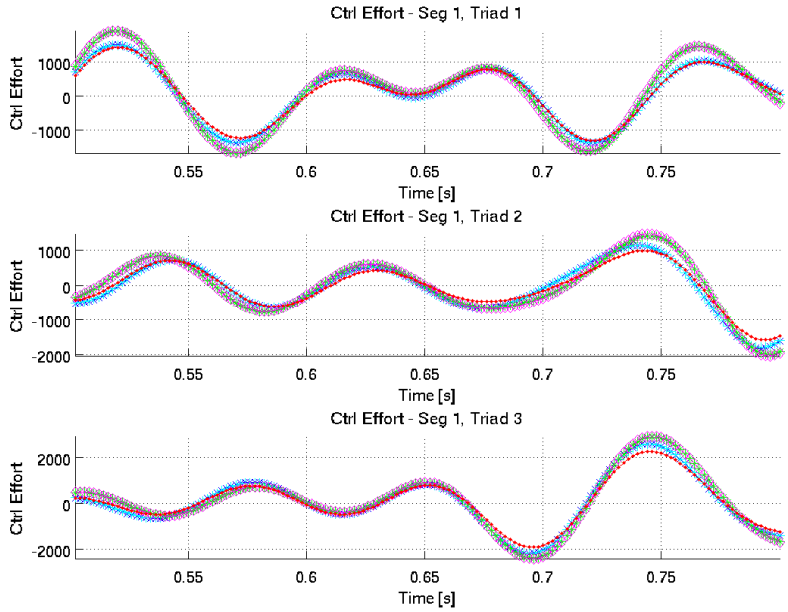

Figure 103. Random Force 6.25 to 12.5 Hz Control Effort, Segment 1

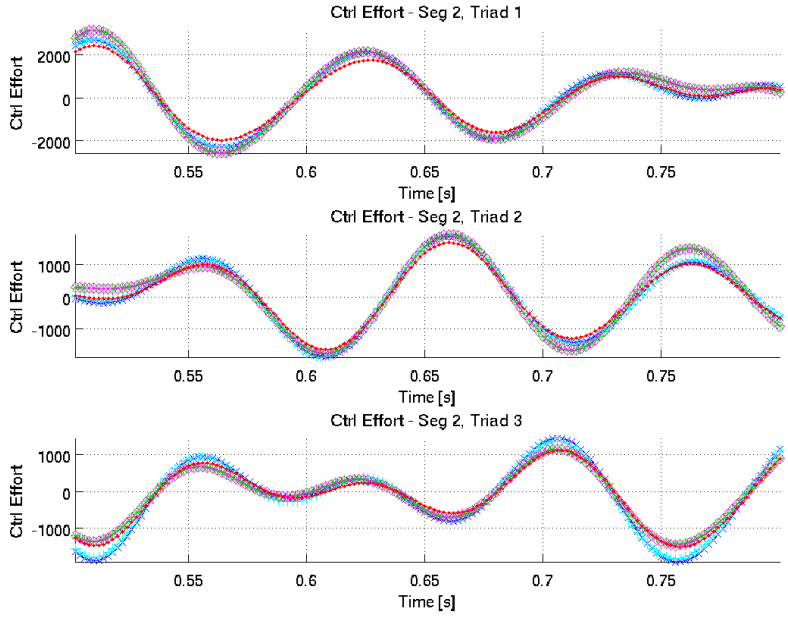

Figure 104. Random Force 6.25 to 12.5 Hz Control Effort, Segment 2

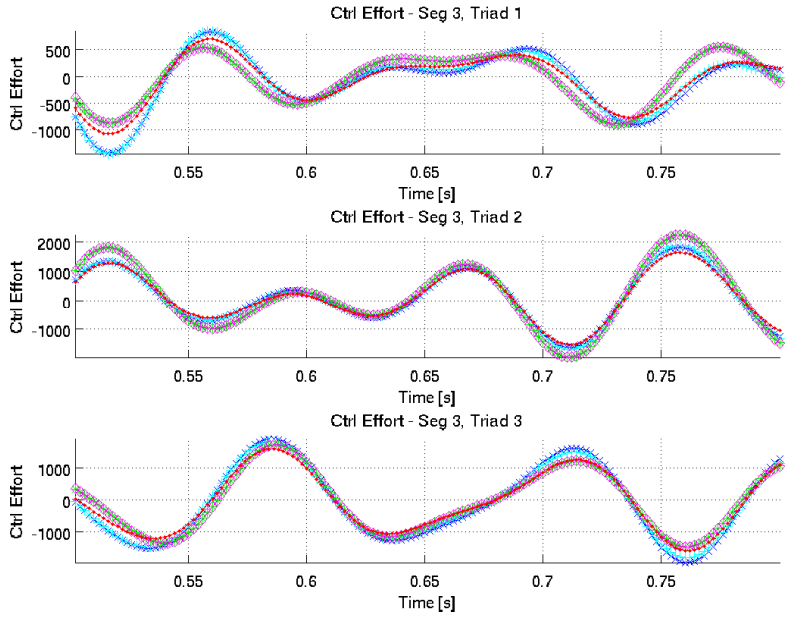

Figure 105. Random Force 6.25 to 12.5 Hz Control Effort, Segment 3

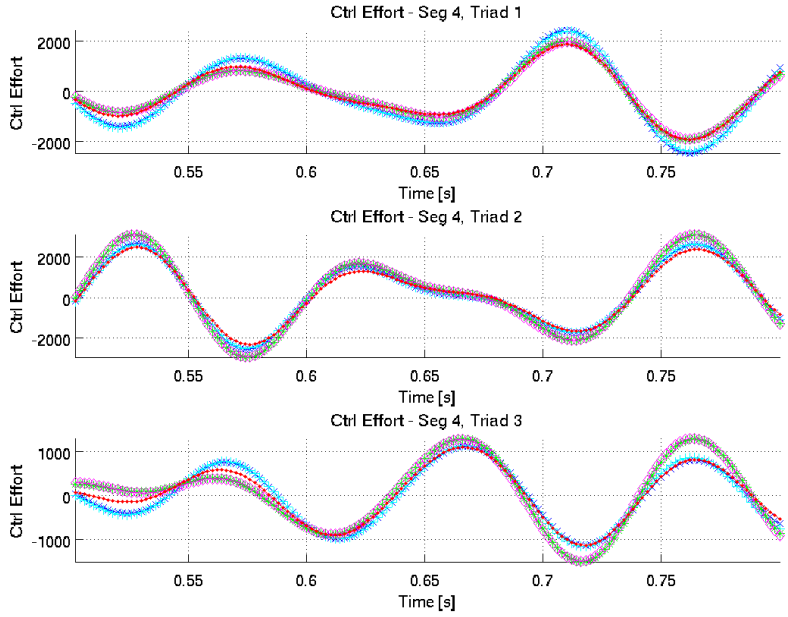

Figure 106. Random Force 6.25 to 12.5 Hz Control Effort, Segment 4

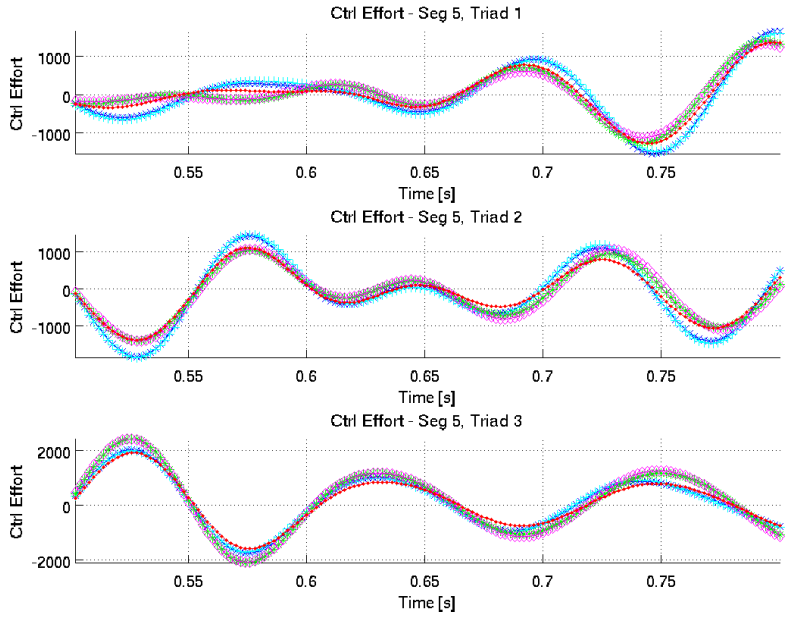

Figure 107. Random Force 6.25 to 12.5 Hz Control Effort, Segment 5

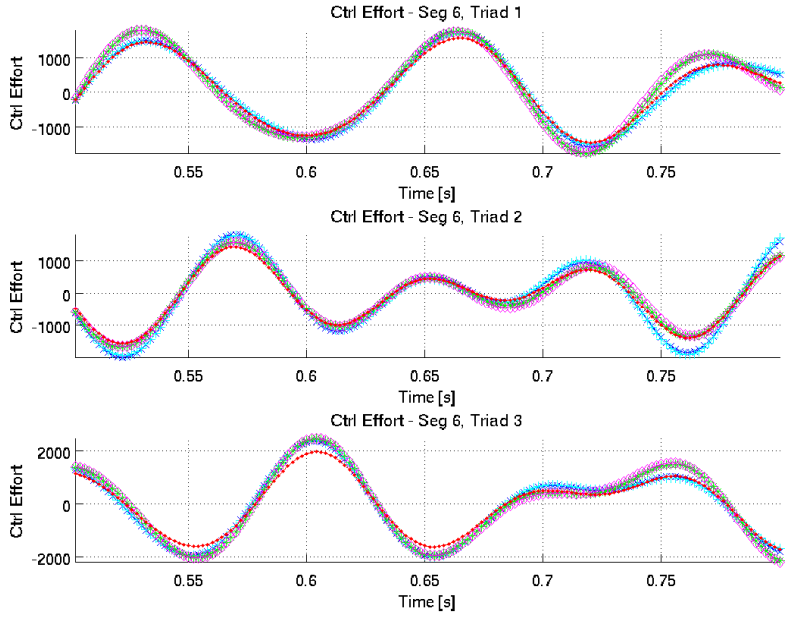

Figure 108. Random Force 6.25 to 12.5 Hz Control Effort, Segment 6

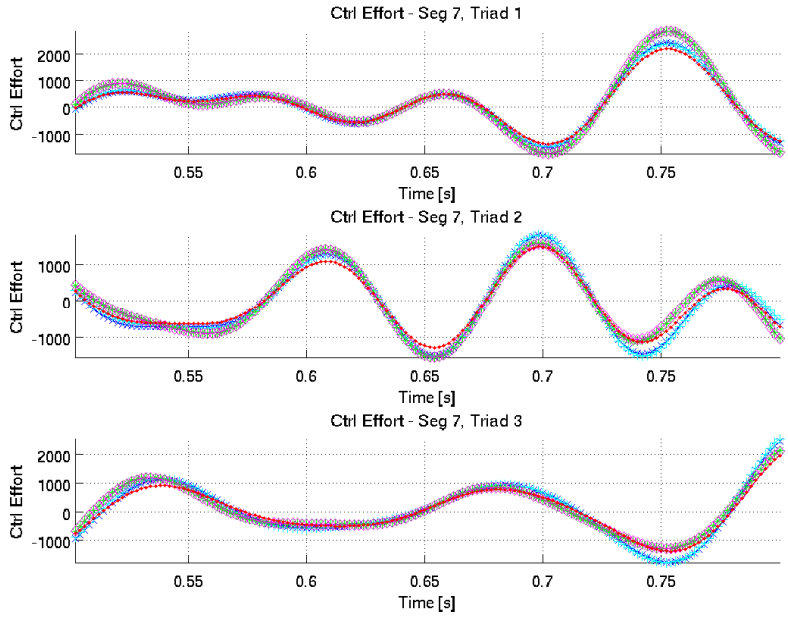

Figure 109. Random Force 6.25 to 12.5 Hz Control Effort, Segment 7

APPENDIX C – Simulation Results for Random Forces, 50 - 100 Hz

Random forces were generated for all nodes in all directions. The force on each node was temporally filtered by a second order bandpass butterworth filter with cutoff frequencies at 50 and 100 Hz. The following figures show the spatial RMS of the actuator position errors: overall, sensed modes, and un-sensed modes, respectively. Where the x-axes are time in seconds and the y-axes are spatial RMS of actuator position error in meters.

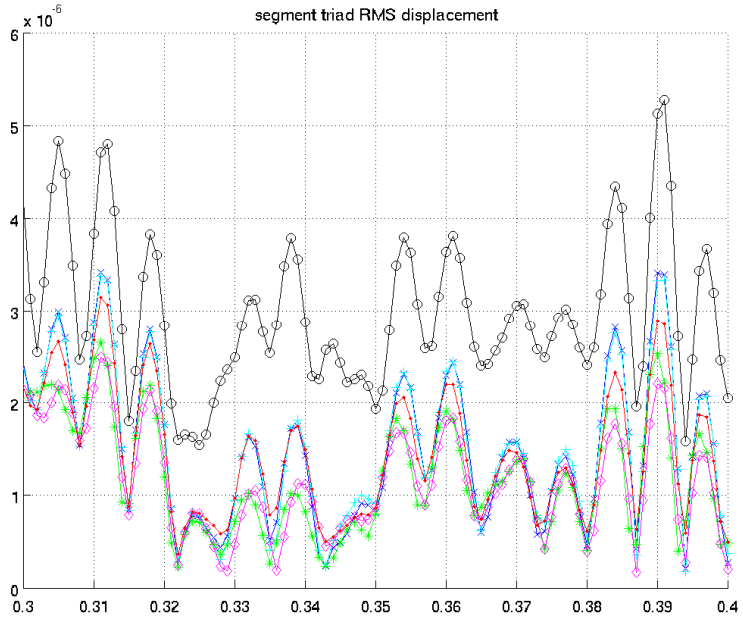

Figure 110. Random Force 50 to 100 Hz RMS Actuator Position Error

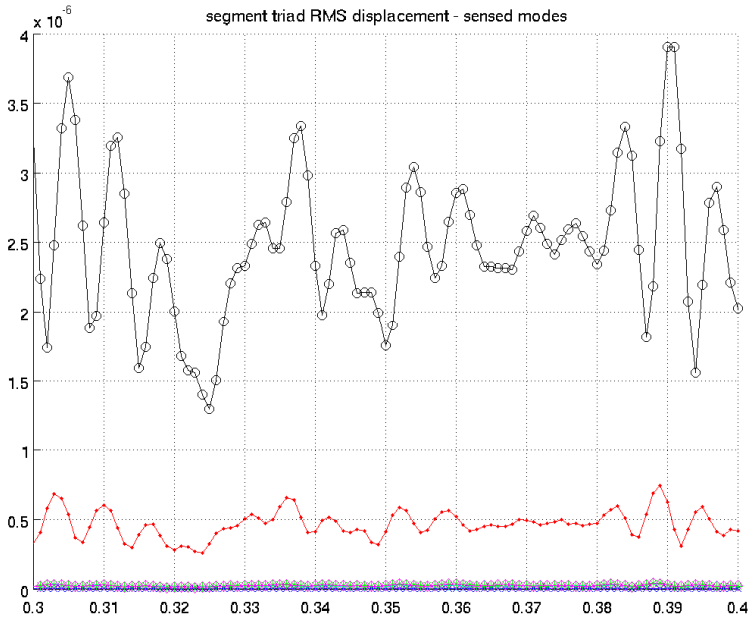

Figure 111. Random Force 50 to 100 Hz RMS Actuator Position Error Sensed Modes

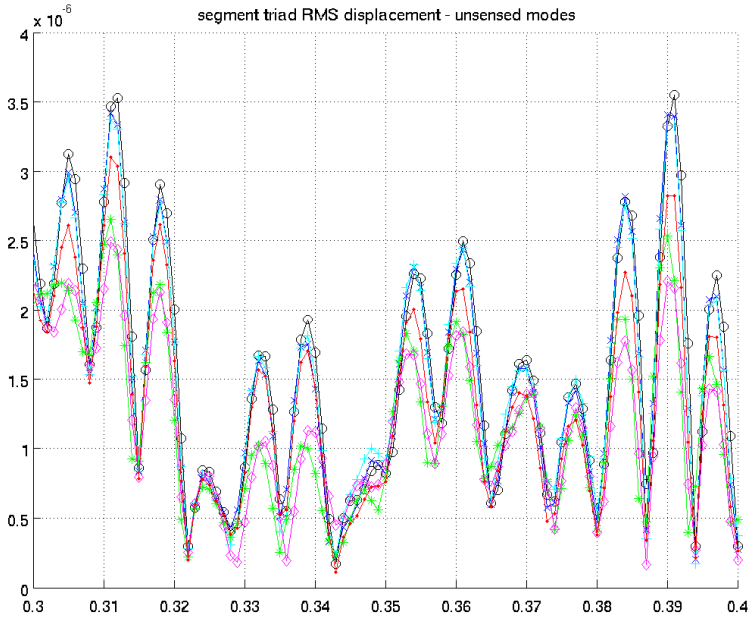

Figure 112. Random Force 50 to 100 Hz RMS Actuator Position Error Unsensed Modes

The following set of figures shows the actuator displacement errors for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator position error in meters.

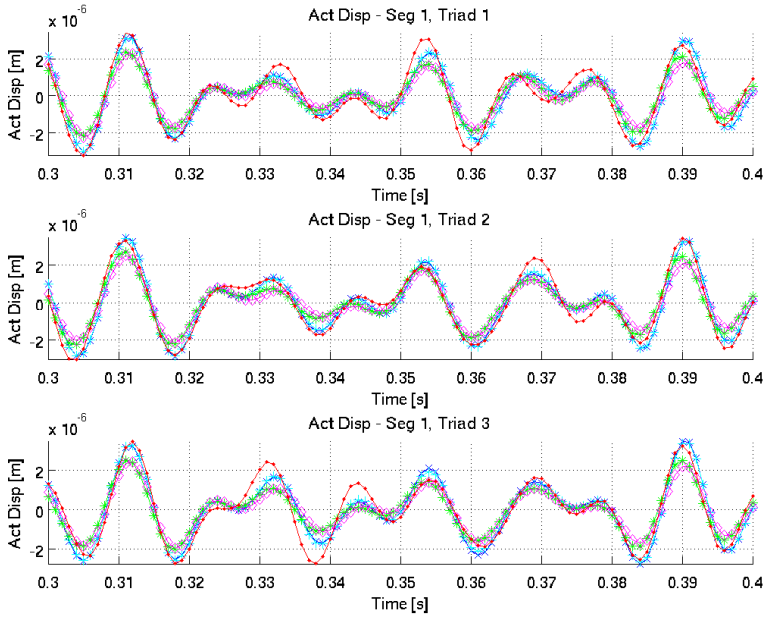

Figure 113. Random Force 50 to 100 Hz Actuator Position Error, Segment 1

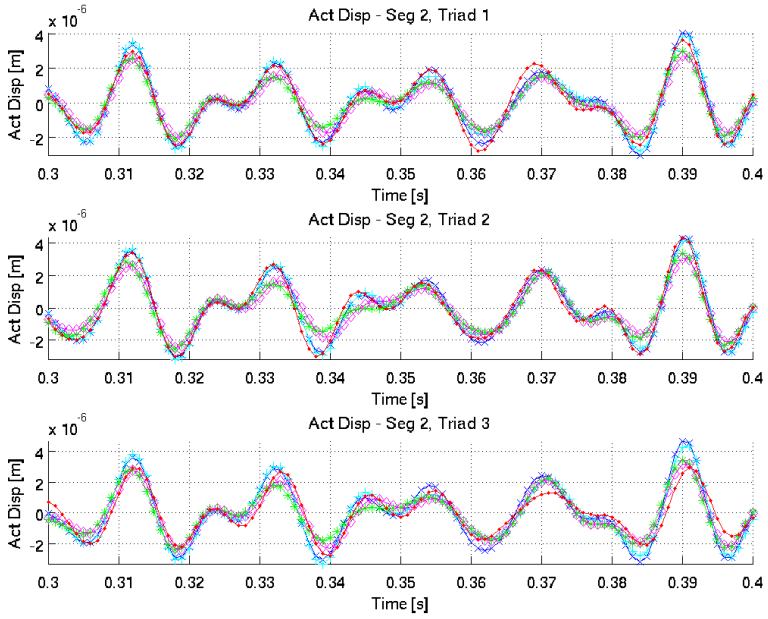

Figure 114. Random Force 50 to 100 Hz Actuator Position Error, Segment 2

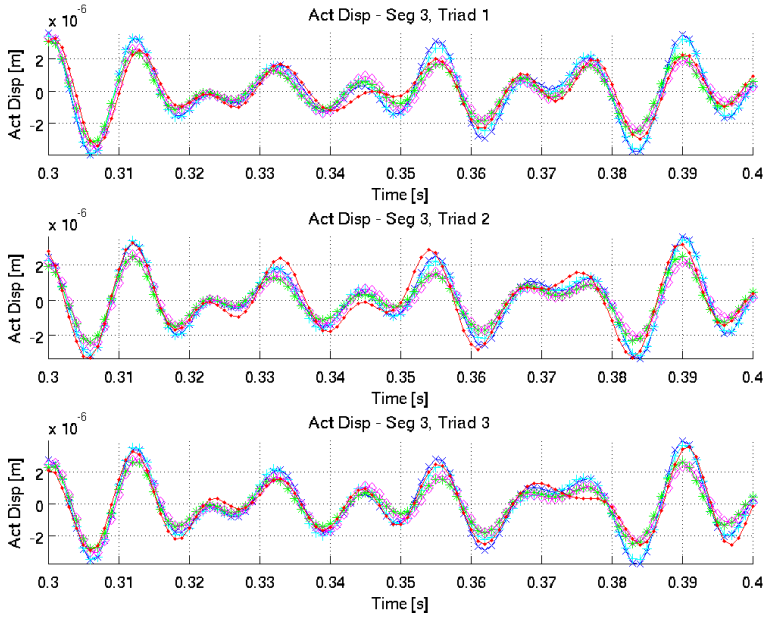

Figure 115. Random Force 50 to 100 Hz Actuator Position Error, Segment 3

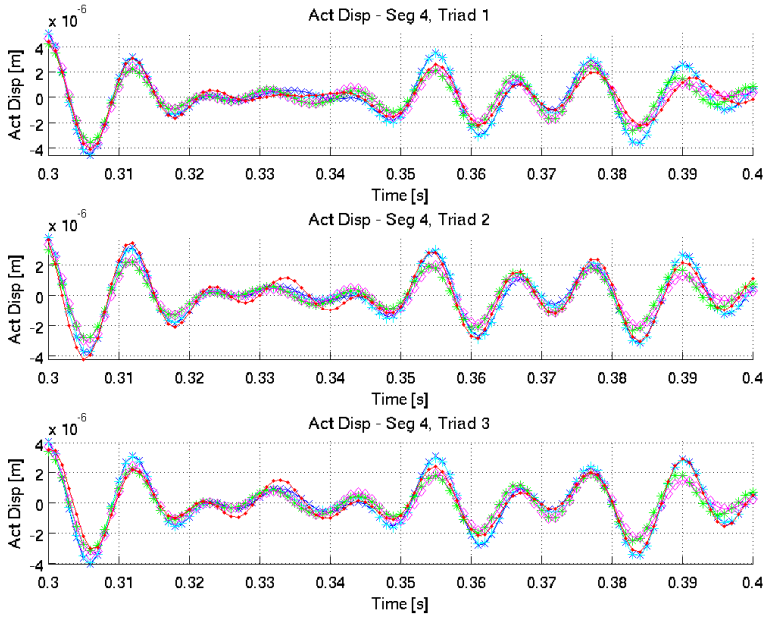

Figure 116. Random Force 50 to 100 Hz Actuator Position Error, Segment 4

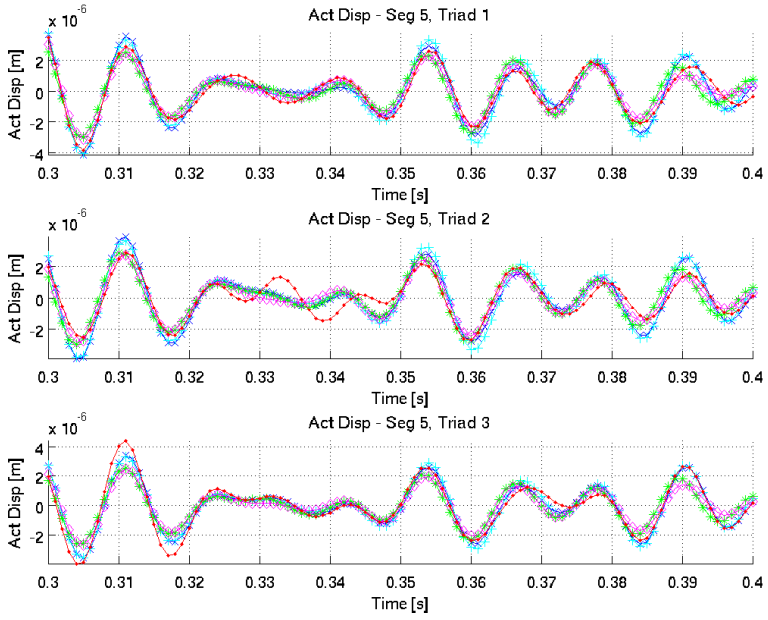

Figure 117. Random Force 50 to 100 Hz Actuator Position Error, Segment 5

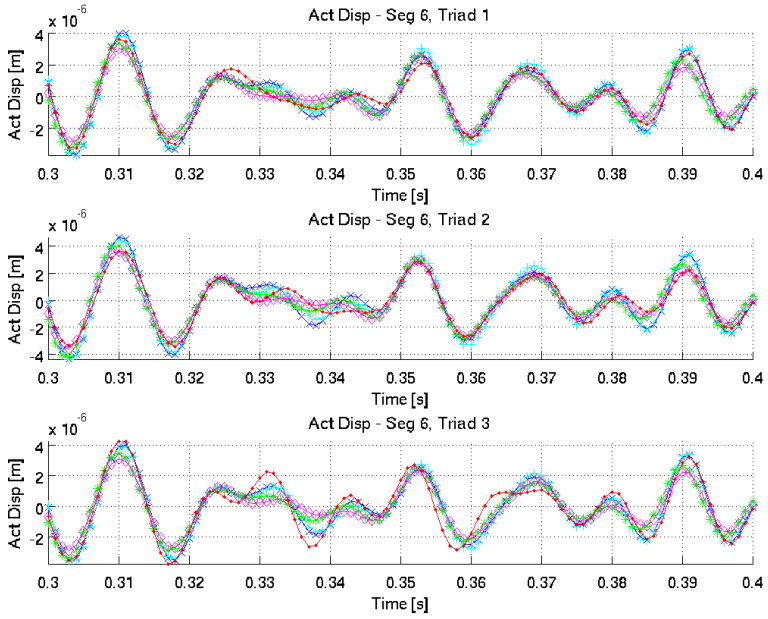

Figure 118. Random Force 50 to 100 Hz Actuator Position Error, Segment 6

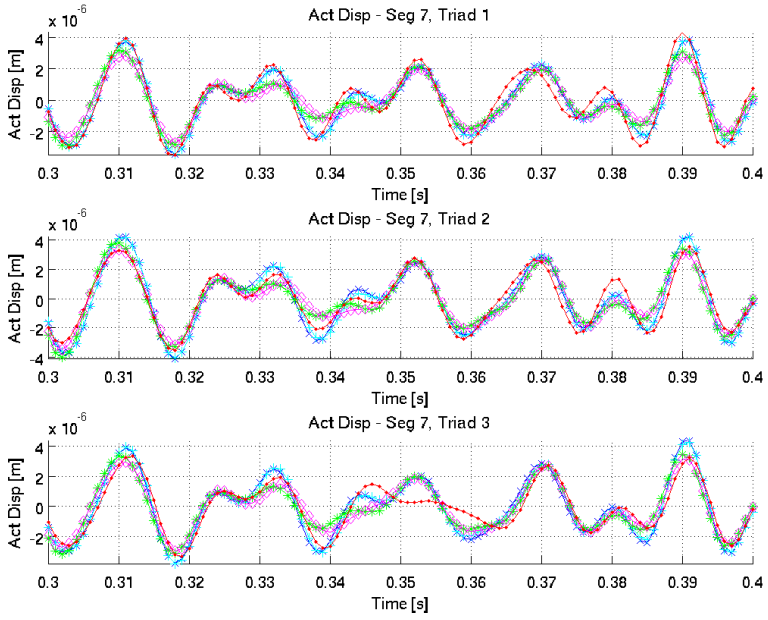

Figure 119. Random Force 50 to 100 Hz Actuator Position Error, Segment 7

The following set of figures shows the actuator control effort for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator control effort signals.

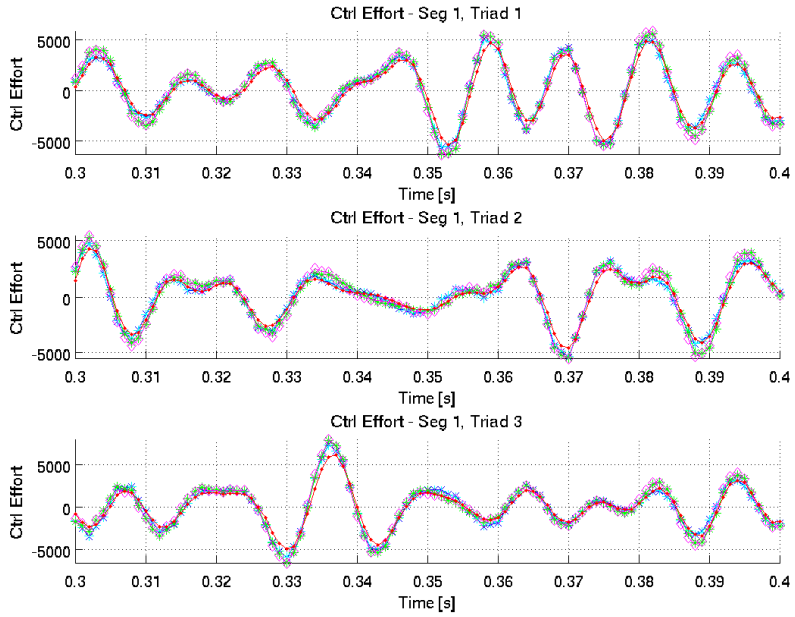

Figure 120. Random Force 50 to 100 Hz Control Effort, Segment 1

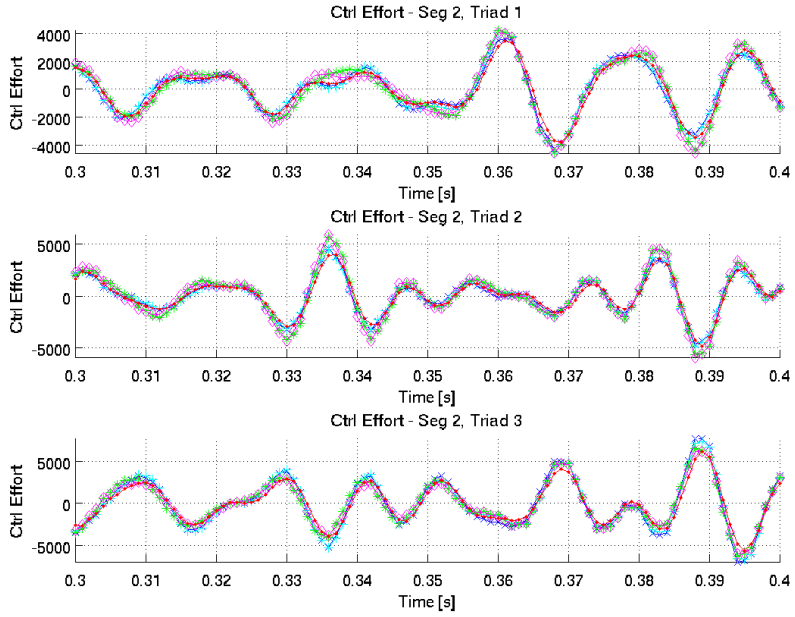

Figure 121. Random Force 50 to 100 Hz Control Effort, Segment 2

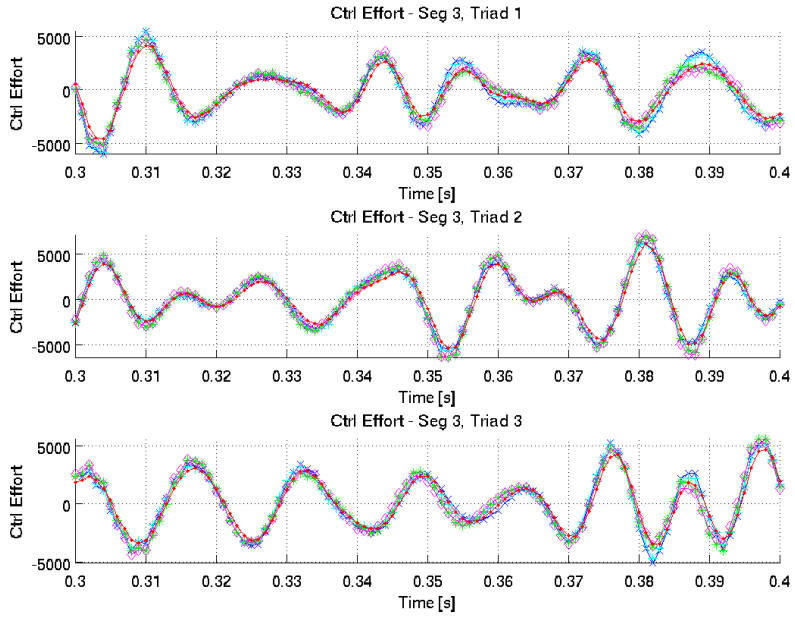

Figure 122. Random Force 50 to 100 Hz Control Effort, Segment 3

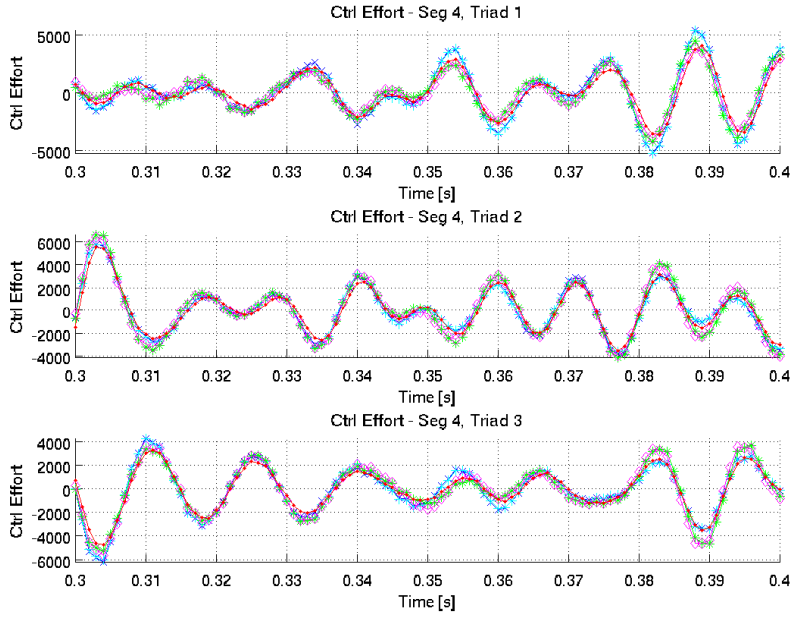

Figure 123. Random Force 50 to 100 Hz Control Effort, Segment 4

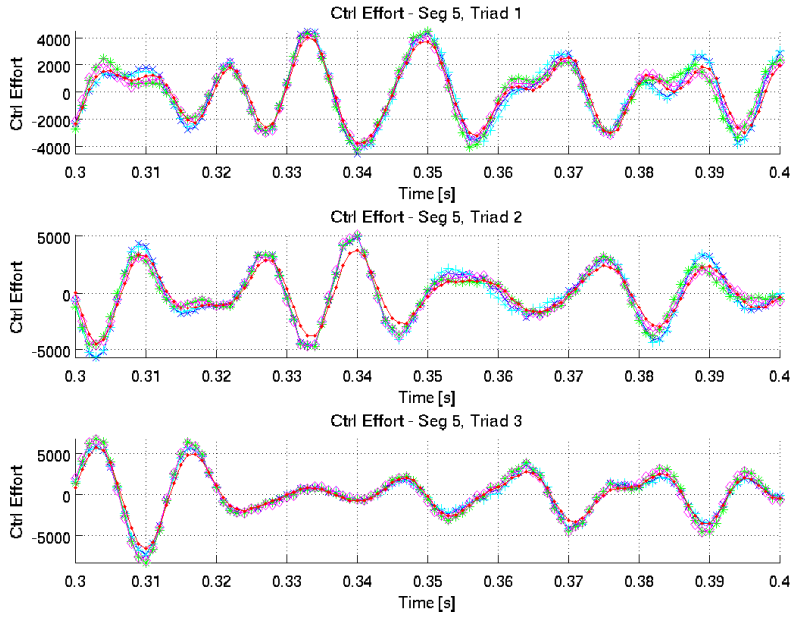

Figure 124. Random Force 50 to 100 Hz Control Effort, Segment 5

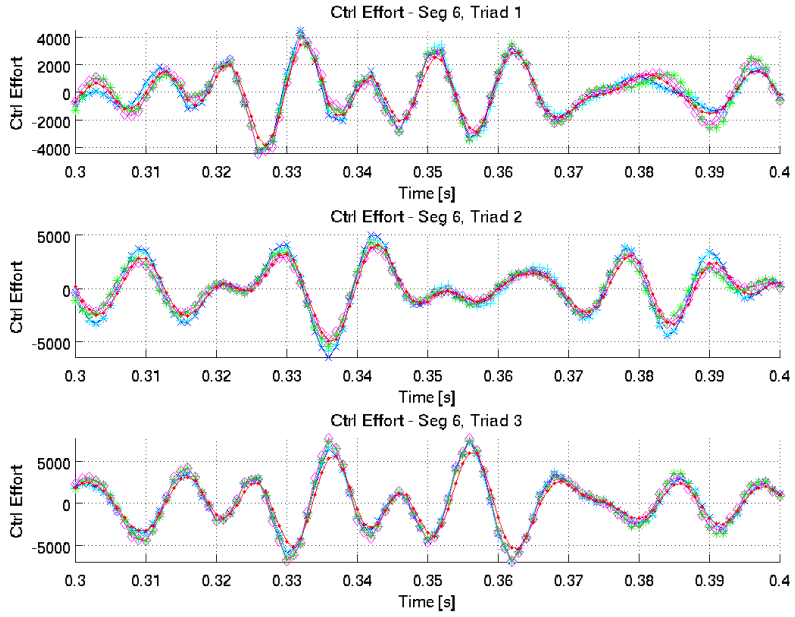

Figure 125. Random Force 50 to 100 Hz Control Effort, Segment 6

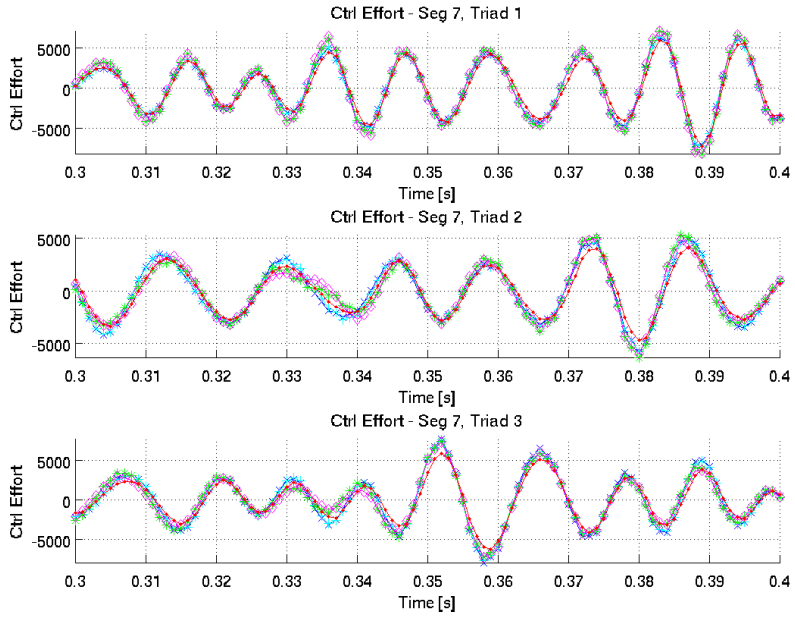

Figure 126. Random Force 50 to 100 Hz Control Effort, Segment 7

APPENDIX D – Simulation Results for Force Screen

A random forces screen was generated and then spatially filtered. The force screen was moved across the surface for the SMCT applying force in the global Z-direction to the system. The following figures show the spatial RMS of the actuator position errors: overall, sensed modes, and un-sensed modes, respectively. Where the x-axes are time in seconds and the y-axes are spatial RMS of actuator position error in meters.

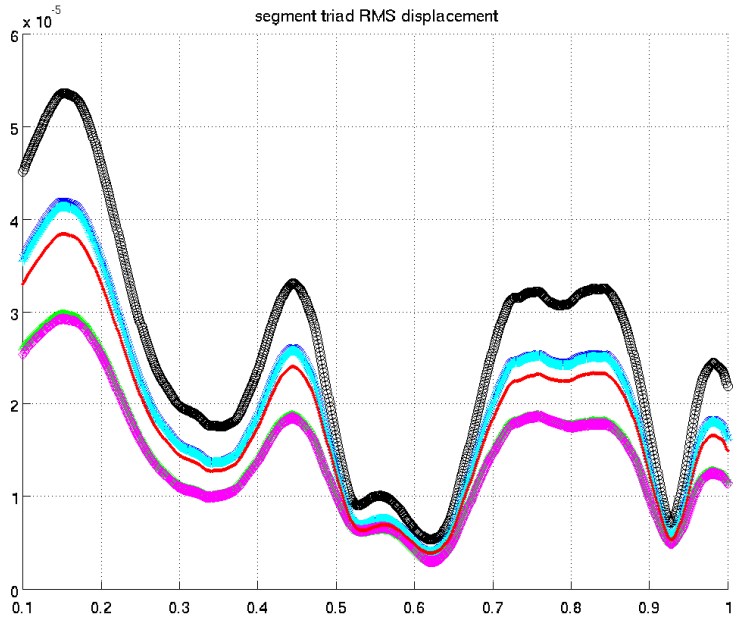

Figure 127. Force Screen RMS Actuator Position Error

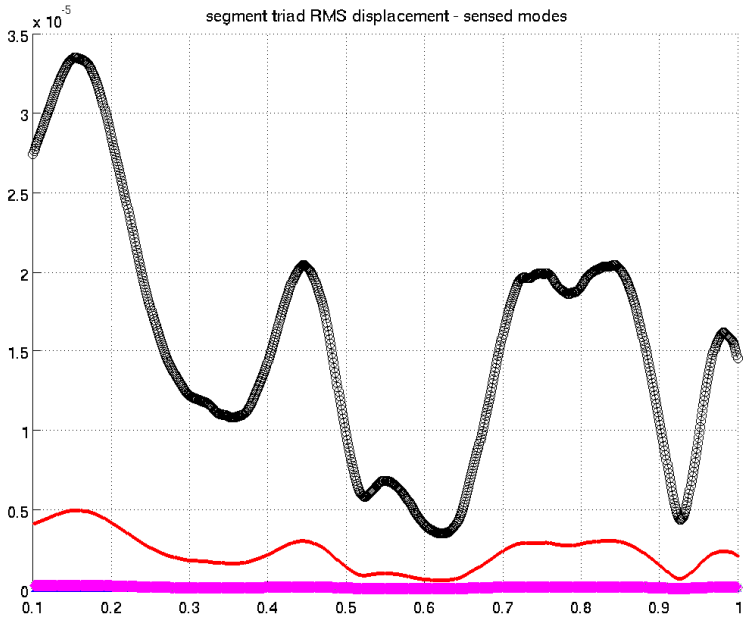

Figure 128. Force Screen RMS Actuator Position Error Sensed Modes

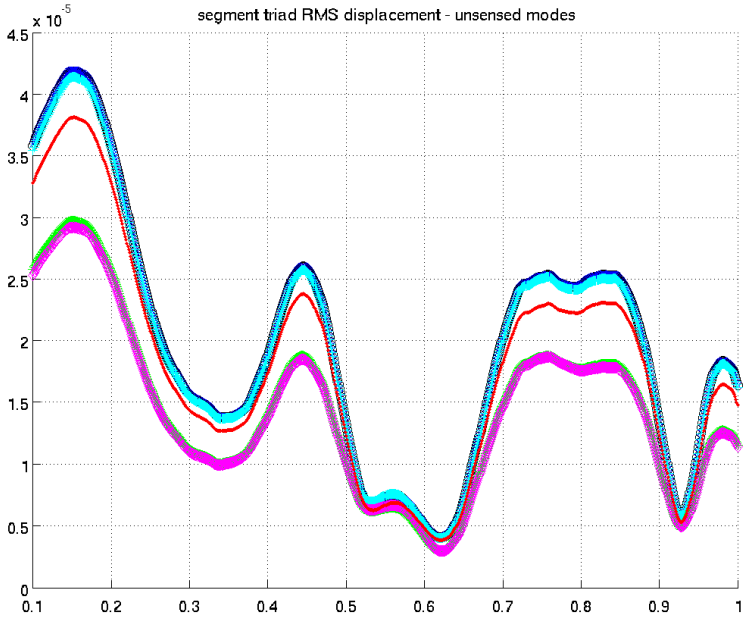

Figure 129. Force Screen RMS Actuator Position Error Unsensed Modes

The following set of figures shows the actuator displacement errors for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator position error in meters.

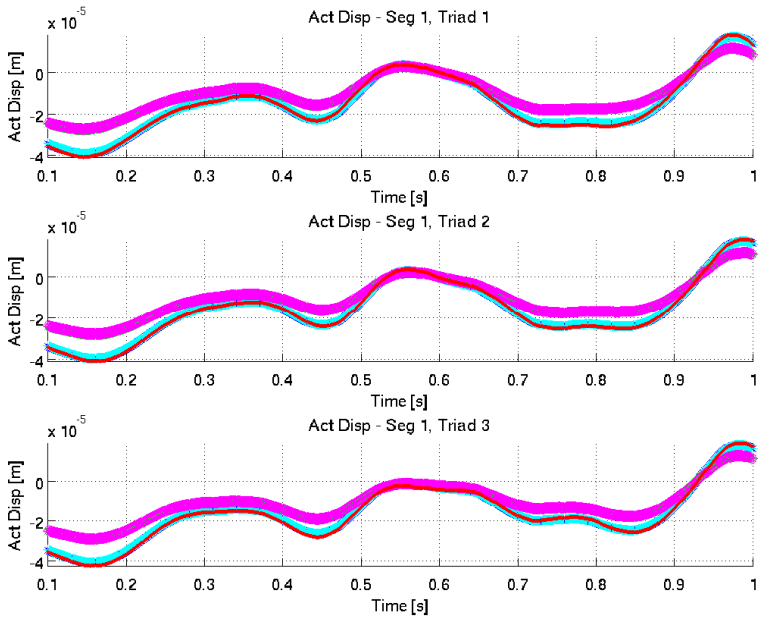

Figure 130. Force Screen Actuator Position Error, Segment 1

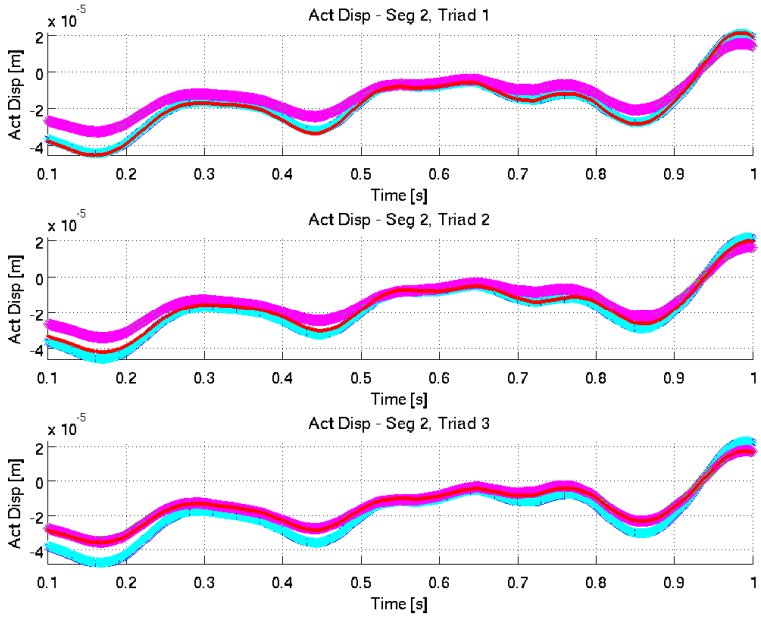

Figure 131. Force Screen Actuator Position Error, Segment 2

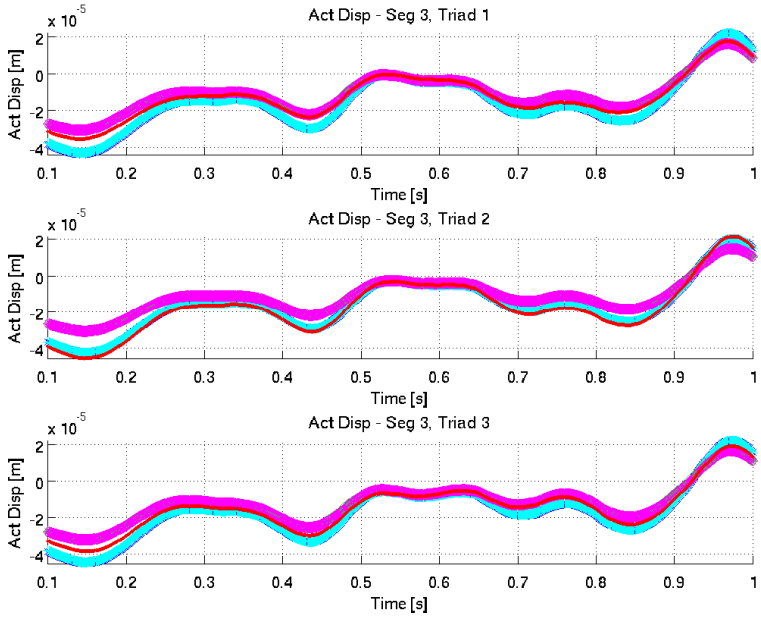

Figure 132. Force Screen Actuator Position Error, Segment 3

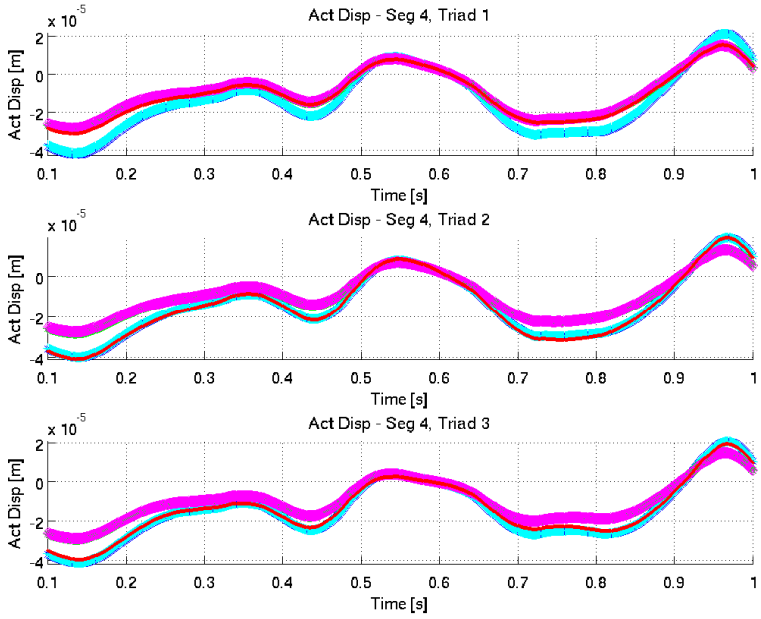

Figure 133. Force Screen Actuator Position Error, Segment 4

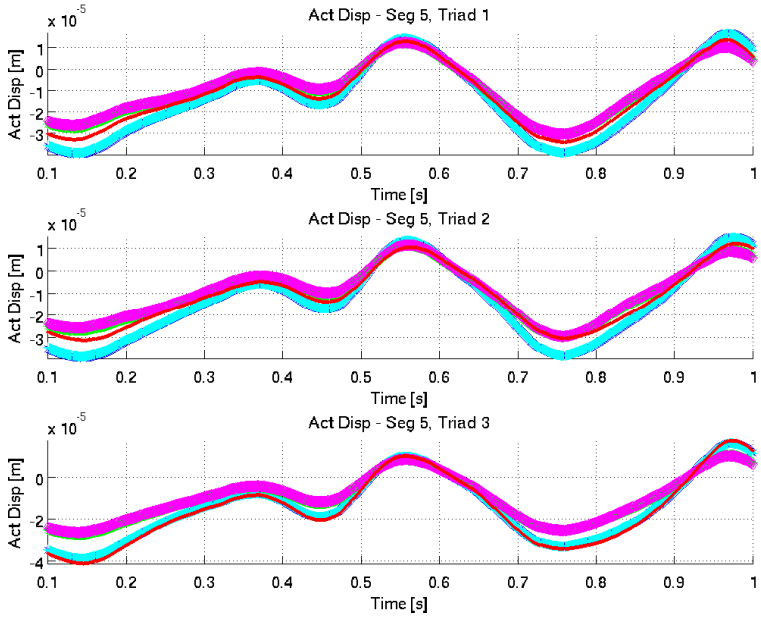

Figure 134. Force Screen Actuator Position Error, Segment 5

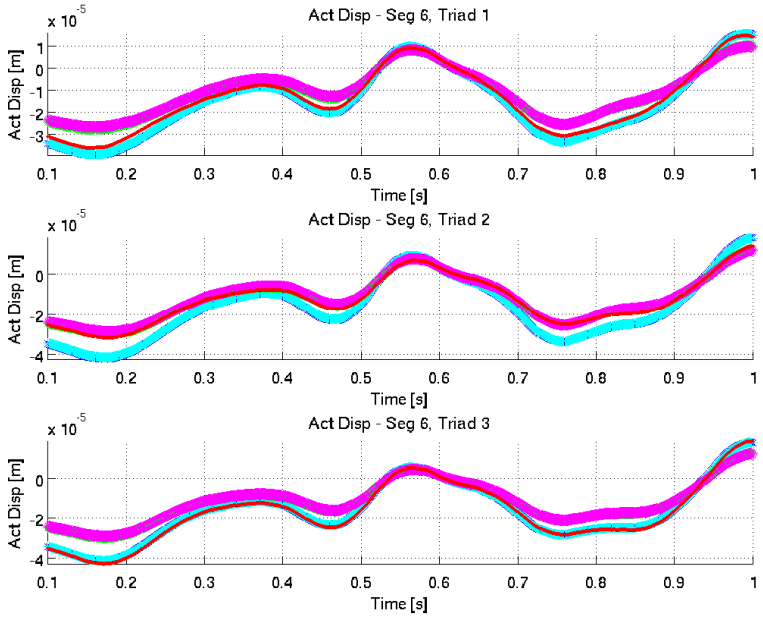

Figure 135. Force Screen Actuator Position Error, Segment 6

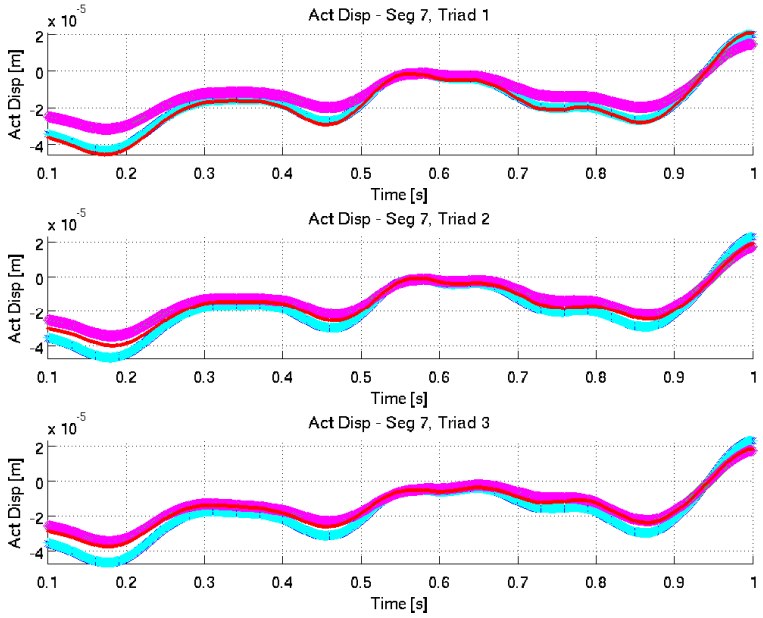

Figure 136. Force Screen Actuator Position Error, Segment 7

The following set of figures shows the actuator control effort for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator control effort signals.

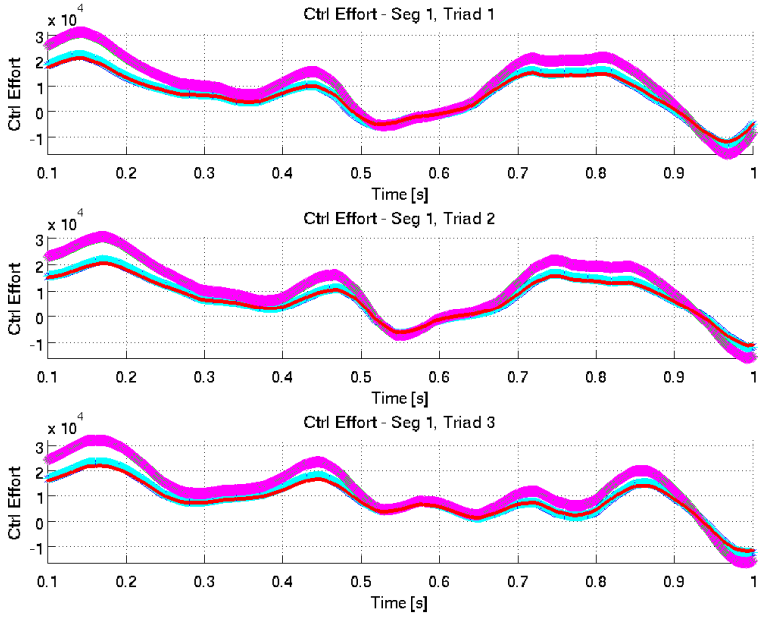

Figure 137. Force Screen Control Effort, Segment 1

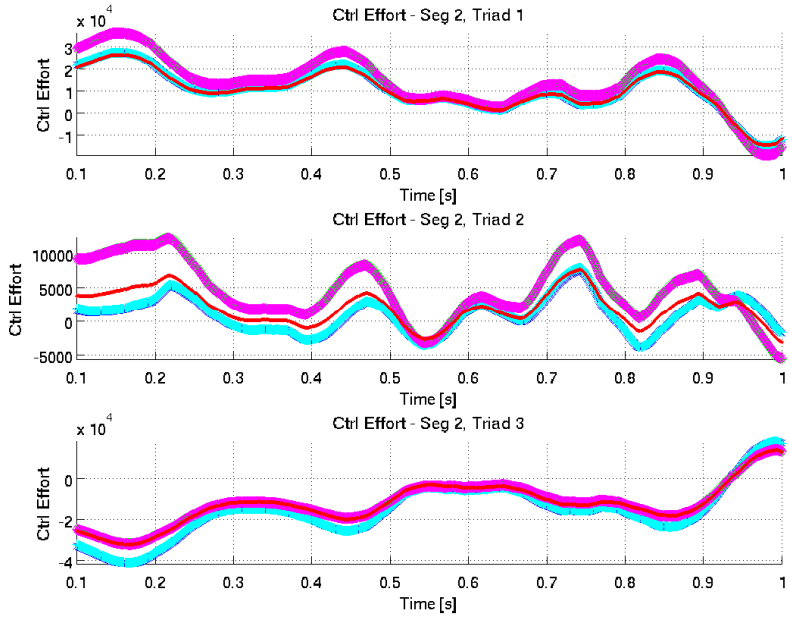

Figure 138. Force Screen Control Effort, Segment 2

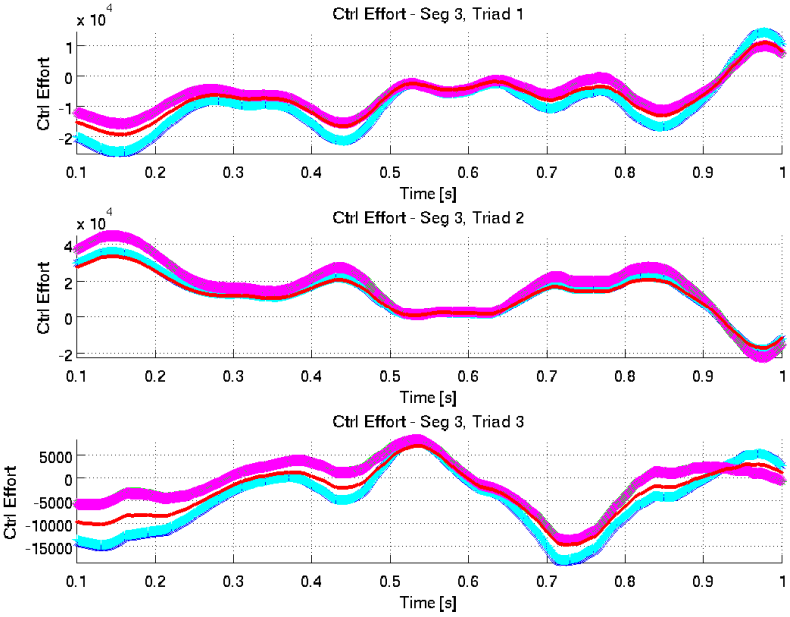

Figure 139. Force Screen Control Effort, Segment 3

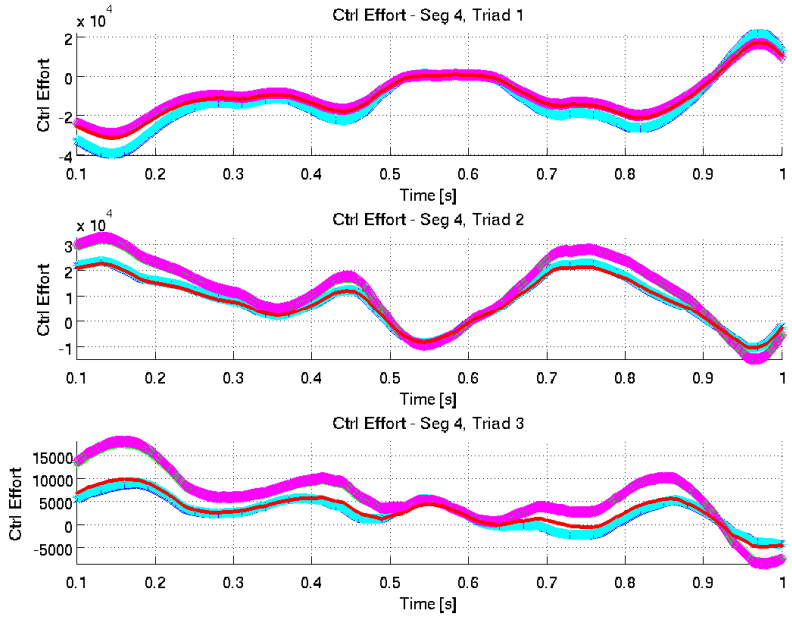

Figure 140. Force Screen Control Effort, Segment 4

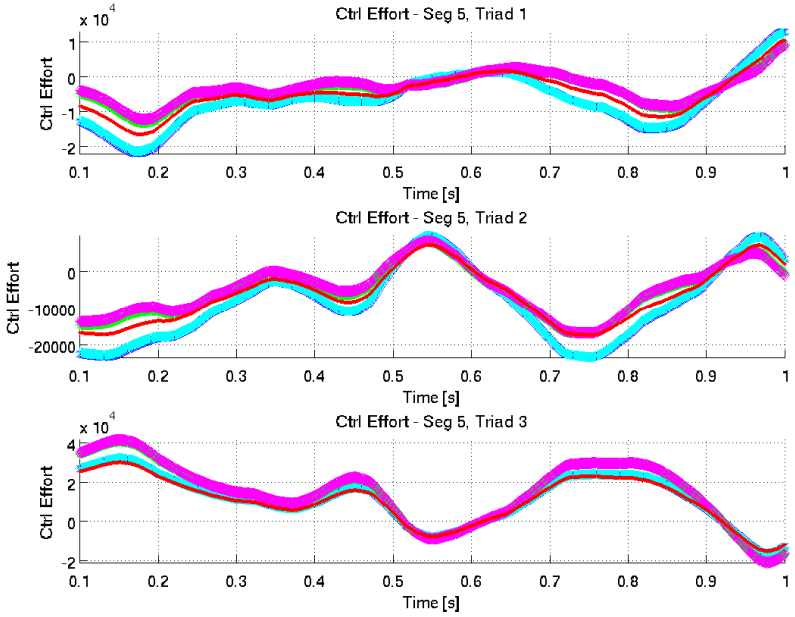

Figure 141. Force Screen Control Effort, Segment 5

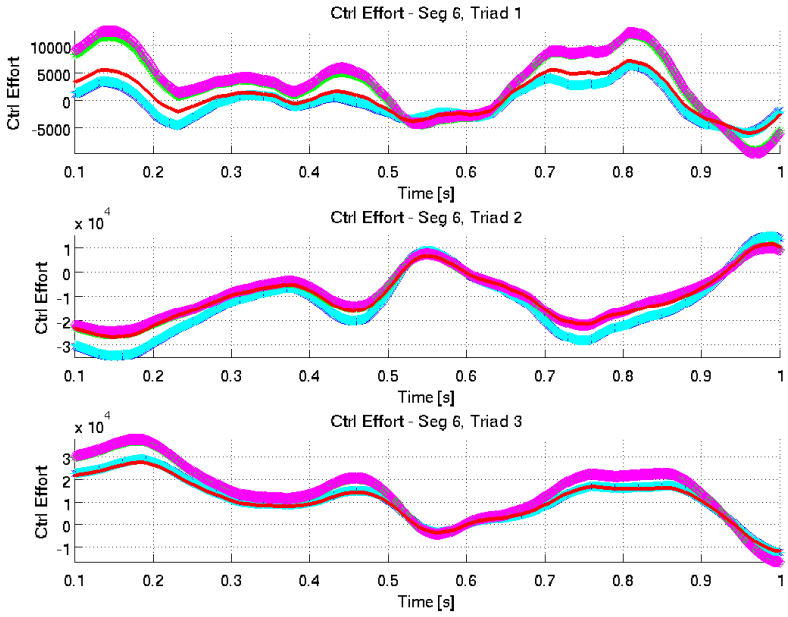

Figure 142. Force Screen Control Effort, Segment 6

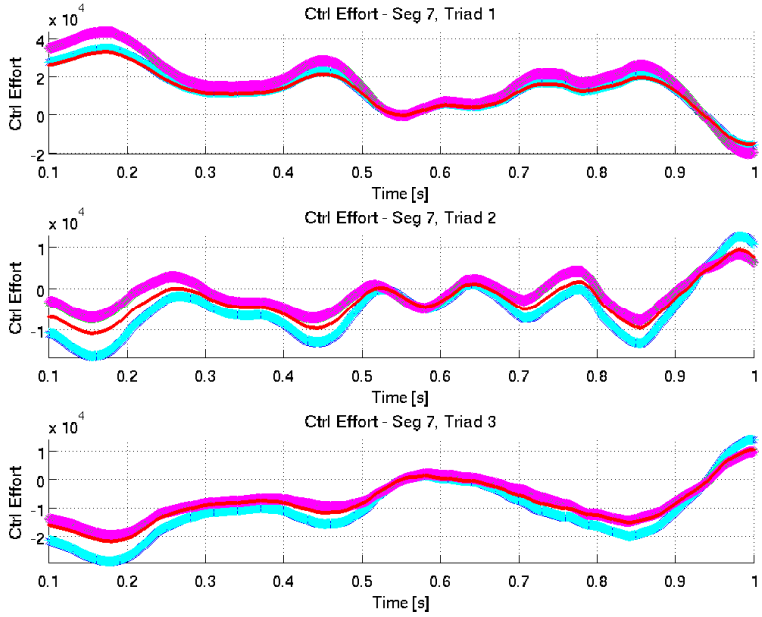

Figure 143. Force Screen Control Effort, Segment 7

APPENDIX E – Simulation Results for Step on Centre Segment

A step force was applied to the centre segment in the global Z-direction. The following figures show the spatial RMS of the actuator position errors: overall, sensed modes, and un-sensed modes, respectively. Where the x-axes are time in seconds and the y-axes are spatial RMS of actuator position error in meters.

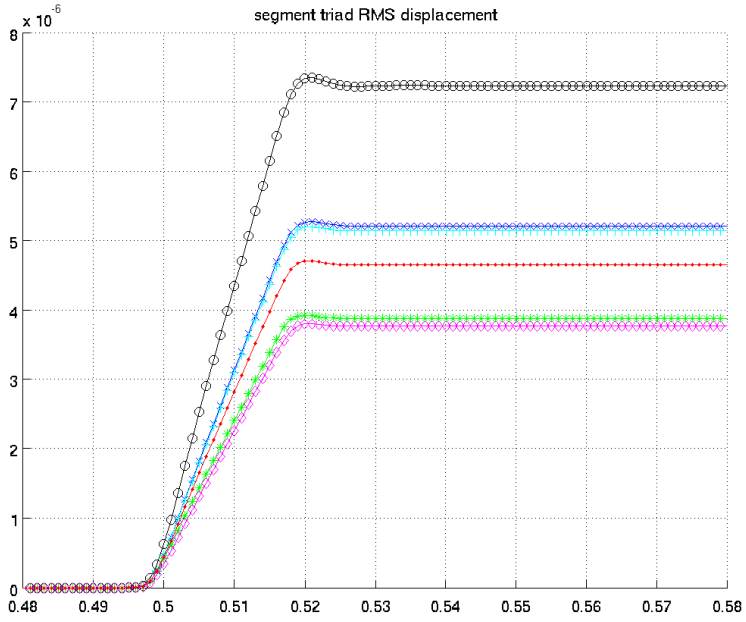

Figure 144. Step Force RMS Actuator Position Error

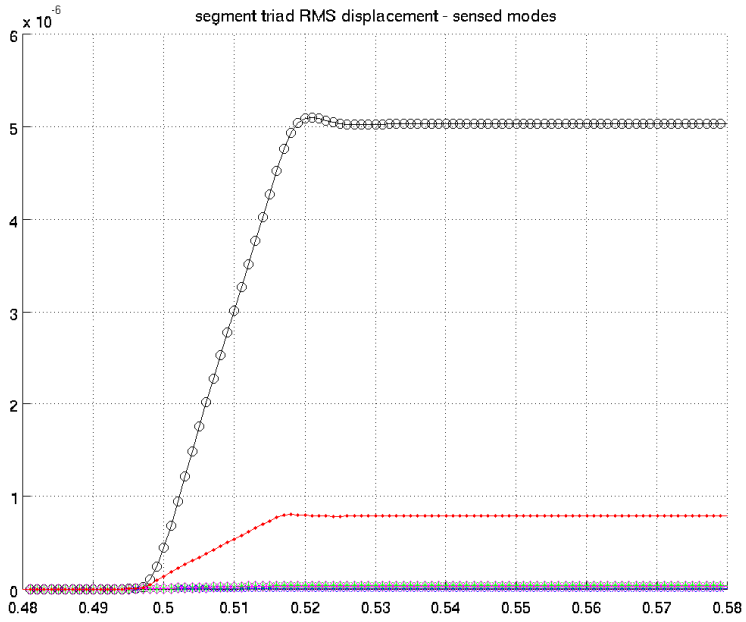

Figure 145. Step Force RMS Actuator Position Error Sensed Modes

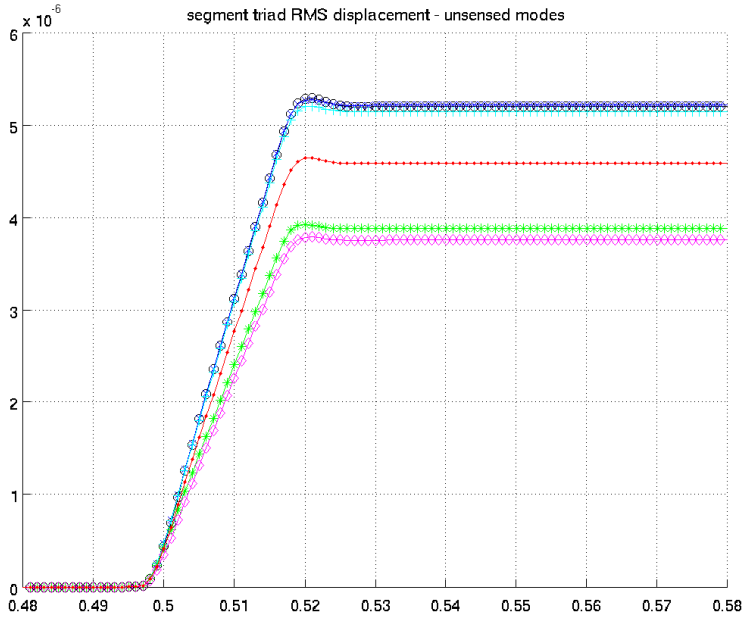

Figure 146. Step Force RMS Actuator Position Error Unsensed Modes

The following set of figures shows the actuator displacement errors for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator position error in meters.

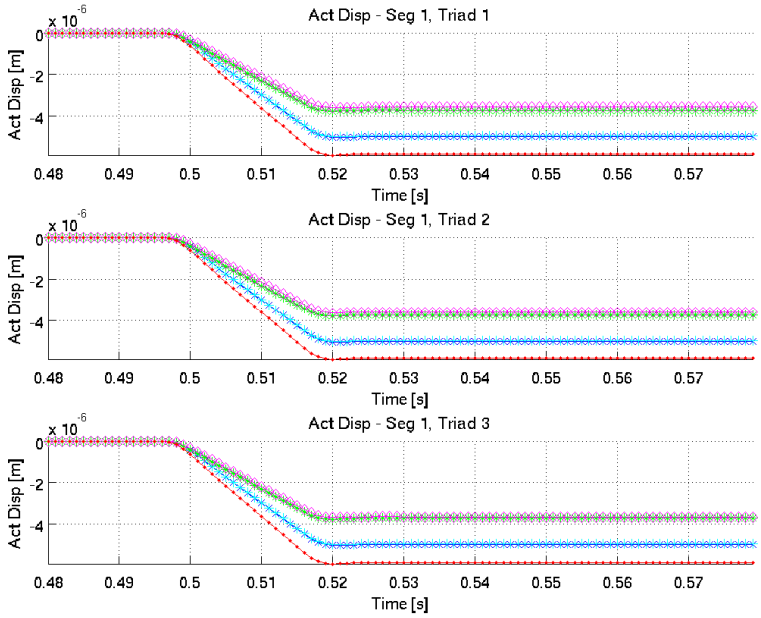

Figure 147. Step Force Actuator Position Error, Segment 1

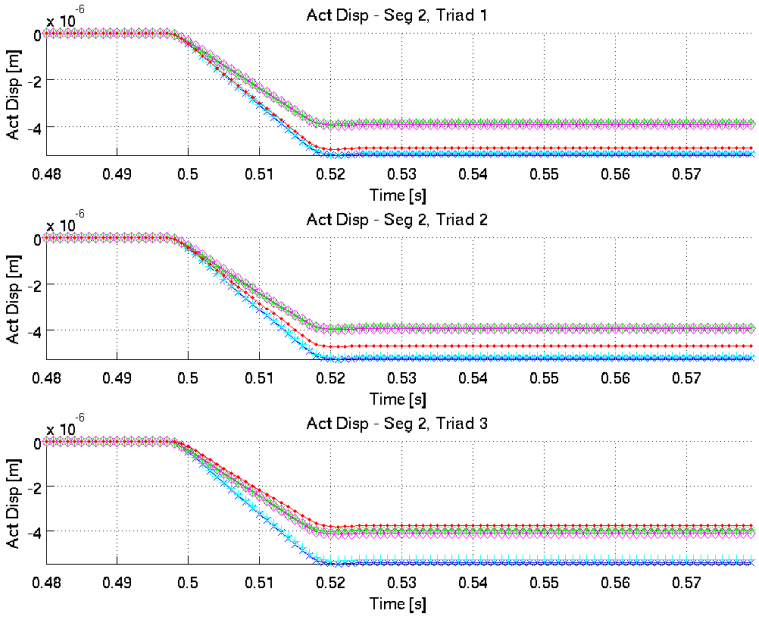

Figure 148. Step Force Actuator Position Error, Segment 2

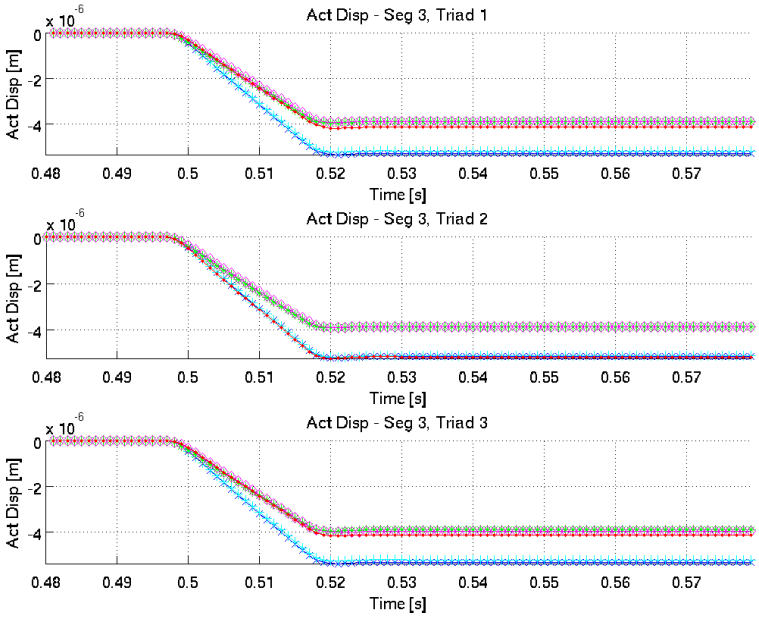

Figure 149. Step Force Actuator Position Error, Segment 3

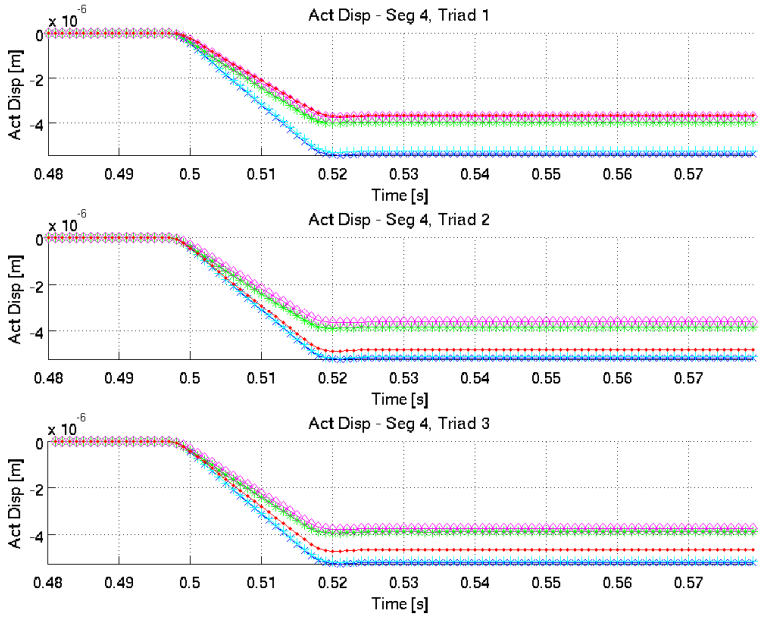

Figure 150. Step Force Actuator Position Error, Segment 4

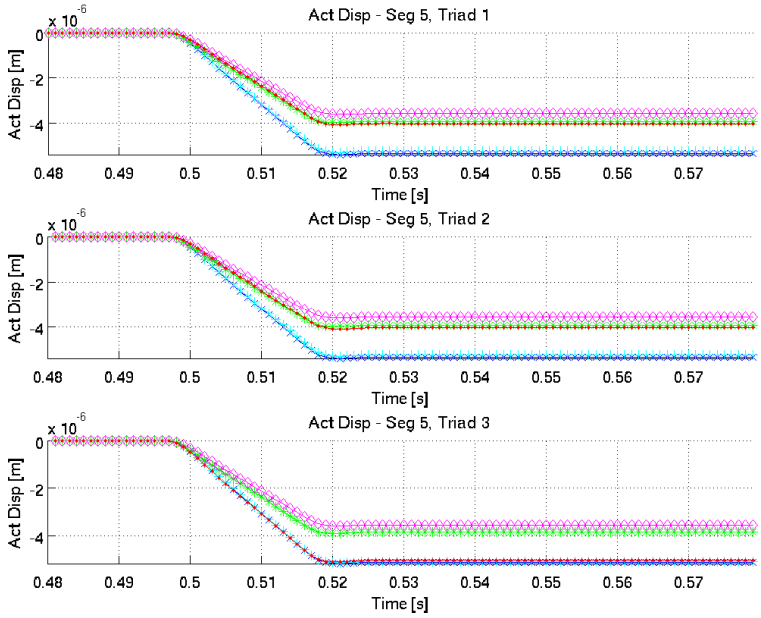

Figure 151. Step Force Actuator Position Error, Segment 5

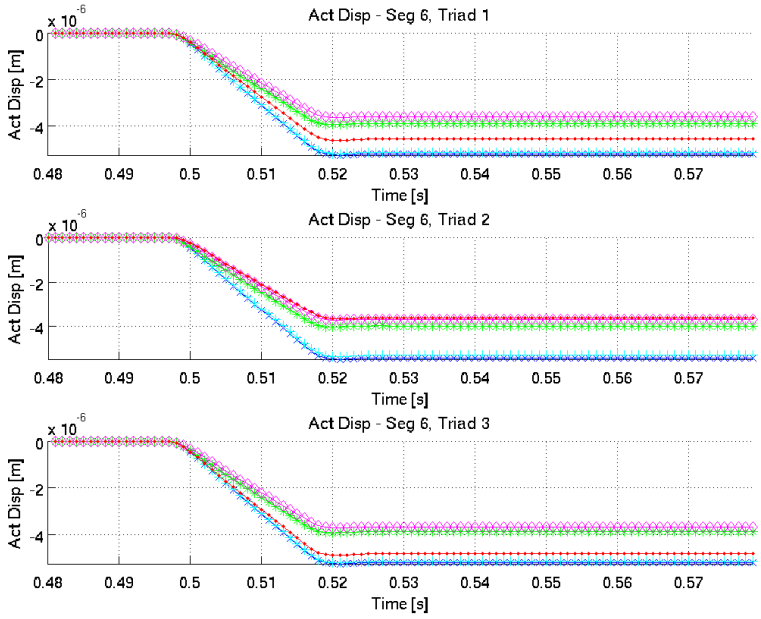

Figure 152. Step Force Actuator Position Error, Segment 6

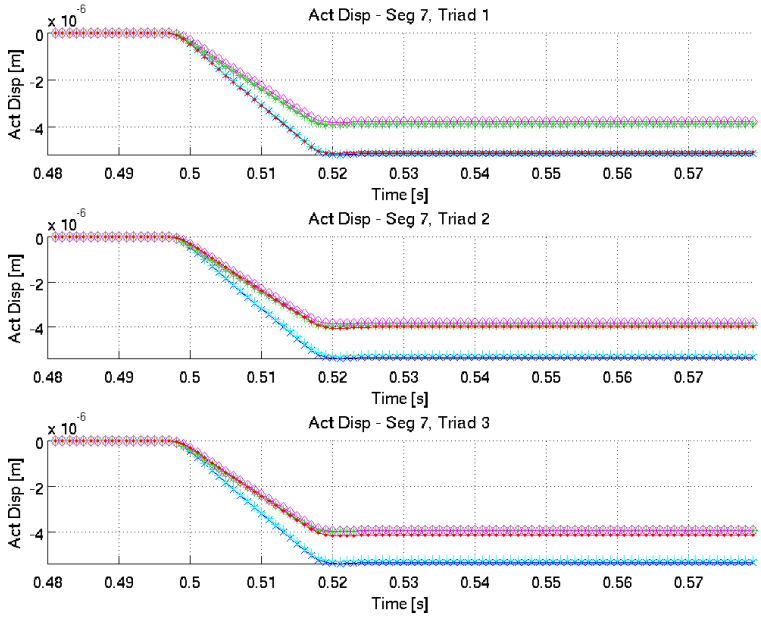

Figure 153. Step Force Actuator Position Error, Segment 7

The following set of figures shows the actuator control effort for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator control effort signals.

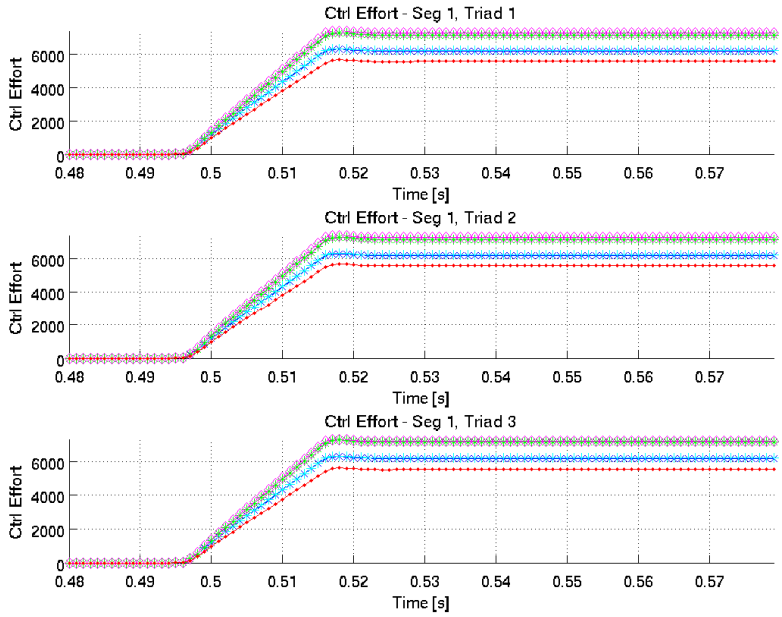

Figure 154. Step Force Control Effort, Segment 1

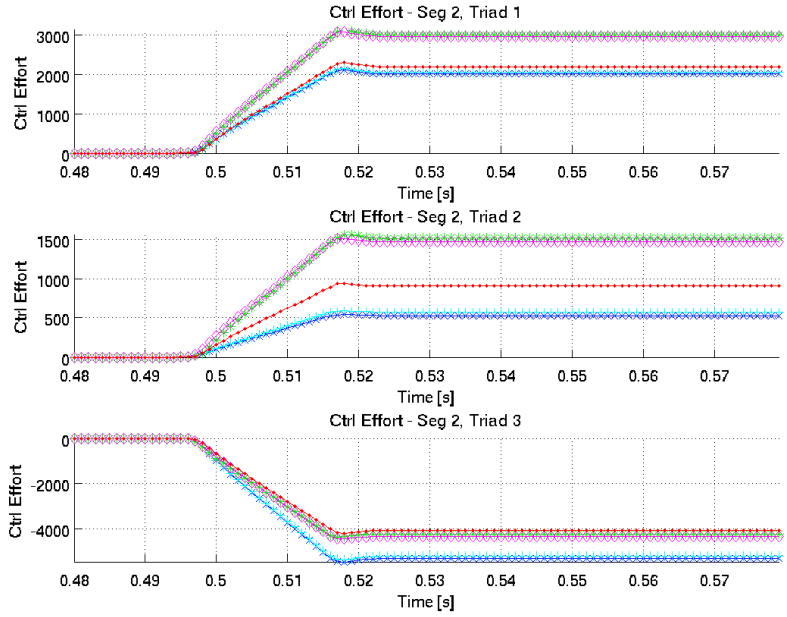

Figure 155. Step Force Control Effort, Segment 2

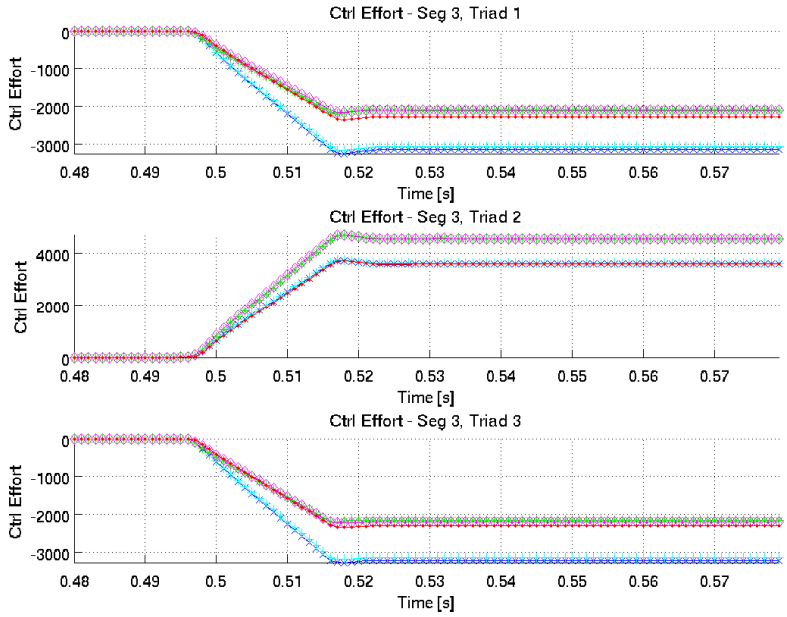

Figure 156. Step Force Control Effort, Segment 3

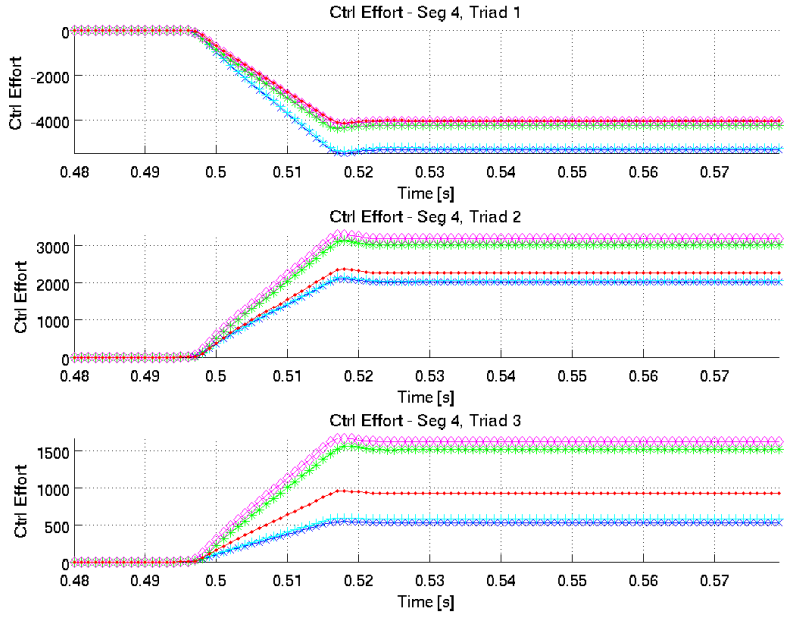

Figure 157. Step Force Control Effort, Segment 4

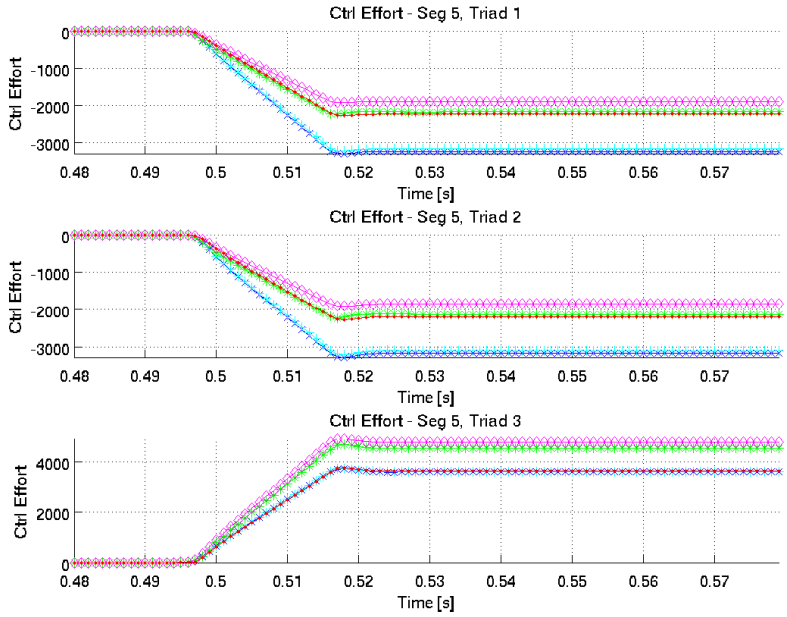

Figure 158. Step Force Control Effort, Segment 5

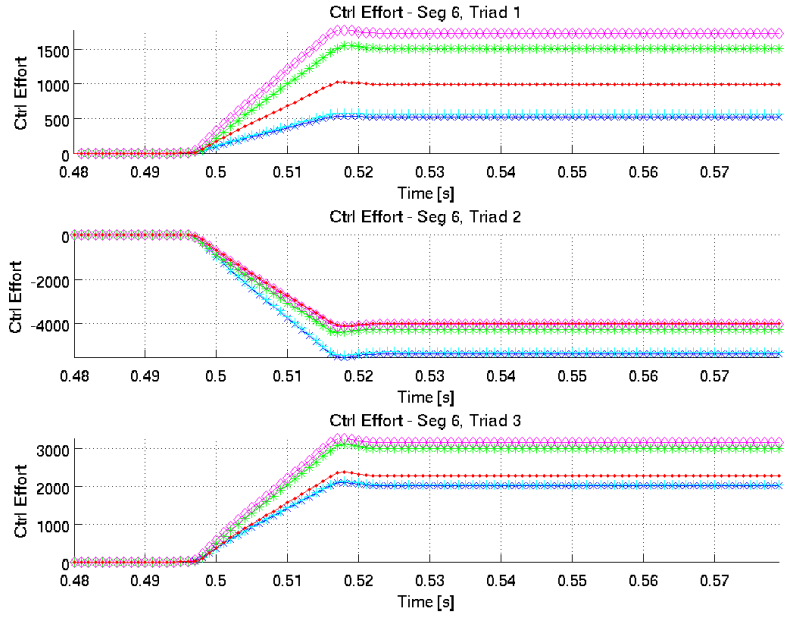

Figure 159. Step Force Control Effort, Segment 6

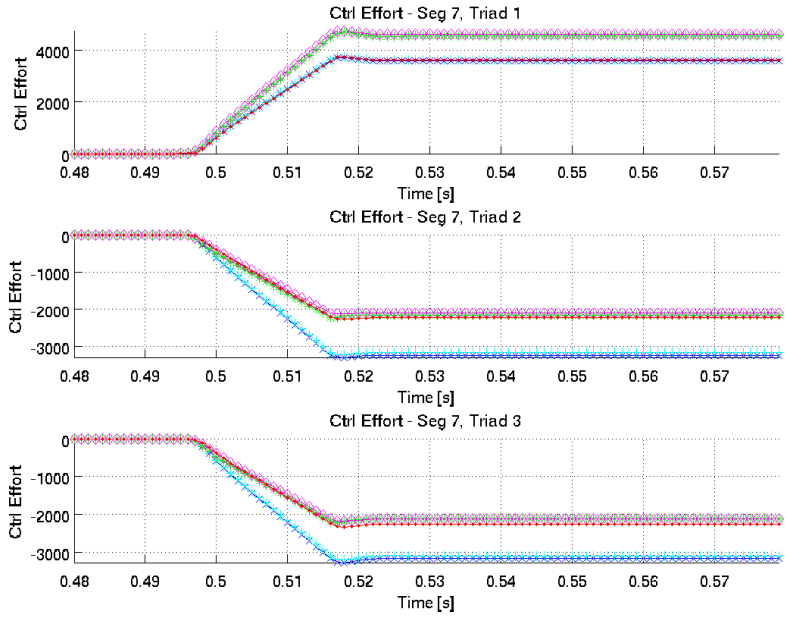

Figure 160. Step Force Control Effort, Segment 7

APPENDIX F – Simulation Results for Impulse on Centre Segment

An impulse force was applied to the centre segment in the global Z-direction. The following figures show the spatial RMS of the actuator position errors: overall, sensed modes, and un-sensed modes, respectively. Where the x-axes are time in seconds and the y-axes are spatial RMS of actuator position error in meters.

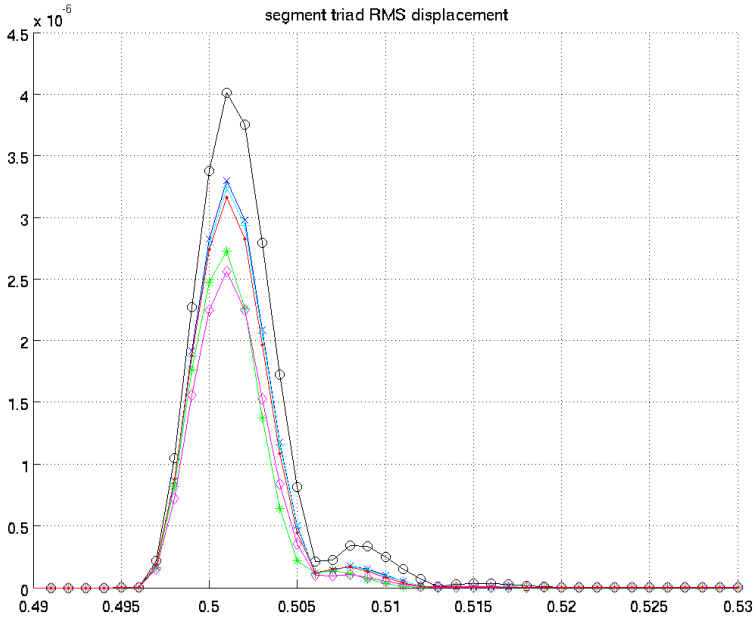

Figure 161. Impulse Force RMS Actuator Position Error

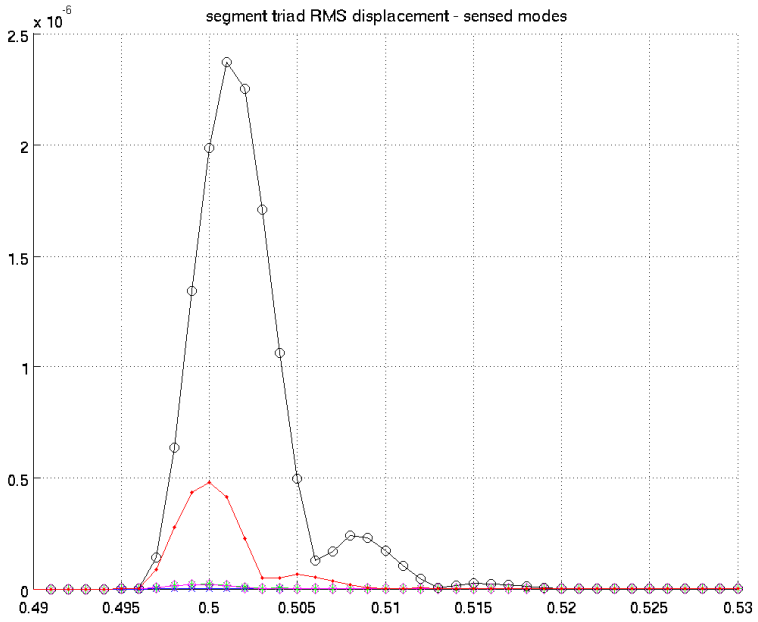

Figure 162. Impulse Force RMS Actuator Position Error Sensed Modes

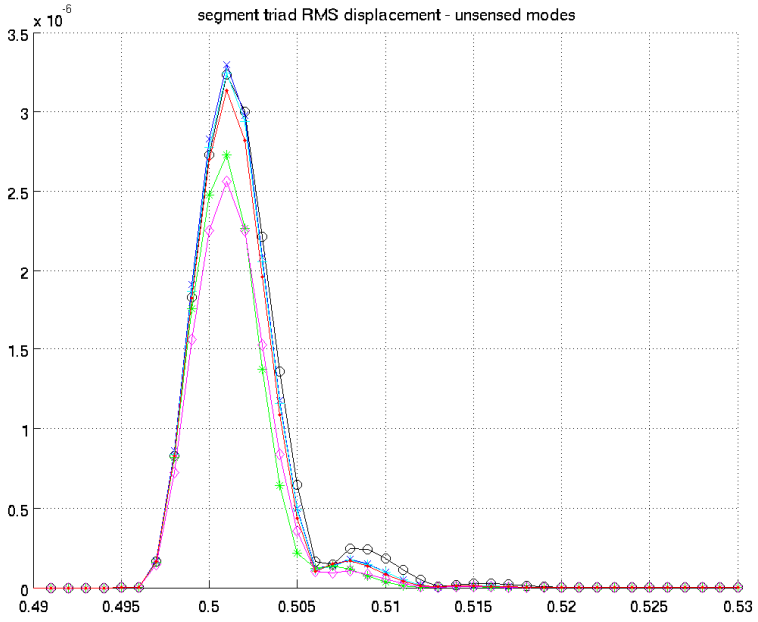

Figure 163. Impulse Force RMS Actuator Position Error Unsensed Modes

The following set of figures shows the actuator displacement errors for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator position error in meters.

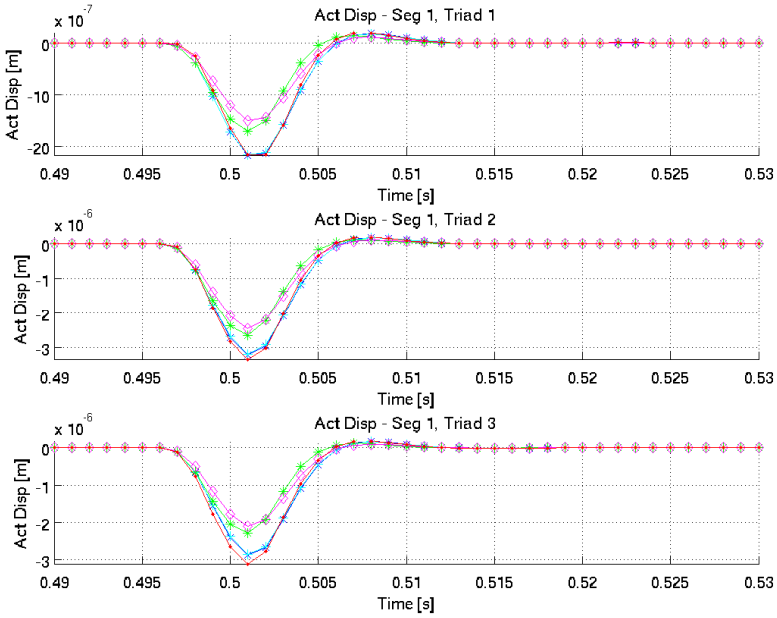

Figure 164. Impulse Force Actuator Position Error, Segment 1

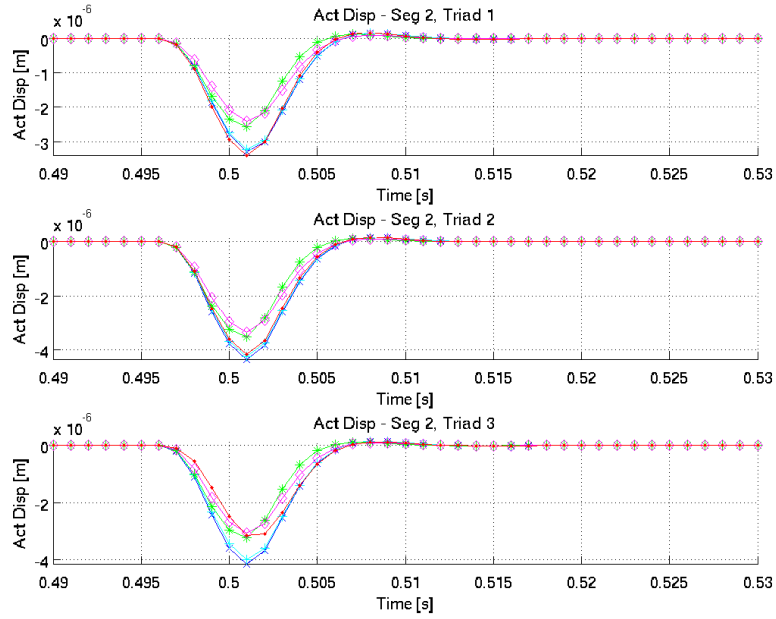

Figure 165. Impulse Force Actuator Position Error, Segment 2

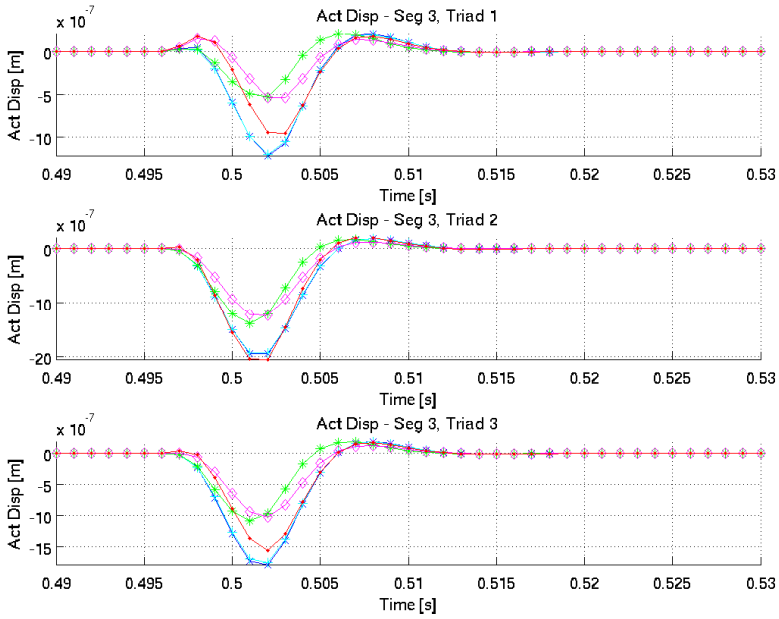

Figure 166. Impulse Force Actuator Position Error, Segment 3

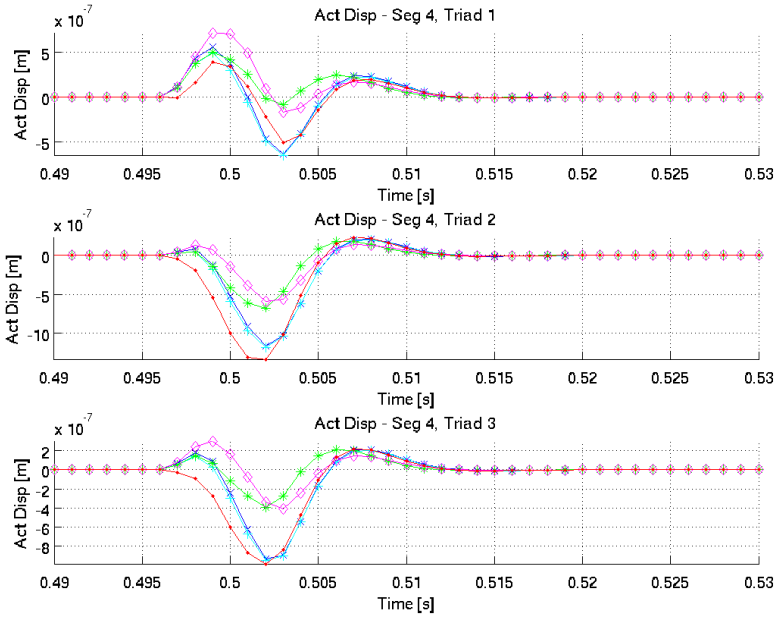

Figure 167. Impulse Force Actuator Position Error, Segment 4

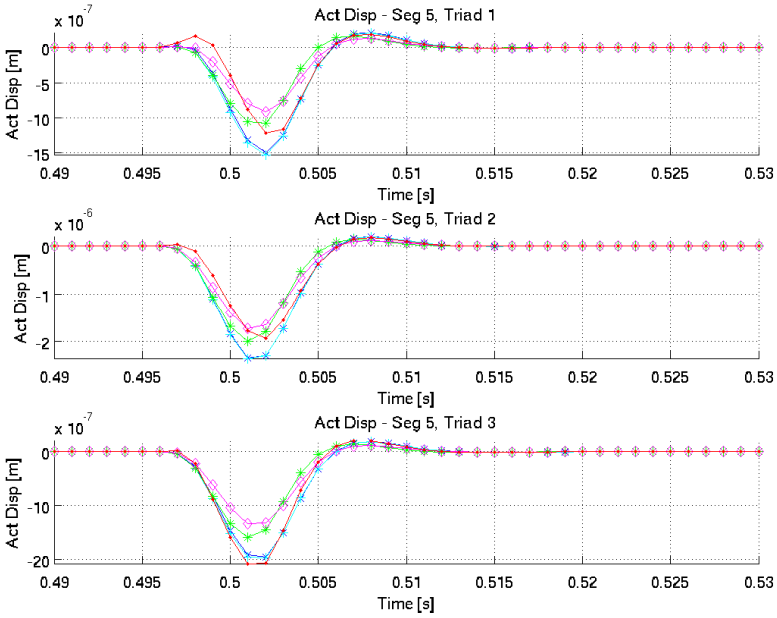

Figure 168. Impulse Force Actuator Position Error, Segment 5

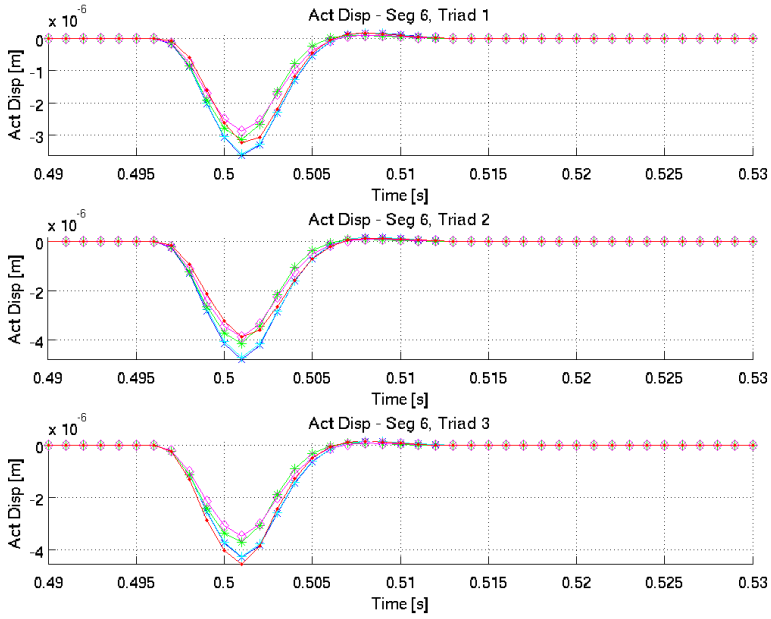

Figure 169. Impulse Force Actuator Position Error, Segment 6

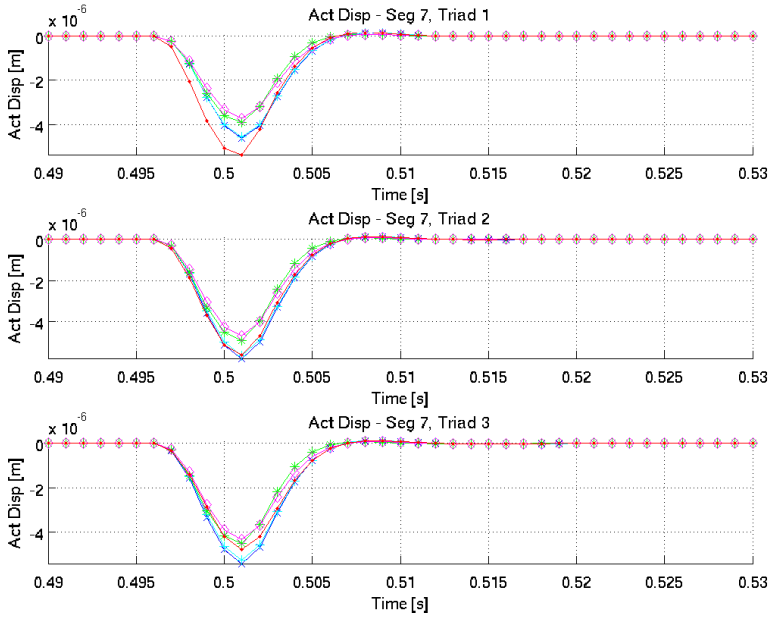

Figure 170. Impulse Force Actuator Position Error, Segment 7

The following set of figures shows the actuator control effort for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator control effort signals.

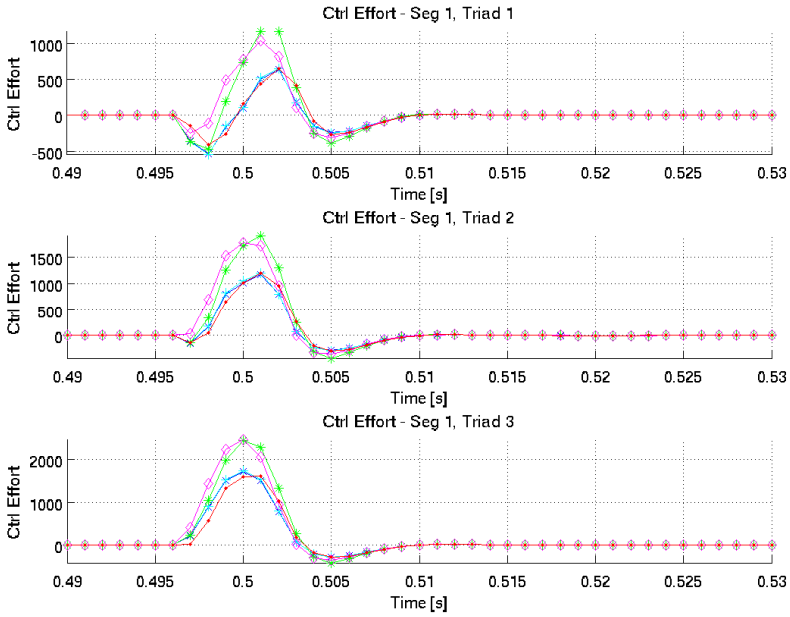

Figure 171. Impulse Force Control Effort, Segment 1

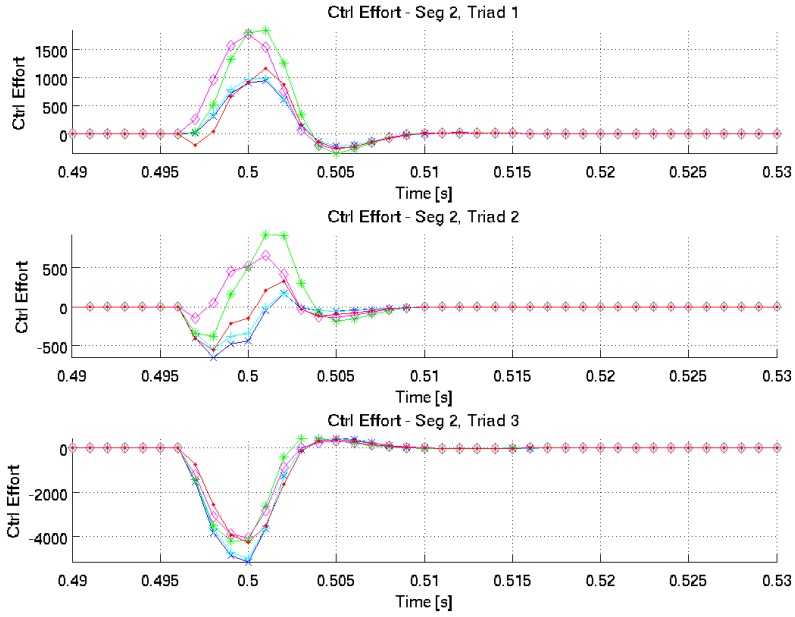

Figure 172. Impulse Force Control Effort, Segment 2

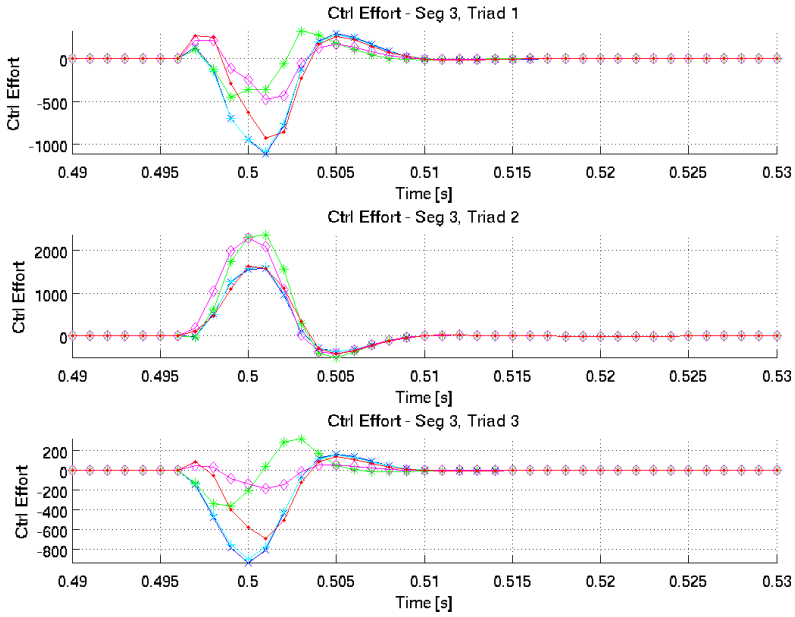

Figure 173. Impulse Force Control Effort, Segment 3

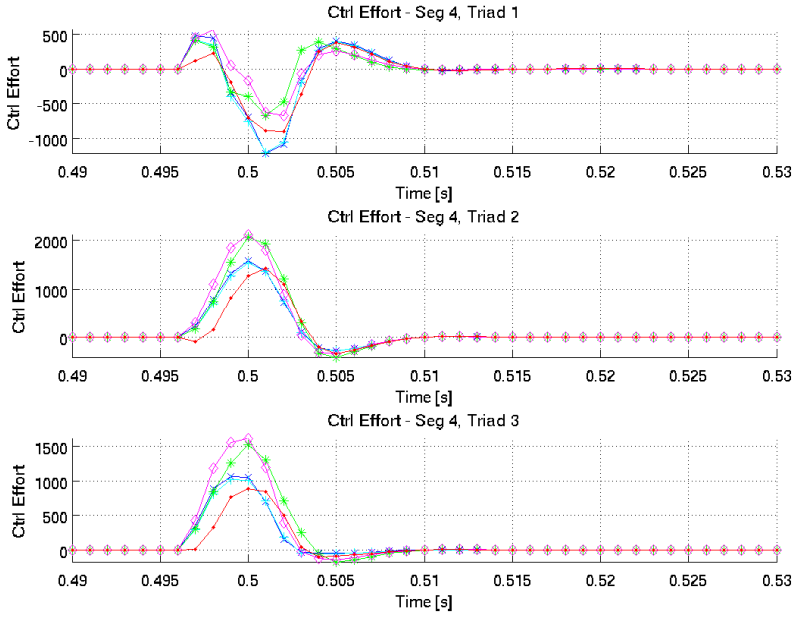

Figure 174. Impulse Force Control Effort, Segment 4

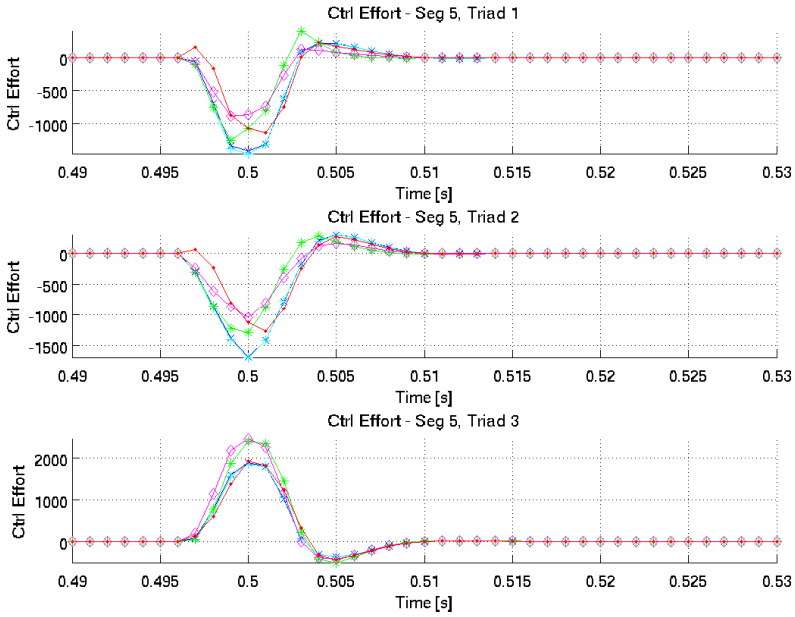

Figure 175. Impulse Force Control Effort, Segment 5

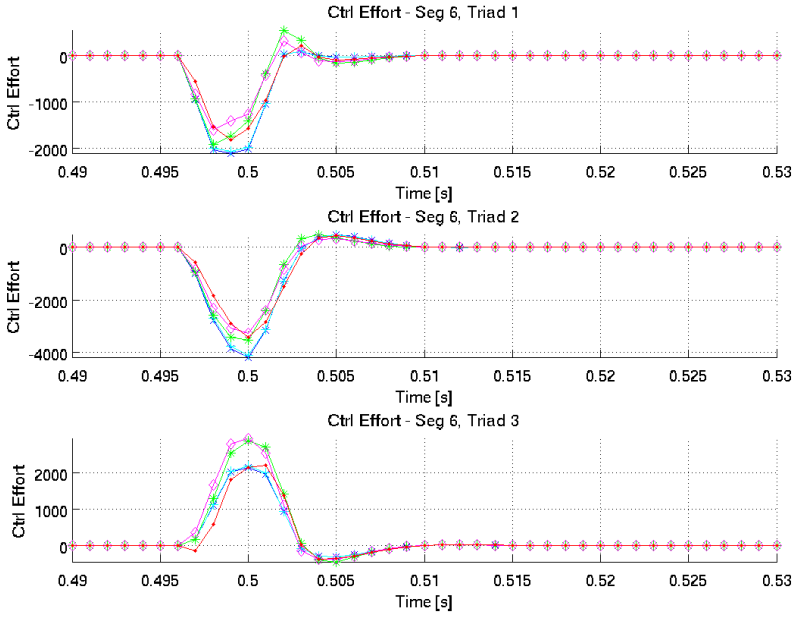

Figure 176. Impulse Force Control Effort, Segment 6

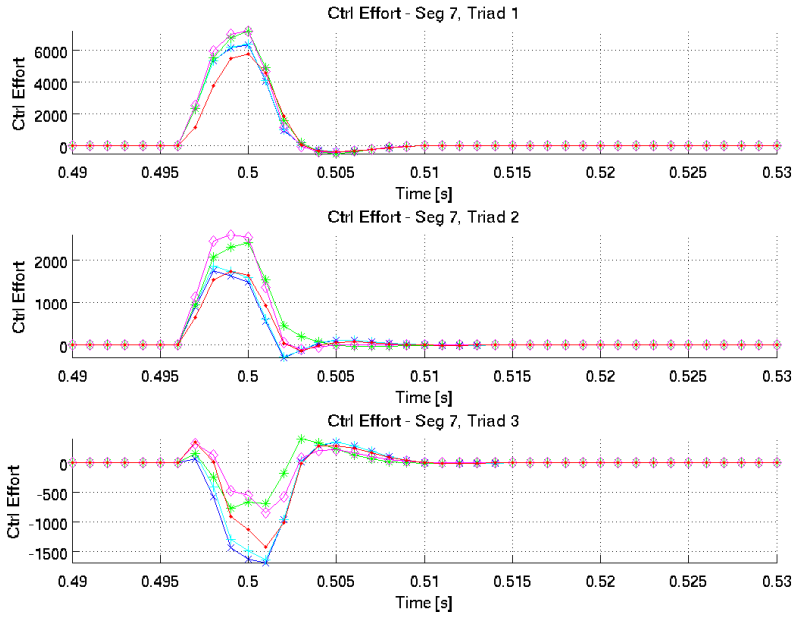

Figure 177. Impulse Force Control Effort, Segment 7

APPENDIX G – Simulation Results for 10 Hz Sinusoidal Force

A 10 Hz sinusoidal force was applied to all nodes in the global Z-direction. The following figures show the spatial RMS of the actuator position errors: overall, sensed modes, and un-sensed modes, respectively. Where the x-axes are time in seconds and the y-axes are spatial RMS of actuator position error in meters.

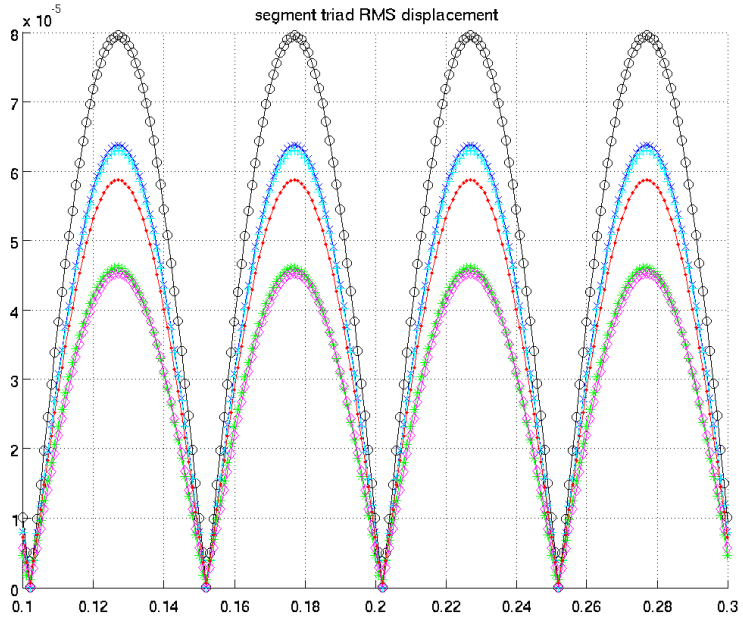

Figure 178. 10 Hz Sinusoidal Force RMS Actuator Position Error

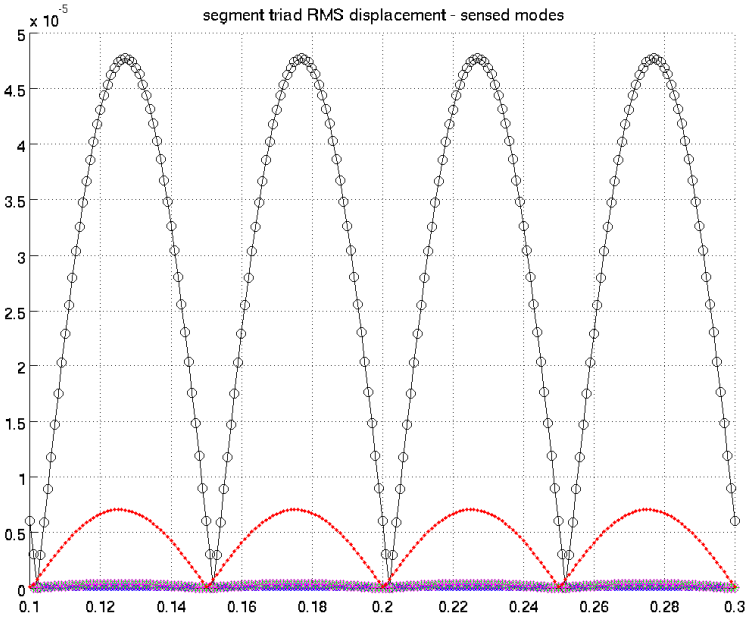

Figure 179. 10 Hz Sinusoidal Force RMS Actuator Position Error Sensed Modes

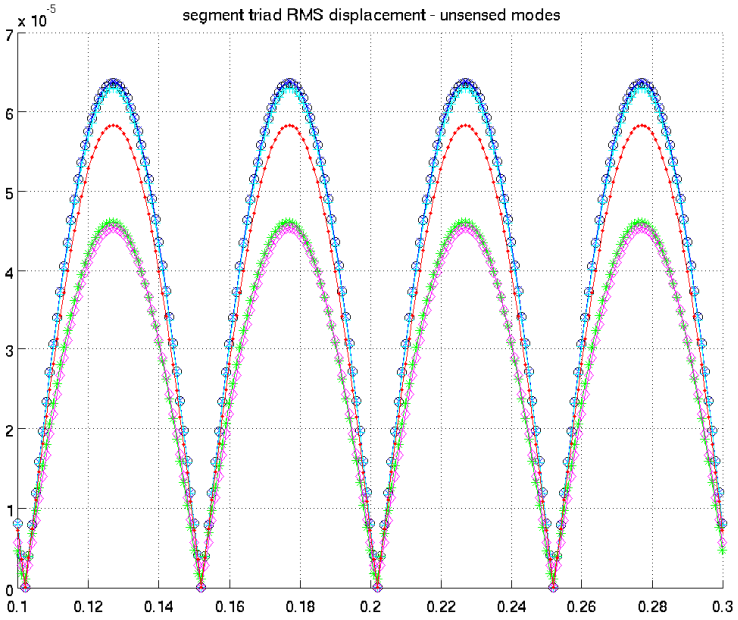

Figure 180. 10 Hz Sinusoidal Force RMS Actuator Position Error Unsensed Modes

The following set of figures shows the actuator displacement errors for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator position error in meters.

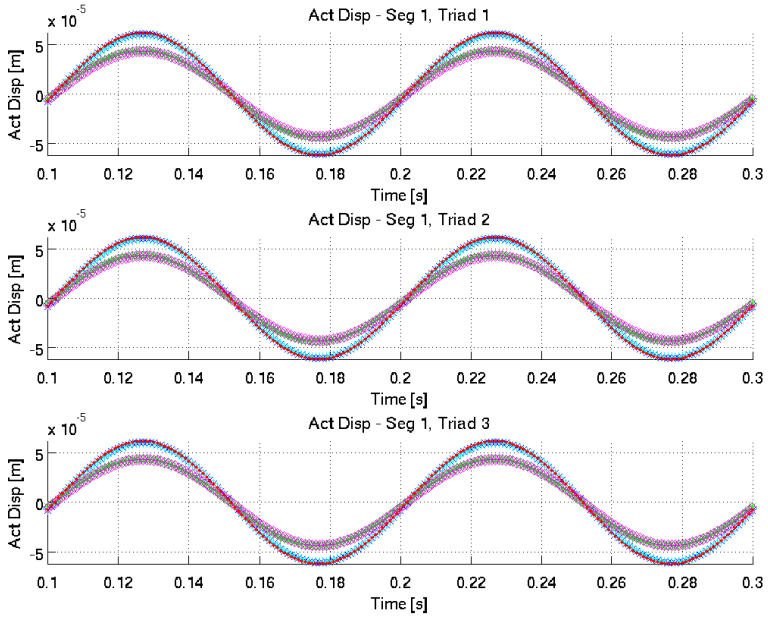

Figure 181. 10 Hz Sinusoidal Force Actuator Position Error, Segment 1

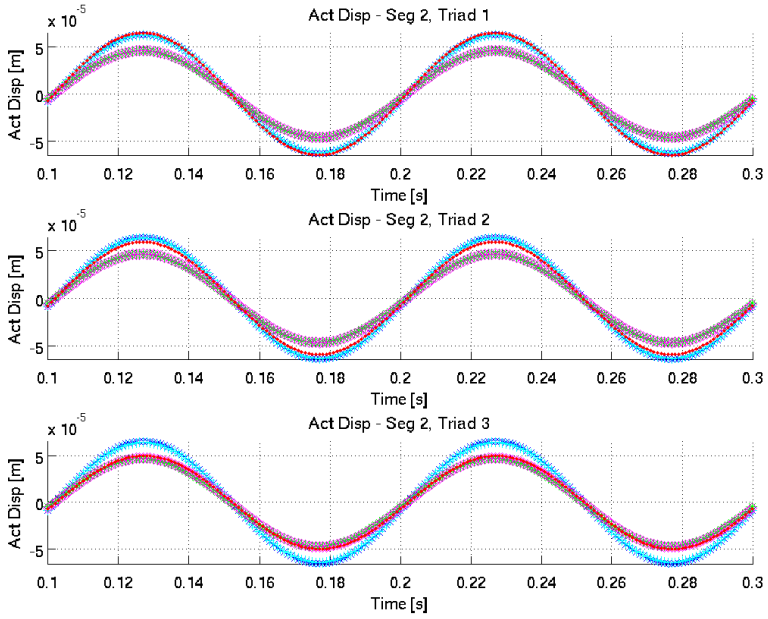

Figure 182. 10 Hz Sinusoidal Force Actuator Position Error, Segment 2

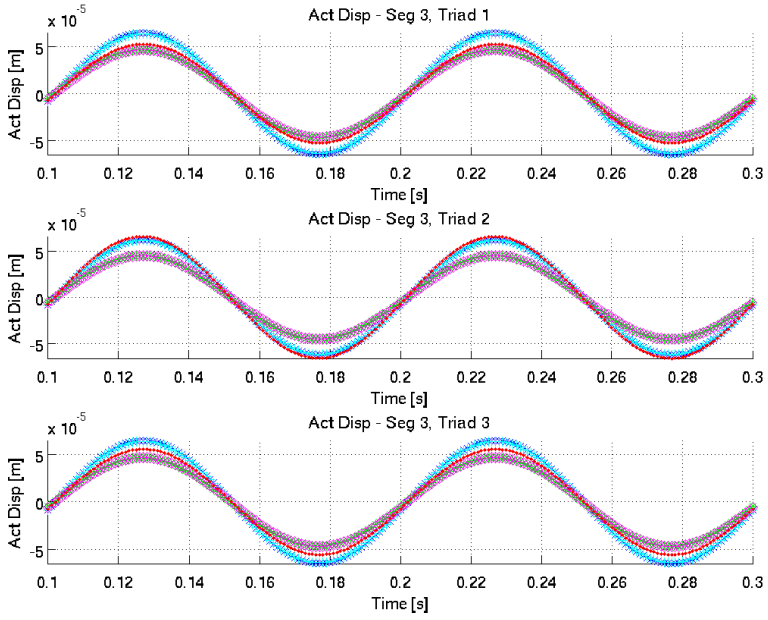

Figure 183. 10 Hz Sinusoidal Force Actuator Position Error, Segment 3

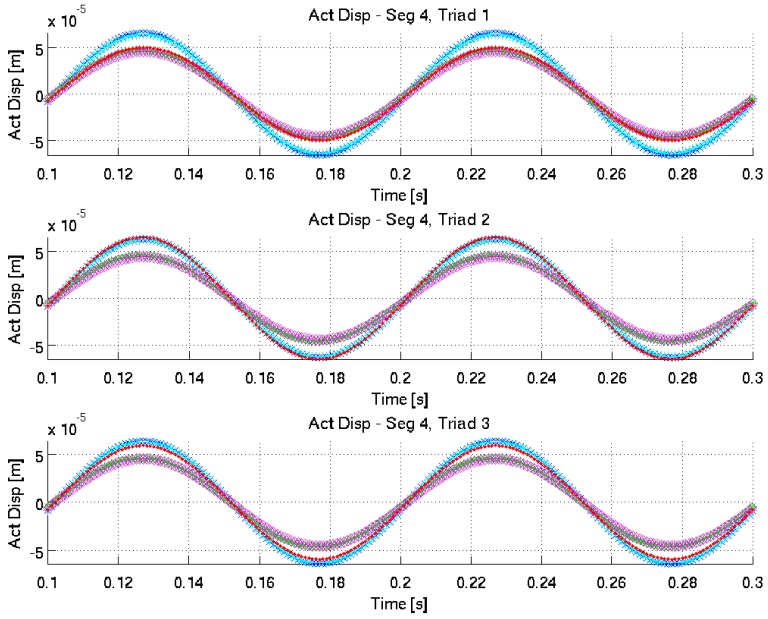

Figure 184. 10 Hz Sinusoidal Force Actuator Position Error, Segment 4

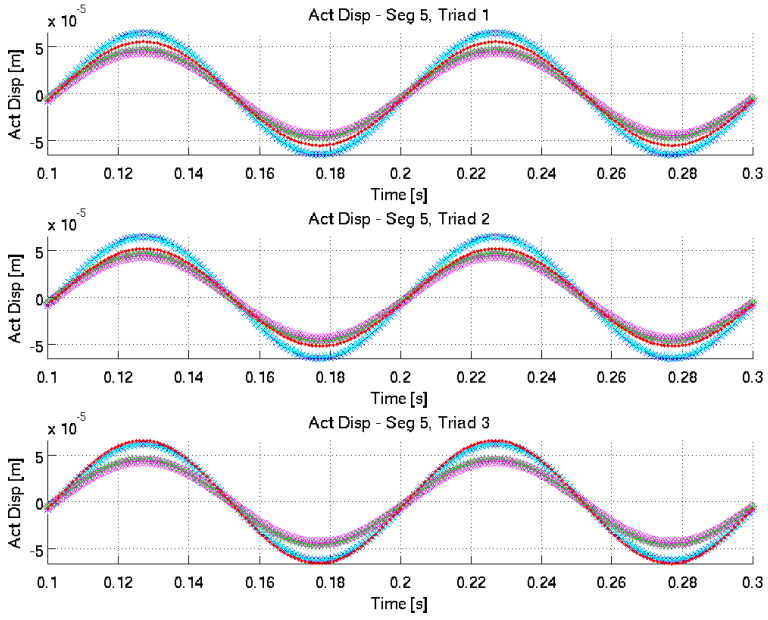

Figure 185. 10 Hz Sinusoidal Force Actuator Position Error, Segment 5

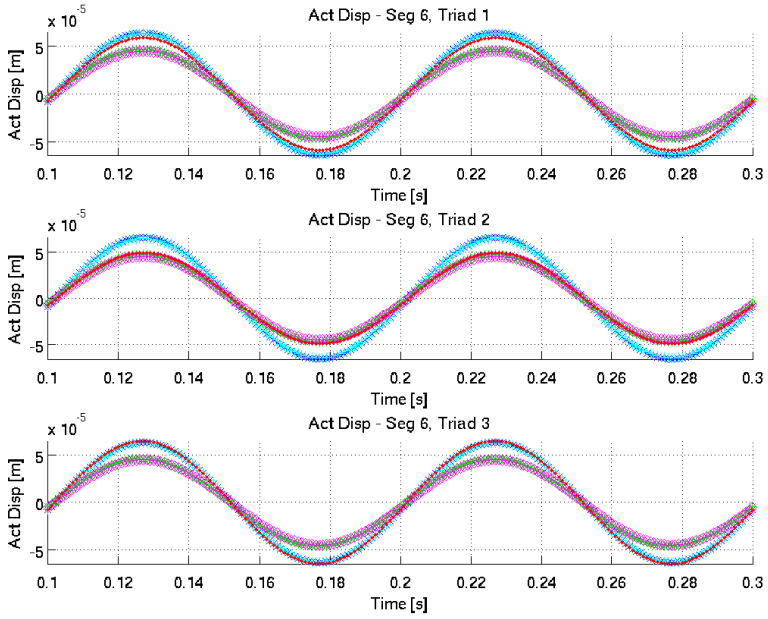

Figure 186. 10 Hz Sinusoidal Force Actuator Position Error, Segment 6

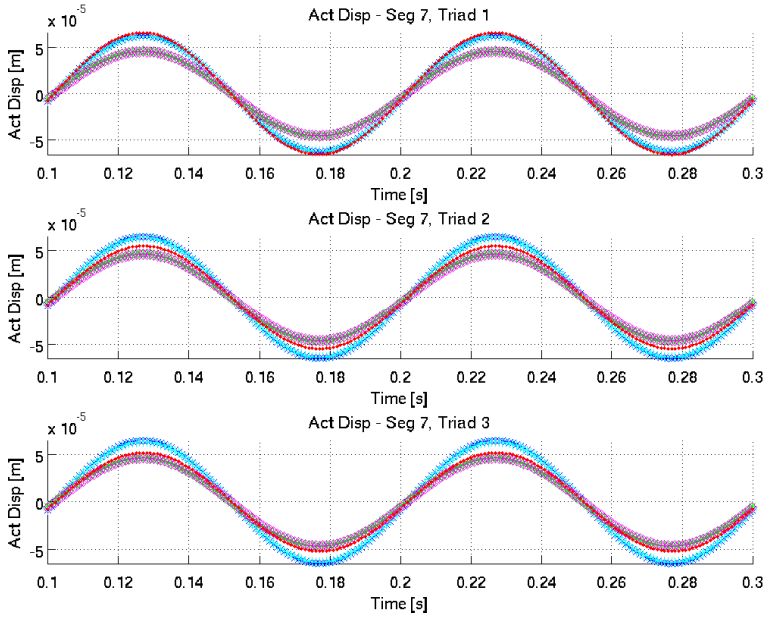

Figure 187. 10 Hz Sinusoidal Force Actuator Position Error, Segment 7

The following set of figures shows the actuator control effort for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator control effort signals.

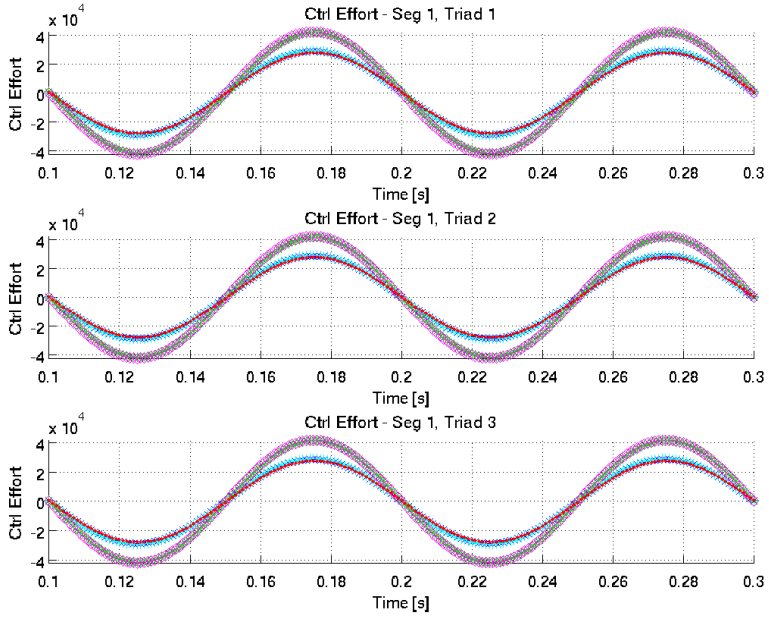

Figure 188. 10 Hz Sinusoidal Force Control Effort, Segment 1

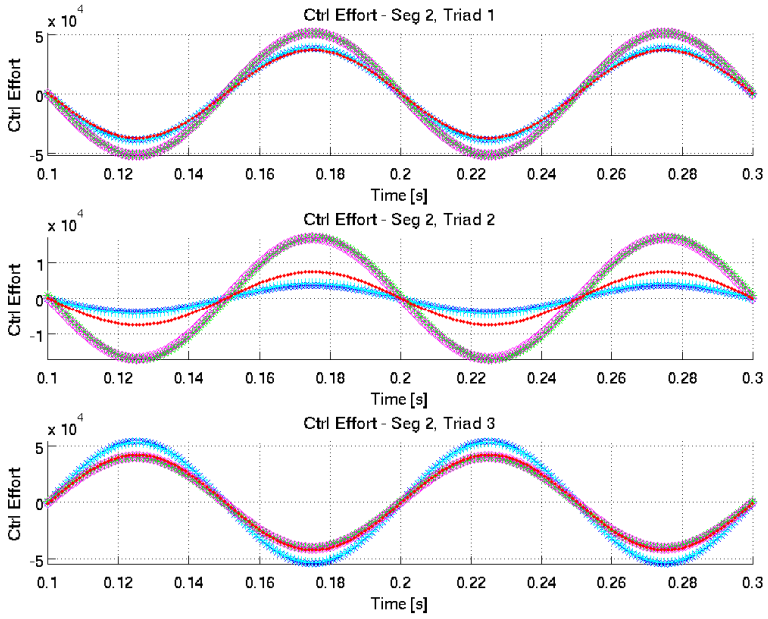

Figure 189. 10 Hz Sinusoidal Force Control Effort, Segment 2

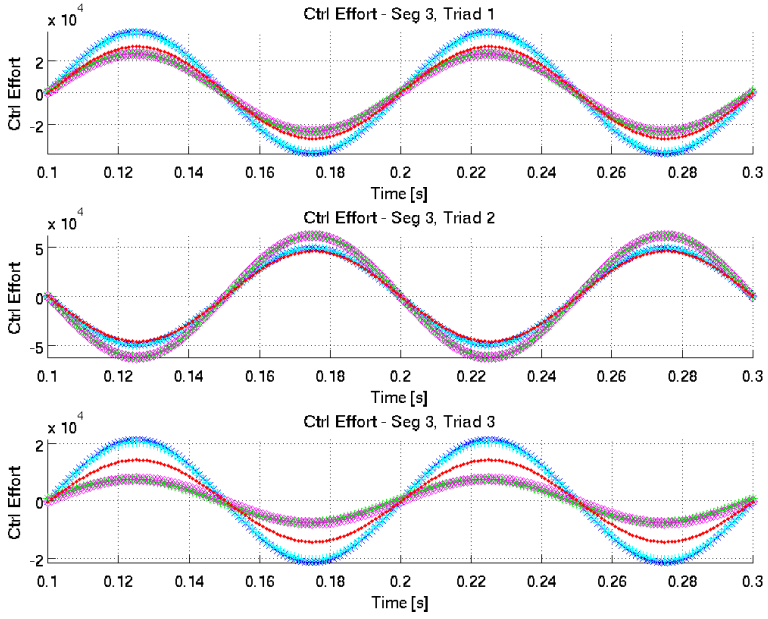

Figure 190. 10 Hz Sinusoidal Force Control Effort, Segment 3

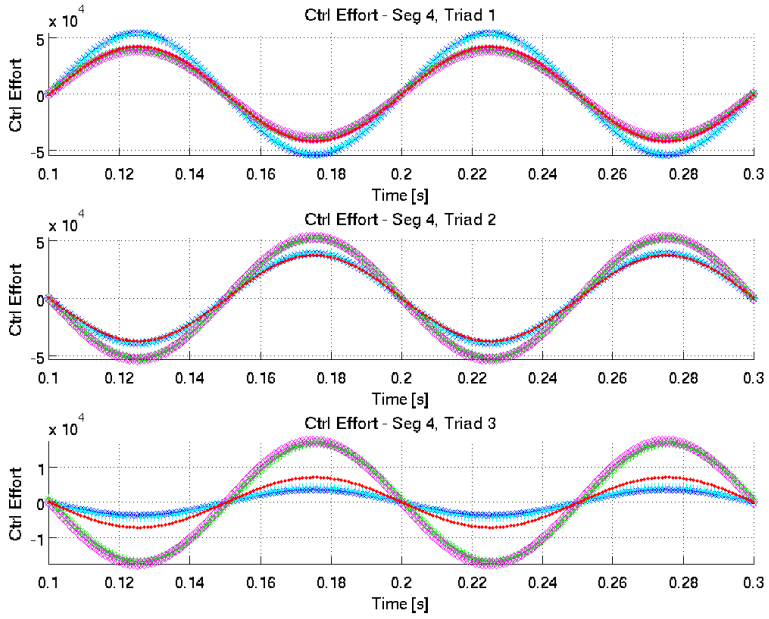

Figure 191. 10 Hz Sinusoidal Force Control Effort, Segment 4

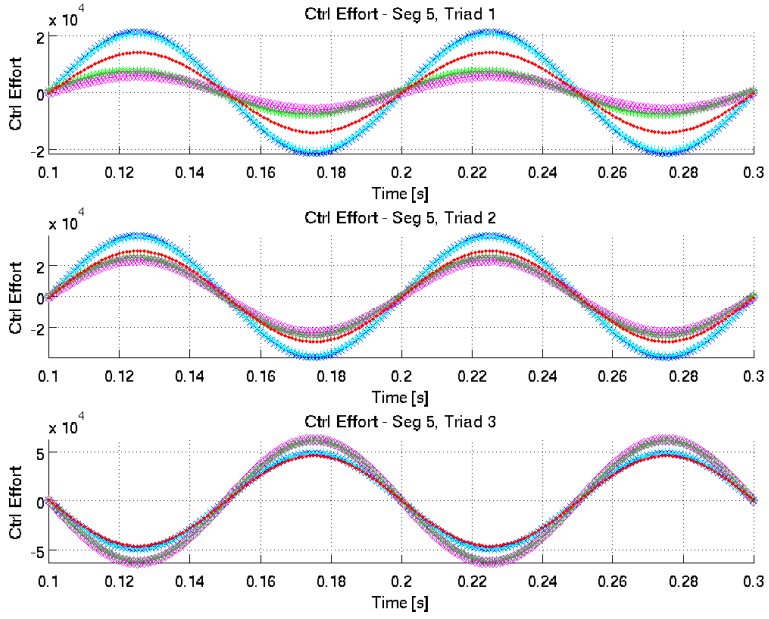

Figure 192. 10 Hz Sinusoidal Force Control Effort, Segment 5

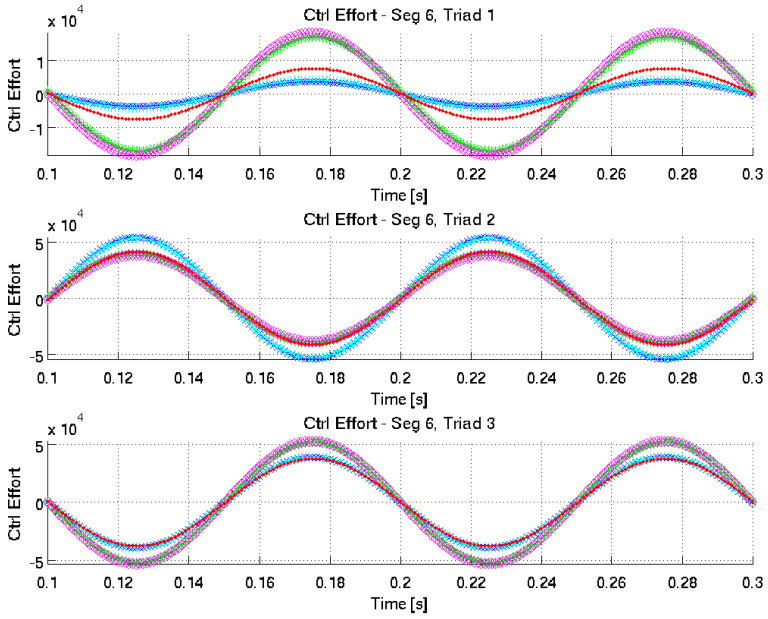

Figure 193. 10 Hz Sinusoidal Force Control Effort, Segment 6

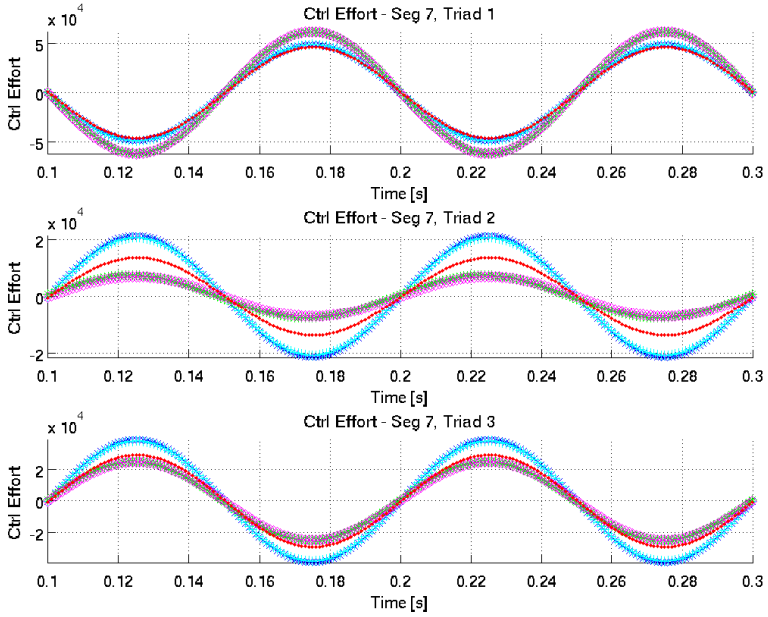

Figure 194. 10 Hz Sinusoidal Force Control Effort, Segment 7

APPENDIX H – Simulation Results for 60 Hz Sinusoidal Force

A 60 Hz sinusoidal force was applied to all nodes in the global Z-direction. The following figures show the spatial RMS of the actuator position errors: overall, sensed modes, and un-sensed modes, respectively. Where the x-axes are time in seconds and the y-axes are spatial RMS of actuator position error in meters.

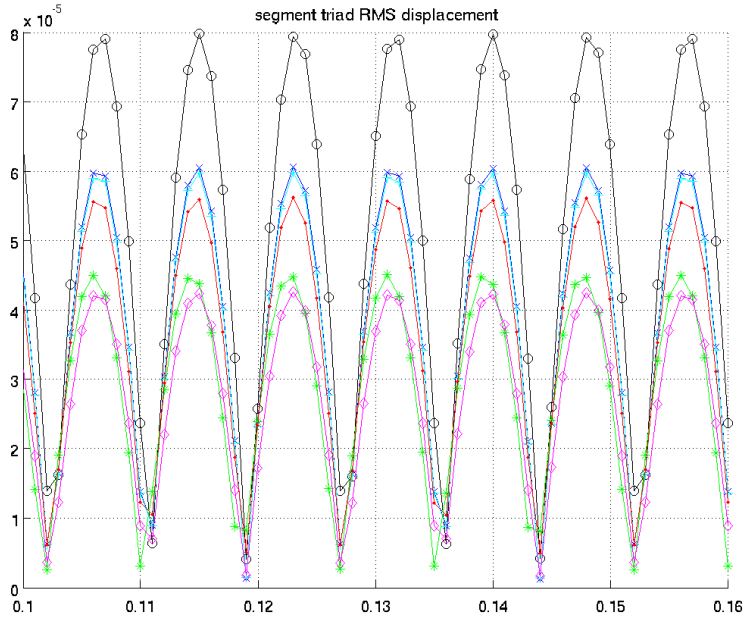

Figure 195. 60 Hz Sinusoidal Force RMS Actuator Position Error

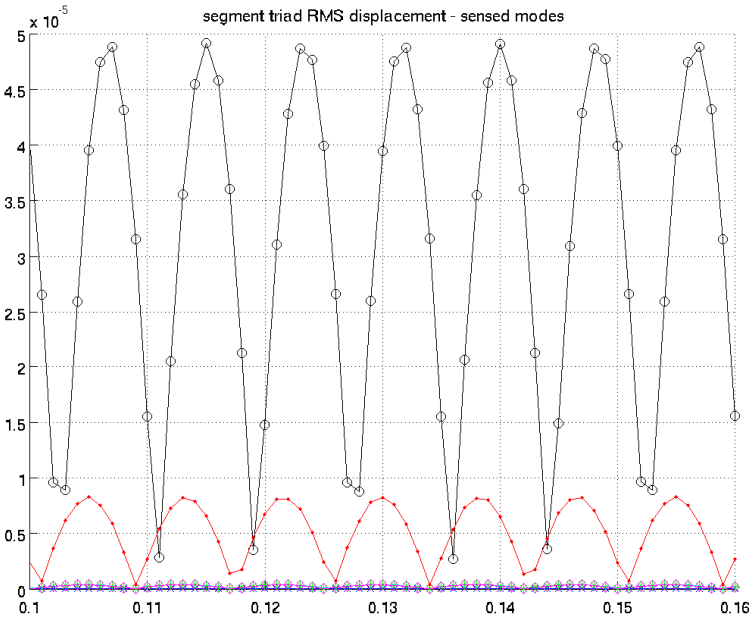

Figure 196. 60 Hz Sinusoidal Force RMS Actuator Position Error Sensed Modes

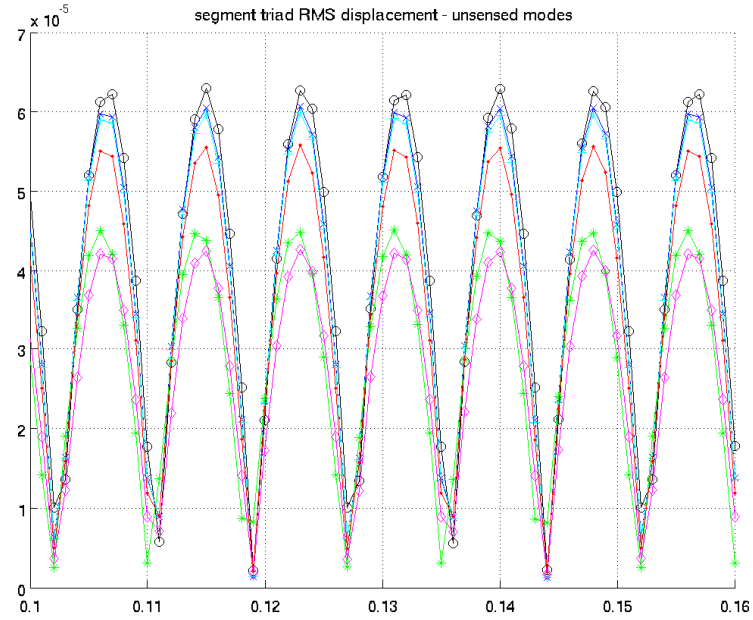

Figure 197. 60 Hz Sinusoidal Force RMS Actuator Position Error Unsensed Modes

The following set of figures shows the actuator displacement errors for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator position error in meters.

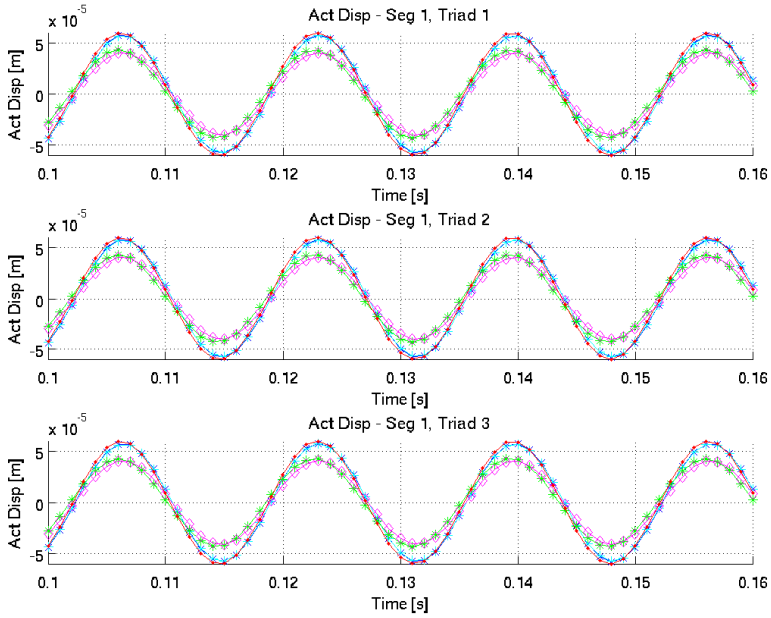

Figure 198. 60 Hz Sinusoidal Force Actuator Position Error, Segment 1

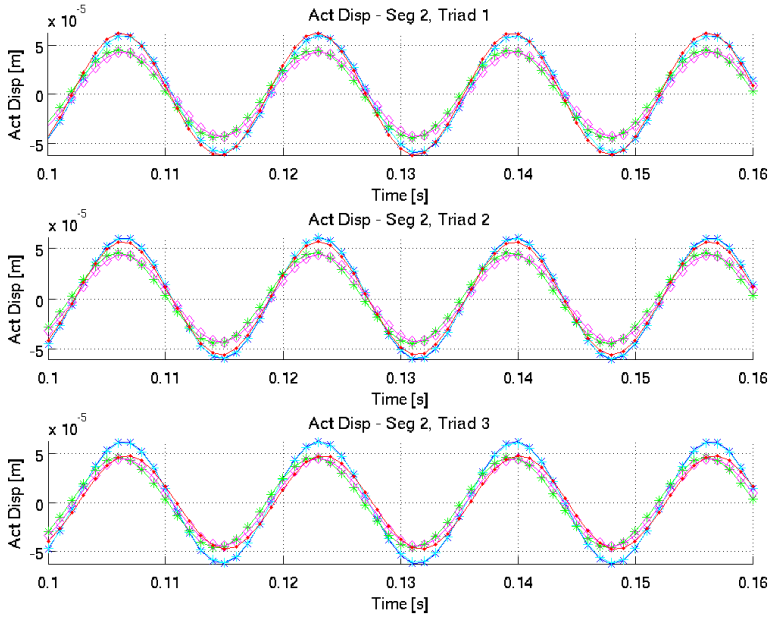

Figure 199. 60 Hz Sinusoidal Force Actuator Position Error, Segment 2

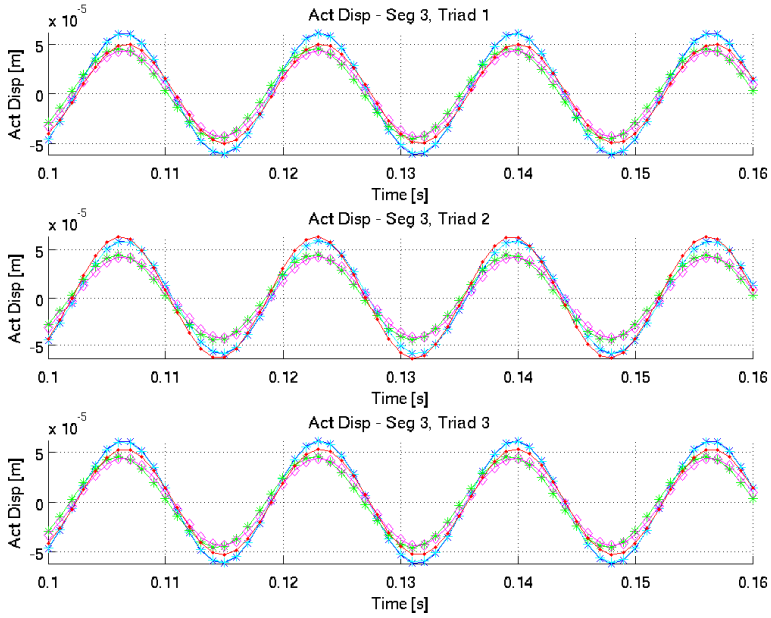

Figure 200. 60 Hz Sinusoidal Force Actuator Position Error, Segment 3

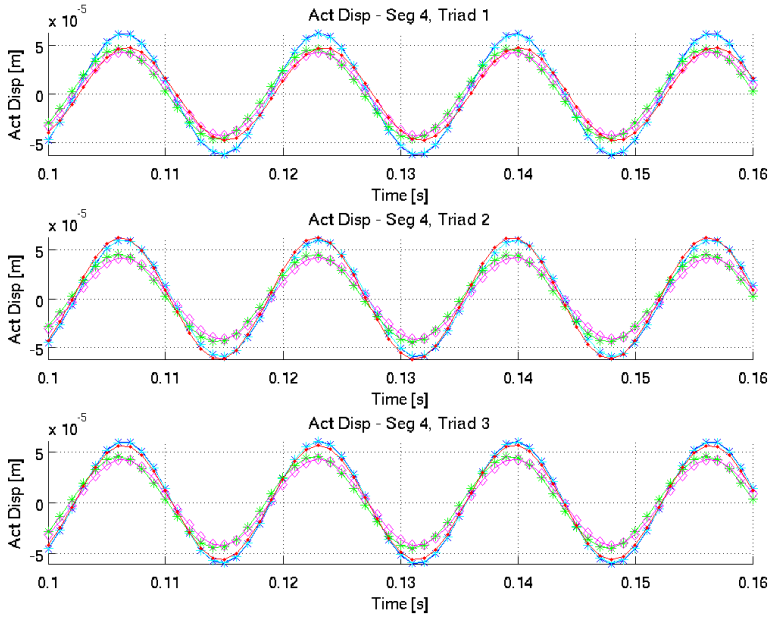

Figure 201. 60 Hz Sinusoidal Force Actuator Position Error, Segment 4

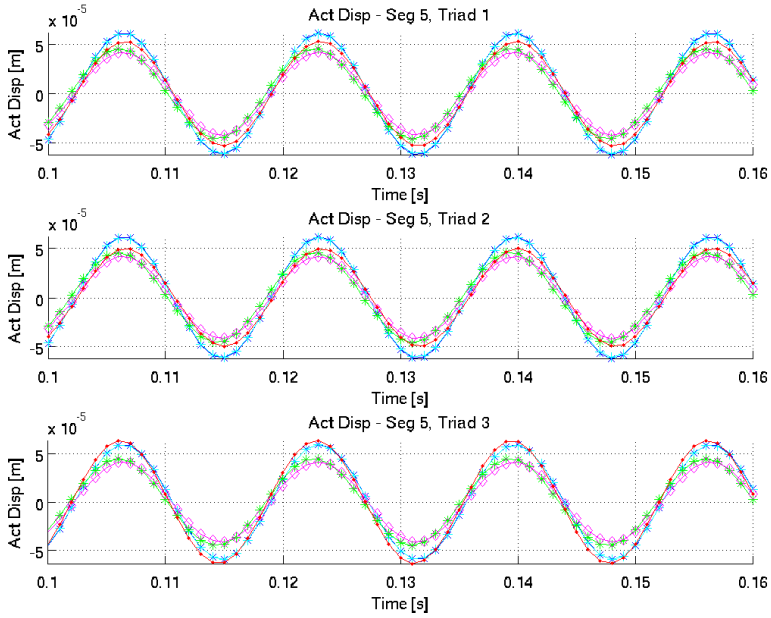

Figure 202. 60 Hz Sinusoidal Force Actuator Position Error, Segment 5

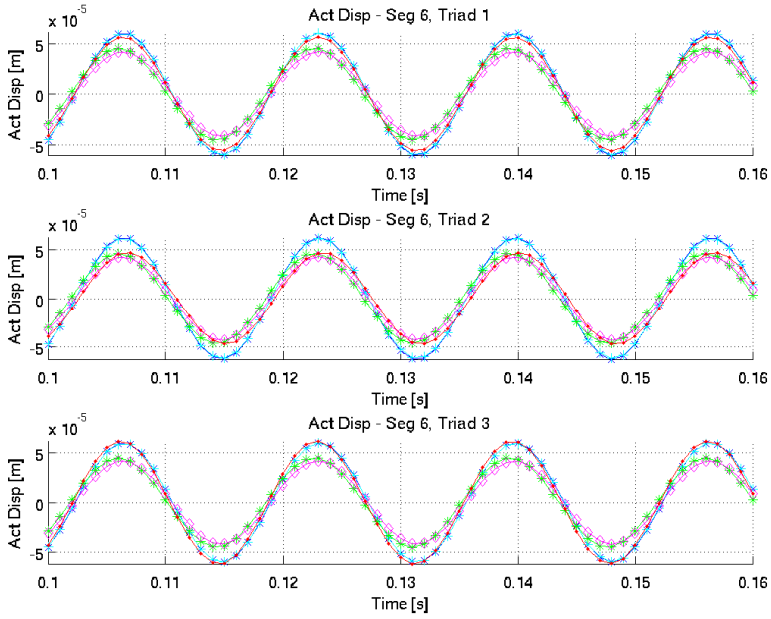

Figure 203. 60 Hz Sinusoidal Force Actuator Position Error, Segment 6

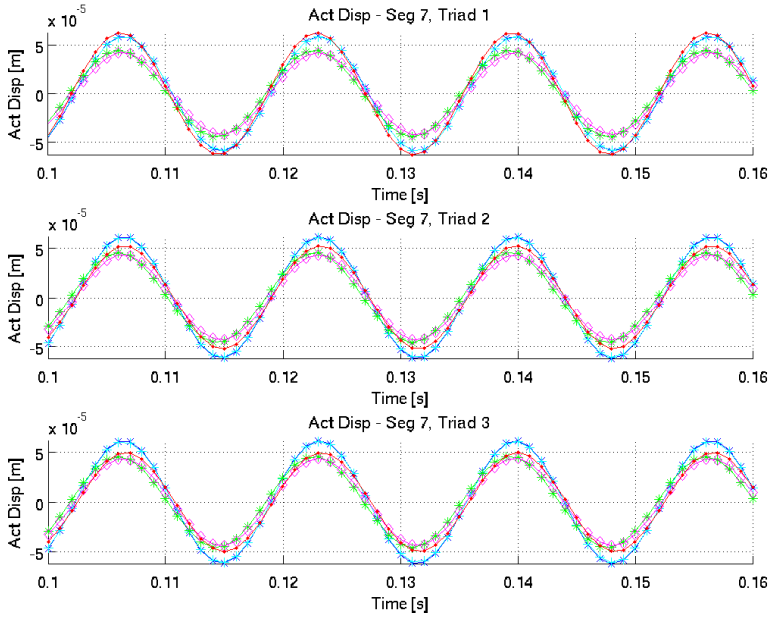

Figure 204. 60 Hz Sinusoidal Force Actuator Position Error, Segment 7

The following set of figures shows the actuator control effort for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator control effort signals.

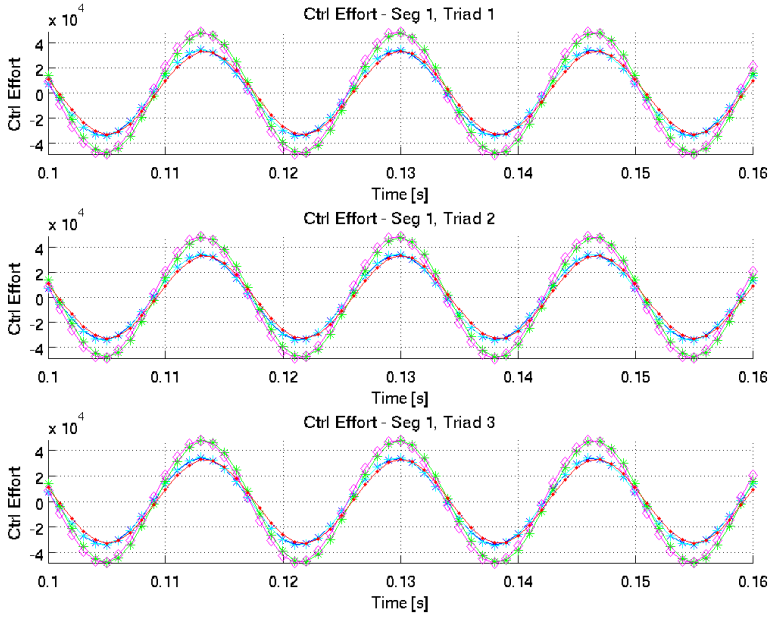

Figure 205. 60 Hz Sinusoidal Force Control Effort, Segment 1

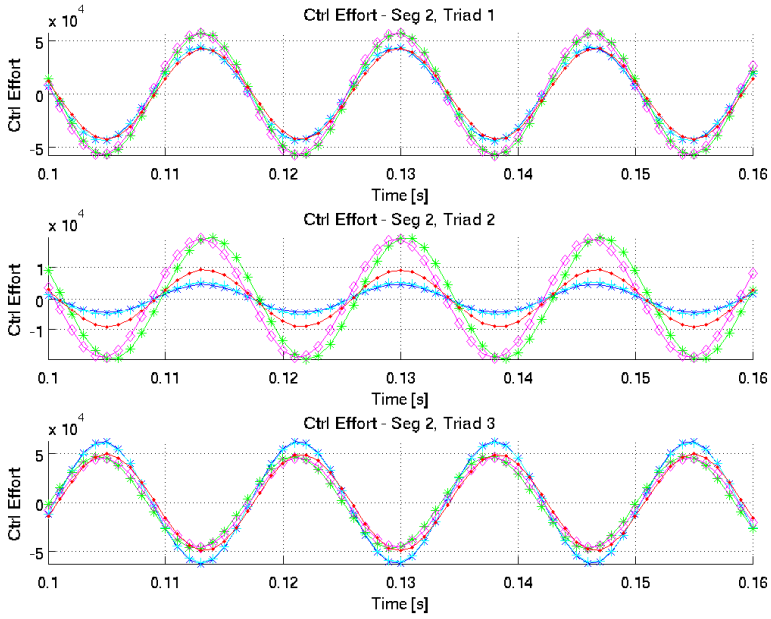

Figure 206. 60 Hz Sinusoidal Force Control Effort, Segment 2

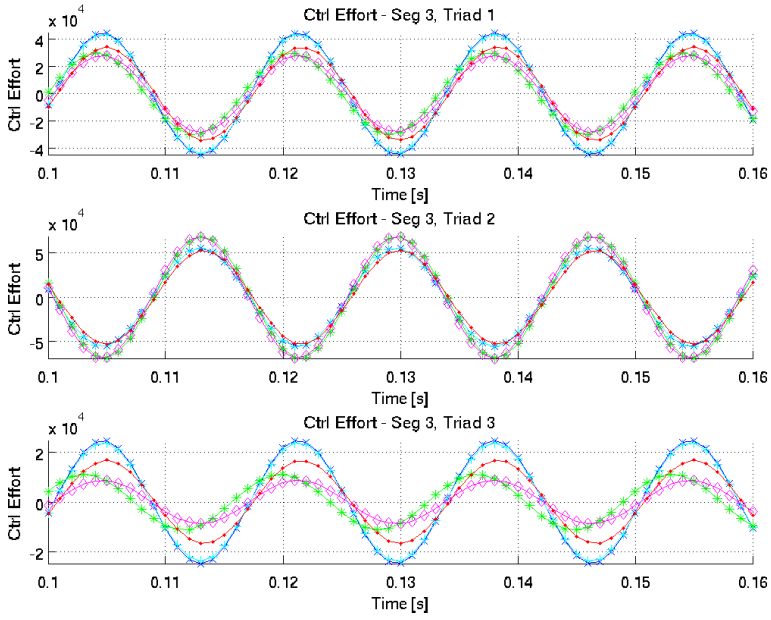

Figure 207. 60 Hz Sinusoidal Force Control Effort, Segment 3

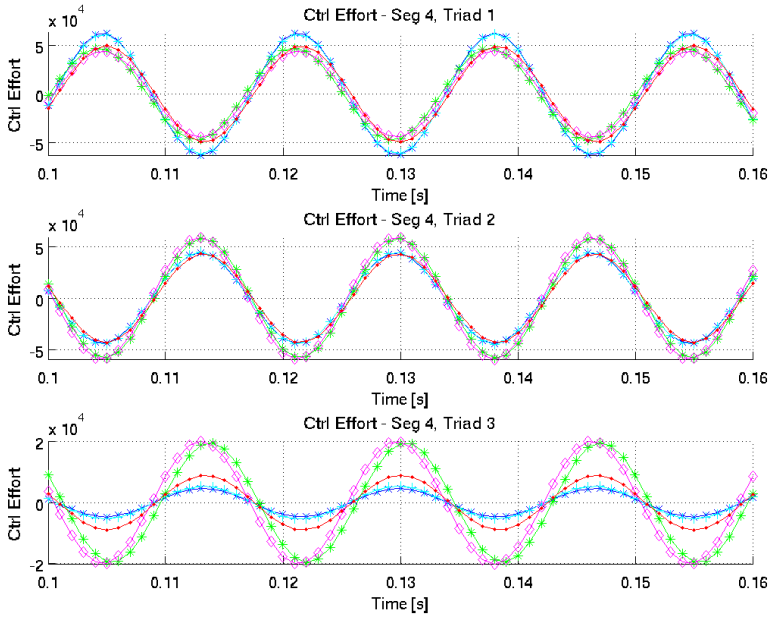

Figure 208. 60 Hz Sinusoidal Force Control Effort, Segment 4

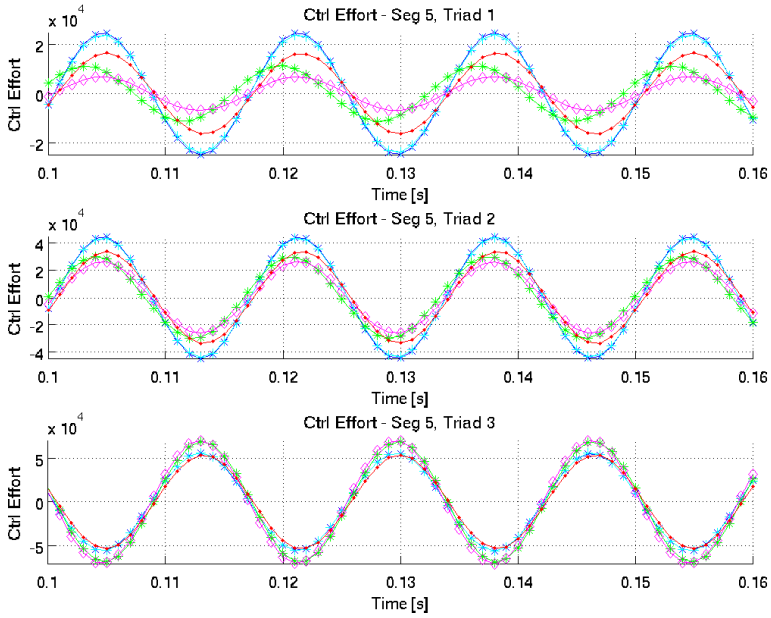

Figure 209. 60 Hz Sinusoidal Force Control Effort, Segment 5

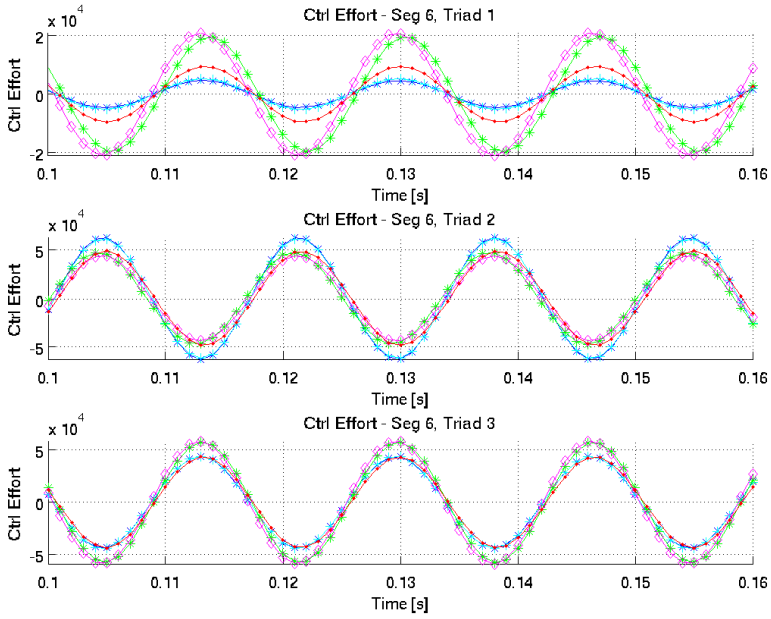

Figure 210. 60 Hz Sinusoidal Force Control Effort, Segment 6

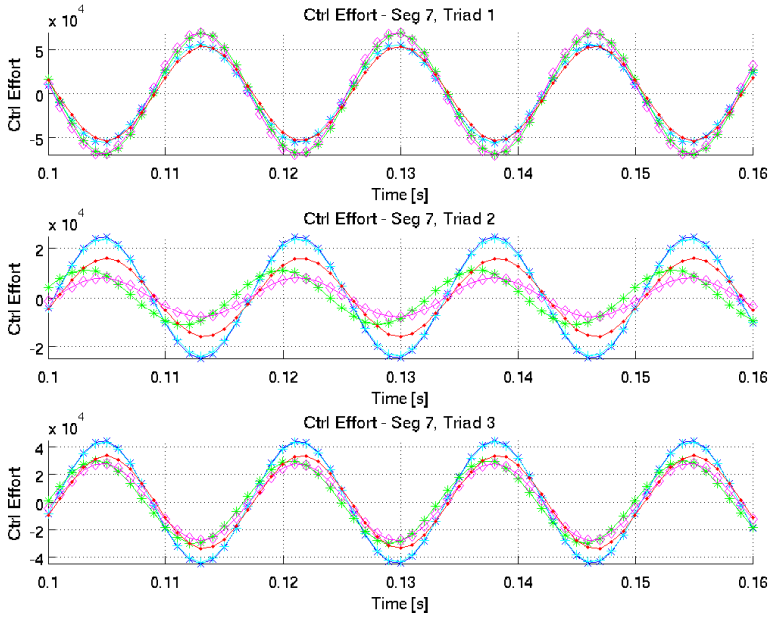

Figure 211. 60 Hz Sinusoidal Force Control Effort, Segment 7

APPENDIX I – Simulation Results for 100 Hz Sinusoidal Force

A 100 Hz sinusoidal force was applied to all nodes in the global Z-direction. The following figures show the spatial RMS of the actuator position errors: overall, sensed modes, and un-sensed modes, respectively. Where the x-axes are time in seconds and the y-axes are spatial RMS of actuator position error in meters.

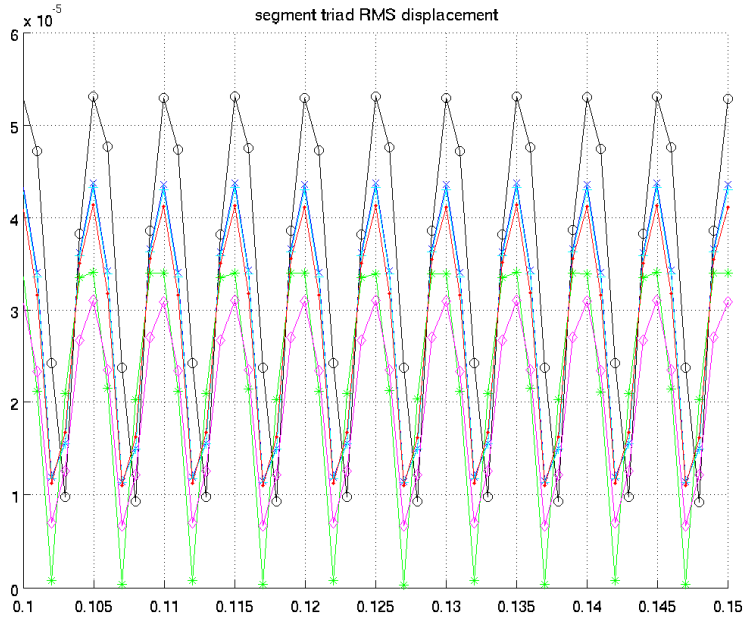

Figure 212. 100 Hz Sinusoidal Force RMS Actuator Position Error

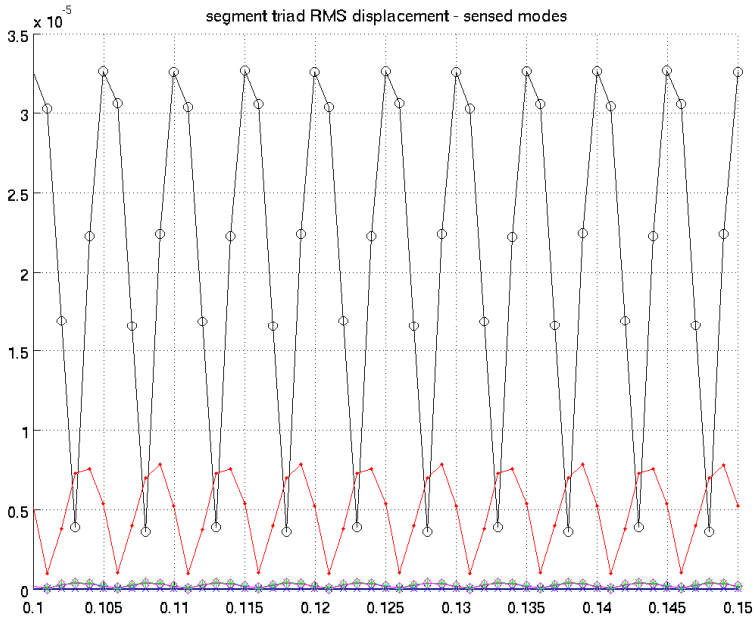

Figure 213. 100 Hz Sinusoidal Force RMS Actuator Position Error Sensed Modes

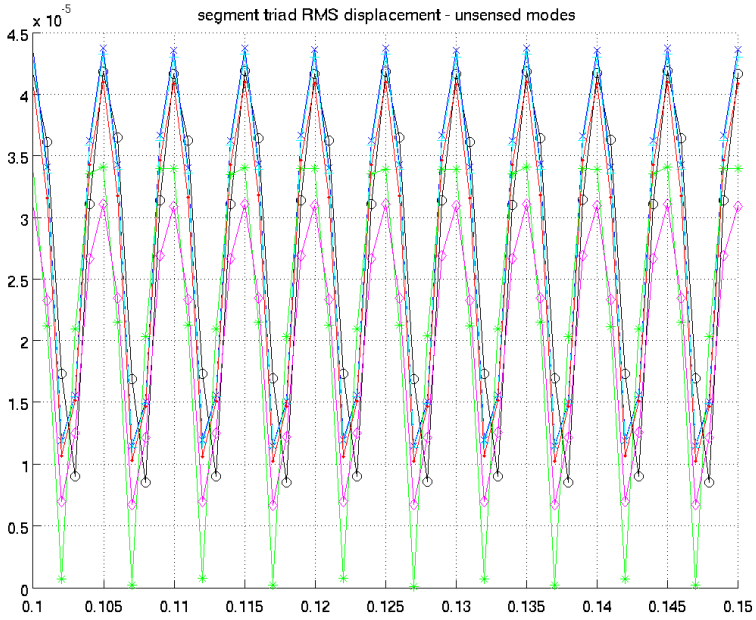

Figure 214. 100 Hz Sinusoidal Force RMS Actuator Position Error Unsensed Modes

The following set of figures shows the actuator displacement errors for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator position error in meters.

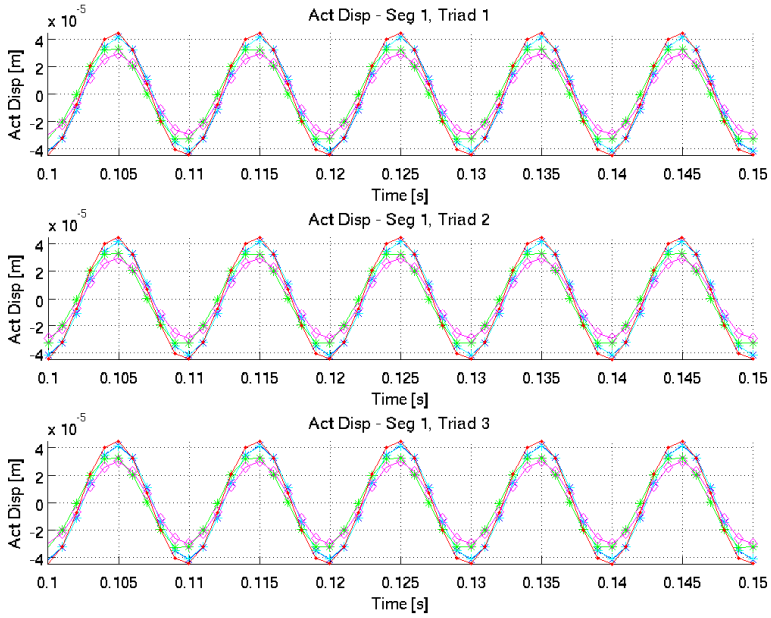

Figure 215. 100 Hz Sinusoidal Force Actuator Position Error, Segment 1

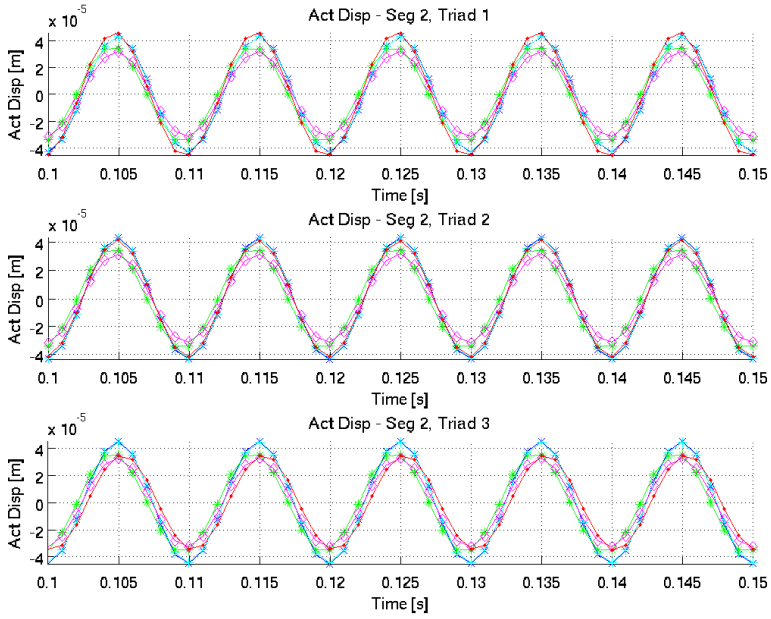

Figure 216. 100 Hz Sinusoidal Force Actuator Position Error, Segment 2

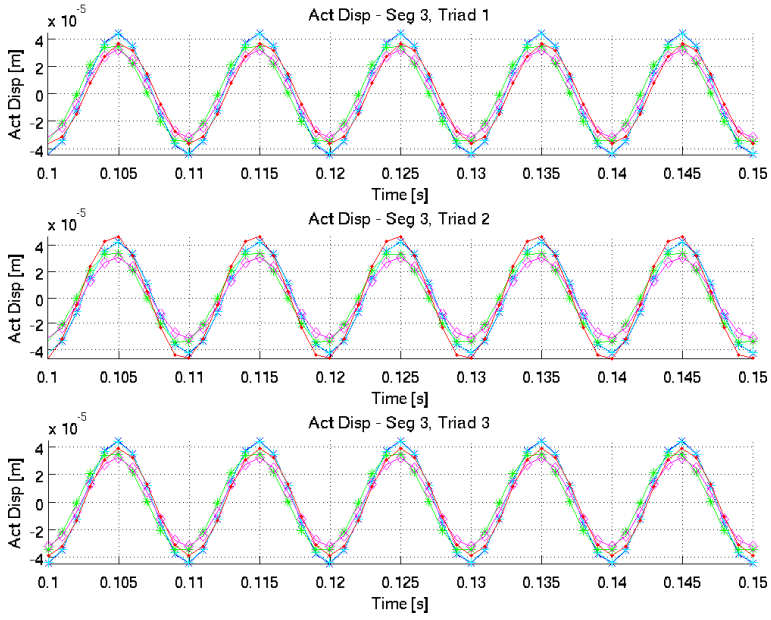

Figure 217. 100 Hz Sinusoidal Force Actuator Position Error, Segment 3

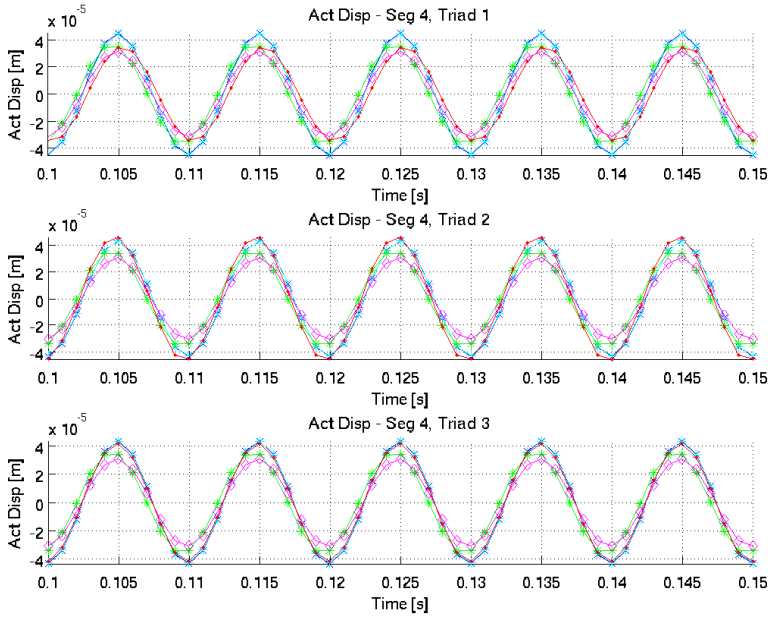

Figure 218. 100 Hz Sinusoidal Force Actuator Position Error, Segment 4

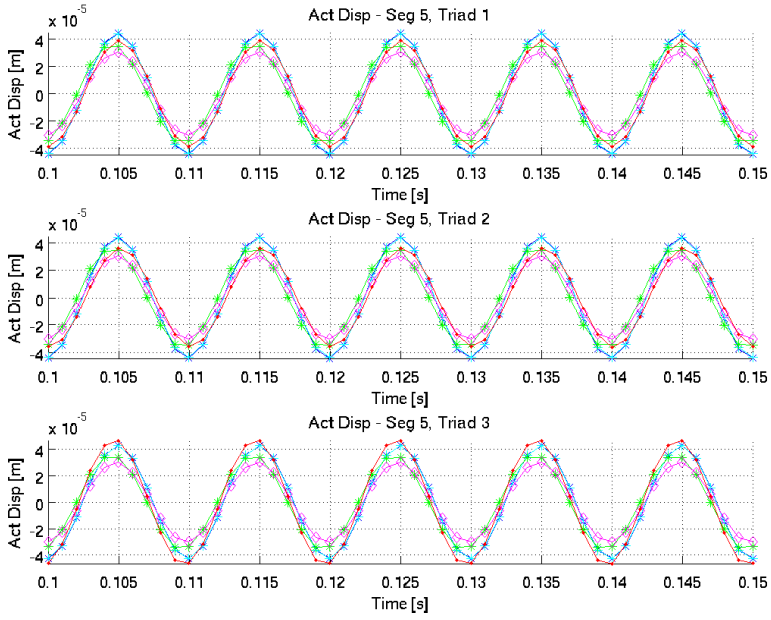

Figure 219. 100 Hz Sinusoidal Force Actuator Position Error, Segment 5

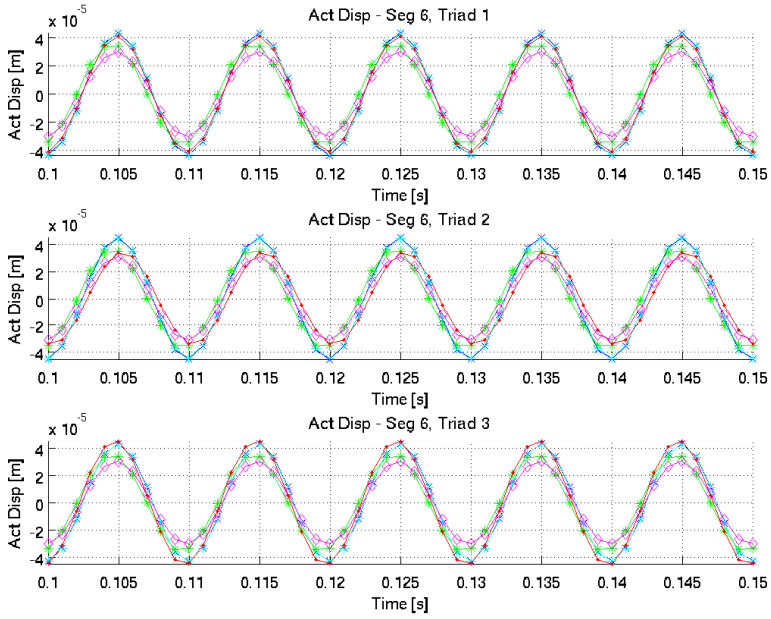

Figure 220. 100 Hz Sinusoidal Force Actuator Position Error, Segment 6

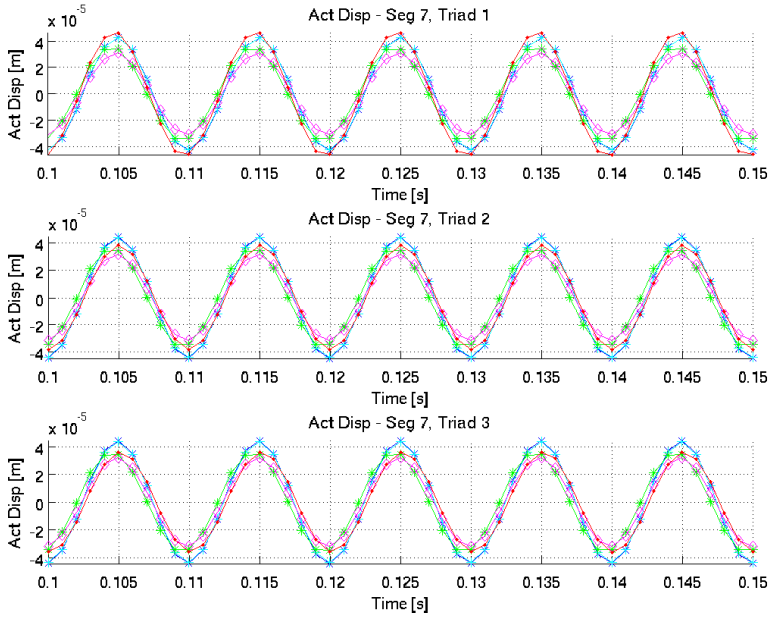

Figure 221. 100 Hz Sinusoidal Force Actuator Position Error, Segment 7

The following set of figures shows the actuator control effort for each of the seven segments. Where the x-axes are time in seconds and the y-axes are actuator control effort signals.

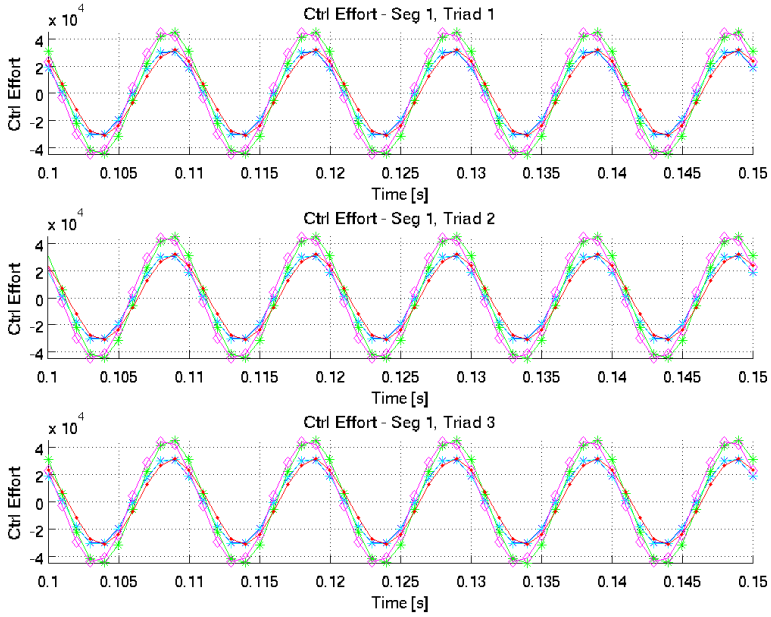

Figure 222. 100 Hz Sinusoidal Force Control Effort, Segment 1

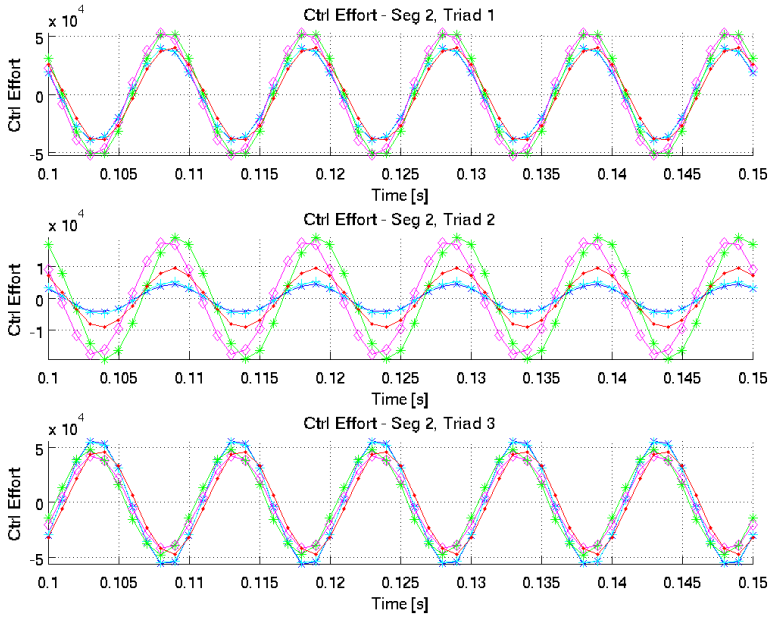

Figure 223. 100 Hz Sinusoidal Force Control Effort, Segment 2

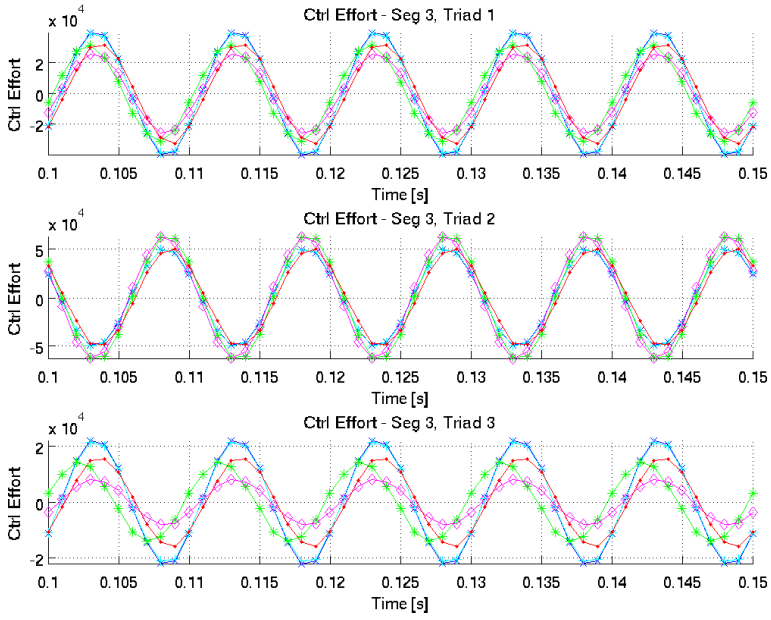

Figure 224. 100 Hz Sinusoidal Force Control Effort, Segment 3

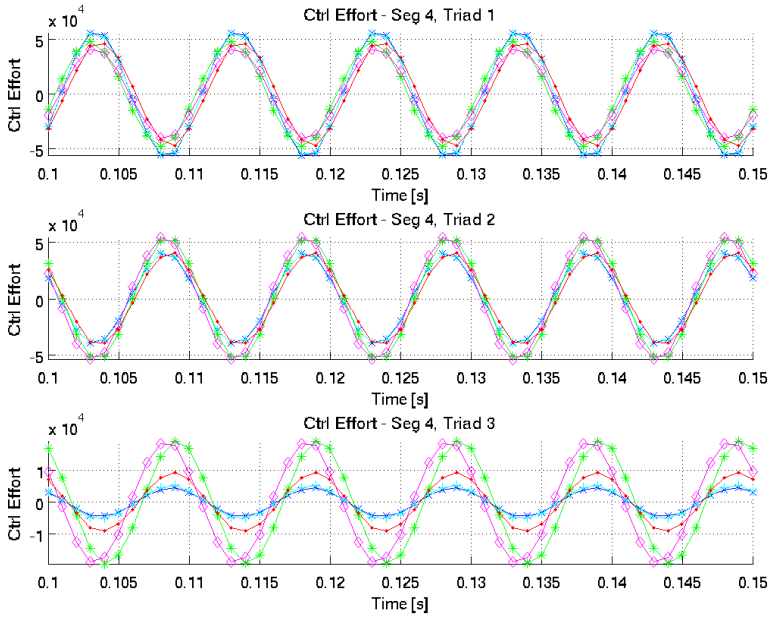

Figure 225. 100 Hz Sinusoidal Force Control Effort, Segment 4

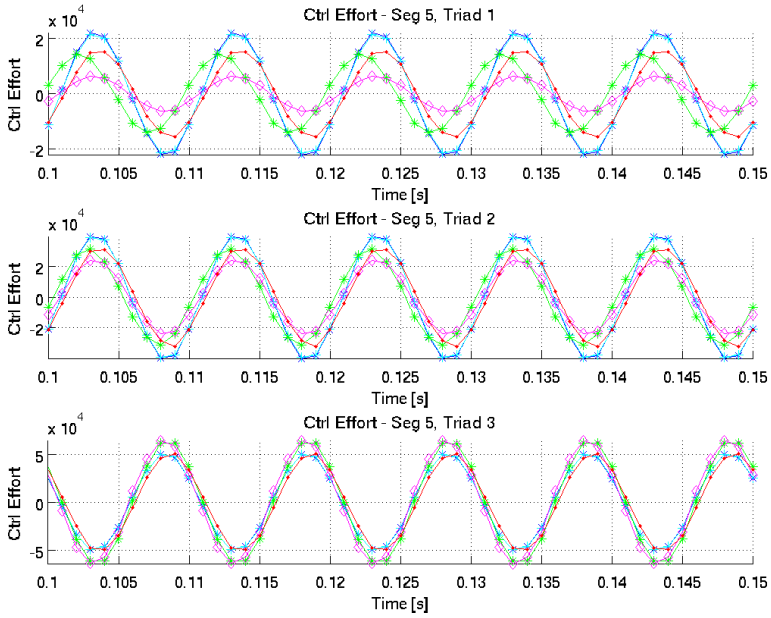

Figure 226. 100 Hz Sinusoidal Force Control Effort, Segment 5

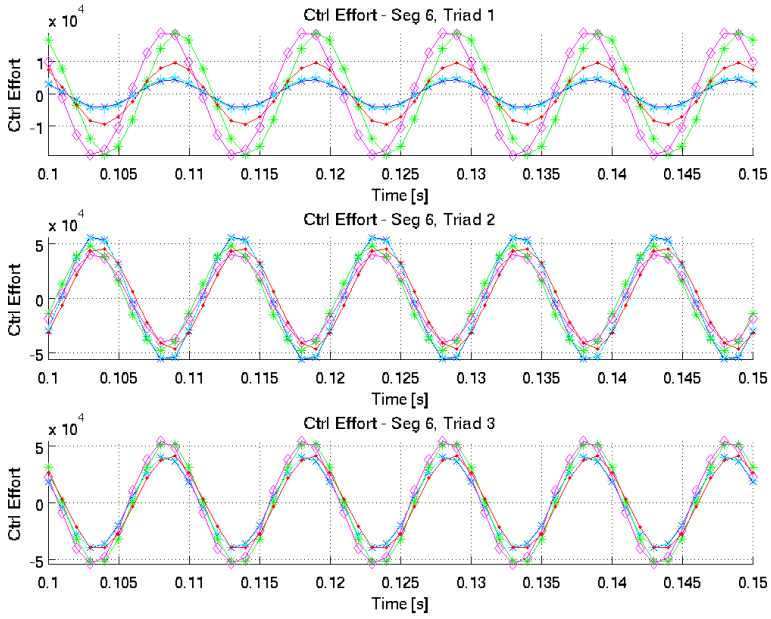

Figure 227. 100 Hz Sinusoidal Force Control Effort, Segment 6

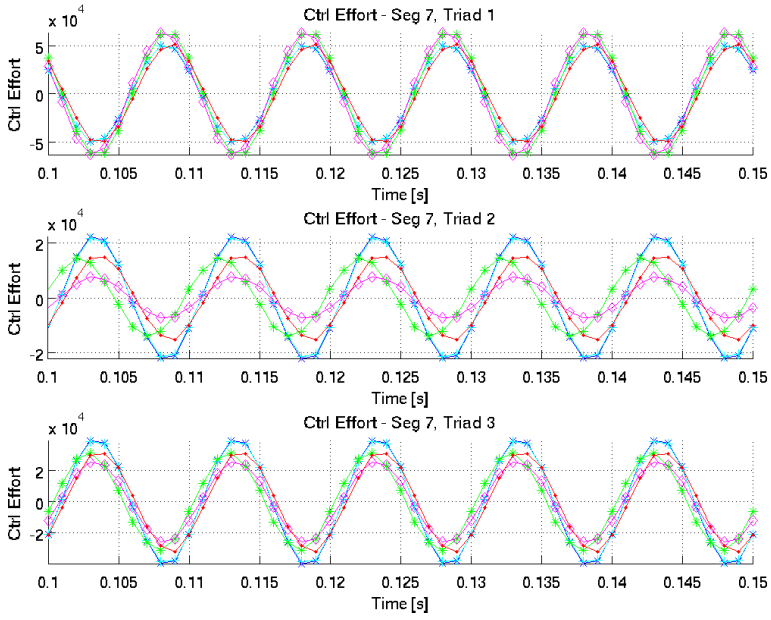

Figure 228. 100 Hz Sinusoidal Force Control Effort, Segment 7

BIBLIOGRAPHY

[1] Thirty Meter Telescope Project, Online: www.tmt.org (2008)

[2] G. Angeli, M. Cho, M. Whorton, "Active Optics and Control Architecture for a Giant Segmented Mirror Telescope", Proc. SPIE, Vol. 4840, pp. 129-139, (2003)

[3] E. Brunetto, P. Dierickx, R. Gilmozzi, M. Louarn, F. Koch, L. Noethe, C. Verinaud, N. Yaitskova, "Progress of ESO's 100-m OWL Optical Telescope Design" Proc. SPIE, Vol. 5382, pp. 159-168, (2004)

[4] D. MacMynowski, P. Thompson, M. Sirota, "Analysis of TMT Primary Mirror Control-Structure Interaction", Proc. SPIE, Vol. 7017, No 41 pp. (2008)

[5]. R. Minor, A. Arthur, G. Gabor, H. Jackson, R. Jared, T. Mast, B. Schaefer, "Displacement sensors for the primary mirror of the W. M. Keck telescope" Proc. SPIE, Vol. 1236, 1009 (1990)

[6]. D. Roziere, S. Buous, A. Courteville, "Position sensors for segmented mirror", Proc. SPIE, Vol. 5495, 660 (2004)

[7] R. D'Andrea, G. Dullerud, "Distributed Control Design for Spatially Interconnected Systems", IEEE Trans. on Automatic Control, Vol. 48, No. 9, pp. 1478- 1495, (2003)

[8] J. Llacer, R.C. Jared, J.M. Fuertes, "Analysis of the W.M. Keck telescope primary mirror control loop" Proc. SPIE, Vol. 1236, Segmented Mirror Control II, LBL-2759, (1990)

[9] J. Brink, A. Charles, C. Hettlage, T. Husser, A. Koeslag, E. Romero-Colmenero, "The SALT observation control system", Proc. SPIE, Vol. 7019, 70190N (2008)

[10]. P. Palunas, J. Fowler, J. Booth, G. Damm, G. Ames, "Control of the Hobby-Eberly Telescope primary mirror array with the segment alignment maintenance system", Proc. SPIE, Vol. 5496, pp. 659-666 (2004)

[11]. S. Padin, W. Davison, “Model of image degradation due to wind buffeting on an extremely large telescope”, Applied Optics, Vol. 43, Issue 3, pp. 592-600, (2004)

[12]. D. MacMynowski, G. Angeli, K. Vogiatzis, J. Fitzsimmons, S. Padin, “Parametric modeling and control of telescope wind-induced vibration”, Proc. SPIE, Vol. 5497, pp. 266-277 (2004)

[13] D. MacMynowski, C. Blaurock, G. Angeli, “Initial Control Results for the Thirty Meter Telescope”, AIAA Guidance, Navigation, and Control Conference, AIAA 2005-6075, San Francisco, CA, (2005)

[14]. M. Johns, R. Angel, S. Shectman, R. Bernstein, D. Fabricant, P. McCarthy, M. Phillips, “Status of the Giant Magellan Telescope (GMT) Project”, Proc. SPIE, Vol. 5489, 441 (2004)

[15]. M. Dimmler, T. Erm, B. Bauvir, B. Sedghi, H. Bonnet, M. Müller, A. Wallander, “E-ELT Primary Mirror Control System” Proc. SPIE, Vol. 7012 (2008)

[16] J. Kulkarni, R. D’Andrea, B. Brandl, “Application of distributed control techniques to the adaptive secondary mirror of Cornell’s Large Atacama Telescope”, Proc. SPIE, Vol. 4839, 750 (2003)

[17] M. Morales, M. Mormorani, H. Boussalis, “Design, Simulation & Control of a Segmented Reflector Test-bed”, Proc. Mediterranean Conference on Control and Automation, MED99, (1999)

[18]. F. Gonté, C. Dupuy, C. Frank, C. Araujo, R. Brast, R. Frahm, R. Karban, L, Andolfato, R. Esteves, M. Nylund, B. Sedghi, G. Fischer, L. Noethe, F. Derie, “The First Active Segmented Mirror at ESO”, The Messenger, Vol. 128, pp 23

[19] J. Dunn, S. Roberts, D. Kerley, J. Fitzsimmons, J. Pazder, G. Herriot, M. Smith, “TMT/VLOT Integrated Modeling”, Proc. SPIE, Vol. 5579, pp. 323-332, (2004)

[20] S. Roberts, S. Sun, D. Kerley, “Optical Performance Analysis and Optimization of Large Telescope Structural Designs”, Proc. SPIE, Vol. 5867, pp. 200-211, (2005)

[21] D. Kerley, E. Park, J. Dunn, “Distributed Modeling and Decentralized H_∞ Control of a segmented Telescope Mirror”, *Proc. of IMECE 44145*, (2007)

[22] A. Preumont, R. Bastaits, G. Rodrigues, “Active Optics for Large Segmented Mirrors: Scale Effects” IV ECCOMAS Thematic Conference on Smart Structures and Materials, (2009)

[23] S. C. West, S. Callahan, F. H. Chaffee, W. Davison, S. DeRigne, D. Fabricant, C. B. Foltz, J. M. Hill, R. H. Nagel, A. Poyner, and J. T. Williams, “Toward first light for the 6.5-m MMT Telescope,” Proc. SPIE, Optical Telescopes of Today and Tomorrow 2871, (1996)

[24]. D. Buckley P. Charles, K. Nordsiek, D. O’Donghue, “The South African Large Telescope Project”, Proc. IAU, No. 232 (2005)

[25]. C. Baffes, T. Mast, J. Nelson, E. Ponslet, V. Stephens, L. Stepp, E. Williams, “Primary Mirror Segmentation Studies for the Thirty Meter Telescope”, Proc. SPIE, Vol. 7018, (2008)

[26]. M. Troy, G. Chanan, E. Sirko, E. Leffert, “Residual Misalignments of the Keck Telescope Primary Mirror Segments: Classification of Modes and Implications for Adaptive Optics”, Proc. SPIE, Vol. 3352, 307 (1998)

[27] T. Mast, G. Chanan, J. Nelson, R. Minor, R. Jared, “Edge Sensor Design for the TMT”, Proc. SPIE, Vol. 6267, 62672S (2006)

[28]. D. Logan, “A First Course in the Finite Element Method”, 3rd edition, University of Wisconsin, Platteville (2002)

[29] E. Ponslet, D. Blanco, M. Cho, T. Mast, J. Nelson, RJ Ponchione, M. Sirota, V. Stephens, L. Stepp, A. Tubb, E. Williams, “Development of the Primary Mirror Segment Support Assemblies for the Thirty Meter Telescope”, Proc. SPIE, Vol. 6273, 627319 (2006)

[30] K. Glover, “All Optimal Hankel Norm Approximation of Linear Multivariable Systems, and Their Lμ – error Bounds”, Int. J. Control, Vol. 39, No. 6, pp 1145-1193 (1984)

[31] M. Safonov, R. Chiang, “A Schur Method for Balanced Model Reduction”, IEEE Trans. on Automatic Control, Vol. 34, No. 7, pp. 729-733, (1989)

[32] B. Recht, “Distributed Control of Systems Over Discrete Groups” IEEE Trans. on Automatic Control, Vol. 49, No. 9, pp. 1446- 1452, (2004)

[33] R. Chandra, J Fowler, R D’Andrea, “Control of interconnected systems of finite spatial extent”, Proc. of 41st IEEE Conference on Decision and Control, Vol. 1, pp. 238- 239, (2002)

[34]. M. Safonov, D. Limbeer, “Simplifying the H-infinity theory via looping shifting”, Proc. IEEE Conf. Decision Control, (1998)

[35] R. Yu, S. Roberts, I. Sharf, “Model Order Reduction of Structural Dynamics of a Very Large Optical Telescope”, Proc. SPIE, Vol. 5497, 611 (2004)

[36]. G. Chanan, D. MacMartin, J. Nelson, T. Mast, “Control and alignment of segmented-mirror telescopes: matrices, modes, and error propagation”, Applied Optics, Vol. 43, Issue 6, pp. 1223-1232 (2004)

[37]. D. MacMartin, G. Chanan, “Measurement accuracy in control of segmented-mirror telescopes”, Applied Optics, Vol. 43, Issue 3, pp. 608-615 (2004)

[38] P. Gahinet, P. Apkarian, “A Linear Matrix Inequality Approach to H_∞ Control” International Journal of Robust and Nonlinear Control, Vol. 4, pp. 421-448, (1994)

[39]. L. Shampine, M. Reichelt, "The MATLAB ODE Suite", SIAM Journal on Scientific Computing, Vol. 18, 1997, pp 1-22 (1997)

[40]. Y. Nesterov, A. Nemirovski, “Interior Point Polynomial Methods in Convex Programming: Theory and Applications”, SIAM, Philadelphia (1994)

[41]. A. Nemirovski, P. Gahinet, "The Projective Method for Solving Linear Matrix Inequalities," Proc. Amer. Contr. Conf., Baltimore, Maryland, pp. 840-844 (1994)

Title of Thesis:

Distributed Control of a Segmented Telescope Mirror

Author : Dan Kerley

Signed: