ABSTRACT

Dissertation Title: ON THE POINTING AND JITTER CHARACTERIZATION OF MEMS TWO-AXIS (TIP-TILT) MIRRORS

Clinton Lee Edwards, Ph.D., 2007

Directed By: Professor Christopher C. Davis
Department of Electrical and Computer Engineering

This dissertation presents three first-principles analytic, closed-form models that describe the pointing characteristics of MEMS Two-Axis (Tip-Tilt) Mirrors: (1) a 2D Torque Model, (2) a Micro-mirror Pointing Model (MPM), and (3) a Micro-mirror Jitter Model (MJM).

The 2D Torque Model accounts for all of the fundamental electrodynamics inherent in the operation of a MEMS Two-Axis (Tip-Tilt) Mirror. The 2D Torque Model is utilized in the MPM model and the MJM model and is a function of both axis angles. These three models provide explicit relationships between MEMS mirror physical, electrical, and environmental design parameters, and mirror performance.

The MPM model, consisting of coupled damped harmonic oscillators with the 2D Torque Model as an input, is used to analyze the dynamics of the mirror. This

formulation is imposed by Euler equations and the mirror's rigid structure. A generalized torque function, "G", is presented that utilizes symmetry in the torsional expressions to facilitate software implementation. A methodology is explained for determining the dynamic constants for the mirror as well as an "effective length" which accounts for electric field fringing. Since MEMS fabrication leads to variations in physical properties, the MPM model can be calibrated for a particular mirror to compensate for this variation. Experimental measurements and the MPM model results are in close agreement for steady-state and transient mirror dynamics.

The MJM model was created using the MPM model to address the effects of mirror facet jitter. The MJM model provides an explicit relationship between noise sources and the resulting mirror jitter. It can be used to simulate the effects of mirror jitter as a function of the originating noise sources which are: (1) control voltage fluctuation, (2) platform vibration, (3) Brownian motion noise. A methodology is developed to validate the MJM model. Measurements from the resulting experimental apparatus support the model. Additionally, the experimental apparatus permitted pressure dependent measurements to be made. Mirror jitter was recorded and analyzed for varying pressure and tip-tilt angles. Damping constants (for both axes) were measured. Brownian motion generated jitter was isolated and its variance observed to be pressure invariant as the model predicted.

ON THE POINTING AND JITTER CHARACTERIZATION OF MEMS TWO-AXIS (TIP-TILT) MIRRORS

By

Clinton Lee Edwards

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2007

Advisory Committee:
Professor Christopher C. Davis, Chair
Professor William S. Levine
Professor Andre Tits
Professor Reza Ghodssi
Professor Robert W. Gammon

Dedication

To my wife, Corinne Preston Edwards, and my children Sheely Anne, Xavier Lee, and Preston Spencer, who made it all worthwhile.

To my mother, Patricia Hogan Edwards, who made it possible; and to my father, friend and colleague, Dr. M. Lee Edwards, who made it fun.

Acknowledgements

I would like to acknowledge the support of several individuals who helped in this research effort. I wish to thank Dr. Christopher Davis for his continuing guidance over the past six years as my Ph.D. advisor. I must thank Dr. Bradley Boone for being my on-site advisor for the past three years while working at the Johns Hopkins University, Applied Physics Laboratory (JHU-APL). Dr. Boone correctly identified this research area as one rich with potential. Dr. Boone has also endured editing many of my publication drafts that we have co-authored. Dr. William Levine also was extremely helpful with the stochastic process aspects of this research and his influence can be seen throughout this text as well as our other publications. The experimental measurements could not have been done without the support of my group supervisor, Dr. Isaac Bankman. I want to recognize JHU-APL and the Janney Fellowship Committee as well as the Space Department's RF Engineering Group (SER) for their funding and facility support of this research.

Though, I have recognized her in my Dedication of this dissertation, I want again to acknowledge the support of my wife, Corinne, who has been my companion through these many years of schooling. She has shared with me every emotional high and low that accompanied this work. She has constantly encouraged me to continue my education even when it meant her being a "single-mom" with three children for months on end. Thank you, Corinne.

Table of Contents

List of Tables

List of Figures

List of Publications

The following are peer-reviewed conference and journal publications that are derived from this dissertation research.

1. C. L. Edwards, B. G. Boone, W. S. Levine and C. C. Davis, "Modeling of Electrostatically Actuated Two-Axis (Tip-Tilt) MEMS Torsion Micro-Mirrors for Laser Beam Steering", *Proceedings of the SPIE*, Vol. 6555, No. 10, April, 2007
2. C. L. Edwards, B. G. Boone, W. S. Levine and C. C. Davis, "An Analytic Torque Model for Two-Axis MEMS Mirrors", *SPIE Journal of Micro/Nanolithography, MEMS, and MOEMS*, Vol. 6, No. 4, 2007.
3. C. L. Edwards, B. G. Boone, W. S. Levine and C. C. Davis, "First-Principles Jitter Characterization of Two-Axis MEMS Mirrors", Submitted for Publication at SPIE *Journal of Micro/Nanolithography, MEMS, and MOEMS*, July, 2007.

Chapter 1: Introduction

Overview of Dissertation

Micro-Electro-Mechanical System (MEMS) mirror devices are certain to be important for advanced optical communication and LIDAR/imaging systems. These devices have attractive features such as compact size, low-mass, and low power consumption. However, much work remains in modeling of these devices. This dissertation contains the development of the *first closed-form set of models that are analytic yet contain all of significant factors fundamental to describing the electrodynamics and motion of a MEMS Two-Axis (Tip-Tilt) mirror.* Three models have been developed: the 2D Torque Model, the Micro-mirror Pointing Model (MPM), and the Micro-mirror Jitter Model (MJM). The 2D Torque Model relates the applied voltages and the mirrors tip and tilt angles to instantaneous torque. The MPM model utilizes the torque model to describe the dynamic effects of a MEMS 2D mirror in responding to various control voltages. Both the 2D Torque Model and MPM model are supported by experimental measurements and it is shown that the models are accurate in describing steady-state as well as transient motions.

The Micro-mirror Jitter Model (MJM) addresses mirror facet jitter and is based upon the MPM model. The MJM model can be used to simulate mirror jitter as a function of the originating noise sources which are: (1) control voltage fluctuations, (2)

platform vibration, and (3) Brownian motion (mechanical-thermal) noise. The MJM model is supported by high-precision experimental measurements.

Overview of Chapters

Chapter 2 summarizes MEMS 2D mirror technology. Methods of actuation, examples of commercial-off-the-shelf (COTS) two-axis scanning mirrors, as well as the need for analytic modeling are discussed. Special effort is taken in the last section to review the prior work in modeling MEMS Two-Axis (Tip-Tilt) Mirrors.

Chapter 3 documents the development of the analytic 2D Torque Model and MPM model as well as simulations in SIMULINK and MATLAB. Removable singularities in the torque functions are considered. The torque expression with respect to the four voltage pads is implemented in software utilizing mirror symmetry, and a generalized torque function. Examples of mirror dynamics and operational sensitivities are illustrated through simulations.

Chapter 4 consists of the experimental measurements that validate the 2D Torque Model and MPM model for a noise-free environment. The experimental apparatus and methodology are also described in detail. The analytic MPM model results are compared to the experimental measurements for different mirror angles and found to be in close agreement.

Chapter 5 contains the development of the MJM model. This is done by expanding the MPM model using a first-order multidimensional Taylor expansion around a deterministic operating point. The MJM model relates the power spectral densities (PSD) of the mirror jitter to the PSD of the noise sources. Input noise sources are related to the physical, electrical and environmental conditions.

Chapter 6 presents the experimental measurements supporting the MJM model. A methodology is developed for acquiring mirror jitter measurements and validating the MJM model. The precision of the data acquisition process is analyzed. The different sources of jitter are estimated based on fundamental physical relationships or measured. Because no measurements of Brownian motion noise for the two-axis tip-tilt mirror exist in the open literature, an experimental apparatus was optimized to minimize other jitter sources and thereby directly measure the jitter induced by Brownian motion noise. The MJM model results are compared to experimental measurements and found to be in close agreement.

Chapter 7 contains the conclusions of this research detailing the *new and significant* contributions of this work relating to the 2D Torque Model, MPM model, MJM model and Systems Implications.

The 2D Torque Model and MPM model are analytic and closed-form showing functional dependence on physical, electrical, and environmental parameters. Mirror 2D pull-in expressed in terms of poles in the S-plane. The Euler equations formulation is used to motivate the use of coupled damped harmonic oscillators. The MPM model is validated with experimental measurements.

The MJM model estimates mirror jitter based of originating sources of jitter. Jitter is related by transfer functions and power spectral densities of noise sources. The damping constants and mirror "Q" are measured for both axes. Curves of damping versus pressure are provided. Jitter variance for both axes of rotation is invariant with pressure. The MJM model agrees with experimental measurements. The experimental apparatus is able to quantify jitter sources including the Brownian motion noise.

Simulations of 2D Torque, MPM, and MJM models enable design engineers to evaluate system tolerances in terms of expected mirror dynamics and jitter. System engineers can determine components and device noise requirements to ensure that the entire system stays within specification. Given a set of components in a system, system engineers can predict jitter of mirror facet.

Chapter 2: Introduction to MEMS and the Two-Axis (Tip-Tilt) Mirror

MEMS vs. MOEMS

By definition a Micro-Electrical-Mechanical System (MEMS) is a miniature structure that contains parts that are created through microfabrication. The device is designed to have mechanical motion or action induced by electrical stimulus [106], [107]. A subclass of MEMS devices is Micro-Optical-Electrical-Mechanical Systems (MOEMS) which have the additional specification of being in an optical system.

MOEMS have the advantage over other MEMS devices of moving, guiding or directing light which is massless. As the MOEMS device decreases in size the inherent limitations are the wavelength of light, the model accuracy, and the reliability of the fabrication process.

MEMS devices occupy less space, use less energy and material than non-miniaturized devices. Because of this it is expected that they will develop similarly to integrated circuits (IC) and become much cheaper due to mass fabrication and processing [106].

MEMS Fabrication Process

Since this dissertation deals with modeling and experimental measurements of a specific type of MEMS device, the Two Dimensional (2D) or Two-Axis MEMS Electrostatic Tip-Tilt Mirrors, it seems appropriate that we begin with a description of the MEMS fabrication process. MEMS technology design has three distinct stages as shown in Figure 2.1.

The first step is to develop the functional design of the MEMS device. This consists of the geometric shape of the device and its performance criteria. If it is an optical device then it also has additional optical specifications as to smoothness, reflectivity and other properties.

The second process is the electromechanical design consisting of mechanical, electrical and material calculations relating to stress and strain on the device as well as resistivity, conductivity and thermal capacitance.

The third step is the microfabrication process which consists of the masking of the sacrificial layer, material deposition and etching.

Figure 2.1 Steps of the MEMS Design Process

The rapid growth in MEMS technology has also created a need for accurate modeling of the dynamics of MEMS devices in terms of electrical, physical and environmental design parameters that can be varied in the fabrication process as specified by performance criteria. This research deals specifically with models for the dynamics of a MOEMS device, the Two-Axis Tip-Tilt Micro-mirror for laser and optical beamsteering.

Optical Communications and LIDAR/Imaging Applications

MEMS mirrors potentially offer a very compact and high-speed beamsteering capability for multiple applications [6]. The two-dimensional (2D) Tip-Tilt scanning mirrors offer continuous high precision pointing over a solid angle region. Applications can generally be divided into two categories: Optical Communications and LIDAR/Imaging Applications.

Optical communication applications involve the transmission and routing of information encoded at optical wavelengths. MEMS Optical Communication technologies can be divided into two classes: signal routing hardware and free-space

optical (FSO) communication links. Much work has been done in the last ten years in beam pointing and steering for both fiber based switching and FSO transmission [6].

For optical networking and signal routing technologies, optical MEMS devices replace traditional optical-electronic hardware that converts photons to electrons. Traditionally, the electrons are processed and routed and then reconverted to photons for retransmission along existing fiber. Fiber optic networks are limited in data transmission by the slowest components which are electron based. MEMS 2D Tip-Tilt Scanning mirrors could replace traditional electron devices or enhance existing network capability including cross-connect switches, on-demand capacity configurable networks and variable optical attenuators.

Cross-connect switches route signals from one optical fiber to a number of different fibers connecting different locations [29], [42], [52], [53], [56], [67], [84], [87]. A notional concept of an all optical cross connect is shown in Figure 2.2. In this figure, two arrays of Tip-Tilt Scanning mirrors are utilized to steer optical beams to a specific fiber. Lucent Technologies has made a multibillion-dollar investment in developing such optical switches.

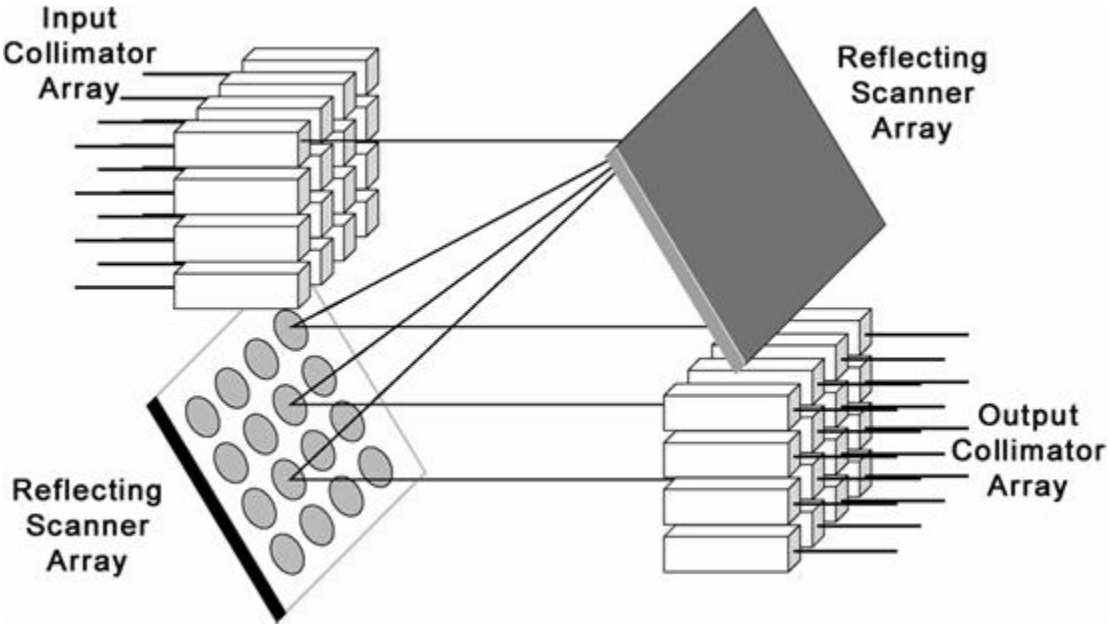


Figure 2.2 Optical Cross-Connect Networking with Two MEMS Mirror Scanning Arrays Routing Light to Designated Fibers

While cross-connect switches route one signal to one fiber, capacity configurable optical networks use scanning mirrors to direct light from multiple single-mode fibers into a multimode fiber. This network has an attractive bandwidth-on-demand feature that can be allocated in real-time. Currently 99% of optical communications uses a fixed capacity point-to-point fiber optic network.

While, precise pointing of 2D mirrors is needed in cross-connect switches and capacity configurable networks to ensure that the light is transmitted to the appropriate fiber, Variable Optical Attenuators (VOA) ensure a specified portion of the light is not transmitted through a optical fiber. In optical transmission systems, especially multimode fibers, the power spectrum intensity should be uniform for the

transmitted wavelengths [106]. Variable optical attenuators direct a portion of the light beam that will not be within the numerical aperture of another the fiber and therefore will not propagate in the fiber as shown in Figure 2.3. Scanning MEMS mirrors are ideal for this application since they are continuously steerable yielding precise pointing and attenuation [10], [67], [84].

While MEMS 2D scanning mirrors could replace network and signal routing hardware components, the mirrors could replace the "fiber" component of the optical fiber infrastructure. Free-Space Optical (FSO) Communications involves transmitting laser light through the air instead of in a fiber. FSO Communications holds the promise of network links that are reconfigurable, mobile and quickly installable. FSO communication links are divided into two classes; long-haul and short-haul links.

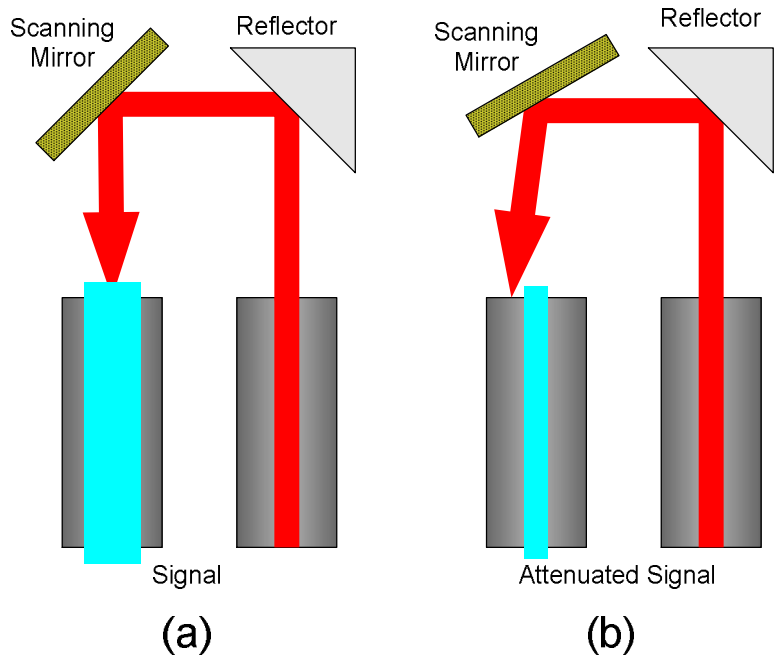


Figure 2.3 (a)-(b) Illustration of Variable Optical Attenuator

MEMS 2D Tip-Tilt scanning mirrors could play a key role in FSO long-haul applications such as satellite-to-satellite separated by hundreds of kilometers. Pointing specifications for such links have been investigated and generally require microradian precision and stability [2], [23], [24], [27], [33], [51], [93], [99], [105]. Deep-space FSO links have also been proposed and even greater precision is required in the sub-microradian regime.

Another long-haul application for optical communications is geosynchronous earth orbit (GEO) to ground downlinks. These links could be designed to mitigate availability limits due to cloud cover or adverse weather. In this case, different 2D MEMS Tip-Tilt scanning mirrors point to distinct areas on the ground. If one receiver

location was shut down due to weather or cloud cover, an adjacent ground area might be unobscured. Tip-Tilt Scanning mirrors could point to alternate locations to maintain link connectivity. Analytical work has been done to establish the availability for these links but not specifically for MEMS mirrors [111].

For more than a decade the military has had an interest in the concept of "total battlefield connectivity." This involves both long-haul and short-haul links to connect multiple assets such as troops, vehicles, low flying unmanned aerial vehicles (UAVs), high altitude aerostats, airplanes and satellites by means of FSO links as shown in Figure 2.4.

Also hybrid microwave/optical systems for battlefields are an attractive solution because of extremely hostile electromagnetic interference (EMI) environment [102]. Because high data rate transmission and mobility are considered "force multipliers," FSO links and perhaps the 2D MEMS mirror may play critical roles in future military tactical systems.

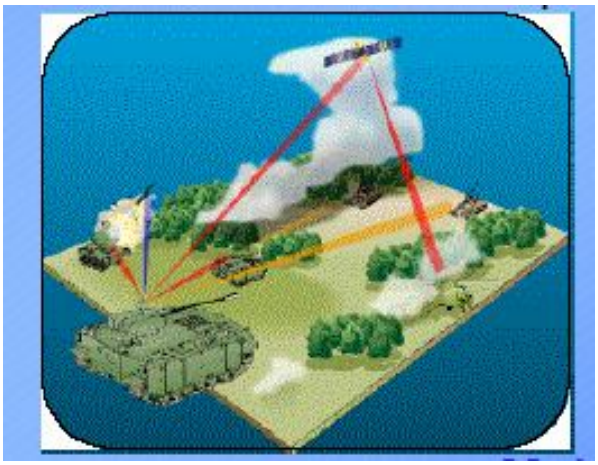


Figure 2.4 Military Concept of Total Battlefield Connectivity

Short-haul applications for FSO links involving 2D MEMS scanning mirrors include replacements for rigid RF and microwave cables in testing. In some testing environments the presence of rigid cables is not feasible. FSO links also give a degree of flexibility that a cable with a fixed length does not.

Perhaps the most commercially viable short-haul solution involving a 2D MEMS Scanner is in the so-called "Last Mile Problem." While the optical fiber infrastructure is only available to a small number of businesses, many businesses and residences in urban areas are within a mile of a fiber optic network. FSO communications could provide distribution nodes for a fiber optical network. MEMS 2D scanners would be small low-power devices that have a built in fast pointing and tracking features capable of compensating for some dynamics of the FSO link.

In addition for optical communications, 2D scanning mirrors have a promising future as components of LIDAR and imaging systems. While optical communications encode data onto light, LIDAR and imaging applications use light for sensing applications. LIDAR and imaging applications both use radiated optical sources for sensing applications, only the sensor employed is different. LIDAR uses a detector as a sensor and timed release of radiation to determine range. Imaging applications use the human eye as the sensor.

LIDAR applications for the 2D scanning mirror include resident space object (RSO) inspection in which one satellite scans another space-based object to determine characteristics of it such as telemetry, rotation, vibration, heading and other information. Additionally, laser radar could employ miniature beamsteerers to be used for landing and docking in space as shown in Figure 2.5.

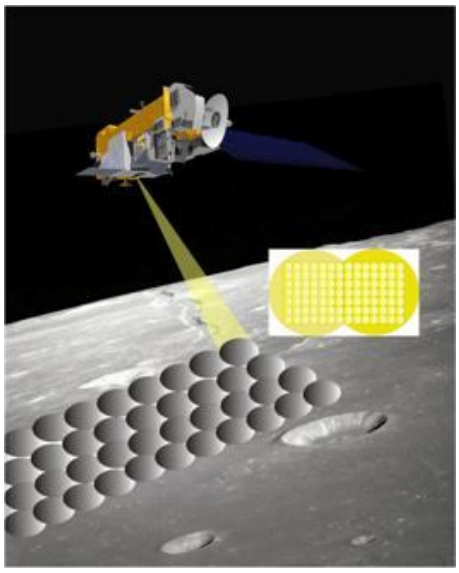


Figure 2.5 LIDAR Scanning System Proposed Lunar Mission in 2018

Since the Space Shuttle Columbia disaster, 2D MEMS scanning have been proposed as part of a system to inspect heat shield tiles of the space shuttle before reentry as shown in Figure 2.6.

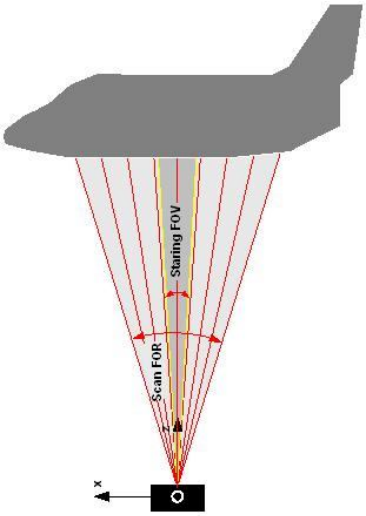


Figure 2.6 Mini-Satellite Scanning Shuttle Tiles

Medical applications that have been considered utilizing the 2D MEMS scanning mirrors consist of both medical [103], [104] applications [6], [45], [46], [70], [71]. Two of the most prominent are for endoscopes and confocal microscopes. Endoscope travel through the body can provide diagnostic information about tissue.

A contrast to the 2D scanning mirror's continuous range of pointing is the Digital Micro-mirror Device (DMD), shown in Figure 2.7, which only points in two directions: one considered the "on" position and the other the "off" position [54]. The DMD technology developed by Dr. Larry Hornbeck and Texas Instruments is the most commercially successful micro-mirror technology to date as the fundamental component of Digital Light Projection Television or DLP TV. Arrays of DMD

mirrors are used for high resolution pictures. The mirrors project onto different areas of the television screen. Varying the mirror's time spent in the "on" position allows for shades of the primary colors to be synthesized.

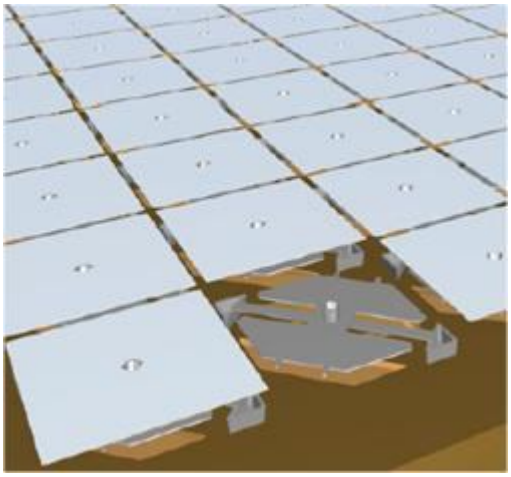


Figure 2.7 Texas Instruments DMD Array

From the previous wide range of applications of the 2D MEMS mirrors discussed it is clear that these mirrors will play a role in future technology in space, air and ground based technologies both for the military and civilian applications. They represent more compact, flexible, lower power, and a high-bandwidth capability compared to traditional technologies.

Precise Pointing: A Common Requirement

The critical system parameter shared by the above applications is the precise pointing capability of the micro-mirror. Having high fidelity models that describes steady-

state and dynamic behavior as well as micro-mirror jitter is critical to improving pointing performance [3], [4], [13], [26], [58]. A jitter model would have to be a function of originating noise sources such as control voltage fluctuations, platform vibration, and Brownian motion noise. These models could be used to improve system pointing accuracy and to reduce jitter.

Precise pointing permits for less power to be transmitted to the receiver and narrower beams in FSO applications as shown in Figure 2.8. Less power on the mirror minimizes mirror warping due to thermal heating. Minimizing the jitter leads to greater power efficiency which is required for space-based operations. Additionally constant signal-to-noise ratios (SNR) lower bit-error-rates (BER).

Investigations have been conducted into characterizing and minimizing jitter in MEMS devices especially MEMS oscillators and resonators [16], [17], [18], [20], [26], [28], [38], [47], vibration sensors [13], [28] and specifically mirrors [14], [23], [27], [45], [51], [55], [62]. Though some have addressed controls issues and compensation, none have developed explicit relationships between originating noise sources and the mirror jitter.

Previously, intersatellite and deep space jitter requirements haves been estimated [2], [23], [24], [27], [33], [51], [93], [99], [105]. These requirements of sub-microradian pointing and minimal spreading of the beam can only be achieved with a fundamental

first-principles understanding of the 2D mirror dynamics. The mirror dynamics results from deterministic torques as well as random stochastic noise sources that cause the mirror facet to jitter.

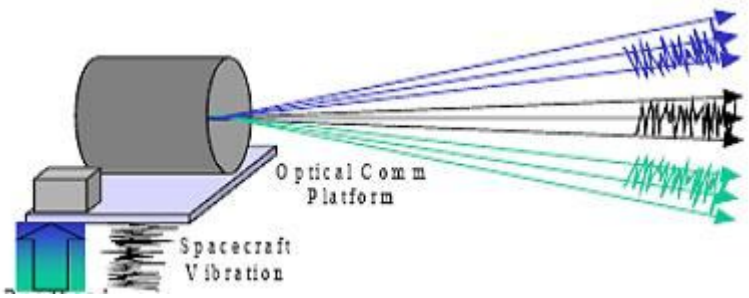


Figure 2.8 Beam Jitter Causing Signal-to-Noise Ratio to Vary

Electrostatic Actuation

Electrostatic actuation (ESA) of MEMS between the mirror facet and electrode pads located below the facet has the advantage of increasing effectiveness as the scale of the device is decreasing [39], [40], [41], [42], [43]. Additionally, electrostatic actuation (ESA) consumes negligible power, which is important for applications in space where power must be conserved. Electrostatic actuation also is desirable because of relatively low voltages compared to piezo materials and a minimum hysteresis from charge accumulation if the mirror facet is well grounded.

Examples of Commercial Tip-Tilt Micro-Mirrors

Four commercially available mirrors are shown in Figures 2.9-2.12, made by four different companies. While they come in different shapes and sizes the 2D tip-tilt mirrors have something in common: A pair of orthogonal axes that allow the mirror to tip in one direction and tilt in another. Electrostatic actuation is the most common form of inducing mechanical movement and is a natural choice for the MEMS designer.

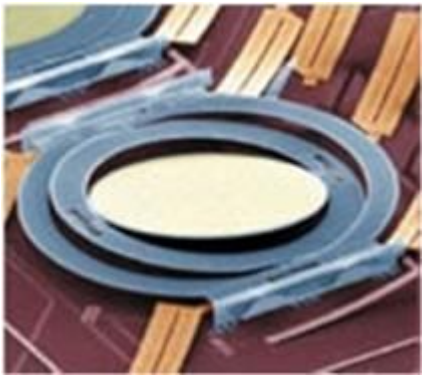


Figure 2.9 Lucent Technologies 2D Scanning Mirror. Mirror Facet Width 700µm.

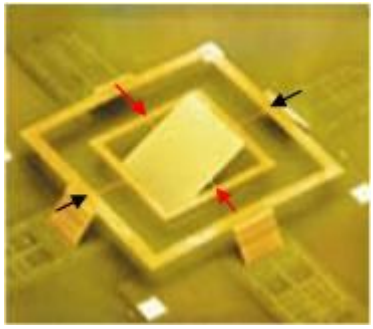


Figure 2.10 Optical Micro-Machines with Arrows Denoting Hinges. Mirror facet width is 200µm.

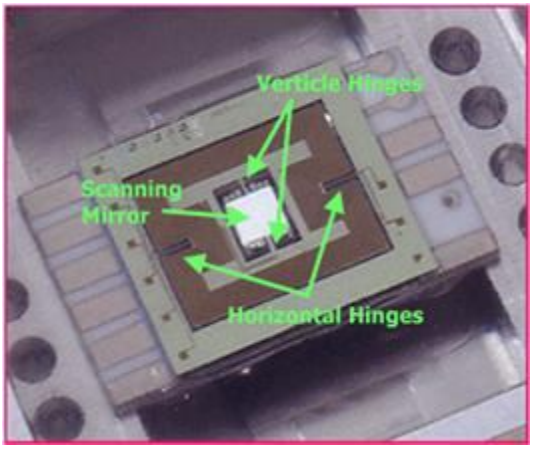


Figure 2.11 Microvision Inc., Arrows Denoting Hinges

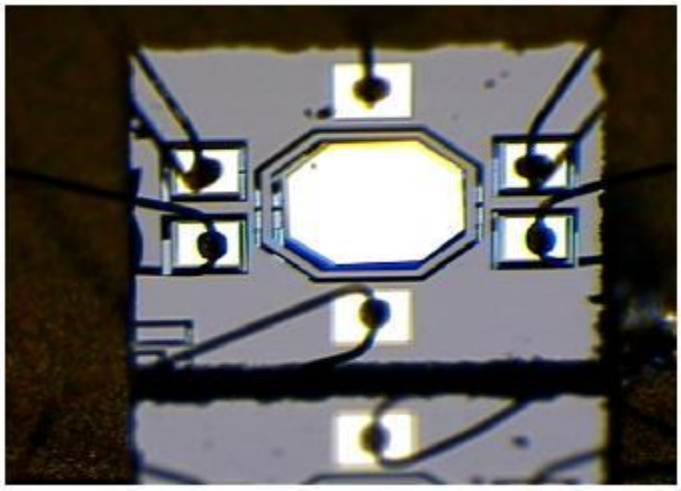


Figure 2.12 MEMS Optical Inc Micro-Mirror. Facet Width is 520µm.

Common Bias Configuration

The most common voltage configuration of 1D and 2D mirrors utilizes a common voltage bias (V_B) for all electrode pads and then applies (adds or subtracts) an

actuation voltage, v, around this bias to induce movement. For the single-axis case, as shown in Figure 2.13, only one actuating voltages is required but in the two-axis case two actuating voltages are required. The common bias configuration has been established as the way to minimize mirror warping for the 2D tip-tilt mirror [57]. Figure 2.14 shows the MEMS Optical Inc. micro-mirror with the facet removed revealing the four electrode pads underneath the mirror that control mirror pointing.

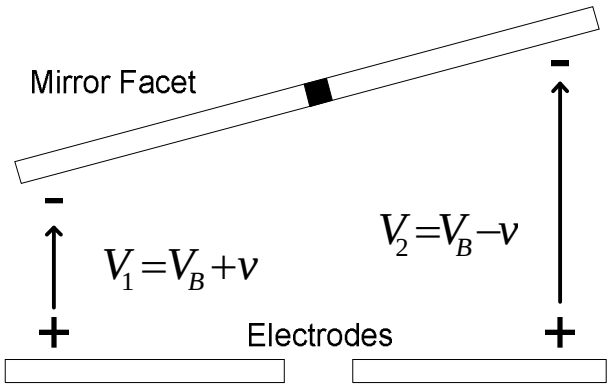


Figure 2.13 1D MEMS Electrostatic Mirror

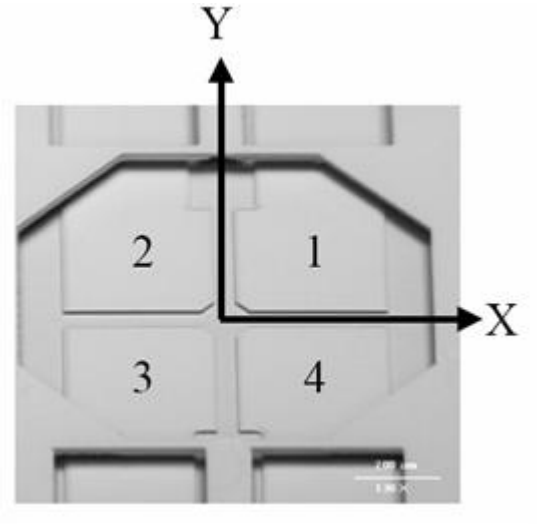


Figure 2.14 MEMS Optical Inc. Mirror's Four Voltage Pads

Micro-Mirror Modeling

A great deal of research has been done in modeling micro-mirror actuation [9], [12], [23], [24], [29], [31], [33], [34], [57], [63], [67], [70], [100], though none have dealt with the electrostatically actuated two-axis (2D) tip-tilt mirrors by developing a physically complete closed-form analytic model from electrical and mechanical first-principles, until now [108], [109], [110]. Additionally, no publication discusses the sources of mirror jitter that ultimately limit pointing performance. There have also been no published measurements of mirror facet jitter.

Much of the previous work in modeling of single-axis and two-axis tip-tilt mirrors has relied on Finite Element Methods (FEM), Finite Element Analysis (FEA), Boundary Element Methods (BEM) [1],[14], [20], [29], [30], [32], [34], [39], [44], [50], [57], [68], [69], [77], [90] or other numerical techniques [21], [11], [36], [57],

[80] to solve the electro-magnetic fields between the mirror facet and the electrode pads below. Other models relate the MEMS mirror behavior to lumped element models that can then be simulated in commercial software tools like P-SPICE [1], [14], [30], [48]. Polynomial fitting has been used in several cases to relate the applied voltages to the rotation angle induce for the 1D and 2D mirrors [21], [29], [90]. While these plots are helpful to the end user, who will integrate the mirrors into a system, the designer, who has a set of performance requirements and must create a mirror design that meets these requirements, does not have an explicit relationship between the mirror physical parameters and mirror performance.

The pull-in region for a MEMS micro-mirror is where the restoring torque of the mirror hinge is approximately equal to the electrostatic torque. As a result this region is unstable and is very sensitive to disturbances. MEMS mirrors, functioning as sensors, near the "pull-in" region have been modeled in detail [8], [11], [30], [44], [76], [77], [80], [90].

Some researchers have made assumptions that allow the torque expressions, and also the harmonic oscillator equations, to decouple and thus simplify the model. However, this is not physically appropriate. In fact, the mirror motions derived from generalized Euler equations result in a pair of damped harmonic oscillators that coupled through the torsional expression. The electrical-mechanical design for the

2D scanner with four electrode pads and two actuating voltages requires torsional expressions that are functions of both rotation angles.

Though the 1D analytical expression for the torque induced on a single-axis torsional mirror has been solved [8], [30], [35], [42], [72], [73], [74], because of the complicated geometry involved in solving the 2D case tip-tilt mirror, this problem has not been previously solved in an analytic closed-form.

Virtually all of the models attempt to relate the capacitance between the mirror and electrodes to the torque on the mirror causing it to tip and tilt [8], [13], [21], [25]. The dynamic response for the 1D and 2D cases is modeled as simple harmonic oscillator model[5], [7], [14], [16], [18], [19], [21], [25], [35], [36], [37], [41], [44], [48], [49], [50], [52], [61], [65], [66], [69].

An analytical model is more useful to the designer than numerical tables, in that it allows him or her to conceptualize the effects of different design parameters on the mirror [107]. Only first principles analytic closed-form models will allow a designer to relate the physical, electrical and environment design parameters to the dynamic behavior of the mirrors and the induced torque. However, sometimes in order to get a closed-form analytic solution simplifications are made. It is important that the simplifications do not compromise the overall accuracy of the model.

Closed-form torque models for both the one-dimensional (1D) and the two-dimensional (2D) cases are developed. The Micro-mirror Pointing Model (MPM) utilizes the 2D Torque Model as an input to the dynamic equations. Furthermore, the Mirror Pointing Model is linearized in a physically appropriate fashion as to develop the Mirror Jitter Model (MJM) and relate the input noise sources to the output effect on mirror jitter through transfer functions.

Modeling of MEMS Two-Axis Electrostatic Actuated Tip-Tilt Mirrors

In 1980, Peterson et al. [96] authored the first paper that proposed a first-principles closed-form analytic solution for the electrostatically actuated scanning mirror but only for the 1D case. The development of this model is contained in Chapter 3. Peterson's "capacitance model" ignored fringing, mirror warping and the normal termination of the electric field on the conducting mirror facet. Peterson's model remains the standard model for 1D electrostatic actuation for MEMS scanning mirrors to this day.

Peterson's model was used for the first time in the 1991 Huber et al. publication [72], applied to a "MEMS" device. Huber modeled mirror dynamics with a simple harmonic oscillator model with constant coefficients. Huber also measured steady-state characteristics of a micro-mirror.

In 1996 Toshiyoshi et al., [42], proposed enhancing the 1D model to account for curving electric field lines between the tipped mirror and electrode pads. Thus the electric field would be normal to both conducting plates. For small angles, less than 10 degrees, this is a negligible correction and since MEMS mirrors operate at small angles, less than $\pm 3^0$, this correction is very small compared to errors caused by fabrication processes and material's variation.

The first publication attempt in modeling a two-axis (2D) electrostatically actuated MEMS tip-tilt mirror with four electrode pads and rotating frame was Dotzel et al. [95]. Dotzel et al. used two damped harmonic oscillators but ignored cross-axial coupling and numerically solved for the torsional expressions with a FEM analysis.

In 2001, Dr. Ming Wu's seminal publication [57] documented the torsional calculation of the electrostatic torque induced by a mirror at given angles and applied voltages for two-axis tip-tilt micro-mirrors. All other tip-tilt mirror papers that followed utilized this model [1], [63], [68], [71].

Wu established through measurements that a common bias and actuating voltages around the bias produced a minimal amount of warping for a tip-tilt mirror. Thus two actuating voltages are used to create four voltages through mutual addition and subtraction and applied to the four electrodes under the mirror. Wu also established that the differential actuating voltages suppress cross-axial talk. While it may be

minimized, the experimental measurements validating the new Mirror Pointing Model (MPM) of this dissertation will show it to be significant.

Wu accounted for the curving electric field but eliminated the possibility of having a closed-form analytic solution. Wu also neglected the fringing effects of the electric field. As shown in this dissertation, the design of the mirror and operating limitations allow for a physically complete analytic solution to be derived. In the MPM model, the fringing effects and normal termination on a conducting surface of the electric field are accounted for in an "effective length" derived from a sub-set of steady-state measurements used to calibrate each mirror. Having the analytic solution also enables time-domain simulations whereas the previous numerical solutions are computationally expensive for a simulation evaluating the dynamics of the mirror. It is perhaps for this reason that no other publication has evaluated the transient and other time-domain effects of 2D mirrors.

Wu used Euler angles to relate coordinate systems between mirror facet and a space fixed coordinate system. Neither Wu nor any author has yet to develop the two harmonic oscillators with coupled torsional expressions from Euler dynamic equations. The torsional expressions must be functions of both actuating voltages and Euler angles. *The model, with two damped harmonic oscillators and coupled torsional expression, is not arbitrary but is imposed by the Euler equations which are a general expression of the dynamics of a rigid body such as the mirror structure.*

Wu's model also accounted for the mirror's translation in the vertical direction which is appropriate for a surface micro-machined MEMS mirror device. In the surface micro-machining process for MEMS fabrication of tip-tilt mirrors, the facets are thin. Both warping and translation must be accounted for in the models. Bulk processing is now the standard because of the superior optically smooth surface of the mirror facet. Bulk micromachining processing allows MEMS facet's to be thicker and the translational motion of the mirror can be neglected and warping minimized.

The Wu model in [57] appears in every major publication on modeling MEMS two-axis tip-tilt mirrors after 2001. In 2002, MEMS Optical Inc published [69] detailed analytic models of the damping term for the mirror, however, no measurements or curves were provided. With this dissertation, measurements are provided for damping at varying pressures. In 2003 MEMS Optical Inc again published two papers [1], [68] using the Wu Model and detailed their new bulk micromachining process that fabricates facets to any thickness and with a pit-free optically smooth surface. Again in [68], Wu's model was used to study steady-state measurements of applied voltages versus angles but the torsional expressions were decoupled and only a function of a single angle and voltage.

Prior to this dissertation, transients from switching and 2D scanning were not provided for any model in any published paper for the MEMS two-axis tip-tilt mirror.

Jitter was not discussed for any micro-mirror models. Table 2.1 shows a matrix summary of key publications for MEMS electrostatic mirrors since 1980 and the characteristics of those models.

Table 2.1 Summary of Key MEMS Mirror Papers

[Ref], Author, (Year) /	Dim	Model Type	Meas.	Model Output			Electric Field Lines	Pull-In
				Pointing	Jitter	Distortion		
[96] Peterson (1980)	1D	Analytic	Yes	Steady-State only	No	None	Linear	No
[72] Huber (1991)	1D	Numeric	No	Steady-State only	No	None	Linear	Yes
[42] Toshiyoshi (1996)	1D	Analytic(2)	No	Steady-State only	No	None	Linear	No
[95] Dotzel (1997)	2D (1)	Numeric	No	Steady-State only	No	None	Linear	No
[57] Wu (2001)	2D	Numeric	Yes	Steady-State only	No	Warping & Vert Offset	Curved	No
MPM-MJM (2007)	2D	Analytic	Yes	Steady-State & Transients	Yes	None	Linear(3)	Yes

1 Models axis as uncoupled

2 Non closed-form solution

3 Curved electric field lies accounted for by using *effective length*

Chapter 3: 2D Torque Model and Micro-Mirror Pointing Model

Introduction

As shown in Figure 3.1, actuation of a single-axis MEMS micro-mirror results from a potential difference between the grounded mirror facet and the two electrode pads located under the mirror. A common bias voltage is applied to both pads. The actuating voltage is added to the common bias for one pad and subtracted from the common bias for the other pad. The torsional hinge in the center of the mirror provides a restoring torque. The design advantage of utilizing the common bias voltage and separate actuation voltage is that as the torque on one side increases the other must decrease, thus minimizing the mirror facet flexing as the mirror rotates [57].

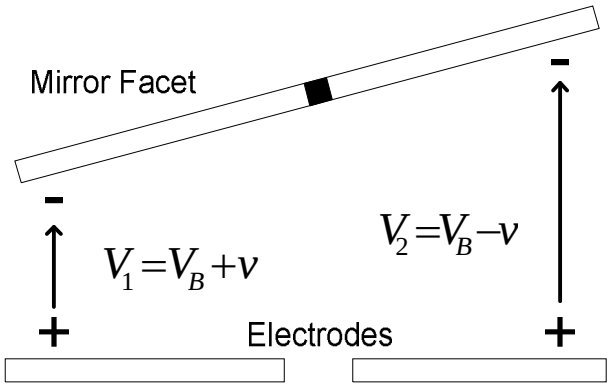


Figure 3.1 Illustration of Common Bias Configuration for 1D Torsional Mirror

A relationship can be developed for the single-axis MEMS micro-mirror that relates the physical parameters of the MEMS mirror, for example its size, thickness, and weight, and the applied voltage, to the resulting torque acting on the mirror. This is

done by relating physical size of the mirror to capacitance (Farads) and relating capacitance to work (Joules). Work is then related to force (Newtons), and then force is used to derive torque (Newton-meters). These physical relationships are illustrated in Figure 3.2.

Following this process, the tilted mirror is treated as a combination of infinitesimal capacitive elements. The resulting forces can be found by integrating over the entire mirror length to determine the total torsion effect. Carefully setting up the integrand for the integral allows for an accurate closed-form analytic solution.

The integration for the 1D case is straight-forward with only a change of variables required. The 2D case is more complicated.

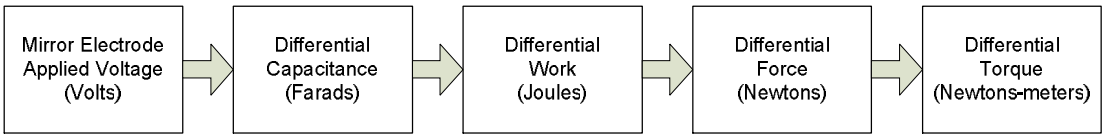


Figure 3.2 Illustration of Physical Process of Torque to Voltage

1-D Torque Development

In the 1D case of Figure 3.3, the capacitance expression, at a point x, for a differential element capacitor is

$$dc = \varepsilon \frac{da}{y(x)} = \varepsilon \frac{\ell_z \cdot dx}{y(x)}$$

where ℓ_z is the length of the pad in the "z" direction, da is the differential area element, dc is the differential capacitance and ε is the permittivity of space. The variables $\hat{x}$, $\hat{y}$, and $\hat{z}$ denote unit-vectors. Differential work is proportional to the squared voltage and is related to the capacitance as follows

$$dW = \frac{1}{2}(dc)V^2 = \frac{1}{2}\left(\varepsilon \frac{\ell_z \cdot dx}{y}\right)V^2$$

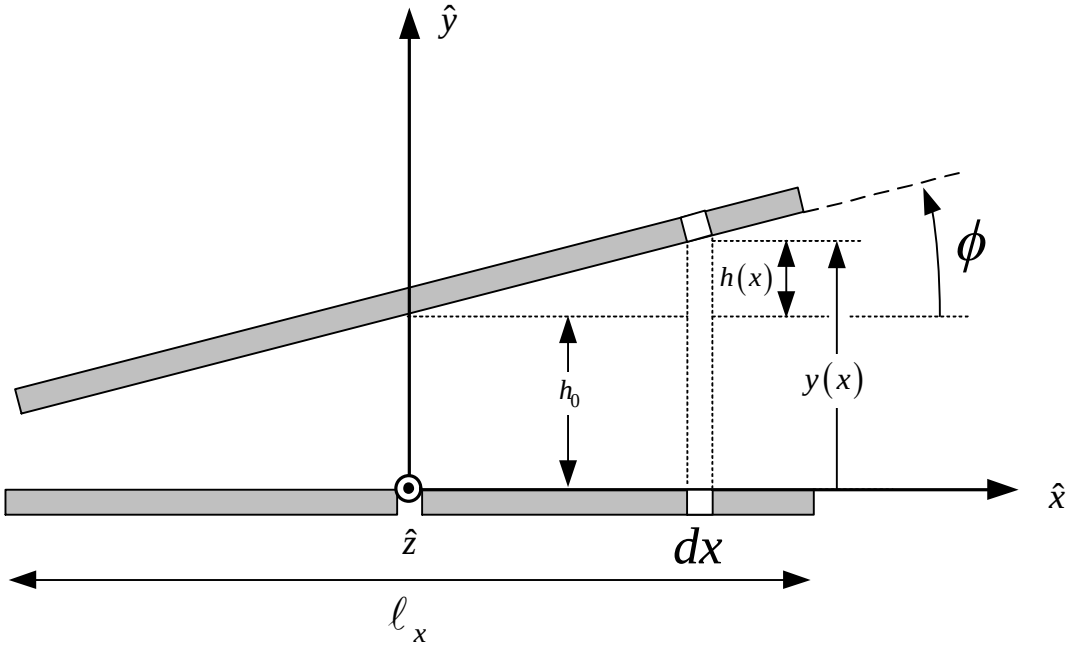


Figure 3.3 Illustration of 1-D Tilted Electrostatic MEMS Mirror

Differential force is the derivative of differential work with respect to y.

$$d\vec{F} = \frac{\partial(dW)}{\partial y}\hat{y} = -\frac{\varepsilon \ell_z \cdot dx}{2y^2}V^2\hat{y}$$

The force is in the $\hat{y}$ direction denoting an attraction between the mirror facet and the electrode pad below it. Also note that both sides of the mirror are attracted to the electrode pad below. The net torque is the sum of the two sides. Each side has an opposite sign of torque. Though the forces are in the same direction, the differential moment arms are opposed to each other as signified by the opposite sign.

The torque, $\vec{\tau}_{zRHS}$, denotes the torque integral from the right-hand-side (RHS) electrode. The "z" indicates that it is around the $\hat{z}$-axis. The differential torque is the cross product of the differential force and the moment arm

$$d\vec{\tau}_{zRHS} = \vec{r} \times d\vec{F} = -\frac{\varepsilon \ell_z \cdot x \cdot dx}{2y^2}V_1^2\hat{z}$$

Noting that:

$$h(x) = x\tan\phi$$

and

$$y = y(x) = h_0 + h(x) = h_0 + x\tan\phi$$

yields,

$$\vec{\tau}_{zRHS} = \int_0^{\frac{\ell_x}{2}} d\vec{\tau}_{z1} = -\frac{\varepsilon \ell_z V_1^2}{2}\int_0^{\frac{\ell_x}{2}} \frac{x \cdot dx}{(h_0 + x\tan\phi)^2}\hat{z}$$

with the appropriate change of variables,

$$u = h_0 + x\tan\phi$$
$$x\tan\phi = u - h_0$$
$$x = \frac{u - h_0}{\tan\phi}$$
$$dx = \frac{1}{\tan\phi}du$$

Thus, integration is along the $\hat{y}$ direction and the limits of integration become

$$B = h_0 + \frac{\ell_x}{2}\tan\phi$$

$$A = h_0$$

The total torque expression can be integrated in closed-form from this new expression.

$$\vec{\tau}_{zRHS} = \int_0^{\frac{\ell_x}{2}} d\tau_{z1} = -\frac{\varepsilon\ell_z V_1^2}{2}\frac{1}{\tan^2\phi}\int_A^B \frac{(u - h_0)\cdot du}{u^2}\hat{z}$$

Integration yields

$$\vec{\tau}_{zRHS} = -\frac{\varepsilon\ell_z V_1^2}{2}\frac{1}{\tan^2\phi}\left[\ln u + h_0 u^{-1}\right]_A^B \hat{z}$$

After substituting the limits the result is:

$$\vec{\tau}_{zRHS} = -\frac{\varepsilon\ell_z V_1^2}{2}\frac{1}{\tan^2\phi}\left[\ln\left(\frac{B}{A}\right) + h_0\left(\frac{1}{B} - \frac{1}{A}\right)\right]\hat{z}$$

Following the same development for the left-hand-side (LHS) of the mirror results in a similar expression

$$\vec{\tau}_{zLHS} = \int_0^{\frac{\ell_x}{2}} d\tau_{z2} = -\frac{\varepsilon \ell_z V_2^2}{2} \int_{-\frac{\ell_x}{2}}^{0} \frac{x \cdot dx}{\left(h_0 + x\tan\phi\right)^2} \hat{z}$$

$$\vec{\tau}_{zLHS} = \int_{-\frac{\ell_x}{2}}^{0} d\tau_{z2} = \frac{\varepsilon \ell_z V_2^2}{2} \int_0^{-\frac{\ell_x}{2}} \frac{x \cdot dx}{\left(h_0 + x\tan\phi\right)^2} \hat{z}$$

Where the upper integration limit changes to

$$B' = h_0 + \frac{\ell}{2}\tan\phi$$

and integration yields:

$$\vec{\tau}_{zLHS} = \frac{\varepsilon \ell_z V_2^2}{2} \frac{1}{\tan^2\phi} \left[\ln\left(\frac{B'}{A}\right) + h_0 \left(\frac{1}{B'} - \frac{1}{A}\right) \right] \hat{z}$$

$$\vec{\tau}_z = \vec{\tau}_{zRHS} + \vec{\tau}_{zLHS}$$

Superposition of the torques from the two sides of the mirror yields the expression for the total torque on the mirror:

$$\vec{\tau}_z = \frac{\varepsilon \ell_z V_2^2}{4} \frac{1}{\tan^2\phi} \left[\ln\left(\frac{h_0 - \frac{\ell_x}{2}\tan\phi}{h_0} \right) + h_0 \left(\frac{1}{h_0 - \frac{\ell_x}{2}\tan\phi} - \frac{1}{h_0} \right) \right] \hat{z}$$

$$- \frac{\varepsilon \ell_z V_1^2}{4} \frac{1}{\tan^2\phi} \left[\ln\left(\frac{h_0 + \frac{\ell_x}{2}\tan\phi}{h_0} \right) + h_0 \left(\frac{1}{h_0 + \frac{\ell_x}{2}\tan\phi} - \frac{1}{h_0} \right) \right] \hat{z}$$

This solution or similar solutions are found in published literature [8], [30], [42], [72], [73], [74] for the single-axis MEMS micro-mirror, although the explicit development of this solution is not. Clear understanding of the 1D solution aids in solving the torsional integral expression for the 2D tip-tilt mirror in finally arriving at the 2D Torque Model. It has been assumed, as done previously by other researchers that the voltage pads extend out to or past the mirror facet. The electric field's fringing effects, however, will be considered in Chapter 4.

The only simplification made to fundamental electrical and physical theory was in assuming that the electric field is parallel with the $\hat{y}$-axis in expressing the differential force. The electric field must terminate normal to a conducting surface as required by Maxwell's Equations as shown in Figure 3.4. Since these mirrors typically only rotate less than three degrees, this simplification is justified since the relative torque error is only 0.01%.

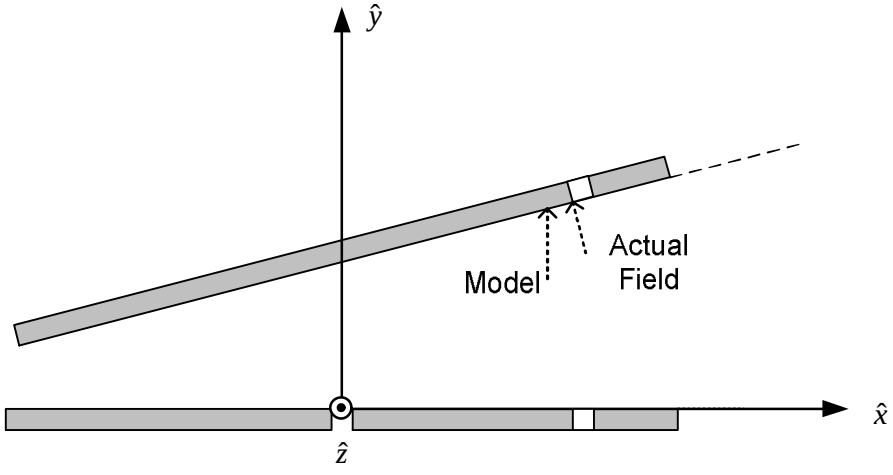


Figure 3.4 Illustration of Electric Field Termination

2D Torque Model

The two-axis tip-tilt mirror construction is based upon the single-axis design. Instead of two pads actuating the mirror, the two-axis mirror has four as shown in Figure 3.5. A bias voltage is still common to all pads but actuating voltages, $\pm u$ and $\pm v$, comprise the voltages for each pad as shown in Figure 3.5.

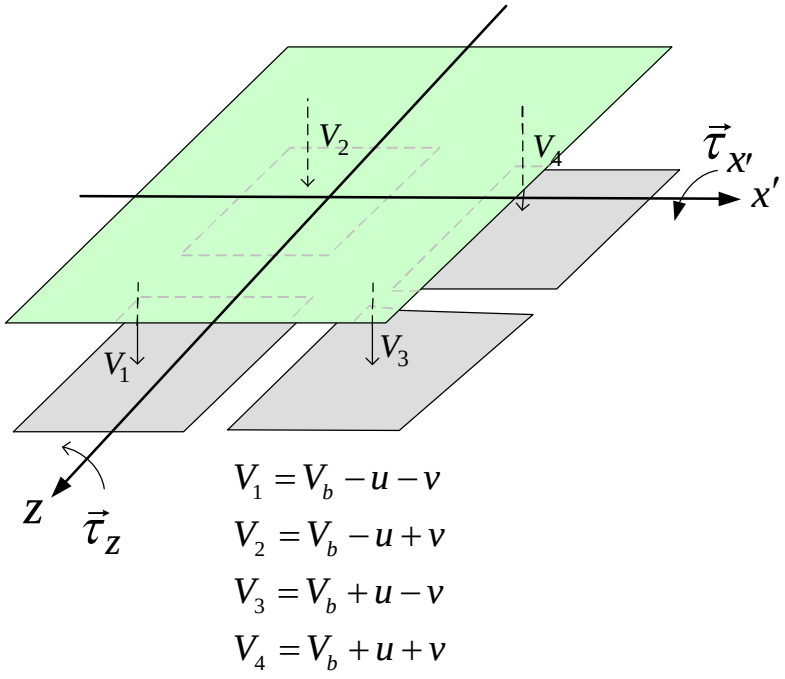


Figure 3.5 Two-Axis MEMS Micro-Mirror with Actuating Voltages

The two axes of rotation and four voltage pad structure is shown in Figure 3.5. The polarities of the voltages are indicated by the arrows which point from the mirror to the pads signifying that the mirror is grounded. Commercial examples of micro-mirrors are in Figures 3.6 and 3.7. The micro-mirror rotation is accomplished by a

torsional frame that surrounds the inner mirror facet. The mirror facet has a torsional hinge that enables rotation in one direction and another torsional hinge on the frame that allows for rotation in an orthogonal direction similar to a two-axis gimbaled structure.

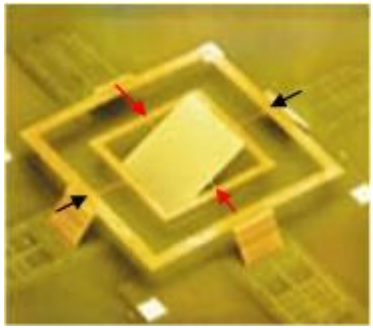


Figure 3.6 Optical Micro-Machines with Arrows Denote Hinges

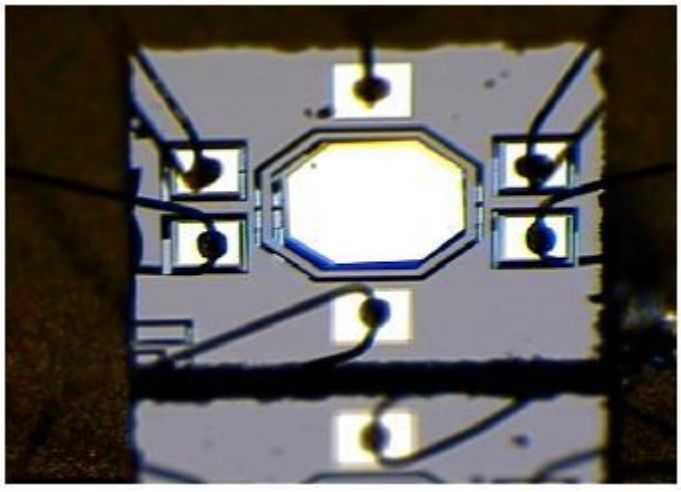


Figure 3.7 MEMS Optical Inc. Micro-mirror with Rotating Frame

Two-Axis (2D) Torque Model Development

Looking first at the geometry and considering only a single quadrant of the MEMS mirror, one can determine the torque induced by the voltage pad below it, as shown in Figures 3.8 and 3.9. Similar to the 1D development, torque on the mirror results from the combined effects of elemental capacitors each with an electrostatic force $d\vec{F}$ that depends upon the spacing between the mirror and the actuator voltage pad.

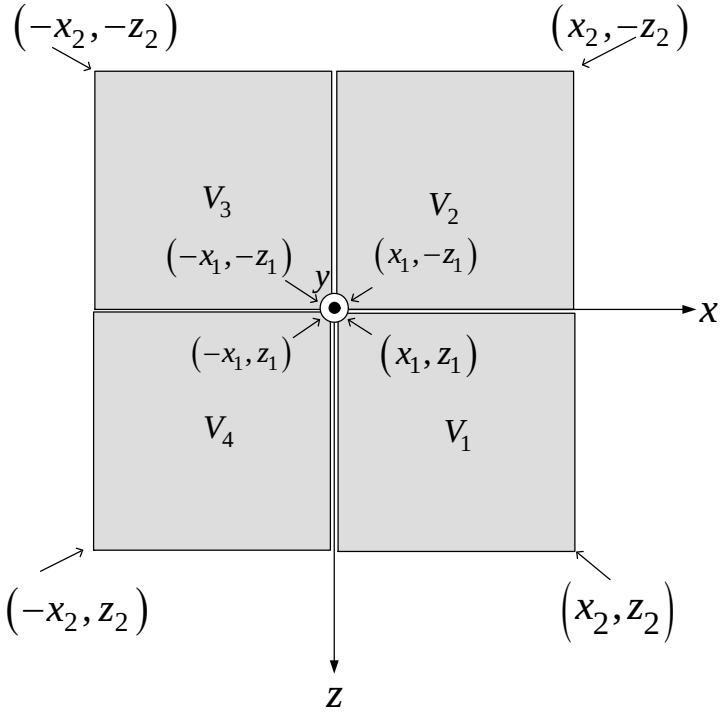


Figure 3.8 Four Voltage Pad Regions

As can be seen in Figure 3.9 one needs to determine the spacing, y, between the mirror and voltage pad as a function of pad position (x,z). The equation describing the mirror as a plane in the translated un-rotated coordinate system (x', y', z'), is given by

$$A'x' + B'y' + C'z' = 0$$

where

$$\begin{aligned} x &= x' \\ y &= y' + h_0 \\ z &= z' \end{aligned}$$

To determine the unknown constants, A', B' and C' one can consider a position vector $\vec{P} = x'\hat{x}' + y'\hat{y}' + z'\hat{z}'$ that specifies a point on the mirror plane. It must always be orthogonal to the normal vector $\hat{n}$, i.e., $\vec{P} \cdot \hat{n} = 0$, where

$$\hat{n} = -\cos\theta \sin\phi \; \hat{x}' + \cos\theta \cos\phi \; \hat{y}' + \sin\theta \; \hat{z}'$$

Consequently, all the points on the mirror facet must satisfy

$$-\cos\theta \sin\phi \; x' + \cos\theta \cos\phi \; y' + \sin\theta z' = 0 .$$

And therefore one sees that

$$\begin{aligned} A' &= -\cos\theta \sin\phi \\ B' &= \cos\theta \cos\phi \\ C' &= \sin\theta \end{aligned}$$

The spacing is given by

$$y = h_0 - Ax - Cz$$

where

$$\begin{aligned} A &= \frac{A'}{B'} = -\tan\phi \\ C &= \frac{C'}{B'} = \tan\theta \sec\phi \end{aligned}$$

At an arbitrary point (x,0,z) the distance to the mirror directly above and normal to the x-z plane is given by y where $y = h_0 + y'$. Consequently the distance is given by , $y = h_0 - Ax - Cz$ where

$$A = -\tan\phi$$
$$C = \tan\theta\sec\phi$$

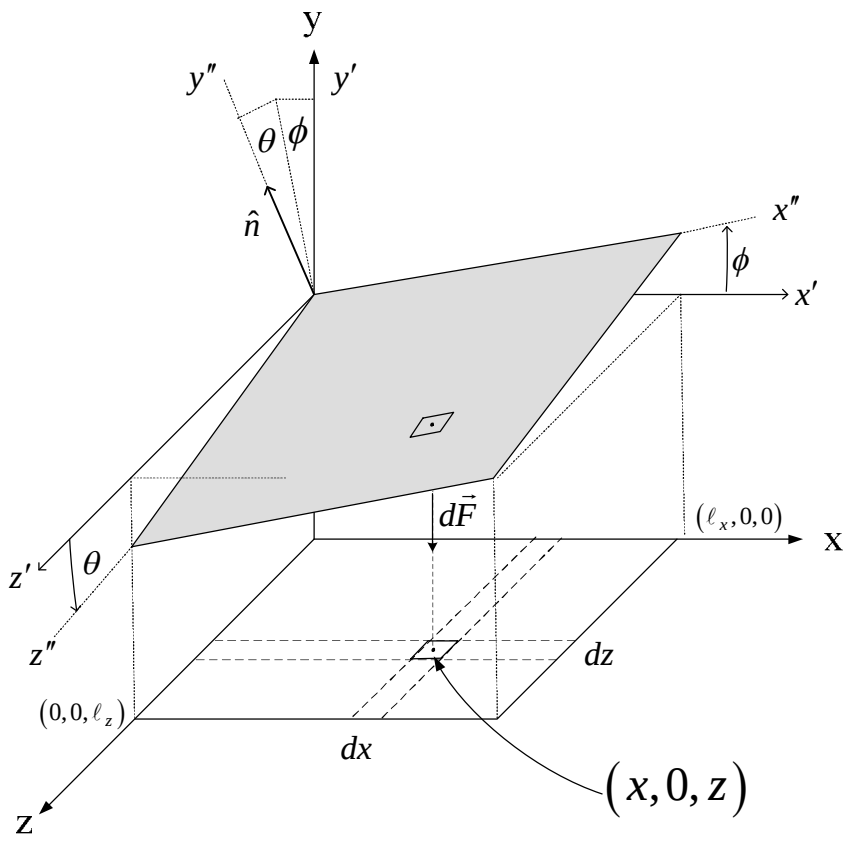


Figure 3.9 Single Quadrant of a Two-Axis MEMS Micro-mirror

As before expressions for the differential capacitance, work, force and torque must be developed. Ignoring that the plates are not quite parallel, an acceptable assumption for small angles, the differential capacitance, dc, equals

$$dc = \varepsilon \frac{da}{y} = \varepsilon \frac{dx \cdot dz}{y}$$

where da is the differential area element, ε is the permittivity of the space, and y is the distance from the differential capacitor to the voltage pad area directly below it. This can in-turn be related to differential work by

$$dW = \frac{1}{2}(dc)V^2 = \frac{1}{2}\left(\varepsilon \frac{dx \cdot dz}{y}\right)V^2$$

Since force is the derivative of work, $d\vec{F}$ can be defined as

$$d\vec{F} = \frac{\partial(dW)}{\partial y}\hat{y} = -\frac{\varepsilon V^2}{2}\frac{dx \cdot dz}{y^2}\hat{y}$$

The torque resulting from the force, $d\vec{F}$ has two vector components, one in the $\hat{z}$ direction and one in the $\hat{x}$ direction. The two torque vectors result from cross products with the respective moment arms from each axis of rotation, viz.,

$$d\vec{\tau}_z = (\textit{moment arm}) \times d\vec{F} = (x''\hat{x}'') \times d\vec{F} = (x\hat{x} + y'\hat{y}) \times d\vec{F} = -\frac{\varepsilon V^2}{2} \cdot \frac{x \cdot dxdz}{y^2}\hat{z}$$

This is illustrated in Figure 3.10. Note that the $\hat{y}$ component has no effect because $d\vec{F}$ has only a $\hat{y}$ component.

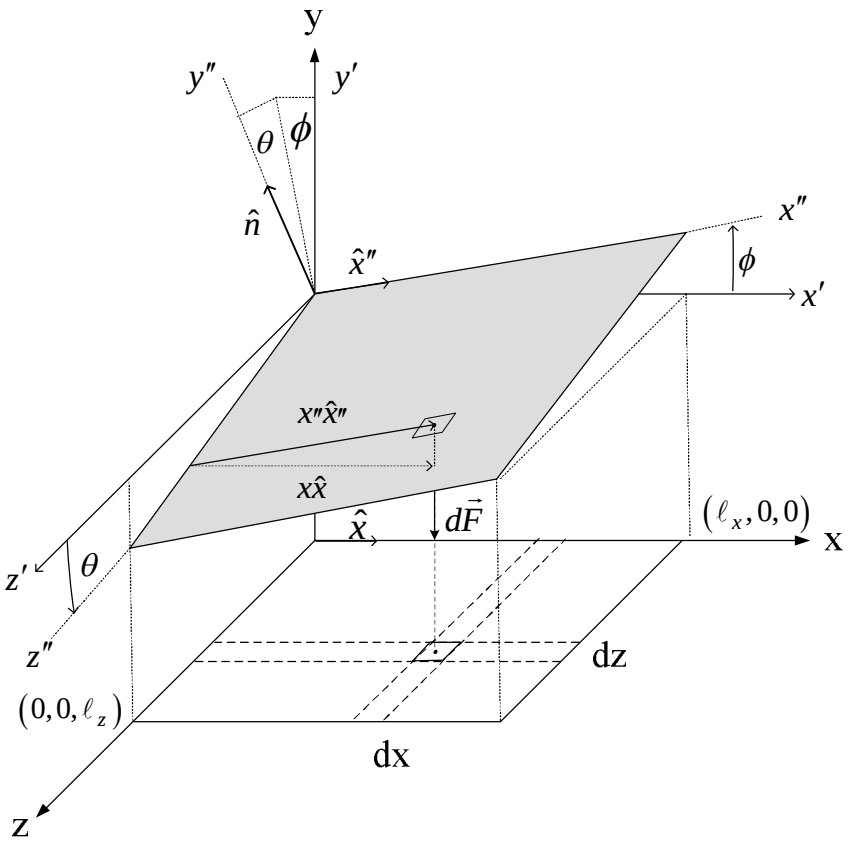


Figure 3.10 Illustration of the torque $d\vec{\tau}_z = -x\left|d\vec{F}\right|\hat{z}$

Assuming that the voltage pad in the x-z plane covers the region from $x = x_1$ to $x = x_2$ and $z = z_1$ to $z = z_2$ then

$$\vec{\tau}_z\left(x_1, x_2, z_1, z_2\right) = -\frac{\varepsilon V^2}{2} \cdot \int_{z_1}^{z_2}\int_{x_1}^{x_2} \frac{x \cdot dxdz}{y^2}\hat{z}$$

Note that z_i and x_i $\left(i = 1, 2\right)$, equals the limits of the overlap of the mirror facet and voltage pads. Some of the notation can be simplified by letting $\vec{\tau}_z = -\frac{\varepsilon V^2}{2} I_z\,\hat{z}$ where

$$I_z\left(x_1, x_2, z_1, z_2\right) = \int_{z_1}^{z_2}\int_{x_1}^{x_2} \frac{x \cdot dxdz}{y^2}$$

or

$$I_z = \int_{z_1}^{z_2}\int_{x_1}^{x_2} \frac{x \cdot dxdz}{\left(h_0 - Ax - Cz\right)^2}$$

Assuming that each integral is bounded, reversing the order of integration simplifies solving this integral equation in closed-form. The change of variable and substitution yields:

$$I_z = \int_{x_1}^{x_2}\int_{z_1}^{z_2} \frac{x \cdot dzdx}{\left(h_0 - Ax - Cz\right)^2} = \int_{x_1}^{x_2} xdx \int_{z_1}^{z_2} \frac{dz}{\left(h_0 - Ax - Cz\right)^2} = \int_{x_1}^{x_2} I'\left(x\right) xdx$$

By changing the variable and making the following substitution

$$u = h_0 - Ax - Cz$$
$$du = -Cdz$$

the inner integral becomes:

$$I'\left(x\right) = \frac{-1}{C}\int_{u_1}^{u_2} \frac{du}{\left(u\right)^2}$$

or

$$I'\left(x\right) = \int_{z_1}^{z_2} \frac{dz}{\left(h_0 - Ax - Cz\right)^2} = \int_{z_1}^{z_2} \left(h_0 - Ax - Cz\right)^{-2} dz = \frac{1}{C}\left[\left(h_0 - Ax - Cz\right)^{-1}\right]_{z_1}^{z_2}$$

$$I'\left(x\right) = \frac{1}{C}\left[\left(h_0 - Ax - Cz_2\right)^{-1} - \left(h_0 - Ax - Cz_1\right)^{-1}\right]$$

$$I'(x)=\frac{1}{C}\left[\frac{1}{(h_0-Ax-Cz_2)}-\frac{1}{(h_0-Ax-Cz_1)}\right]$$

Solving for the second integral is less simple.

$$I_z=\int_{x_1}^{x_2}\frac{1}{C}\left[\frac{1}{(h_0-Ax-Cz_2)}-\frac{1}{(h_0-Ax-Cz_1)}\right]xdx$$

$$I_z=\frac{1}{AC}\int_{x_1}^{x_2}\left[\frac{Ax}{(h_0-Ax-Cz_2)}-\frac{Ax}{(h_0-Ax-Cz_1)}\right]dx$$

Note that the integrand can be written

$$\frac{Ax}{(h_0-Ax-Cz_2)}=-1+\frac{h_0-Cz_2}{(h_0-Ax-Cz_2)}$$

As a result the above equation can be rewritten as

$$I_z=\frac{1}{AC}\int_{x_1}^{x_2}\left[-1+\frac{h_0-Cz_2}{(h_0-Ax-Cz_2)}+1-\frac{h_0-Cz_1}{(h_0-Ax-Cz_1)}\right]dx$$

removing the x term from the numerator and which simplifies to

$$I_z=\frac{1}{AC}\int_{x_1}^{x_2}\left[\frac{h_0-Cz_2}{(h_0-Ax-Cz_2)}-\frac{h_0-Cz_1}{(h_0-Ax-Cz_1)}\right]dx$$

Now all the constants can be moved outside of the integral, and the integral is in a more familiar form.

$$I_z=\frac{(h_0-Cz_2)}{AC}\int_{x_1}^{x_2}\frac{dx}{(h_0-Ax-Cz_2)}-\frac{(h_0-Cz_1)}{AC}\int_{x_1}^{x_2}\frac{dx}{(h_0-Ax-Cz_1)}$$

which becomes

$$I_z=-\frac{(h_0-Cz_2)}{A^2C}\left[\ln(h_0-Ax-Cz_2)\right]_{x_1}^{x_2}+\frac{(h_0-Cz_1)}{A^2C}\left[\ln(h_0-Ax-Cz_1)\right]_{x_1}^{x_2}$$

Substituting in the limits which are the dimensions of the voltage pads results in

$$I_z = -\frac{(h_0 - Cz_2)}{A^2 C}\left[\ln(h_0 - Ax_2 - Cz_2) - \ln(h_0 - Ax_1 - Cz_2)\right] + \frac{(h_0 - Cz_1)}{A^2 C}\left[\ln(h_0 - Ax_2 - Cz_1) - \ln(h_0 - Ax_1 - Cz_1)\right]$$

Since $\vec{\tau}_z = -\frac{\varepsilon V^2}{2} I_z \hat{z}$, then:

$$\vec{\tau}_z(x_1, x_2, z_1, z_2) = \frac{\varepsilon V^2}{2A^2 C}\left\{(h_0 - Cz_2)\left[\ln(h_0 - Ax_2 - Cz_2) - \ln(h_0 - Ax_1 - Cz_2)\right] - (h_0 - Cz_1)\left[\ln(h_0 - Ax_2 - Cz_1) - \ln(h_0 - Ax_1 - Cz_1)\right]\right\}\hat{z} \quad (3.1)$$

To review, $\vec{\tau}_z$ is the torsion that induces the tip angle ϕ. Also note that this expression is for only one voltage pad, but because the other pads can be expressed by changing the limits of integration, it is a straightforward process to determine $\vec{\tau}_z$ as exemplary of all the pads and then to combine similar results and add them vectorially.

Now, turning to the calculation of $\vec{\tau}_x$ or the torsion that induces the second rotation defined by the second angle θ. As shown in Figure 3.10, this angle is caused by torque around the x'' axis.

As in the development of $\vec{\tau}_z$, one first begins by considering only a differential area of a single voltage pad. Note that, because of the rotation ϕ, the projected area of the mirror facet onto the voltage pads below has changed. In this case, ϕ is an Euler angle that relates x'' and x' coordinate systems with

$$d\vec{\tau}_{x''} = \cos\phi \, d\vec{\tau}_{x'} \text{ and } d\vec{\tau}_{x'} = d\vec{\tau}_x$$

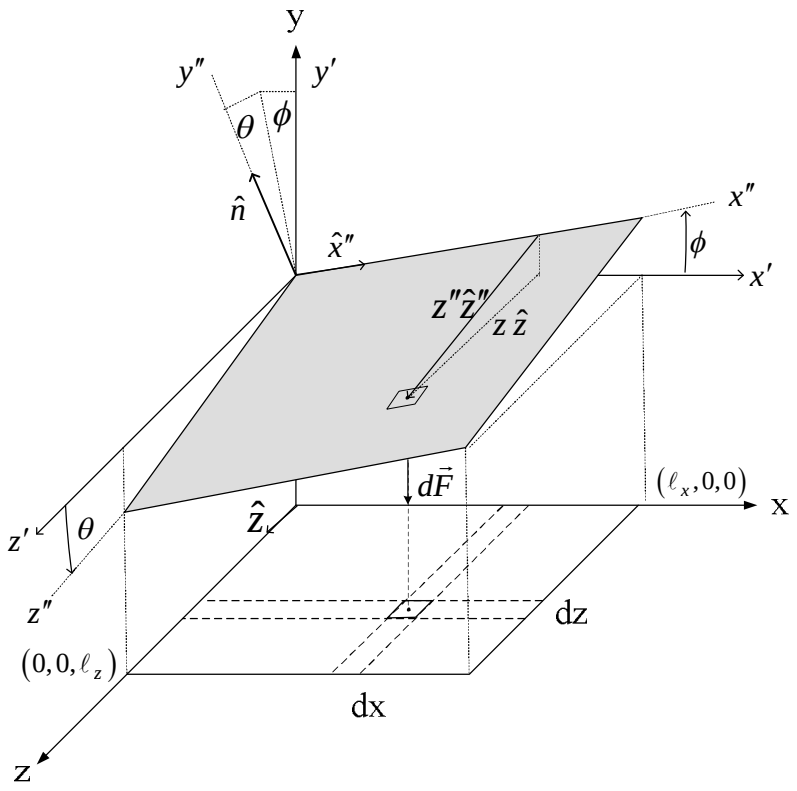


Figure 3.11 Rotation about the x'' axis

Thus, the differential torque around the x axis is expressed as

$$d\vec{\tau}_x = \left(moment\ arm\right)\times d\vec{F} = \left(z''\hat{z}''\right)\times d\vec{F} = \left(y'\hat{y}+z\hat{z}\right)\times d\vec{F}$$

There is no contribution from $\hat{y}'$ because $d\vec{F}$ is parallel to $\hat{y}$. Hence,

$$d\vec{\tau}_x = \left(y\hat{y}+z\hat{z}\right)\times d\vec{F} = \frac{\varepsilon V^2}{2}\cdot\frac{z\cdot dxdz}{y^2}\hat{x}$$

The torque is therefore

$$\vec{\tau}_x\left(x_1,x_2,z_1,z_2\right) = \frac{\varepsilon V^2}{2}\cdot\int_{z_1}^{z_2}\int_{x_1}^{x_2}\frac{z\cdot dxdz}{y^2}\hat{x}$$

As before, we can define a function I_x such that $\vec{\tau}_x = \frac{\varepsilon V^2}{2} I_x\,\hat{x}$

where

$$I_x\left(x_1,x_2,z_1,z_2\right) = \int_{z_1}^{z_2}\int_{x_1}^{x_2}\frac{z\cdot dxdz}{y^2}$$

or

$$I_x = \int_{z_1}^{z_2}\int_{x_1}^{x_2}\frac{z\cdot dxdz}{\left(h_0-Ax-Cz\right)^2}$$

$$I_x = \int_{z_1}^{z_2} zdz\int_{x_1}^{x_2}\frac{dx}{\left(h_0-Ax-Cz\right)^2} = \int_{z_1}^{z_2} I''\left(z\right)zdz$$

As before, switching the limits integration and with a similar substitution and change of variables yields:

$$I''\left(z\right) = \int_{x_1}^{x_2}\frac{dx}{\left(h_0-Ax-Cz\right)^2} = \int_{x_1}^{x_2}\left(h_0-Ax-Cz\right)^{-2}dx = \frac{1}{A}\left[\left(h_0-Ax-Cz\right)^{-1}\right]_{x_1}^{x_2}$$

Substituting in the limits of integration yields

$$I''(z)=\frac{1}{A}\left[\left(h_0-Ax_2-Cz\right)^{-1}-\left(h_0-Ax_1-Cz\right)^{-1}\right]$$

or

$$I''(z)=\frac{1}{A}\left[\frac{1}{\left(h_0-Ax_2-Cz\right)}-\frac{1}{\left(h_0-Ax_1-Cz\right)}\right]$$

Moving to the outer integral

$$I_x=\int_{z_1}^{z_2}\frac{1}{A}\left[\frac{1}{\left(h_0-Ax_2-Cz\right)}-\frac{1}{\left(h_0-Ax_1-Cz\right)}\right]zdz$$

$$I_x=\frac{1}{AC}\int_{z_1}^{z_2}\left[\frac{Cz}{\left(h_0-Ax_2-Cz\right)}-\frac{Cz}{\left(h_0-Ax_1-Cz\right)}\right]dz$$

As in the first integral, one can remove the linear term in the numerator with a similar substitution.

$$I_x=\frac{1}{AC}\int_{z_1}^{z_2}\left[-1+\frac{h_0-Ax_2}{\left(h_0-Ax_2-Cz\right)}+1-\frac{h_0-Ax_1}{\left(h_0-Ax_1-Cz\right)}\right]dz$$

$$I_x=\frac{\left(h_0-Ax_2\right)}{AC}\int_{z_1}^{z_2}\frac{1}{\left(h_0-Ax_2-Cz\right)}dz-\frac{\left(h_0-Ax_1\right)}{AC}\int_{z_1}^{z_2}\frac{1}{\left(h_0-Ax_1-Cz\right)}dz$$

The integration is again straightforward and results in

$$I_x=-\frac{\left(h_0-Ax_2\right)}{AC^2}\left[\ln\left(h_0-Ax_2-Cz\right)\right]_{z_1}^{z_2}+\frac{\left(h_0-Ax_1\right)}{AC^2}\left[\ln\left(h_0-Ax_1-Cz\right)\right]_{z_1}^{z_2}$$

Substituting in the limits of integration (as determined by the dimensions of the voltage pads) results in

$$I_x = -\frac{(h_0 - Ax_2)}{AC^2}\left[\ln(h_0 - Ax_2 - Cz_2) - \ln(h_0 - Ax_2 - Cz_1)\right] + \frac{(h_0 - Ax_1)}{AC^2}\left[\ln(h_0 - Ax_1 - Cz_2) - \ln(h_0 - Ax_1 - Cz_1)\right]$$

This in turn is related to $\vec{\tau}_x$ by

$$\vec{\tau}_x = \frac{\varepsilon V^2}{2} I_x \hat{x}$$

or

$$\vec{\tau}_x(x_1, x_2, z_1, z_2) = -\frac{\varepsilon V^2}{2}\left\{\frac{(h_0 - Ax_2)}{AC^2}\left[\ln(h_0 - Ax_2 - Cz_2) - \ln(h_0 - Ax_2 - Cz_1)\right] - \frac{(h_0 - Ax_1)}{AC^2}\left[\ln(h_0 - Ax_1 - Cz_2) - \ln(h_0 - Ax_1 - Cz_1)\right]\right\}\hat{x}$$

The resulting torque about the x axis is:

$$\vec{\tau}_x(...) = -\frac{\varepsilon V^2}{2}\left\{\frac{(h_0 - Ax_2)}{AC^2}\left[\ln(h_0 - Ax_2 - Cz_2) - \ln(h_0 - Ax_2 - Cz_1)\right] - \frac{(h_0 - Ax_1)}{AC^2}\left[\ln(h_0 - Ax_1 - Cz_2) - \ln(h_0 - Ax_1 - Cz_1)\right]\right\}\hat{x} \quad (3.2)$$

Euler Equations of Motion

Now that an analytic, physically complete expression for the torques about the two axes of the mirror has been developed, the next step is to determine the mirror response induced by these torques. The total torque vector consists of the sum of its x and z components, i.e.,

$$\vec{\tau}\left(\phi,\theta,u,v\right)=\tau_z\hat{z}+\tau_x\hat{x}$$

where the dependence of the voltage V is shown explicitly in terms of control voltages, u and v, as illustrated in Figure 3.5. To analyze the dynamics of the mirror, it is necessary to determine the torques associated with each of the rotational degrees of freedom. Letting τ_1 and τ_2 be the torques associated with the rotational coordinates ϕ and θ, respectively, then

$$\tau_1=\vec{\tau}\cdot\hat{z}'=\vec{\tau}\cdot\hat{z}=\left(\hat{z}\cdot\hat{z}\right)\tau_z=\tau_z$$
$$\tau_2=\vec{\tau}\cdot\hat{x}''=\left(\hat{x}\cdot\hat{x}''\right)\tau_x=\cos\phi\,\tau_x$$

The torque vector is composed of components aligned with the axes of rotation of the mirror and is given by

$$\vec{\tau}\left(\phi,\theta,u,v\right)=\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}=\begin{bmatrix}\tau_z\\ \tau_x\cos\left(\phi\right)\end{bmatrix}$$

With the two components of torque explicitly defined, and scalar torques τ_1 and τ_2 serving as inputs to the dynamic equations, transient phenomena of the mirror such as, critical damping, and cross-coupling of the axes at different angles can be studied.

It is natural to consider the movement of a MEMS mirror using the Euler equation formulation for the motion. The complete motion of a rigid body is decomposed into

translational and rotational components. Assuming that the mirror is not experiencing translation, one need only consider the expressions for rotational motion. In that case the rotation is described with first-order non-linear coupled differential equations

$$\begin{aligned} I_1\dot{\omega}_1 - \omega_2\omega_3\left(I_2 - I_3\right) &= T_1 \\ I_2\dot{\omega}_2 - \omega_3\omega_1\left(I_3 - I_1\right) &= T_2 \\ I_3\dot{\omega}_3 - \omega_1\omega_2\left(I_1 - I_2\right) &= T_3 \end{aligned}$$

where the ϕ and θ would be augmented with a third orthogonal angular coordinate ψ accounting for twisting of the mirror around the y'' axes. The rotational velocities $(\omega_1, \omega_2, \omega_3)$ are given by $\left(\dot{\phi}, \dot{\theta}, \dot{\psi}\right)$, $\left(I_1, I_2, I_3\right)$ are the moments of inertia, and $\left(T_1, T_2, T_3\right)$ are the torques associated with each of the rotational angular coordinates. Since it is assumed that the mechanical structure does not permit a twisting motion, it follows that ψ is constant and therefore $\omega_3 = 0$. The Euler equations then become

$$\begin{aligned} I_1\dot{\omega}_1 &= T_1 \\ I_2\dot{\omega}_2 &= T_2 \\ \omega_1\omega_2\left(I_1 - I_2\right) &= T_3 \end{aligned}$$

The torques T_1 and T_2 consist of the electrostatic torques described previously reduced by the mechanical restoring torque. The restoring torque is linearly related to the corresponding rotation angles specified by constants $\left(K_{m1}, K_{m2}\right)$ and therefore,

$$\begin{aligned} T_1 &= \tau_1 - K_{m1}\phi \\ T_2 &= \tau_2 - K_{m2}\theta \end{aligned}$$

Substituting these into the Euler rotation equations results in

$$I_1\ddot{\phi} + K_{m1}\phi = \tau_1$$
$$I_2\ddot{\theta} + K_{m2}\theta = \tau_2$$

This is now a pair of second-order differential equations with non-linear forcing functions τ_1 and τ_2.

Coupled Damped Harmonic Oscillators Equations

The equations of motion derived using the Euler formulations assume that the system is lossless. A general model needs to account for losses such as drag. This is accomplished by adding a term proportional to the angular rotation rate under the assumption that the damping is linear. The assumed constant of proportionality is denoted by D_n where $n = 1, 2$ and the equations of motion become

$$I_1\ddot{\phi} + D_1\dot{\phi} + K_{m1}\phi = \tau_1\left(\phi, \theta, u, v\right)$$
$$I_2\ddot{\theta} + D_2\dot{\theta} + K_{m2}\theta = \tau_2\left(\phi, \theta, u, v\right)$$

The applied torque expressions τ_1 and τ_2 have been established, account for cross-coupling of the angular motions. The constants I_1 and I_2 are the mirror facet's moments of inertia with respect to the axes of rotation and can be calculated from the mirror geometry and material density, which are usually provided by the MEMS mirror manufacturer. The damping constants D_1 and D_2 can be estimated from the frequency response of the mirror while only employing a single-axis of rotation. The restoring torsion constants K_{m1} and K_{m2} can be fitted from steady-state mirror

measurements where angles and voltages are again measured employing a single axis of rotation. Restoring torsion constants can also be determined by measuring the resonant frequency of the mirror and the moment of inertia. Since MEMS fabrication processes usually result in a range of restoring torsion constants because of material variation, it is best not to use a specified constant from the manufacturer without first verifying it with measurements.

2D Pull-In Analysis

The damped harmonic oscillator equations coupling through the torque expressions provide a vehicle to study the observed pull-in phenomenon. Pull-in is when the restoring torsion of the mirror hinge is less than or equal to the electrostatic torque attracting the mirror facet to the voltage pad below. To do this it is best to expand the torque functions in a Taylor series around the point (ϕ_0, θ_0) with $u_0 = 0$, and $v_0 = 0$. In that case the angular response described becomes

$$I_1\ddot{\phi} + D_1\dot{\phi} + K_{m1}\phi = K_{1\phi}\phi + K_{1\theta}\theta + K_{1u}u + K_{1v}v$$
$$I_2\ddot{\theta} + D_2\dot{\theta} + K_{m2}\theta = K_{2\phi}\phi + K_{2\theta}\theta + K_{2u}u + K_{2v}v$$

where

$$K_{1\phi} = \frac{\partial \tau_1(\phi_0, \theta_0, 0, 0)}{\partial \phi} \quad K_{1u} = \frac{\partial \tau_1(\phi_0, \theta_0, 0, 0)}{\partial u}$$
$$K_{1\theta} = \frac{\partial \tau_1(\phi_0, \theta_0, 0, 0)}{\partial \theta} \quad K_{1v} = \frac{\partial \tau_1(\phi_0, \theta_0, 0, 0)}{\partial v}$$
$$K_{2\phi} = \frac{\partial \tau_2(\phi_0, \theta_0, 0, 0)}{\partial \phi} \quad K_{2u} = \frac{\partial \tau_2(\phi_0, \theta_0, 0, 0)}{\partial u}$$
$$K_{2\theta} = \frac{\partial \tau_2(\phi_0, \theta_0, 0, 0)}{\partial \theta} \quad K_{2v} = \frac{\partial \tau_2(\phi_0, \theta_0, 0, 0)}{\partial v}$$

Assuming zero initial conditions the Laplace transform of can be written as

$$\left(I_1 s^2 + D_1 s + K_{m1}\right)\Phi(s) = K_{1\phi}\Phi(s) + K_{1\theta}\Theta(s) + K_{1u}U(s) + K_{1v}V(s)$$
$$\left(I_2 s^2 + D_2 s + K_{m2}\right)\Theta(s) = K_{2\phi}\Phi(s) + K_{2\theta}\Theta(s) + K_{2u}U(s) + K_{2v}V(s)$$

Moving the terms $\Phi(s)$ to the LHS of the first equation and likewise $\Theta(s)$ in the second equation yields (the argument "s" is assumed without its explicit display to simplify notation yields.)

$$P_1\Phi = K_{1\theta}\Theta + K_{1u}U + K_{1v}V$$
$$P_2\Theta = K_{2\phi}\Phi + K_{2u}U + K_{2v}V$$

where

$$P_1 = I_1 s^2 + D_1 s + K_{1E}$$
$$P_2 = I_2 s^2 + D_2 s + K_{2E}$$

with $K_{E1} = K_{m1} - K_{1\phi}$ and $K_{E2} = K_{m2} - K_{2\theta}$. These equations can be expressed in matrix form as

$$\begin{pmatrix} P_1 & -K_{1\theta} \\ -K_{2\phi} & P_2 \end{pmatrix}\begin{pmatrix} \Phi \\ \Theta \end{pmatrix} = \begin{pmatrix} K_{1u} & K_{1v} \\ K_{2u} & K_{2v} \end{pmatrix}\begin{pmatrix} U \\ V \end{pmatrix}$$

Multiplying by the inverse matrix gives

$$\begin{pmatrix} \Phi \\ \Theta \end{pmatrix} = \frac{1}{Q}\begin{pmatrix} P_2 & K_{1\theta} \\ K_{2\phi} & P_1 \end{pmatrix}\begin{pmatrix} K_{1u} & K_{1v} \\ K_{2u} & K_{2v} \end{pmatrix}\begin{pmatrix} U \\ V \end{pmatrix}$$

where

$$Q = Q(s) = P_1(s)P_2(s) - K_{1\theta}K_{2\phi}$$

Q(s) is a forth-order polynomial in s. Pull-in can now be studied in terms of the zeros of Q(s), which equate to poles in the Laplace transform of the solution. $K_{1\phi}(\phi_0, \theta_0)$ is an increasing function of ϕ_0 and hence K_{1E} decreases as larger values of ϕ_0 are chosen. $K_{1\theta} > 0$ and $K_{2\phi} > 0$. For sufficiently large ϕ_0 then K_{E1}, then equals zero ,i.e. $\left(K_{1\phi} = K_{m1}\right)$, which means that $P_1(0) = 0$ and $Q(0) < 0$. Since $Q(s) \to +\infty$ as $s \to +\infty$, then $Q(s) = 0$ for a larger real value of $s > 0$, i.e. the solution has a pole in the right half plane. Thus, even with a small voltage, the solution grows and would become unbounded except that it is stopped when the mirror makes contact with the voltage pads. Using the previously described torque function of ϕ_0 and θ_0 values can be calculated that produce $Q(s)$ with right half plane zeros, or equivalently a solution with Laplace transform poles. This set of values for (ϕ, θ) correspond to pull-in angles for the mirror.

"G": The General Torque Function

Because ultimately these functions will be used in simulations and also because of the symmetry in these mirrors, an arbitrary function "G" can be defined that can be called by software to calculate $\vec{\tau}_z$ and $\vec{\tau}_x$ for each voltage pad.

Because of the symmetry between the expression for equations (3.1) and (3.2), we define a general function G as

$$G(r_1, r_2, s_1, s_2, p, q) = \frac{\varepsilon}{2p^2 q}\Big\{(h_0 - qs_2)\big[\ln(h_0 - pr_2 - qs_2) - \ln(h_0 - pr_1 - qs_2)\big] - (h_0 - qs_1)\big[\ln(h_0 - pr_2 - qs_1) - \ln(h_0 - pr_1 - qs_1)\big]\Big\} \tag{3.3}$$

The torques $\vec{\tau}_x$ and $\vec{\tau}_z$ can now be expressed in terms of G by switching the order of arguments to this function.

$$\vec{\tau}_z(x_1, x_2, z_1, z_2) = V^2 G(x_1, x_2, z_1, z_2, A, C)\hat{z},$$

and

$$\vec{\tau}_x(x_1, x_2, z_1, z_2) = -V^2 G(z_1, z_2, x_1, x_2, C, A)\hat{x}$$

To be explicit, the four quadrants of torque can be expressed in terms of the G function as:

Quad 1: $\vec{\tau}_z(x_1, x_2, z_1, z_2) = V^2 G(x_1, x_2, z_1, z_2, A, C)\hat{z}$
Quad 2: $\vec{\tau}_z(x_1, x_2, -z_1, -z_2) = V^2 G(x_1, x_2, -z_1, -z_2, A, C)\hat{z}$
Quad 3: $\vec{\tau}_z(-x_1, -x_2, -z_1, -z_2) = V^2 G(-x_1, -x_2, -z_1, -z_2, A, C)\hat{z}$
Quad 4: $\vec{\tau}_z(-x_1, -x_2, z_1, z_2) = V^2 G(-x_1, -x_2, z_1, z_2, A, C)\hat{z}$

and

Quad 1: $\vec{\tau}_x\left(x_1, x_2, z_1, z_2\right) = -V^2 G\left(z_1, z_2, x_1, x_2, C, A\right)\hat{x}$

Quad 2: $\vec{\tau}_x\left(x_1, x_2, -z_1, -z_2\right) = -V^2 G\left(-z_1, -z_2, x_1, x_2, C, A\right)\hat{x}$

Quad 3: $\vec{\tau}_x\left(-x_1, -x_2, -z_1, -z_2\right) = -V^2 G\left(-z_1, -z_2, -x_1, -x_2, C, A\right)\hat{x}$

Quad 4: $\vec{\tau}_x\left(-x_1, -x_2, z_1, z_2\right) = -V^2 G\left(z_1, z_2, -x_1, -x_2, C, A\right)\hat{x}$

All the code for the SIMULINK and MATLAB simulation is found in Appendix A.

Removable Singularities in the "G" Function

For completeness, the singularities in the G function are examined next. The numerical behavior of G must be understood to correctly implement it in software such as SIMULINK for simulations. Since the G function appears to have several singularities, it must be determined if they are removable. This next section deals with this analysis.

Singularities in the G function occur when one or both of the angles are zero. To examine the singular cases where $p \to 0$ or $q \to 0$ or $\left(p, q\right) \to \left(0, 0\right)$, the following sets of polynomials are defined:

$$P_{11} = P_{11}\left(p, q\right) = h_0 - pr_1 - qs_1$$
$$P_{12} = P_{12}\left(p, q\right) = h_0 - pr_1 - qs_2$$
$$P_{21} = P_{21}\left(p, q\right) = h_0 - pr_2 - qs_1$$
$$P_{22} = P_{22}\left(p, q\right) = h_0 - pr_2 - qs_2$$

The "G" function can now be represented as

$$G(r_1, r_2, s_1, s_2, p, q) = \frac{\varepsilon}{2p^2 q}\left\{P_{22}(0,q)\left[\ln(P_{22}) - \ln(P_{12})\right] - P_{11}(0,q)\left[\ln(P_{21}) - \ln(P_{11})\right]\right\} \quad (*)$$

and the following identities are evident:

$P_{11}(0,q) = P_{21}(0,q)$, $P_{12}(0,q) = P_{22}(0,q)$, $P_{11}(p,0) = P_{12}(p,0)$, and

$$P_{21}(p,0) = P_{22}(p,0) \quad (**)$$

$$\frac{\partial P_{11}}{\partial p} = \frac{\partial P_{12}}{\partial p} = -r_1, \ \frac{\partial P_{21}}{\partial p} = \frac{\partial P_{22}}{\partial p} = -r_2, \ \frac{\partial P_{21}}{\partial q} = \frac{\partial P_{11}}{\partial q} = -s_1, \text{ and } \frac{\partial P_{22}}{\partial q} = \frac{\partial P_{12}}{\partial q} = -s_1 \quad (***)$$

The singularity at p=0

This is dealt with using L'Hospital's Rule on the "function in (*).

Taking the derivative of the numerator, N, with respect to (WRT) "p" yields

$$\frac{dN}{dp} = \varepsilon\left\{P_{22}(0,q)\left[P_{22}^{-1}\frac{\partial P_{22}}{\partial p} - P_{12}^{-1}\frac{\partial P_{12}}{\partial p}\right] - P_{11}(0,q)\left[P_{21}^{-1}\frac{\partial P_{21}}{\partial p} - P_{11}^{-1}\frac{\partial P_{11}}{\partial p}\right]\right\}$$

Using (***)

$$\frac{dN}{dp} = \varepsilon\left\{P_{22}(0,q)\left[-r_2 P_{22}^{-1} + r_1 P_{12}^{-1}\right] - P_{11}(0,q)\left[-r_2 P_{21}^{-1} + r_1 P_{11}^{-1}\right]\right\}$$

$$\left.\frac{dN}{dp}\right|_{p=0} = \varepsilon\left\{P_{22}(0,q)\left[-r_2 P_{22}^{-1}(0,q) + r_1 P_{12}^{-1}(0,q)\right] - P_{11}(0,q)\left[-r_2 P_{21}^{-1}(0,q) + r_1 P_{11}^{-1}(0,q)\right]\right\}$$

Using (**)

$$\left.\frac{dN}{dp}\right|_{p=0} = \varepsilon\left\{P_{22}(0,q)\left[-r_2P_{22}^{-1}(0,q)+r_1P_{22}^{-1}(0,q\right]-P_{11}(0,q)\left[-r_2P_{11}^{-1}(0,q)+r_1P_{11}^{-1}(0,q)\right]\right\}$$

$$\left.\frac{dN}{dp}\right|_{p=0} = \varepsilon\left\{\left[-r_2+r_1\right]-\left[-r_2+r_1\right]\right\}$$

$$\left.\frac{dN}{dp}\right|_{p=0} = 0$$

Since the denominator $D = 2p^2q$ then $\frac{dD}{dp} = 4pq$ and $\left.\frac{dD}{dp}\right|_{p=0} = 0$ and L'Hospital's rule results in zero over zero. It must be applied a second time, i.e.,

$$\frac{d^2N}{dp^2} = \varepsilon\left\{P_{22}(0,q)\left[(-1)(-r_2)(-r_2)P_{22}^{-2}(p,q)+(-1)(r_1)(-r_1)P_{12}^{-2}(p,q)\right]\right.$$
$$\left.-P_{11}(0,q)\left[(-1)(-r_2)(-r_2)P_{21}^{-2}(p,q)+(-1)(r_1)(-r_1)P_{11}^{-2}(p,q)\right]\right\}$$

$$\left.\frac{d^2N}{dp^2}\right|_{p=0} = \varepsilon\left\{P_{22}(0,q)\left[-r_2^2P_{22}^{-2}(0,q)+r_1^2P_{12}^{-2}(0,q)\right]-P_{11}(0,q)\left[-r_2^2P_{21}^{-2}(0,q)+r_1^2P_{11}^{-2}(0,q)\right]\right\}$$

Using (**)

$$\left.\frac{d^2N}{dp^2}\right|_{p=0} = \varepsilon\left\{P_{22}(0,q)\left[-r_2^2P_{22}^{-2}(0,q)+r_1^2P_{22}^{-2}(0,q)\right]-P_{11}(0,q)\left[-r_2^2P_{11}^{-2}(0,q)+r_1^2P_{11}^{-2}(0,q)\right]\right\}$$

$$\left.\frac{d^2N}{dp^2}\right|_{p=0} = \varepsilon\left\{P_{22}^{-1}(0,q)\left[-r_2^2+r_1^2\right]-P_{11}^{-1}(0,q)\left[-r_2^2+r_1^2\right]\right\}$$

$$\left.\frac{d^2N}{dp^2}\right|_{p=0} = \varepsilon\left(r_1^2-r_2^2\right)\left[P_{22}^{-1}(0,q)-P_{11}^{-1}(0,q)\right] = \varepsilon\left(r_1^2-r_2^2\right)\left[\frac{1}{h_0-qs_2}-\frac{1}{h_0-qs_1}\right]$$

$$\left.\frac{d^2N}{dp^2}\right|_{p=0} = \varepsilon q \frac{(s_2 - s_1)(r_1^2 - r_2^2)}{(h_0 - qs_2)(h_0 - qs_1)}$$

The second derivative of the denominator gives $\left.\frac{d^2D}{dp^2}\right|_{p=0} = 4q$ and L'Hospital's Rule

(the ratio of derivatives) gives $\left.\frac{d^2N}{dp^2}\right|_{p=0} \Big/ \left.\frac{d^2D}{dp^2}\right|_{p=0} = \frac{\varepsilon}{4}(r_1^2 - r_2^2)\frac{(s_2 - s_1)}{(h_0 - qs_2)(h_0 - qs_1)}$

and hence

$$\boxed{G(r_1, r_2, s_1, s_2, 0, q) = \frac{\varepsilon}{4}(r_1^2 - r_2^2)\frac{(s_2 - s_1)}{(h_0 - qs_2)(h_0 - qs_1)}}$$

Thus the first equation is verified.

The singularity at q=0

Evaluating these singularities using L'Hospital's rule but only requires the first derivative with respect to "q" of the numerator and denominator, followed by the substitution q=0, which yields:

$$G(r_1, r_2, s_1, s_2, p, q) = \frac{\varepsilon}{2p^2 q}\left\{P_{22}(0,q)\left[\ln(P_{22}) - \ln(P_{12})\right] - P_{11}(0,q)\left[\ln(P_{21}) - \ln(P_{11})\right]\right\}$$

Thus,

$$N = \varepsilon\left\{P_{22}(0,q)\left[\ln(P_{22}) - \ln(P_{12})\right] - P_{11}(0,q)\left[\ln(P_{21}) - \ln(P_{11})\right]\right\}$$

Using the product rule for derivatives

$$\frac{dN}{dq}=\varepsilon\left\{P_{22}(0,q)\left[-s_2P_{22}^{-1}(p,q)+s_2P_{12}^{-1}(p,q)\right]-P_{11}(0,q)\left[-s_1P_{21}^{-1}(p,q)+s_1P_{11}^{-1}(p,q)\right]\right.$$
$$\left.-s_2\left[\ln\left(P_{22}(p,q)\right)-\ln\left(P_{12}(p,q)\right)\right]+s_1\left[\ln\left(P_{21}(p,q)\right)-\ln\left(P_{11}(p,q)\right)\right]\right\}$$

$$\left.\frac{dN}{dq}\right|_{q=0}=\varepsilon\left\{P_{22}(0,0)\left[-s_2P_{22}^{-1}(p,0)+s_2P_{12}^{-1}(p,0)\right]-P_{11}(0,0)\left[-s_1P_{21}^{-1}(p,0)+s_1P_{11}^{-1}(p,0)\right]\right.$$
$$\left.-s_2\left[\ln\left(P_{22}(p,0)\right)-\ln\left(P_{12}(p,0)\right)\right]+s_1\left[\ln\left(P_{21}(p,0)\right)-\ln\left(P_{11}(p,0)\right)\right]\right\}$$

$$\left.\frac{dN}{dq}\right|_{q=0}=\varepsilon\left\{h_0\left[-s_2P_{22}^{-1}(p,0)+s_2P_{12}^{-1}(p,0)\right]-h_0\left[-s_1P_{21}^{-1}(p,0)+s_1P_{11}^{-1}(p,0)\right]\right.$$
$$\left.-s_2\left[\ln\left(P_{22}(p,0)\right)-\ln\left(P_{12}(p,0)\right)\right]+s_1\left[\ln\left(P_{21}(p,0)\right)-\ln\left(P_{11}(p,0)\right)\right]\right\}$$

$$\left.\frac{dN}{dq}\right|_{q=0}=\varepsilon\left\{h_0\left[-s_2P_{22}^{-1}(p,0)+s_2P_{11}^{-1}(p,0)\right]-h_0\left[-s_1P_{22}^{-1}(p,0)+s_1P_{11}^{-1}(p,0)\right]\right.$$
$$\left.-s_2\left[\ln\left(P_{22}(p,0)\right)-\ln\left(P_{11}(p,0)\right)\right]+s_1\left[\ln\left(P_{22}(p,0)\right)-\ln\left(P_{11}(p,0)\right)\right]\right\}$$

$$\left.\frac{dN}{dq}\right|_{q=0}=\varepsilon\left\{h_0(s_1-s_2)P_{22}^{-1}(p,0)-h_0(s_1-s_2)P_{11}^{-1}(p,0)+(s_1-s_2)\ln\left(P_{22}(p,0)\right)-(s_1-s_2)\ln\left(P_{11}(p,0)\right)\right\}$$
$$\left.\frac{dN}{dq}\right|_{q=0}=\varepsilon(s_1-s_2)\left\{h_0P_{22}^{-1}(p,0)-h_0P_{11}^{-1}(p,0)+\ln\left(P_{22}(p,0)\right)-\ln\left(P_{11}(p,0)\right)\right\}$$

With the denominator, $D=2p^2q$ the derivative is $\left.\frac{dD}{dq}\right|_{q=0}=2p^2$. Then the

L'Hospital's rule is

$$\left.\frac{dN}{dq}\right|_{q=0}\Big/\left.\frac{dD}{dq}\right|_{q=0}=\frac{\varepsilon(s_1-s_2)}{2p^2}\left\{h_0P_{22}^{-1}(p,0)-h_0P_{11}^{-1}(p,0)+\ln\left(P_{22}(p,0)\right)-\ln\left(P_{11}(p,0)\right)\right\}$$

$$G(r_1, r_2, s_1, s_2, p, 0) = \frac{\varepsilon(s_1 - s_2)}{2p^2}\left\{\frac{h_0}{h_0 - pr_2} - \frac{h_0}{h_0 - pr_1} + \ln(h_0 - pr_2) - \ln(h_0 - pr_1)\right\}$$

Thus the second equation verified.

Singularity when q=0 and p=0

Using L'Hospital's rule requires two derivatives and yields:

$$N = \varepsilon(s_1 - s_2)\left\{h_0 P_{22}^{-1}(p,0) - h_0 P_{11}^{-1}(p,0) + \ln\left(P_{22}(p,0)\right) - \ln\left(P_{11}(p,0)\right)\right\}$$

$$\frac{dN}{dp} = \varepsilon(s_1 - s_2)\left\{(-1)(-r_2)h_0 P_{22}^{-2}(p,0) - (-1)(-r_1)h_0 P_{11}^{-2}(p,0) - r_2 P_{22}^{-1}(p,0) + r_1 P_{11}^{-1}(p,0)\right\}$$

$$\frac{dN}{dp} = \varepsilon(s_1 - s_2)\left\{r_2 h_0 P_{22}^{-2}(p,0) - r_1 h_0 P_{11}^{-2}(p,0) - r_2 P_{22}^{-1}(p,0) + r_1 P_{11}^{-1}(p,0)\right\}$$

$$\left.\frac{dN}{dp}\right|_{p=0} = \varepsilon(s_1 - s_2)\left\{r_2 h_0 P_{22}^{-2}(0,0) - r_1 h_0 P_{11}^{-2}(0,0) - r_2 P_{22}^{-1}(0,0) + r_1 P_{11}^{-1}(0,0)\right\}$$

$$\left.\frac{dN}{dp}\right|_{p=0} = \varepsilon(s_1 - s_2)\left\{r_2 h_0 h_0^{-2} - r_1 h_0 h_0^{-2} - r_2 h_0^{-1} + r_1 h_0^{-1}\right\} = 0 \text{ Since } D = 2p^2, \text{ then}$$

$\left.\frac{dD}{dp}\right|_{p=0} = 0$ and the ratio gives zero over zero. Second derivatives must then be

determined, i.e.,

$$\frac{d^2N}{dp^2} = \varepsilon(s_1 - s_2)\left\{-2r_2(-r_2)h_0 P_{22}^{-3}(p,0) - (-2)r_1(-r_1)h_0 P_{11}^{-3}(p,0)\right.$$
$$\left. -(-1)r_2(-r_2)P_{22}^{-2}(p,0) + (-1)r_1(-r_1)P_{11}^{-2}(p,0)\right\}$$

$$\frac{d^2N}{dp^2} = \varepsilon\left(s_1 - s_2\right)\left\{2r_2^2 h_0 P_{22}^{-3}\left(p,0\right) - 2r_1^2 h_0 P_{11}^{-3}\left(p,0\right) - r_2^2 P_{22}^{-2}\left(p,0\right) + r_1^2 P_{11}^{-2}\left(p,0\right)\right\}$$

$$\left.\frac{d^2N}{dp^2}\right|_{p=0} = \varepsilon\left(s_1 - s_2\right)\left\{2r_2^2 h_0 P_{22}^{-3}\left(0,0\right) - 2r_1^2 h_0 P_{11}^{-3}\left(0,0\right) - r_2^2 P_{22}^{-2}\left(0,0\right) + r_1^2 P_{11}^{-2}\left(0,0\right)\right\}$$

$$\left.\frac{d^2N}{dp^2}\right|_{p=0} = \varepsilon\left(s_1 - s_2\right)\left\{2r_2^2 h_0 h_0^{-3} - 2r_1^2 h_0 h_0^{-3} - r_2^2 h_0^{-2} + r_1^2 h_0^{-2}\right\}$$

$$\left.\frac{d^2N}{dp^2}\right|_{p=0} = \varepsilon\frac{\left(s_1 - s_2\right)\left(r_2^2 - r_1^2\right)}{h_0^2} \text{ and } \left.\frac{d^2D}{dp^2}\right|_{p=0} = 4$$

L'Hospital's ratio (of derivatives) gives

$$\left.\frac{d^2N}{dp^2}\right|_{p=0} \Big/ \left.\frac{d^2D}{dp^2}\right|_{p=0} = \varepsilon\frac{\left(s_1 - s_2\right)\left(r_2^2 - r_1^2\right)}{4h_0^2} \text{ and therefore}$$

$$G\left(r_1, r_2, s_1, s_2, 0, 0\right) = \varepsilon\frac{\left(s_1 - s_2\right)\left(r_2^2 - r_1^2\right)}{4h_0^2}$$

Thus the third equation verified.

Note that it doesn't matter whether you let $p \to 0$ and then $q \to 0$ or if you let $q \to 0$ and then $p \to 0$. Hence, it doesn't matter which order you take the limits.

Thus, it can be shown that when $P_{ij} = 0$, it means that the mirror has made contact with one of the voltage pads. Hence, a knowledge of which polynomial has vanished will let one determine exactly which pad the mirror has contacted.

Let u be the control voltage that produces rotation around the z' axis measured by the angle, ϕ, and v be the control voltage that causes rotation around the x'' axis measured by the angle θ.

The control voltage is combined with a bias voltage, V_b, and the combination of control is applied to the various pads as follows:

$$
\begin{aligned}
V_1 &= V_b - u + v \\
V_2 &= V_b - u - v \\
V_3 &= V_b + u - v \\
V_4 &= V_b + u + v
\end{aligned}
$$

The torque in the z direction produced by the appropriate applied voltages is:

$$
\begin{aligned}
\vec{\tau}_{z1}(\ x_1,\ x_2,\ z_1,\ z_2) &= V_1^2 G(\ x_1,\ x_2,\ z_1,\ z_2, A, C\ )\hat{z} = G_{11} V_1^2 \hat{z} \\
\vec{\tau}_{z2}(\ x_1,\ x_2, -z_2, -z_1) &= V_2^2 G(\ x_1,\ x_2, -z_2, -z_1, A, C)\hat{z} = G_{12} V_2^2 \hat{z} \\
\vec{\tau}_{z3}(-x_2, -x_1, -z_2, -z_1) &= V_3^2 G(-x_2, -x_1, -z_2, -z_1, A, C)\hat{z} = G_{13} V_3^2 \hat{z} \\
\vec{\tau}_{z4}(-x_2, -x_1,\ z_1,\ z_2) &= V_4^2 G(-x_2, -x_1,\ z_1,\ z_2, A, C)\hat{z} = G_{14} V_4^2 \hat{z}
\end{aligned}
$$

$$
\vec{\tau}_z(\phi, \theta, u, v) = \vec{\tau}_{z1} + \vec{\tau}_{z2} + \vec{\tau}_{z3} + \vec{\tau}_{z4} = \left(V_1^2 G_{11} + V_2^2 G_{12} + V_3^2 G_{13} + V_4^2 G_{14}\right)\hat{z}
$$

The torque in the x direction is likewise given by

$$\vec{\tau}_{x1}(\ x_1,\ \ x_2,\ \ z_1,\ \ z_2) = -V_1^2 G(\ z_1,\ \ z_2,\ x_1,\ \ x_2, C, A)\,\hat{x} = -G_{21}V_1^2\hat{x}$$
$$\vec{\tau}_{x2}(\ x_1,\ \ x_2, -z_2, -z_1) = -V_2^2 G(-z_2, -z_1,\ x_1,\ \ x_2, C, A)\,\hat{x} = -G_{22}V_2^2\hat{x}$$
$$\vec{\tau}_{x3}(-x_2, -x_1, -z_2, -z_1) = -V_3^2 G(-z_2, -z_1, -x_2, -x_1, C, A)\,\hat{x} = -G_{23}V_3^2\hat{x}$$
$$\vec{\tau}_{x4}(-x_2, -x_1,\ \ z_1,\ \ z_2) = -V_4^2 G(\ z_1,\ \ z_2, -x_2, -x_1, C, A)\,\hat{x} = -G_{24}V_4^2\hat{x}$$

$$\vec{\tau}_x(\phi,\theta,u,v) = \vec{\tau}_{x1} + \vec{\tau}_{x2} + \vec{\tau}_{x3} + \vec{\tau}_{x4} = -\left(V_1^2 G_{21} + V_2^2 G_{22} + V_3^2 G_{23} + V_4^2 G_{24}\right)\hat{x}$$

$$\vec{\tau}(\phi,\theta,u,v) = \begin{bmatrix} \vec{\tau}_1 \\ \vec{\tau}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \hat{z}' \\ \tau_2 \hat{x}'' \end{bmatrix}$$

where

$$\tau_1(\phi,\theta,u,v) = \vec{\tau}_z \cdot \hat{z}' = \left[G_{11}\ G_{12}\ G_{13}\ G_{14}\right]\left[V_1^2\ V_2^2\ V_3^2\ V_4^2\right]^t \tag{3.4}$$

$$\tau_2(\phi,\theta,u,v) = \vec{\tau}_x \cdot \hat{x}'' = -\cos(\varphi)\left[G_{21}\ G_{22}\ G_{23}\ G_{24}\right]\left[V_1^2\ V_2^2\ V_3^2\ V_4^2\right]^t \tag{3.5}$$

Thus the Euler equations are

$$I_1\ddot{\phi} + D_1\dot{\phi} + K_{m1}\phi = \tau_1(\phi,\theta,u,v)$$
$$I_2\ddot{\theta} + D_2\dot{\theta} + K_{m2}\theta = \tau_2(\phi,\theta,u,v)$$

with the appropriate substitutions.

Simulations

Using the MATLAB and SIMULINK software packages, distributed by Mathworks, the MEMS mirror dynamics are modeled. SIMULINK provides a rapid modeling and design capability, which allows parameters of the mirror to be varied and the performance of the mirror to be analyzed. The generalized torque function, "G" was used in the graphical programming of the dynamics equations as shown in Figure 3.13. SIMULINK also calls an initialization file, shown in Figure 3.12, that defines the physical, electrical and environmental parameters, and hence a structure with these parameters as elements.

The SIMULINK graphical program of the MEMS micro-mirror has three levels. The first level has the inputs and the outputs listed. The inputs are the actuating voltages u and v. The outputs are the tip and tilt angles as a function of time. The second level is the graphical programming of the second order dynamic equations with the coupled 2D Torque Model. The parameters of the mirror enter into the programming at this point as shown in Figure 3.14. The third layer of programming is the 2D Torque Model which is an embedded m-file. Both the voltage inputs and the voltage outputs are a function of time. Initial conditions are also specified for the angular velocity and acceleration. After the model has been validated, SIMULINK is expected to be an accurate model of the mirror dynamics.

Once the Micro-mirror Pointing Model and 2D Torque Model have been validated, SIMULINK allows the designer to vary different parameters and determine the effects on mirror dynamics. Examples of parameters that could be varied are: the mirror height, length, bias voltage, torsional restoring constant, damping constant, and moment of inertia about each axis. Additionally, the mirror dynamics has a positional dependence because of the electrostatic pull-in.

The following are selected observations that can be made from the SIMULINK simulations of the MPM model with the embedded 2D Torque Model and MEMS-scale two-axis tip-tilt mirrors.

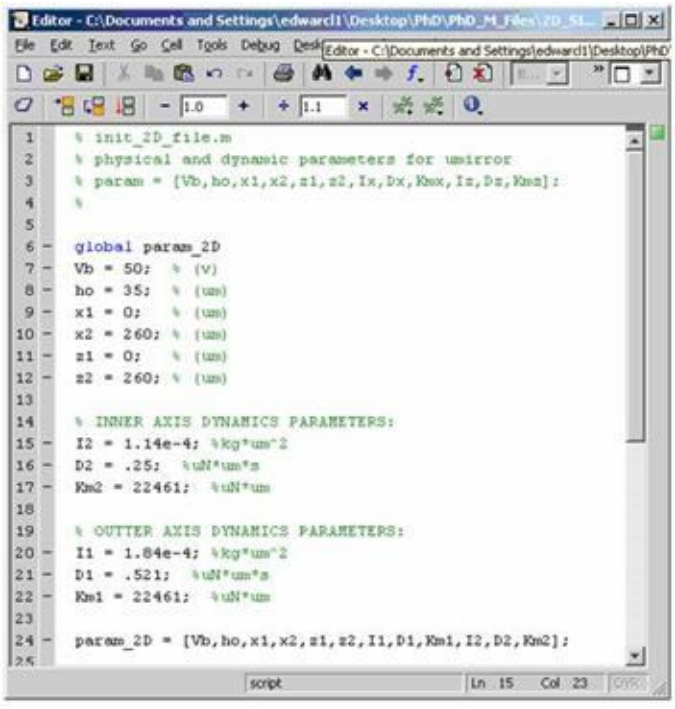


Figure 3.12 Initialization File for MEMS Mirror Simulation

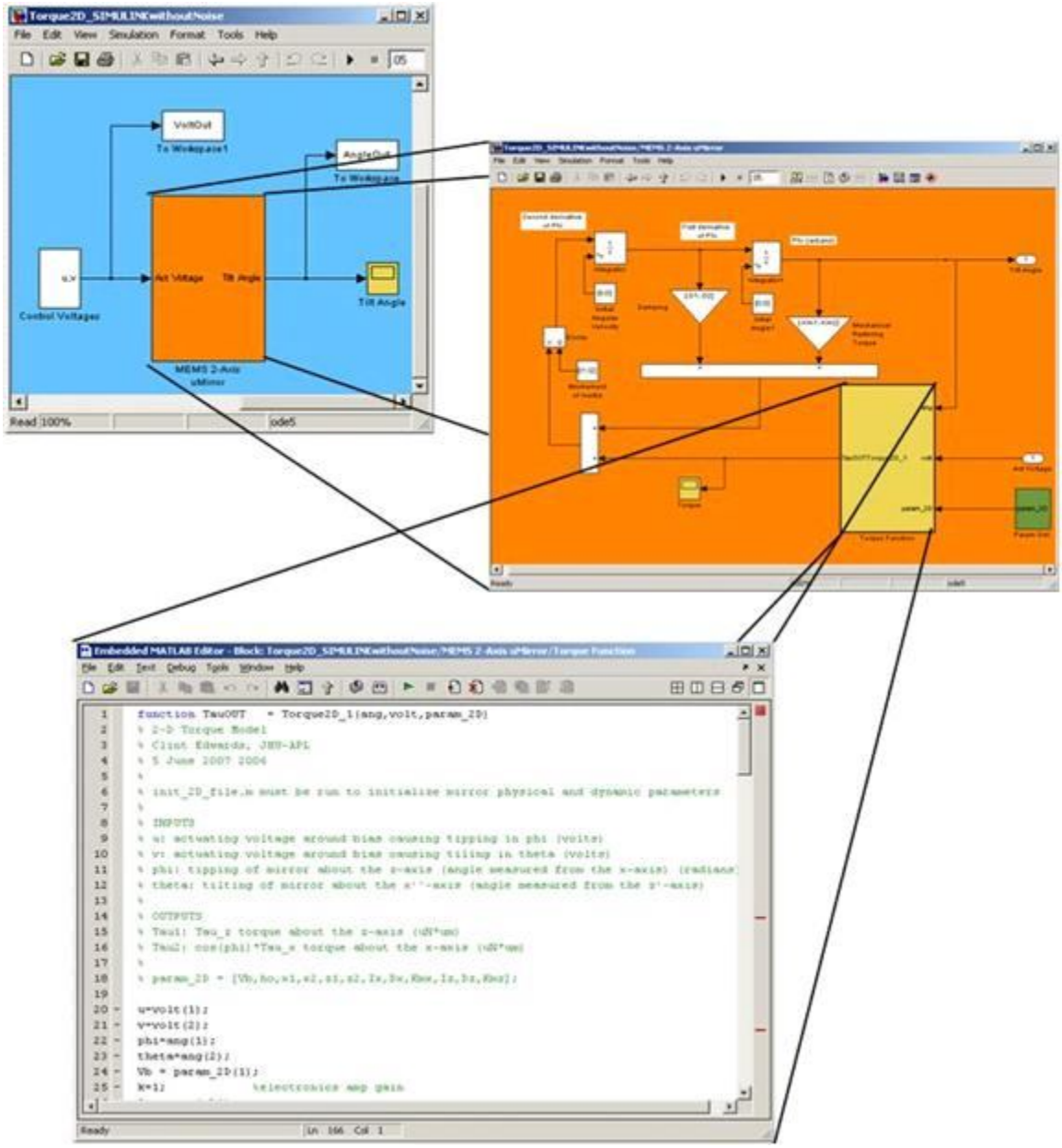


Figure 3.13 Graphical Programming of Dynamic Equations

Cross-axial Coupling

Time-series plots can be created as shown in Figure 3.14, where the first actuating voltage (u) is modulated by a square wave with amplitude of 3 volts, and the second

actuating voltage (v) is held constant at 5 volts. The bias voltage is 50 volts. Figure 3.14 shows a tip-tilt mirror with only one actuating voltage employed in a switching mode. Note the slight cross coupling between the axes as shown in Figure 3.14(a) and 3.14(b). The ϕ is in blue and the θ is in red. Figure 3.14(b) is an enlarged region of Figure 3.14(a). Both plots show the mirror cross-coupling and mirror overshoot. This was observed experimentally as accurate and supports the importance of having a physically complete and accurate torque model that accounts for axial coupling.

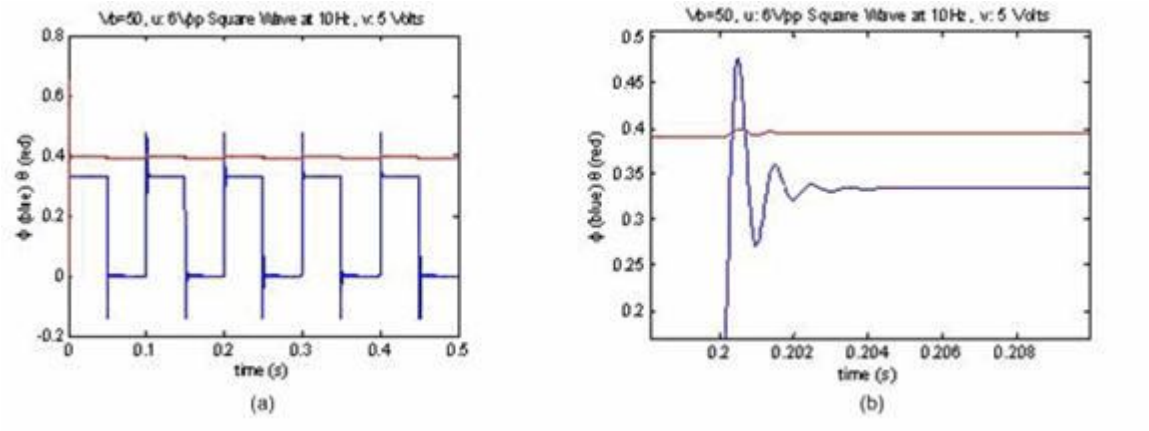


Figure 3.14 (a) Time-series Data from Simulink (b) Mirror Overshoot and Cross-Coupling. The ϕ is in Blue and the θ is in Red. Angles Measured in Degrees.

By increasing the damping of this mirror, critical damping can be achieved. SIMULINK allows us to look at transient effects that are of importance to optical switching or laser beamsteering, where fast precise pointing and control are critical to

overall system performance. Also, when the mirror is operating near the pull-in region, transients can destabilize the mirror and cause it to malfunction or become damaged. For all these reasons a capability to model the mirror before fabrication allows the designer to consider the dynamics before investing in costly manufacturing. Furthermore, having an analytical model illustrates functional dependence on physical, electrical and environmental parameters.

Effects of changing the bias Voltage

Figure 3.15 illustrates a MEMS two-axis scanning mirror with a bias voltage of 50 and another with a bias voltage of 30. If a 5 voltage peak-to-peak sine wave, at 100 Hz, is used to modulate the mirror, it is clear the lower bias voltage causes the mirror to tip at greater angles than the mirror with a bias voltage of 50 volts. Figure 3.16 is the same illustration with the average angle removed so the curves can be compared easier.

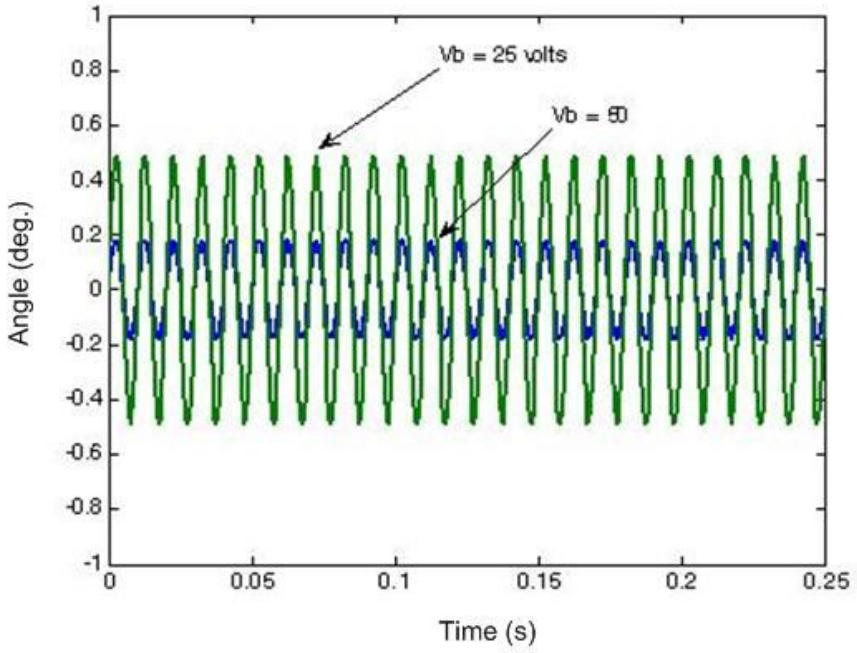


Figure 3.15 Illustration of Effects of Varying Mirror Bias Voltage

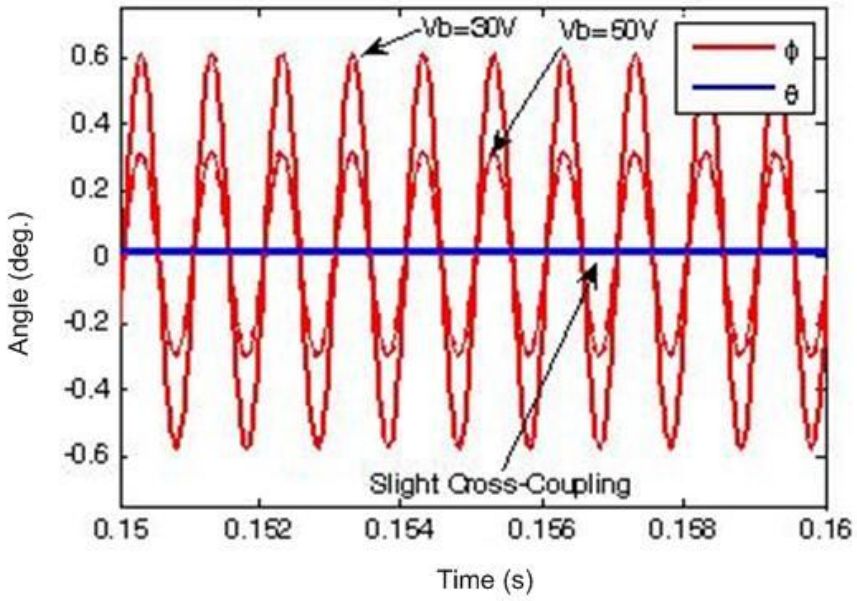


Figure 3.16 Effects of Mirror Bias with Average Angle Removed

Positional Dependence in Scanning Mode

As shown in Figures 3.17 and 3.18, when the mirror is operating in scanning mode the position of the mirror affects the magnitude of the angular excursions for the same inputs voltage single. The blue curve's average angle is 0.5 degrees and the green curves average angel is 1.8 degrees. When the mirror is operating at 1.8 degrees it is closer to pull-in and the same 2 volt sinusoid results in greater angular excursions. When the mirror voltage is $v = 15 + 2\sin\left(2\pi 100t\right)$ the excursions are greater than when they are with $v = 5 + 2\sin\left(2\pi 100t\right)$. Figure 3.18 has the average angular values removed and the differences are more apparent. The voltage bias is constant at 50 volts.

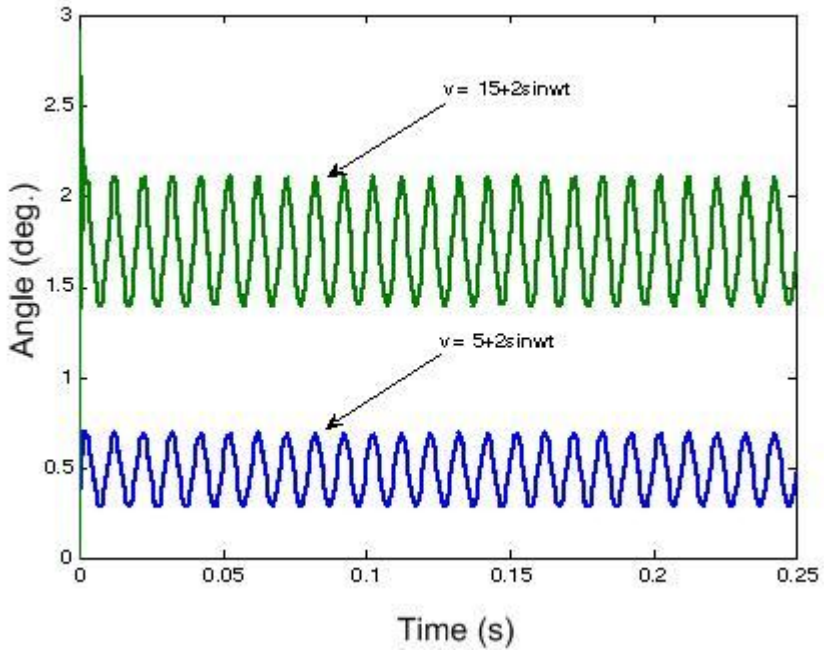


Figure 3.17 Illustration of Positional Dependence of Mirror in Scanning Mode.

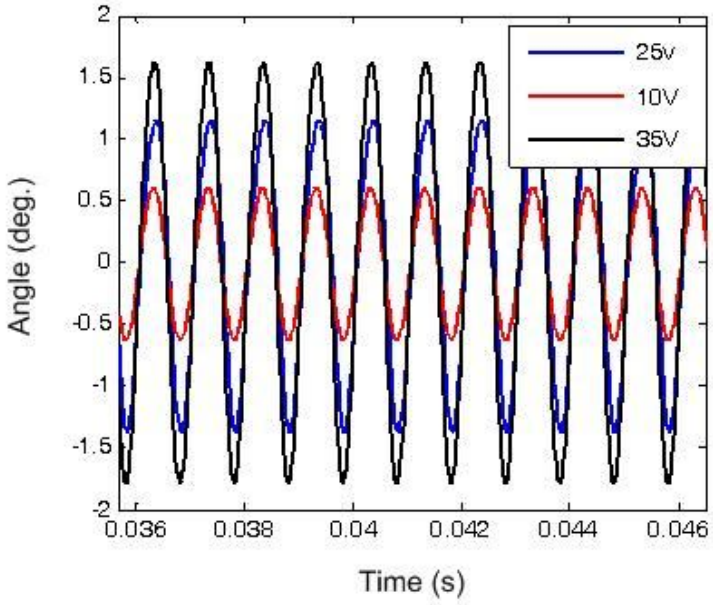
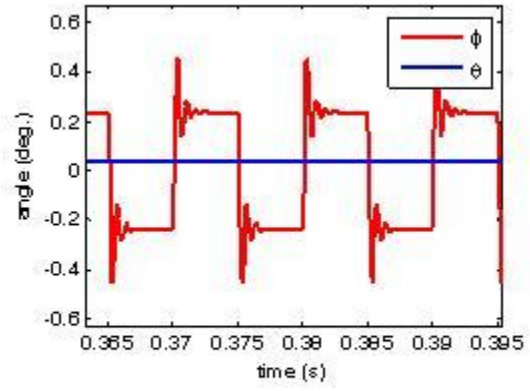


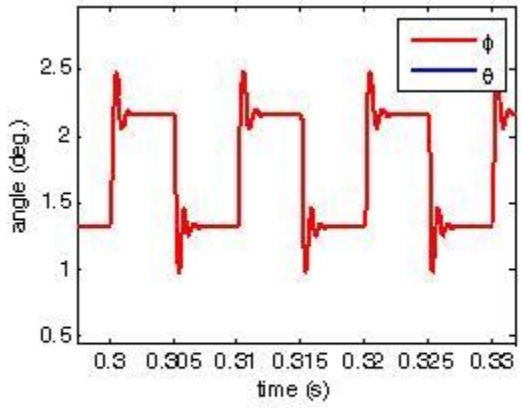
Figure 3.18 Positional Dependence in Scanning Mode (Average Angle Removed)

Mirror in Switching Mode

For optical switching and other applications, it is critical that transient effects are considered in the design of the MEMS mirror. Switching model consists of the mirror moving between two fix angular positions as shown in Figure 3.19(a)-(b). Because the mirror responds differently based upon proximity to pull-in, the transients are greater closer to pull-in. Figure 3.19(a)-(b) illustrates the effects of position (proximity to pull-in) on the transients in the switching mode. For Figure 3.19(a), the mirror is closer to pull-in and the maximum transient is about twice as large as in Figure 3.19(a).



(a)



(b)

Figure 3.19 Illustration of the Positional Effects for Switching Applications

Varying Micro-Mirror Damping of Mirror

For switching and routing applications very fast switching of beams are required but the mirror must be is critically damped. Critical damping can be defined as the minimum damping that does not produce periodic overshoot. It is also the maximum

slew rate that does not produce periodic overshoot. Figure 3.20 shows three curves with different damping values.

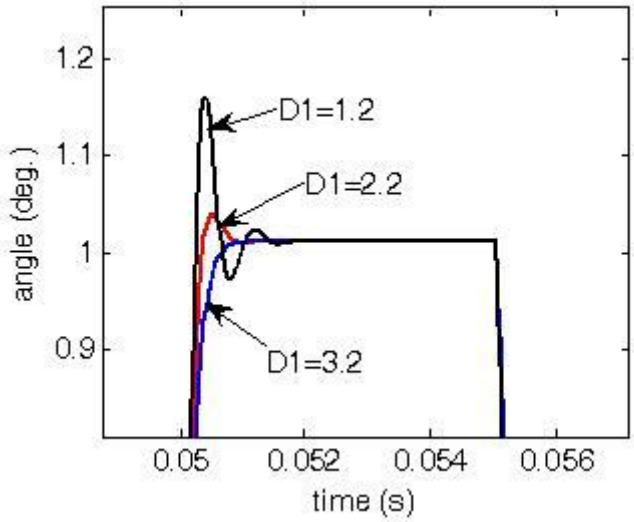


Figure 3.20 Illustration of Different Damping Factors Effects on Transients. Units of Damping are µN*µm*s.

Resonance Frequency

By using a chirp signal as input to the simulations, the resonance can be observed as shown in Figure 3.21. The chirps signal has an increasing frequency of oscillation. Therefore the time that resonance occurs on the horizontal axis of Figure 3.21 can be related to a frequency that is the resonant frequency of the mirror.

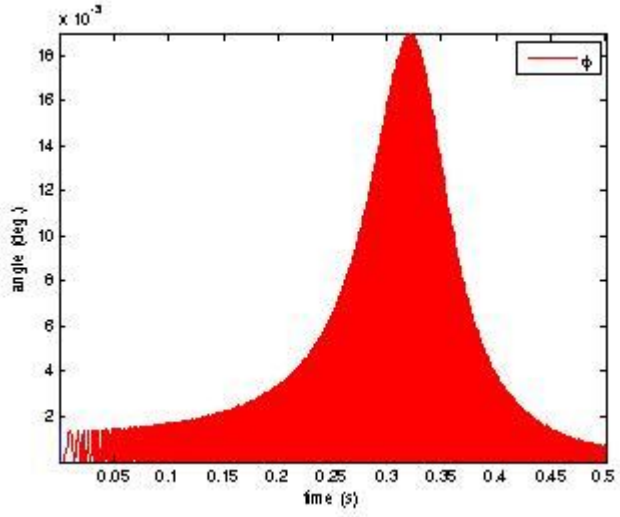


Figure 3.21 Chirp Signal Control Voltage Showing Mirror Resonance

Pull-In

Pull-in can be observed from having a ramping function as the input for only of the voltages. Figure 3.22 shows the pull-in occurring at 0.19 seconds and at an angle close to 2.5 degrees. At pull-in, the mirrors electrostatic attraction is greater than the mirrors hinge's restoring torque. The result is the mirror facet makes contact with the electrode over some other part of the mirror structure.

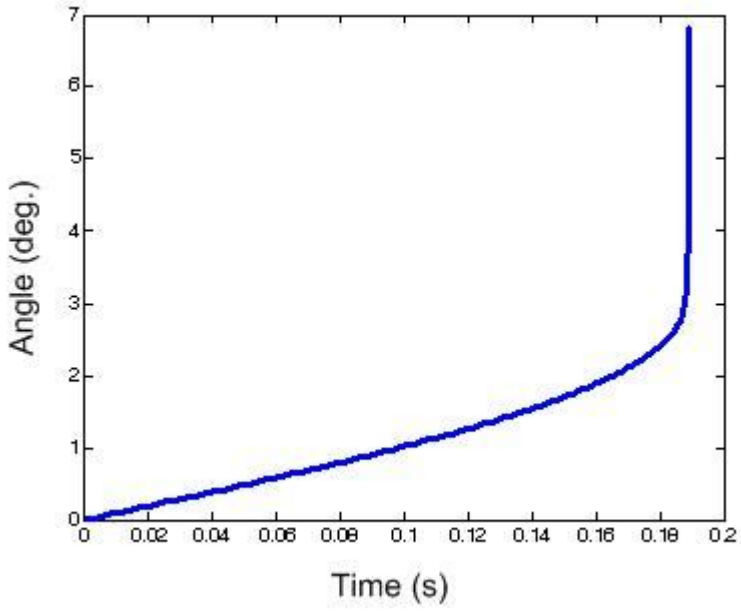


Figure 3.22 Illustration of Pull-In Induced by a Ramping Control Voltage

Conclusions

In this chapter a first principles analytic 2D Torque Model and Micro-Mirror Pointing Model have been developed and simulated. The 2D Torque Model is expressed in terms of electrical, physical, and environmental design parameters. A generalized torque function, "G", was developed for software implementation of the 2D Torque Model.

Euler dynamic equations have been used to derive the harmonic oscillator model with coupled torsional expressions for the two axes that is used in the Micro-Mirror Pointing Model. Pull-in was expressed in terms of the poles of the Euler equations

and the 2D Torque Model. The singularities of G, when either or both of the angles of the mirror are zero, have been analyzed. It is found that the singularities are removable and the model is valid at these limiting points. The torque model is physically complete, working over the full range of angles.

Finally, samples of SIMULINK simulations are shown illustrating the effects of position as well as varying design parameters on mirror performance.

Chapter 4: Validation of the Micro-Mirror Pointing Model

Experimental Apparatus and Measurements

Measurements to validate the 2D Torque Model and MPM model were taken in the Space Department's RF Engineering Group Optical Communications Lab of the Johns Hopkins University's Applied Physics Laboratory in Laurel, Maryland. Figure 4.1 is a diagram and Figure 4.2 is a photo of the experimental set-up. The measurement system included a vacuum chamber housing a MEMS micro-mirror and other instruments for controlling, measuring, and inspecting the mirror. While model validation only required measurements at standard temperature and pressure, the vacuum system configuration insured structural stability and cleanliness. As can be seen in the diagram, a laser beam passed through a pin-hole aperture and continued through a quartz window into the vacuum chamber and reflected off of the MEMS two-axis tip-tilt mirror. The laser beam then exited via a quartz window and reflected off of a mirror fixed to a goniometer finally hitting a quad cell detector. The MEMS mirror angular orientation responds to a drive voltage controller originally designed for Piezoelectric actuated high-voltage requirements. Modulation of the control voltage was accomplished for this MEMS device with a function generator capable of synthesizing a variety of signals. As shown in Figure 4.1, the mirror and the quad cell detector voltage levels were monitored and recorded. This was done using National Instruments (NI) LabView software working in concert with a NI data acquisition (DAQ) card. Figure 4.2 shows the vacuum chamber with the Plexiglas lid

removed and the actual components that were used to validate the 2D Torque Model and related dynamics (MPM model.) Table 4.1 gives a list of experimental components and their part numbers and significant specifications.

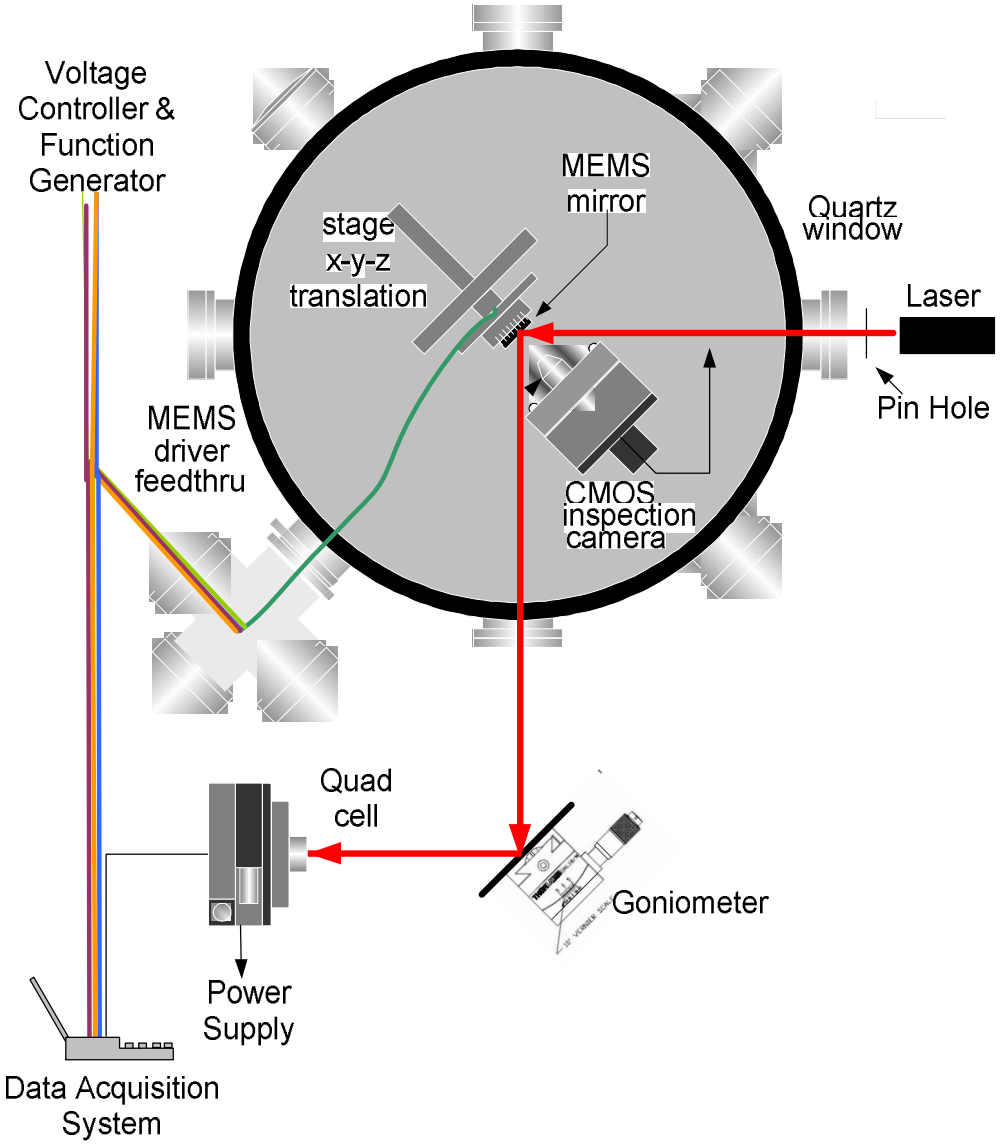


Figure 4.1 Diagram of the Experimental Set-up

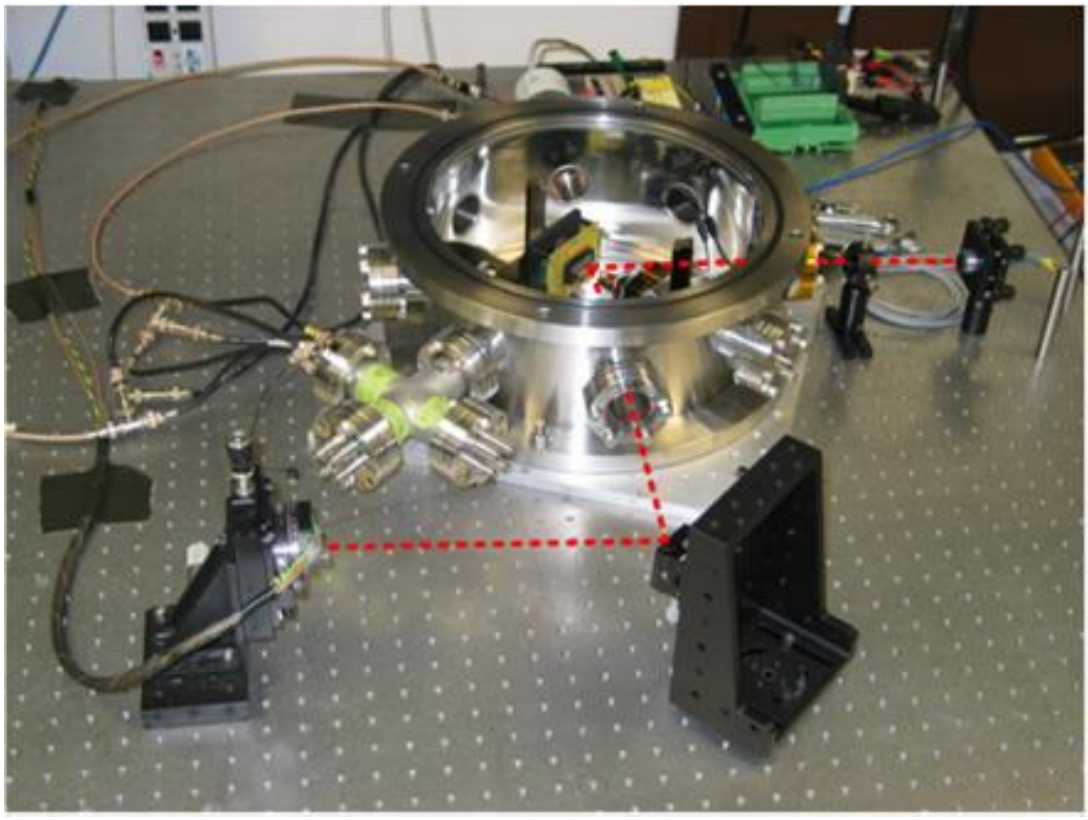


Figure 4.2 Photo of the Experimental Set-up

Table 4.1 Experimental Apparatus

COMPONENT	**PART NUMBER**
LASER	3.3mW, 670nm Lasiris StockerYale laser diode with single-mode optical fiber and collimator
PIN HOLE	Thorlab 1000 μm, Circular
MEMS MIRROR	MEMS Optical Inc. Two-Axis (Tip-Tilt) 520 μm Mirror
QUAD CELL	PSS-QP50-6SD, Pacific Silicon Quadrant Array Diode
GONIOMETER	Thorlab GNL 30, Angular Resolution: 5 μ rad
DATA ACQUISITION SYSTEM	National Instruments LabView and AI-16XE-50 Data-Acquisition-Card
POWER SUPPLY (QUAD CELL)	Agilent 3630A Programmable DC Power Supply, Range: 0-15V
VOLTAGE CONTROLLER	Thorlab MDT693A Piezo Controller, Range 0-100V, Resolution 0.1V
CMOS INSPECTION CAMERA	SONY MP4900 Camera
FUNCTION GENERATOR	Agilent 33250A Function/Arbitrary Waveform Generator, Voltage 0-5V, 1MHz Max. Frequency

Component Descriptions

A 670nm Lasiris StockerYale 3.3mW laser was coupled via a single-mode optical fiber, as shown in Figure 4.3. It was mounted such that it illuminated a MEMS Optical Inc. micro-mirror in the vacuum chamber and then reflected off the gold facet of the mirror onto a quad cell, as shown in Figure 4.1 and Figure 4.2.

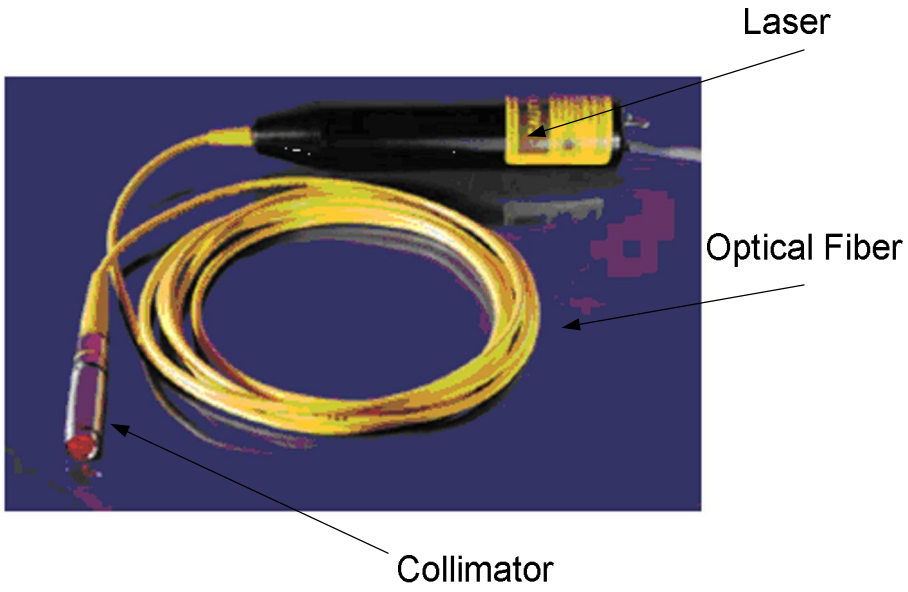


Figure 4.3 670nm Lasiris StockerYale 3.3mW laser

The MEMS Optical Inc. two-axis (tip-tilt) mirror, as shown in Figure 4.4, has an octagonal shape with reflective facet diameter of 520µm. The rotating frame and hinges are also visible in Figure 4.4. The electrical bonding pads can also be seen in this picture.

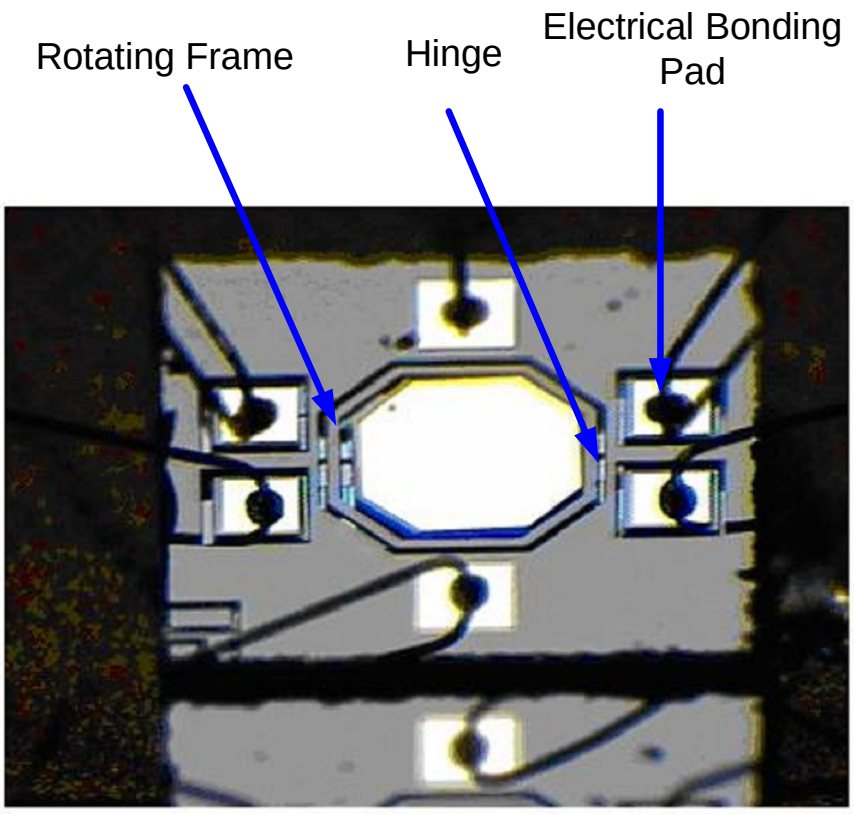


Figure 4.4 MEMS Optical Micro-mirror. Mirror Facet 520µm Across.

The mirror was biased at 50 volts with a Thorlab MDT693A Piezo Controller shown in Figure 4.5. The voltage controller received the function generator's signal through the BNC connectors on the front of the chassis as shown in Figure 4.5. The function generator's signal was multiplied by 10 and added to the DC signal of the voltage controller. An Agilent 33250A Function/Arbitrary Waveform Generator, shown in Figure 4.6, supplied modulating signals such as square or sine waves.

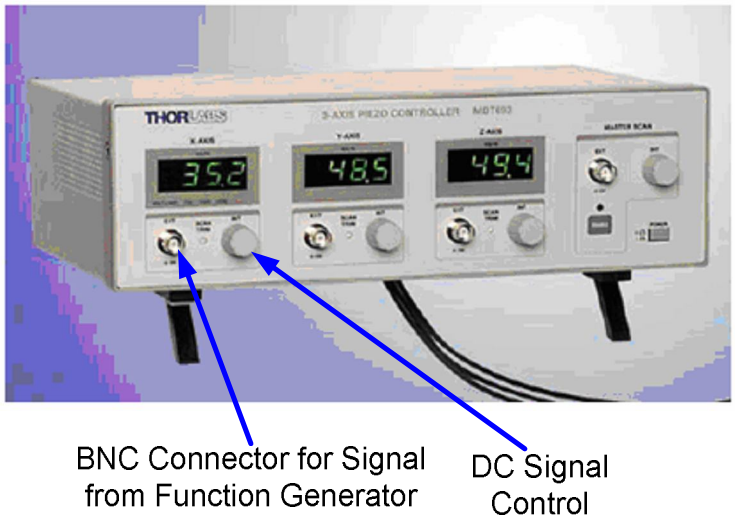


Figure 4.5 Thorlab MDT693A Piezo Controller

Figure 4.6 Agilent 33250A Function/Arbitrary Waveform Generator

As the mirror was actuated, the beam moved on the quad cell. The PSS-QP50-6Sd Quad Cell, shown in Figure 4.7, was manufactured by Pacific Silicon Sensor. The movement on the quad cell was recorded in the form of time-series data by LabView using a National Instruments AI-16XE-50 Data-Acquisition-Card, which had a

maximum sampling rate of 200k samples-per-second (S/s.) For this experiment sampling of each channel into LabView was held constant at 50kS/s.

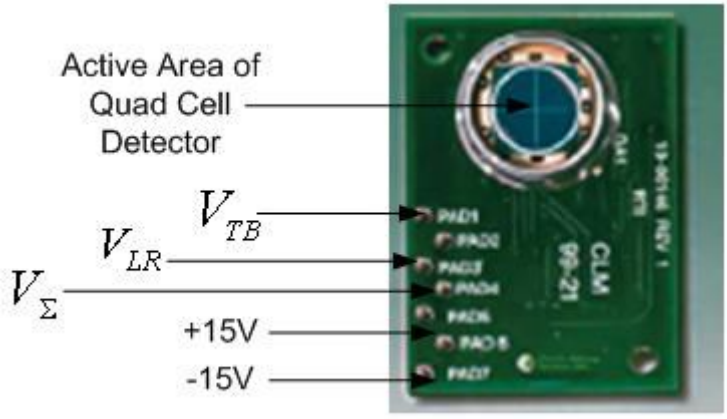


Figure 4.7 Pacific Silicon Sensor's PSS-QP50-6Sd Quad Cell

The quad cell generates three output voltages as in indicated in Figure 4.7. Two of the voltages are differenced between the top and bottom (TB) and left and right (LR) halves of the quad cell. The third voltage is the sum of all four quadrants of the quad cell. When the laser is perfectly in the center of the quad cell detector, both V_{LR} and V_{TB} are zero. When the laser is off-center of the quad cell, as shown in Figure 4.8, the V_{LR} and V_{TB} are non-zero. The calibration process of the quad cell related the V_{LR} and V_{TB} to displacements on the quad cell as well as calculated the error (or uncertainty) for this measurement.

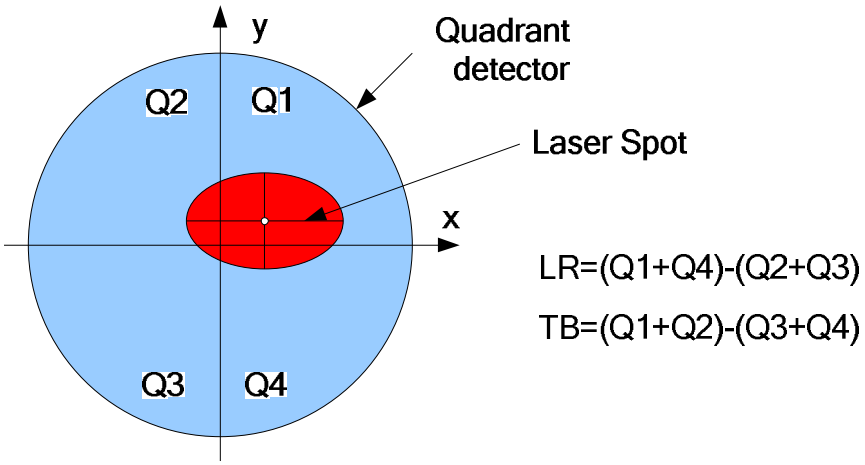


Figure 4.8 Quad Cell with Off-Centered Laser Spot

It was important that the laser light be as highly collimated as possible so as to achieve the optimal spot size on the quad cell. The optimal spot size to maximize range and sensitivity is approximately half of the active area of the detector. As shown in Figure 4.1 and Table 4.1, a 1mm pinhole was placed just before vacuum chamber and MEMS mirror to aperture stop the beam and prevent light from reflecting off other surfaces near the mirror facet. A smaller pinhole would increase the angle of divergence of the beam. The longer the distance, called the optical arm length (OAL), that the laser light travels from the MEMS mirror to the quad cell, the more precisely the tilt angles of the mirror can be measured, but the longer the OAL the larger the spot becomes on the quad cell reducing over all sensitivity. For model validation the experimental apparatus was set-up to provide sub-micro-radian angular precision measurements.

Alignment

Figure 4.1 and 4.2 include several components that aided in the precise alignment of the optical system to reduce systemic errors. These included the X-Y-Z translational stage and the CCD inspection camera. The X-Y-Z stage combined with the inspection camera allowed for positioning of the mirror so that the reflected laser beam was as clean and Gaussian as possible. The inspection camera was linked to a television that showed the mirror's illumination. Figure 4.9 shows the television monitor with the mirror image on the screen. The alignment only needed to be done after each time the components were changed. The goniometer and the X-Y-Z translational stage do not drift over time. This was monitored over several months with no change detected.

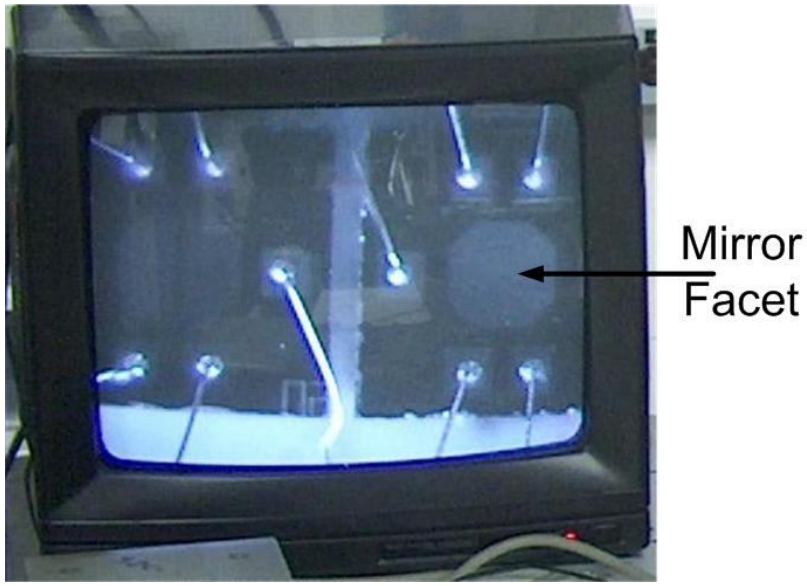


Figure 4.9 MEMS Mirror Viewed From CCD Inspection Camera

Calibration and Precision Procedure

The calibration of the quad cell enables the normalized measured voltage from the quad cell detector to be directly related to the actual displacement of the laser light spot on the quad cell detector surface in both x and y directions.

To relate the output voltage to the displacement, the quad cell was calibrated with a Thorlab GNL30 Goniometer as shown in Figure 4.1 and Figure 4.10. The goniometer with a mirror attached to the rotating surface allowed a precise change in angle to be related to the change in the normalized V_{LR} and V_{TB} of the quad cell. If the distance is precisely measured from the mirror on the goniometer to the quad cell, then a displacement on the face of the quad cell detector can be determined as shown in Figure 4.11. When the laser is perfectly centered in the quad cell, the normalized voltages from the quad cell will be zero. Once a table was generated relating the quad cell's displacement to voltage, it was incorporated into an interpolated look-up table for intermediate values based on Figure 4.12(a)-(b).

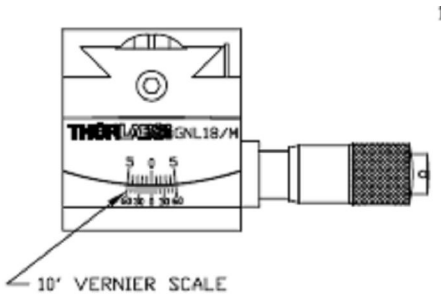


Figure 4.10 Thorlab GNL30 Goniometer

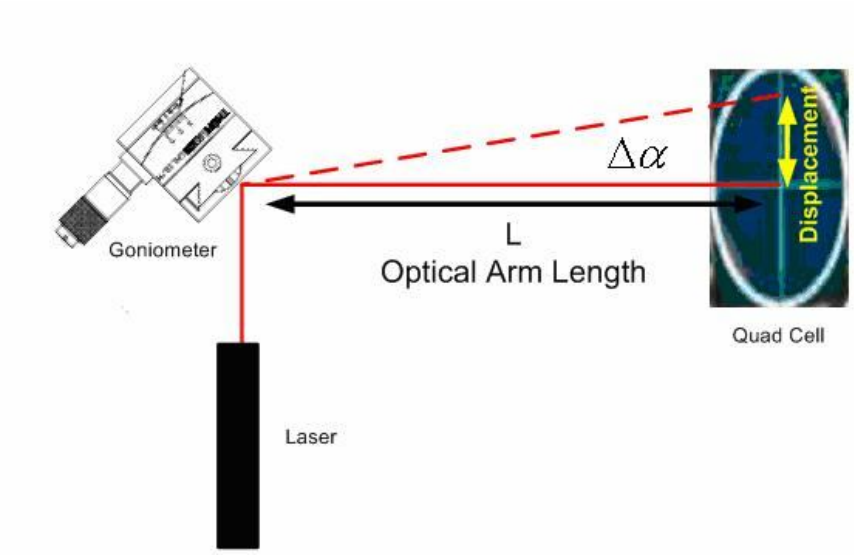
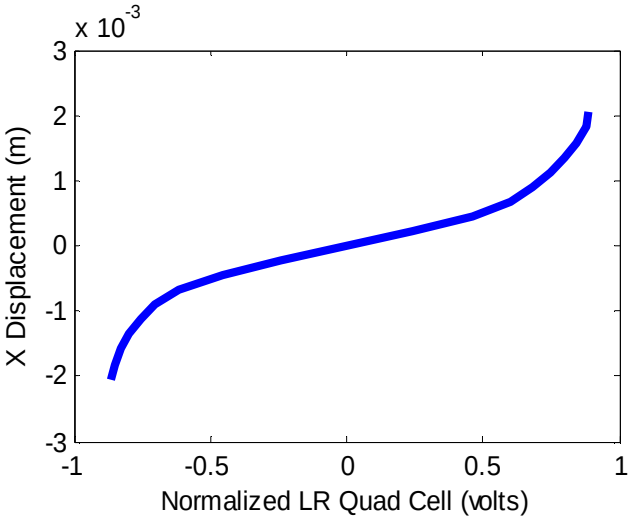
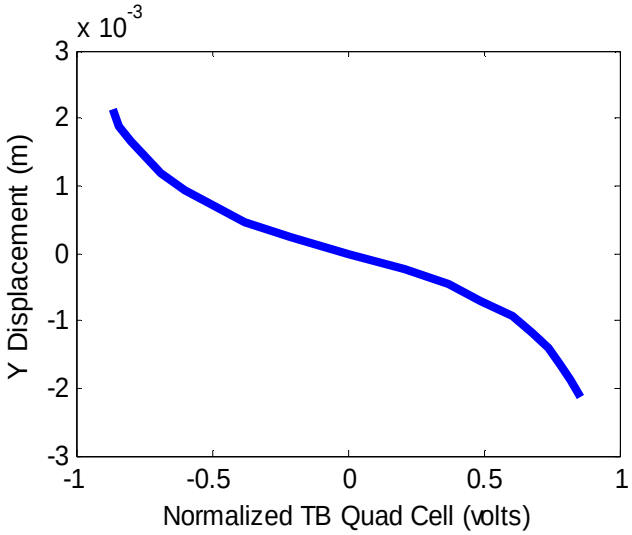


Figure 4.11 Calibration Process for Quad Cell



(a)



(b)

Figure 4.12 (a)-(b) Transfer Functions Relating Displacement on Quad Cell and Normalized (a) Top-Bottom (b) Left-Right Voltages.

Once the quad cell was calibrated, the entire MEMS setup as shown in Figure 4.1 and Figure 4.2 can be calibrated and the precision and errors (or uncertainty) in the measurements determined. The setup was the same as Figure 4.11 with the OAL equal to 0.85 meters.

Calibration Errors

Because no measurement has perfect precision, we must evaluate the error or "uncertainty" of the measurements in the experiment. The error is given in terms of $\Delta\phi$ (Similar calculations can be done for $\Delta\theta$.) In this formulation, all the errors are assumed to be zero mean and Gaussian with standard deviation's $\Delta\phi$ or $\Delta\theta$ for measurements of ϕ and θ respectively.

These are the three sources of error:

1. Optical Arm Length, ΔL
2. Orientation of the quad cell, ΔY
3. Calibration of the quad cell error, ΔX

The errors can be combined in quadrature to yield an angular error of

$$\Delta\phi = \sqrt{\left(\varepsilon_X\right)^2 + \left(\varepsilon_Y\right)^2 + \left(\varepsilon_L\right)^2}$$

where

$$\varepsilon_X = \frac{\partial \phi}{\partial X} \Delta X$$
$$\varepsilon_Y = \frac{\partial \phi}{\partial Y} \Delta Y$$
$$\varepsilon_L = \frac{\partial \phi}{\partial L} \Delta L$$

The partial derivatives are analytic terms whereas the ΔX, ΔY, and ΔL terms are errors that are measured or otherwise derived. The analytic expression is

$$\phi = \tan^{-1}\left(\frac{\sqrt{X^2 + Y^2}}{L}\right)$$

For small angles

$$\phi \cong \frac{\sqrt{X^2 + Y^2}}{L}$$

The partial derivatives of this expression are

$$\frac{\partial \phi}{\partial X} = \left(\frac{1}{2}\right)\frac{2X}{L\sqrt{X^2 + Y^2}}$$

$$\frac{\partial \phi}{\partial Y} = \left(\frac{1}{2}\right)\frac{2Y}{L\sqrt{X^2 + Y^2}}$$

$$\frac{\partial \phi}{\partial L} = \left(-\sqrt{X^2 + Y^2}\right)\frac{1}{L^2}$$

Error ΔY

The ΔY can be minimized by ensuring the axis of the quad cell and the goniometer are mutually orthogonal as shown in Figure 4.1. This is done after centering the laser spot on the quad cell by varying one axis of the goniometer and by changing the orientation of the quad cell until the orthogonal voltage difference is not affected by the variation. Time-series data can be averaged to reduce the error due to the quad cell voltage read-out noise or "ripple."

Error ΔL

The standard deviation or error in the optical arm length, ΔL, is 1mm.

Error ΔX

From the calibration process, we can determine the relationship, $X = R(V_{LR})$ for displacement on the quad cell as a function of the voltage from the quad cell and represented by the plot in Figure 4.12. The error in the displacement, ΔX, was related to error in the left-right voltage by the derivative, $R'(V_{LR})$, which in this case was determined numerically for the curves in Figure 12 (a)-(b). The +/-15V quad cell bias power supply was relatively noisy due to the nature of its switching circuits. This error, measured as a ripple, $\Delta V_{\text{Read-Out}}$, means that the error of the quad cell displacement is

$$\Delta X \approx R'(V_{LR}) \cdot \Delta V_{\text{Read-Out}}$$

Because the voltage read-out noise was 1mV, ε_X was found to be 1.1 μm and was the dominant term as shown in Table 4.2. The term ΔL is 1mm and is larger than ΔX but for displacements on the quad cell and longer OAL, ε_L was reduced well below the dominant error term.

Table 4.2 gives the error terms for:

$$L = 0.85m$$
$$X = 1\times10^{-4} m$$
$$Y = 1\times10^{-7} m$$

For this case the errors or uncertainties were:

$$\Delta L = 0.001 m$$
$$\Delta X = 1\times10^{-6} m$$
$$\Delta Y = 1\times10^{-7} m$$
$$\Delta V_{\text{Read-Out}} = 1mV$$
$$R' = \frac{\Delta X}{\Delta V_{LR}} = 1\times10^{-3} m/V$$

The ϕ error for this example is:

$$\Delta\phi = 1.1\times10^{-6} \text{ rad}$$

Table 4. 2 Error Components for ϕ Calibration

ε_X (m)	1.1×10^{-6}
ε_Y (m)	1.2×10^{-14}
ε_L (m)	1.3×10^{-7}

Methodology

In order to validate the 2D Torque Model and MPM model it must be shown that close agreement exists between experimental measurements and model results. This was done by comparing the experimental results with those obtained by simulating the 2D Torque Model and MPM model for steady-state as well as the transient motions of the mirror. The procedure for this validation was as follows:

1. Determine the effective lengths $\ell_{x_{eff}}$ and $\ell_{z_{eff}}$ and torsional restoring constants K_{m1} and K_{m2} for a mirror. The "effective length" of the mirror accounts for fringing effects of the electric field. Since the torsional restoring constants are influenced by variations in the fabrication process it is also appropriate to determine these constants for each mirror that is tested. By varying the mirror angles using a set of constant actuating voltages, $\{u_m\}$ and $\{v_m\}$, a series of measured steady-state angles $\hat{\phi}_m$ and $\hat{\theta}_m$ versus voltages were determined. Given values for the parameters K_{m1}, $\ell_{z_{eff}}$, K_{m2}, and $\ell_{x_{eff}}$ the 2D Torque and MPM models predicted angles ϕ_m and θ_m which were solutions to the steady-state equations:

$$K_{m1}\phi_m = \tau_1\left(\phi_m, 0, u_m, 0, \ell_{z_{eff}}\right)$$
$$K_{m2}\theta_m = \tau_2\left(0, \theta_m, 0, v_m, \ell_{x_{eff}}\right)$$

The parameters were then varied to produce a least mean square error, i.e.,

$$\left(\hat{K}_{m1}, \hat{\ell}_{z_{eff}}\right) = \left\{\left(K_{m1}, \ell_{z_{eff}}\right) such\ that\ \sum_m \left(\phi_m - \hat{\phi}_m\right)^2\ minimized\right\}$$

$$\left(\hat{K}_{m2}, \hat{\ell}_{x_{eff}}\right) = \left\{\left(K_{m2}, \ell_{x_{eff}}\right) such\ that\ \sum_m \left(\theta_m - \hat{\theta}_m\right)^2\ minimized\right\}$$

In this work the best-fits for "effective lengths" and torsional restoring constants were determined using the Matlab least-mean-squared (LMS) fitting simplex search algorithm.

2. The moment of inertia, I_i, can be calculated from the mirror size, material composition or from the manufacturer specification sheet or estimated from the dynamics.

3. The damping constants D_1 and D_2 were determined by examining the frequency response of the mirror. The damping constants of the system are equal to the full-width-at-half-max (FWHM) of the resonance peak, $\Delta\omega$, multiplied by the moment of inertia for that axis of the mirror, or

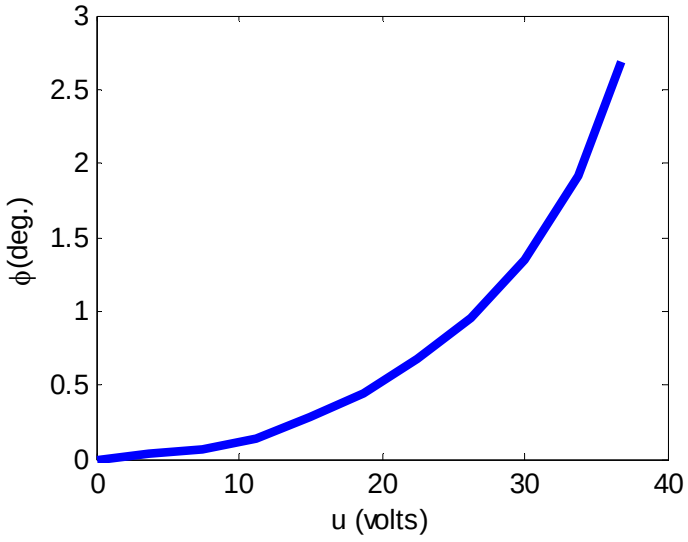
$$D_i = \Delta\omega_i I_i$$

4. With all of the dynamic constants determined for each axis of the mirror, the experimental time-series data were compared to the MPM model simulated results using MATLAB and SIMULINK. Transient effects in the experimental data and simulation were also compared.

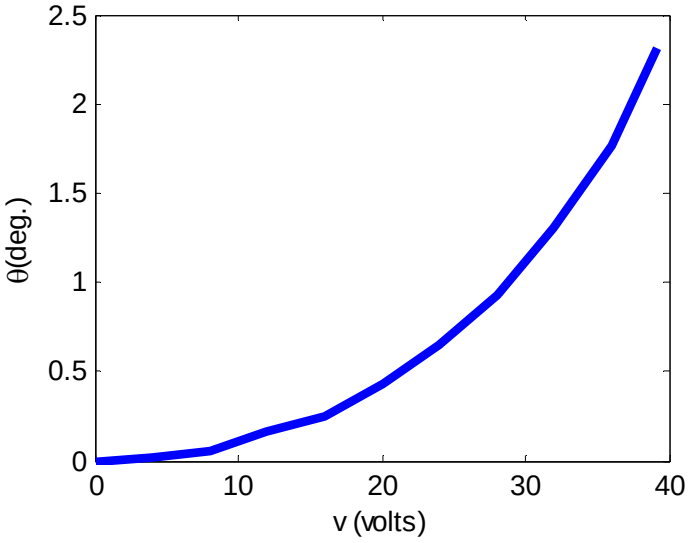
Physical Parameter Determination

Two sets of measurements were taken on a mirror manufactured by MEMS Optical Inc., who provided manufacturing specifications for these mirrors [1], [7]. The measured mirror angle in the ϕ and θ directions versus voltage is shown in Figure 4.13(a)-(b) respectively. The "effective lengths" of the mirror as well as the torsional restoring constants for the mirrors are given in Table 4.3. The "effective lengths" were longer than the 520 μm physical facet lengths to account for fringe effects. The effects are relatively small but are accounted for. Plots relating the steady-state angles ϕ and θ versus the applied voltage for the MPM model versus experimental measurements are shown in Figure 4.14(a)-(b)

Although the effective length and restoring constants K_{mi} are "fitted" using steady-state measurements, the formulation for the 2D Torque Model is still explicitly in terms of the physical parameters. This is different from other attempts to model MEMS mirrors with polynomial fitting [21], [29], [90]. In these models, the coefficients of polynomials have no explicit relationship with fundamental physics and give the designer no indication of how varying one physical parameter may affect the entire device's performance.



(a)



(b)

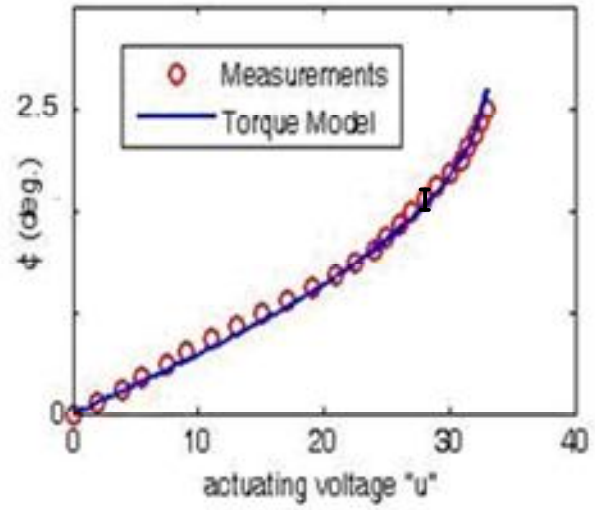
Figure 4.13 (a)-(b) Measurement Based Relationships Between the Angles ϕ and θ Versus the Actuating Voltages u and v

Table 4.3 Mirror Calibrated Torsional Restoring Constant and Effective Length

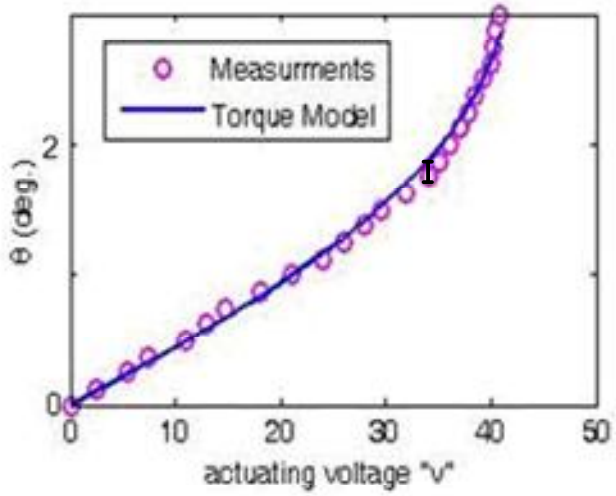
$K_{m1}\left(\mu N \mu m\right)$	22605	$K_{m2}\left(\mu N \mu m\right)$	18801
$\ell_{z_{eff}}\left(\mu m\right)$	525	$\ell_{x_{eff}}\left(\mu m\right)$	522

Allowing for the fitting of physical parameters, such as torsional restoring constants and effective lengths, highlights a strength of the analytical 2D Torque Model and the dynamic equations. Since the MEMS fabrication process inherently results in variations in performance based on physical dimension variations, the model is essentially being calibrated for each mirror to account for these variations. However, through the 2D Torque and MPM model, an explicit relationship still exists between the mirror parameters and the torque expression or angles. Calibration is based on a limited set of measurements.

Additionally, in determining these physical parameters from measurements, the designer learns more about the variability of the fabrication processes. The model then can be used to "reverse-engineer" the process, i.e. starting with critical design parameters once can determine the required fabrication process tolerances.



(a)



(b)

Figure 4.14 (a)-(b) Measurements Vs. Model with Optimal K_m and ℓ_{eff} .

The damping constants are determined next. This is done for both axes but explicitly shown for only the ϕ angle in Figure 4.15. To do this the function generator

provided the control voltage and swept a sine wave over a range of frequencies, and the magnitude of the response to that frequency was recorded with Labview similar to Figure 4.15.

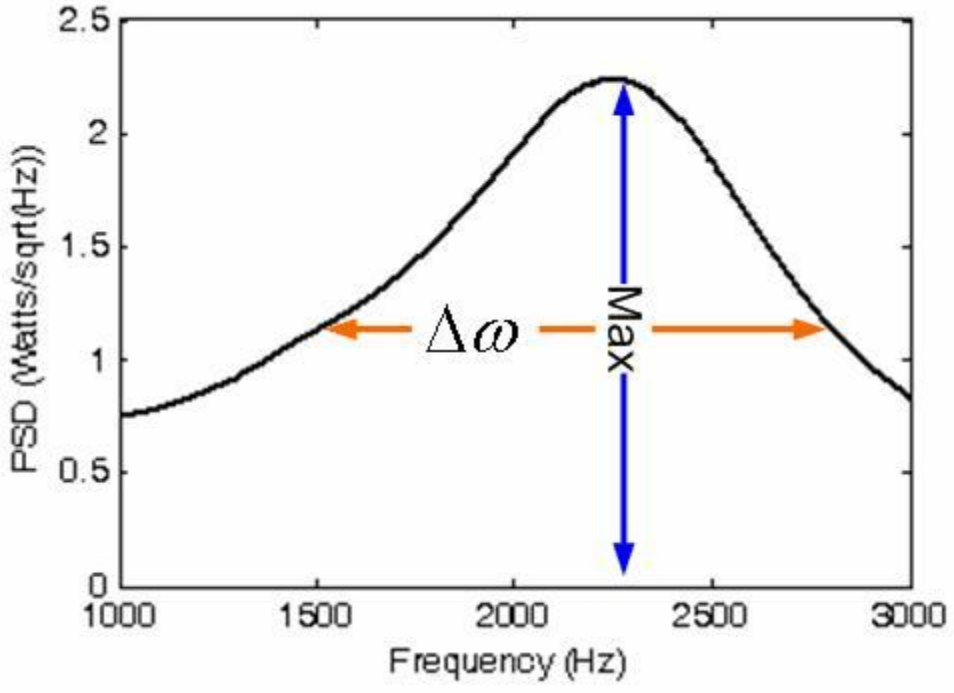


Figure 4.15 Mirror ϕ Frequency Response

From plots similar to Figure 4.15, the FWHM (half power points) value, $\Delta\omega_i$, was determined. This value is multiplied by the moment of inertia to calculate the damping constant $\left(D_i = \Delta\omega_i I_i\right)$ for one axis. The MEMS Optical Inc. spec. sheet gives the moments of inertia are given in Table 4.4, and the estimated damping constants are given in Table 4.5.

Mirror Dynamics Constants

With the dynamic constants determined, measured transient effects of the mirror facet angles (ϕ,θ) for a given set of actuating voltages, (u,v) were compared to simulated results using the MPM model. The mirror was operated at multiple angles with a bias voltage of 50 volts $(V_B = 50)$.

Table 4.4 MEMS Mirror Moment of Inertia

$I_1\left(\mu N \mu m\ s^2\right)$	1.84e(-4)
$I_2\left(\mu N \mu m\ s^2\right)$	1.14e(-4)

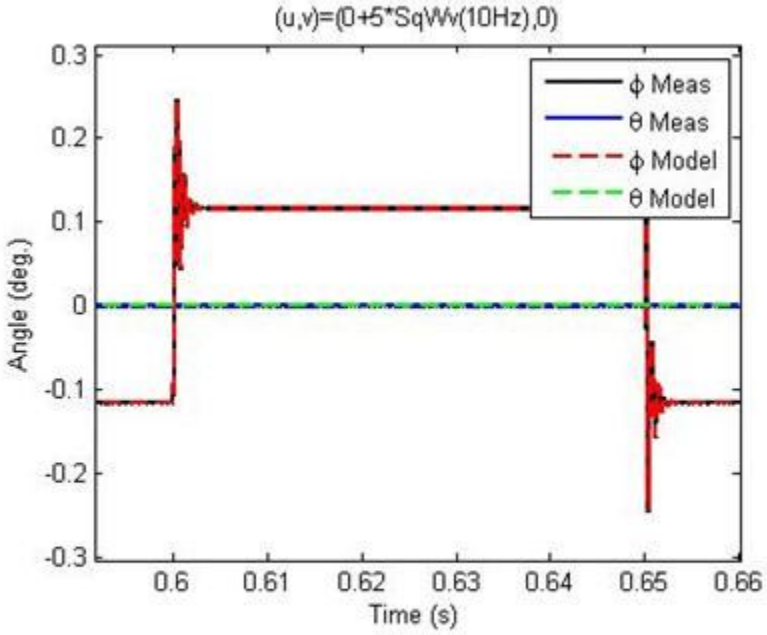
Table 4.5 Mirror Damping Constants

$D_1\ (\mu N\ \mu m\ \mathrm{s})$	0.615
$D_2\ (\mu N\ \mu m\ \mathrm{s})$	0.285

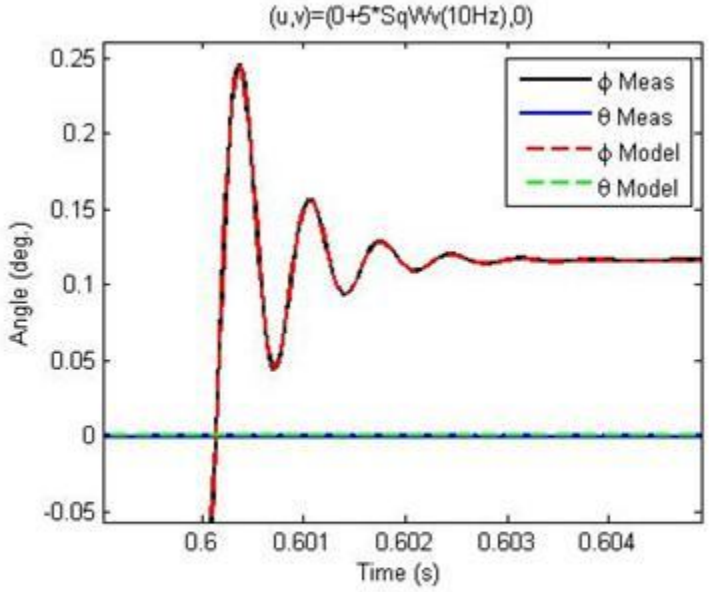
In addition to a constant actuating voltage (u,v), the mirror was also driven at 10 Hz with a square wave with a peak-to-peak voltage of 5 volts. The results of these measurements were compared to the MPM model simulated results. The modulation frequency of 10Hz is low enough that both transient and steady-state effects can be observed.

Measurements Vs. Model

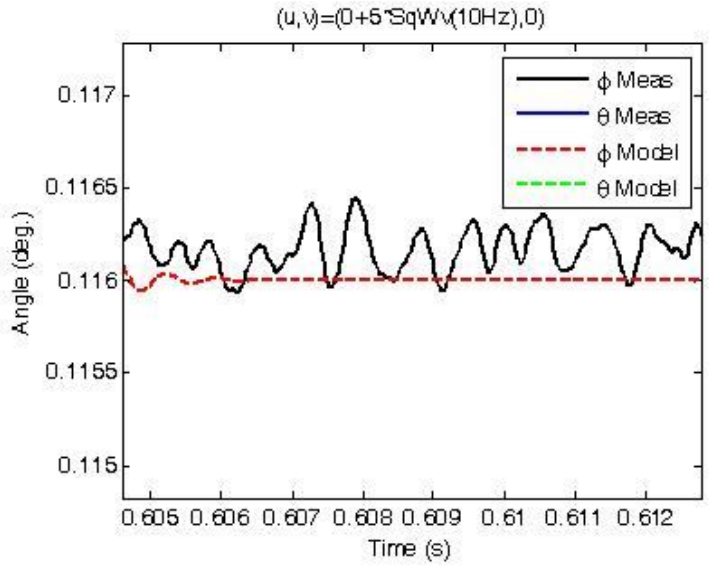
A complete set of measurements versus MPM model simulated results are found in Appendix A. This chapter primarily highlights mirror operations in the full 2D mode. Figure 4.14 illustrated the precision of the fitting process. Figure 4.16(a) shows an entire cycle of the square wave. Figure 4.16(b) illustrates a close-up view of the transient. The model and the measurements were still indistinguishable at this enlarged view. At greater magnification, Figure 4.16(c) distinctly shows the difference in the steady-state values of the MPM model and experimental measurements. The mirror continues to jitter for reasons that will be discussed in Chapters 5 and 6. Figure 4.16(d) shows the peak of the first transient oscillation overshoot and indicates that the damping constant is slightly smaller than the measurements. As is shown in the complete set of measurements and model comparisons, the model accurately predicts the transients as well as the steady-state values for both the single-axis and 2D actuation. Only near pull-in did the non-linearities of the mirror begin to increase, and show slight differences between the experimental measurements and the simulated MPM model.


(u,v)=(0+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0.6
0.61
0.62
0.63
0.64
0.65
0.66

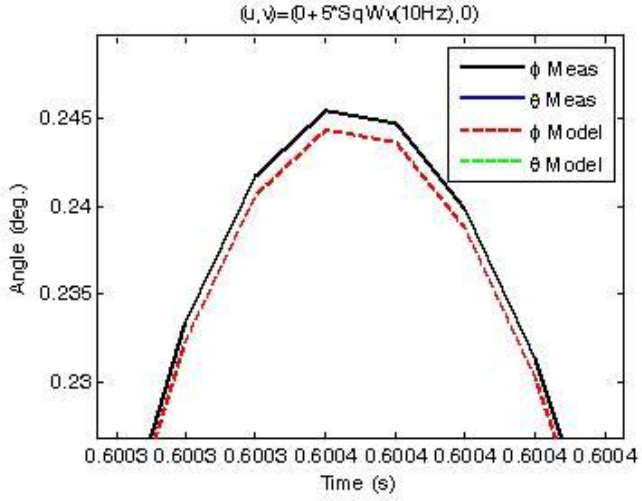
(a)


(u,v)=(0+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.25
0.2
0.15
0.1
0.05
0
-0.05
0.6
0.601
0.602
0.603
0.604

(b)



(c)



(d)

Figure 4.16 (a)-(d), Plot of Model vs. Measurements for 5 Volt Peak-to-Peak Square Wave Modulating ϕ Axis at 10 Hz (a) Single Cycle of Square Wave (b) Enlarged view of Transients (c) Enlarged View of Steady-State (d) Enlarged View of Peak Transient.

The following figures illustrate the MPM model and compare the model to experimental measurements. Figures 4.17 (a)-(b) illustrate the mirror operating in the 2D mode. Note that in Figure 4.17(b) the model correctly follows, or predicted, the slight cross coupling in the θ-axis caused by the actuation in the ϕ direction. As the mirror tilts further, the cross-axial coupling becomes greater as shown in Figures 4.17-4.22. The model still in Figure 4.23 and 4.24 follows closely the mirror transient and steady-state behavior until where the mirror is close to pull-in as shown. Differences between measurements and model then become observable.

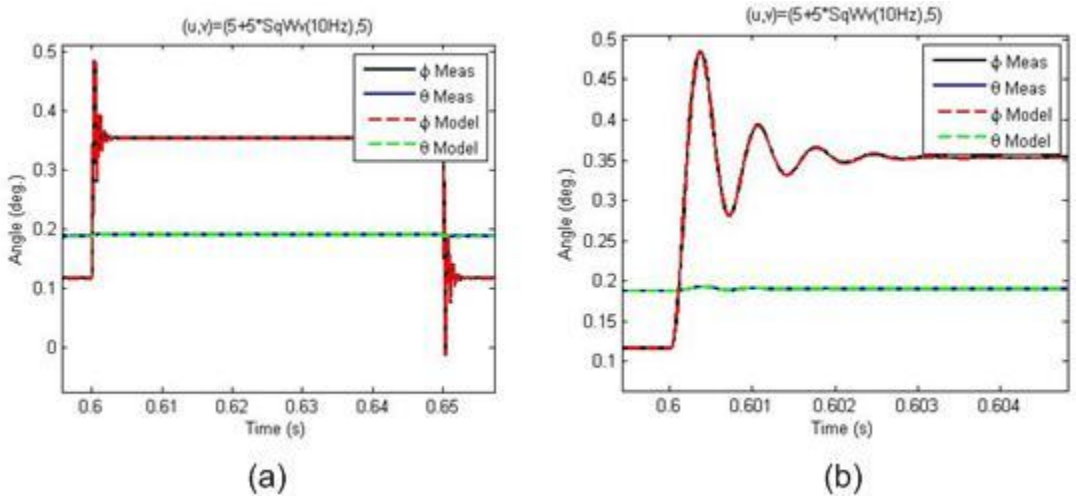


Figure 4.17 (a)-(b) Illustration of 2D Actuation with u=5 and v=5 with Square Wave Modulation in u Voltage(or the ϕ-Axis.)

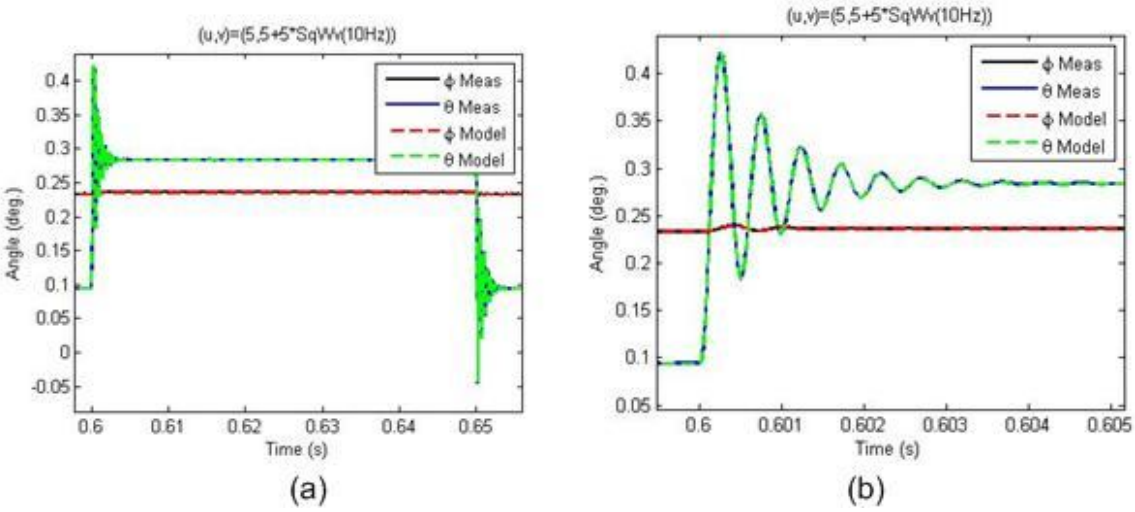


Figure 4.18 (a)-(b) Illustration of 2D Actuation with u=5 and v=5 with Square Wave Modulation in v Voltage (or the θ-Axis.)

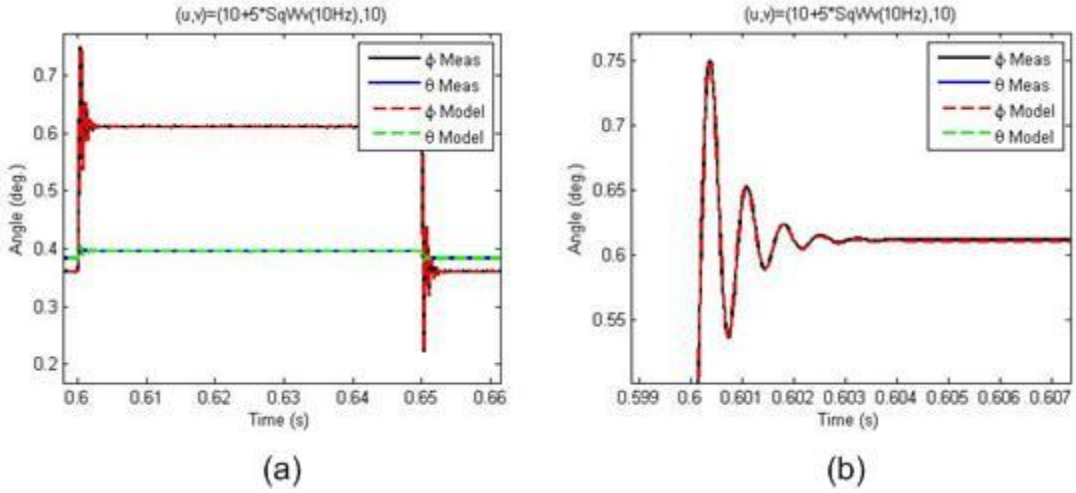


Figure 4.19 (a)-(b) Illustration of 2D Actuation with u=10 and v=10 with Square Wave Modulation in u Voltage (or the ϕ-Axis.)

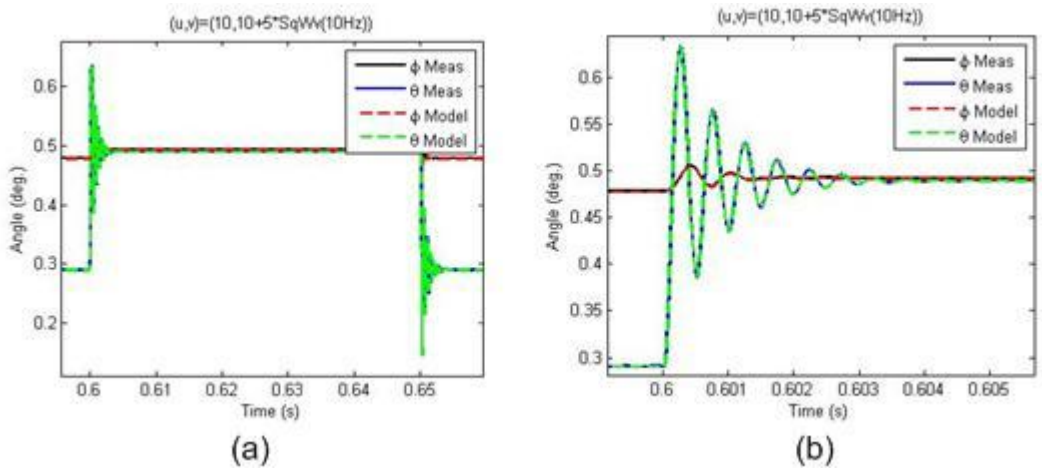


Figure 4.20 (a)-(b) Illustration of 2D Actuation with u=10 and v=10 with Square Wave Modulation in v Voltage (or the θ-Axis.)

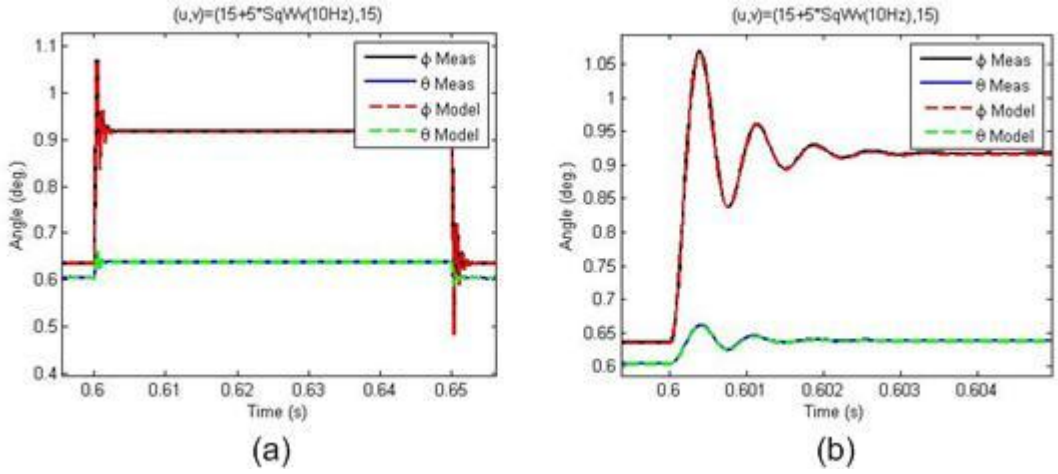


Figure 4.21 (a)-(b) Illustration of 2D Actuation with u=15 and v=15 with Modulation in u Voltage (or the ϕ-Axis.)

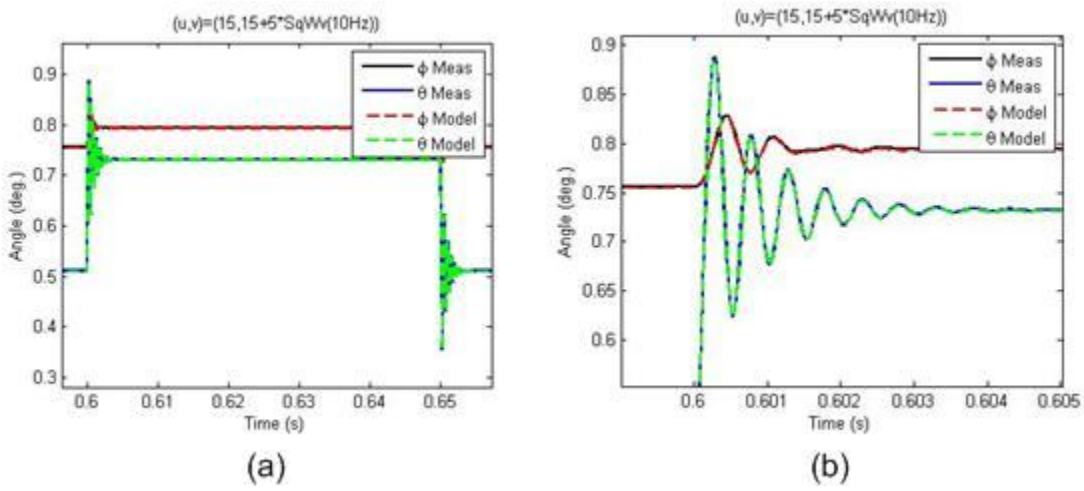


Figure 4.22 (a)-(b) Illustration of 2D Actuation with u=15 and v=15 with Square Wave Modulation in v Voltage (or the θ-Axis.)

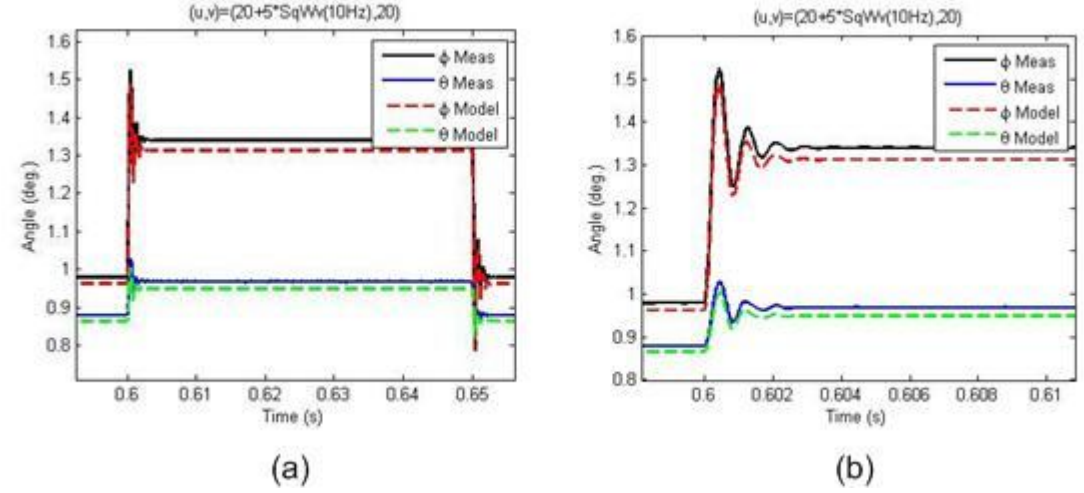


Figure 4.23 (a)-(b) Illustration of 2D Actuation with u=20 and v=20 with Square Wave Modulation in u Voltage (or the ϕ-Axis.)

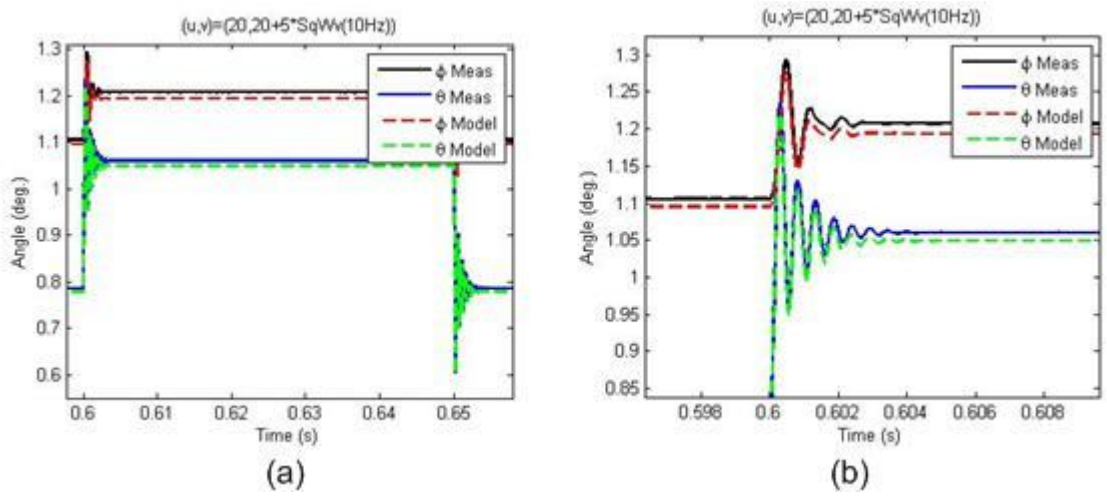


Figure 4.24 (a)-(b) Illustration of 2D Actuation with u=20 and v=20 with Square Wave Modulation in v Voltage (or the θ-Axis.)

Conclusions

In this chapter measurements were presented that support the first-principles analytic, closed-form 2D Torque Model and Micro-mirror Pointing Model (MPM) for the two-axis (2D) tip-tilt mirror. The MPM model accurately describes the mirror's motions for a wide range of operations. An experimental apparatus was described that validated the 2D Torque and MPM models for a noise-free environment.

A methodology was also developed for determining the dynamic constants for the mirror as well as an effective length, which accounts for fringing electric fields. Allowing for the fitting of physical parameters such as torsional restoring constants

and “effective lengths” highlights the strengths of the MPM model and calibration process.

The model is in close agreement with the measurements for both single axis and 2D movement. As the mirror approaches pull-in, the nonlinear nature of pull-in causes slight deviations that differences become visible. For all other operations, the model performs remarkably well compared to measurements of transients, steady-state and cross coupling effects. This is due to the fundamental nature of the model combined with the calibration of the mirror (fitting to the effective length of the mirror and the torsional restoring constant for each axis.)

While MEMS fabrication does lead to inherent variations in mirror physical properties, the model can be calibrated for each mirror to compensate for this. In determining these physical parameters, knowledge is gained about the fabrication processes, while still leaving the model in terms of an explicit relationship between mirror parameters and performance. The resulting high-fidelity models could enable designers to optimize mirror operations for applications where fast and precise pointing is critical.

Chapter 5: Micro-Mirror Jitter Model

Overview

In Chapter 3 torque expressions were developed based on fundamental physical relationships for a MEMS Two-Axis (Tip-Tilt) mirror structure. This model related its orientation to applied voltages. The torque expressions combined with derived equations of rotational motion represent a new model that predicts micro-mirror orientation in response to control voltages. Since orientation determines the mirror's pointing geometry for a laser system, the model is referred to as the Micro-mirror Pointing Model or MPM model. The MPM model was validated in Chapter 4 by comparing the model results with experimental measurements for deterministic (noise-free) control voltages. As was expected, after the transients had decayed, stochastic fluctuations continued in the mirror's angle. It has been recognized that this "jitter" is due to three primary noise sources: (1) control voltage fluctuations, (2) platform vibration, and (3) Brownian motion (or mechanical-thermal) noise as illustrated in Figure 5.1. These three sources of noise have been studied in other MEMS devices and oscillators but not for micro-mirrors [58], [62] [65], [66], [67], [68], [69]. Brownian motion noise will be modeled as a zero-mean white Gaussian stochastic process.

In this chapter the 2D Micro-mirror Pointing Model (MPM) from Chapter 3 is used to create a Two-Axis (Tip-Tilt) Micro-mirror Jitter Model (MJM). The MJM model

consists of statistical relationships between the mirror jitter and its originating sources. Among other things the MJM model along with the MPM model would permit the system engineer to evaluate expected jitter for MEMS device components and environments or to determine if components are acceptable given system-level pointing and jitter performance requirements.

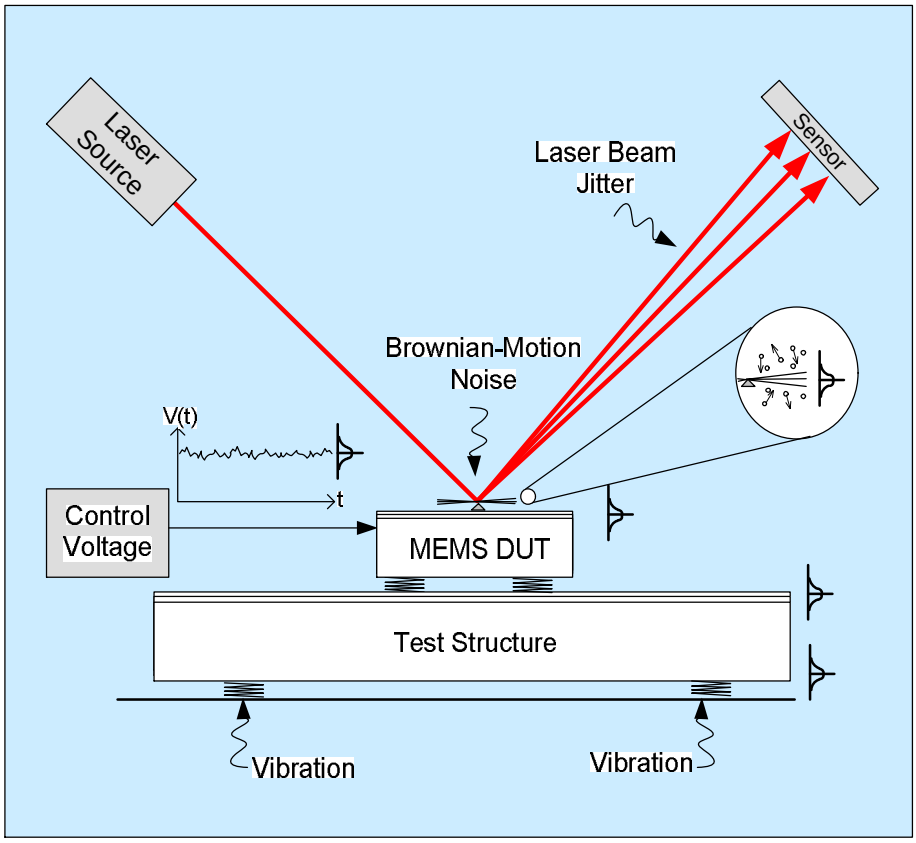


Figure 5.1 MEMS Noise Sources and Resulting Beam Pointing Jitter

Furthermore, the dominating sources of jitter can be determined and analyzed. For instance, the dominant source of jitter in measurements validating the MPM model was the control voltage noise due to noisy switching power supplies. For the MJM model measurements (Chapter 6), the dominant source of jitter is Brownian motion

noise. This experimental set-up was designed to measure the Brownian motion noise. The MEMS mirror and quad cell were biased by batteries to minimize voltage noise. The measurements were made by a high-resolution data acquisition system with 24 bits of precision over a 20 volt range and excellent inter-channel isolation.

Micro-Mirror Jitter Model

In developing the two-axis Micro-mirror Jitter Model (MJM) stochastic (noise) sources must be added to the MPM model. These noise sources represent voltage fluctuations in control circuits, platform and supporting structure vibration, and Brownian motion of air molecules bombarding the mirror facet. The angular rotation of the mirror ϕ, and θ, is described by equations (5.1).

$$I_1\ddot{\phi} + D_1\dot{\phi} + K_{1m}\phi = \tau_1\left(\phi,\ \theta,\ u_0 + u_n,\ v_0 + v_n\right) + \tau_{1vib} + \tau_{1bm} \tag{5.1a}$$

$$I_2\ddot{\theta} + D_2\dot{\theta} + K_{2m}\theta = \tau_2\left(\phi,\ \theta,\ u_0 + u_n,\ v_0 + v_n\right) + \tau_{2vib} + \tau_{2bm} \tag{5.1b}$$

Where:

$\tau_{1vib} \equiv$ contribution to ϕ due to platform vibrations,

$\tau_{1bm} \equiv$ contribution to ϕ due to Brownian motion of molecules impacting the mirror,

$\tau_{2vib} \equiv$ contribution to θ due to platform vibrations,

$\tau_{2bm} \equiv$ contribution to θ due to Brownian motion of molecules impacting the mirror,

$u_n \equiv$ voltage fluctuations on the deterministic control voltage u_0, and

$v_n \equiv$ voltage fluctuations on the deterministic control voltage v_0.

It is helpful to think of the mirror response angles (ϕ,θ) as having two components: one deterministic and one random. To explicitly account for this one can define the tip and tilt orientation as $\phi(t)=\phi_0(t)+\phi_\Delta(t)$ and $\theta(t)=\theta_0(t)+\theta_\Delta(t)$, where ϕ_0 and θ_0 are the angular responses to torques resulting from control voltages u_0 and v_0. The angles θ_Δ and ϕ_Δ are responses due to noise sources affecting the mirror. Equation (5.1a) can be written explicitly to show the deterministic and random inputs and output response as:

$$I_1\left(\ddot{\phi}_0+\ddot{\phi}_\Delta\right)+D_1\left(\dot{\phi}_0+\dot{\phi}_\Delta\right)+K_{1m}\left(\phi_0+\phi_\Delta\right)=\tau_1\left(\phi_0+\phi_\Delta,\ \theta_0+\theta_\Delta,\ u_0+u_n,\ v_0+v_n\right)+\tau_{1vib}+\tau_{1bm} \tag{5.2}$$

In equation (5.2), distributing the left-hand-side (LHS) dynamic constants and expanding the right-hand-side (RHS) in a multidimensional first-order Taylor expansion around the point $(\phi_0,\theta_0,u_0,v_0)$ results in:

$$I_1\ddot{\phi}_0+D_1\dot{\phi}_0+K_{1m}\phi_0+I_1\ddot{\phi}_\Delta+D_1\dot{\phi}_\Delta+K_{1m}\phi_\Delta\approx \tau_1\left(\phi_0,\theta_0,u_0,v_0\right)+K_{1\phi}\phi_\Delta+K_{1\theta}\theta_\Delta+K_{1u}u_n+K_{1v}v_n+\tau_{1vib}+\tau_{1bm} \tag{5.3}$$

where:

$$\begin{aligned} K_{1\phi}&=\frac{\partial\tau_1\left(\phi_0,\theta_0,u_0,v_0\right)}{\partial\phi}\\ K_{1\theta}&=\frac{\partial\tau_1\left(\phi_0,\theta_0,u_0,v_0\right)}{\partial\theta}\\ K_{1u}&=\frac{\partial\tau_1\left(\phi_0,\theta_0,u_0,v_0\right)}{\partial u}\\ K_{1v}&=\frac{\partial\tau_1\left(\phi_0,\theta_0,u_0,v_0\right)}{\partial v} \end{aligned} \tag{5.4}$$

Since $I_1\ddot{\phi}_0 + D_1\dot{\phi}_0 + K_{1m}\phi_0 = \tau_1\left(\phi_0, \theta_0, u_0, v_0\right)$, equation (5.3) becomes

$$I_1\ddot{\phi}_\Delta + D_1\dot{\phi}_\Delta + K_{1E}\phi_\Delta = K_{1\theta}\theta_\Delta + \tau_{1vib} + \tau_{1bm} + K_{1u}u_n + K_{1v}v_n \tag{5.5}$$

where

$$K_{1E} = K_{1m} - K_{1\phi} \tag{5.6}$$

When the deterministic solution has achieved steady-state, then $u_0 = u_0\left(t\right)$ becomes a constant designated as u_∞. Similarly, $v_0\left(t\right) \to v_\infty$, $\phi_0\left(t\right) \to \phi_\infty$, and $\theta_0\left(t\right) \to \theta_\infty$, and under the steady-state conditions, the K parameters become constants given by:

$$\begin{aligned} K_{1\phi} &= \frac{\partial \tau_1\left(\phi_\infty, \theta_\infty, u_\infty, v_\infty\right)}{\partial \phi} \\ K_{1\theta} &= \frac{\partial \tau_1\left(\phi_\infty, \theta_\infty, u_\infty, v_\infty\right)}{\partial \theta} \\ K_{1u} &= \frac{\partial \tau_1\left(\phi_\infty, \theta_\infty, u_\infty, v_\infty\right)}{\partial u} \\ K_{1v} &= \frac{\partial \tau_1\left(\phi_\infty, \theta_\infty, u_\infty, v_\infty\right)}{\partial v} \end{aligned} \tag{5.4a}$$

Four other parameters are likewise defined as:

$$\begin{aligned} K_{2\phi} &= \frac{\partial \tau_2\left(\phi_\infty, \theta_\infty, u_\infty, v_\infty\right)}{\partial \phi} \\ K_{2\theta} &= \frac{\partial \tau_2\left(\phi_\infty, \theta_\infty, u_\infty, v_\infty\right)}{\partial \theta} \\ K_{2u} &= \frac{\partial \tau_2\left(\phi_\infty, \theta_\infty, u_\infty, v_\infty\right)}{\partial u} \\ K_{2v} &= \frac{\partial \tau_2\left(\phi_\infty, \theta_\infty, u_\infty, v_\infty\right)}{\partial v} \end{aligned} \tag{5.4b}$$

Since equations (5.4) are partial derivatives, in the next section we express them in terms of the generalized torque function "G" developed in Chapter 3.

Partial Derivatives of Voltages "u" and "v":

Recalling the earlier development of equations (3.4) and (3.5) or the generalized torque function:

Then we let

$$\tau_1 = G_{11}V_1^2 + G_{12}V_2^2 + G_{13}V_3^2 + G_{14}V_4^2$$
$$\tau_2 = -\cos\phi\left(G_{21}V_1^2 + G_{22}V_2^2 + G_{23}V_3^2 + G_{24}V_4^2\right)$$

where τ_1 and τ_2 are the total torques on the mirror due to voltages V_1, V_2, V_3, and V_4 on each respective pad and composed of two actuating voltages (u,v) i.e.

$$V_1 = V_b - u - v$$
$$V_2 = V_b - u + v$$
$$V_3 = V_b + u - v$$
$$V_4 = V_b + u + v$$

In general, to calculate

$$\frac{\partial \tau_1}{\partial u} = \sum_{i=1}^{4} 2G_{1i}V_i \frac{\partial V_i}{\partial u}$$

and

$$\frac{\partial \tau_1}{\partial v} = \sum_{i=1}^{4} 2G_{1i}V_i \frac{\partial V_i}{\partial v}$$

and

$$\frac{\partial \tau_2}{\partial u} = -\cos\phi \sum_{i=1}^{4} 2G_{2i}V_i \frac{\partial V_i}{\partial u}$$

and

$$\frac{\partial \tau_2}{\partial v} = -\cos\phi \sum_{i=1}^{4} 2G_{2i}V_i \frac{\partial V_i}{\partial v},$$

where

$$\frac{\partial V_1}{\partial u} = -1,\ \frac{\partial V_1}{\partial v} = 1,\ \frac{\partial V_2}{\partial u} = -1,\ \frac{\partial V_2}{\partial v} = -1,\ \frac{\partial V_3}{\partial u} = 1,\ \frac{\partial V_3}{\partial v} = -1,\ \frac{\partial V_4}{\partial u} = 1,\ \frac{\partial V_4}{\partial v} = 1$$

and therefore the partial derivatives are:

$$K_{1u} = \frac{\partial \tau_1}{\partial u} = 2\left(-G_{11}V_1 - G_{12}V_2 + G_{13}V_3 + G_{14}V_4\right)$$
$$K_{1v} = \frac{\partial \tau_1}{\partial v} = 2\left({}_1G_{11}V - G_{12}V_2 - G_{13}V_3 + G_{14}V_4\right)$$
$$K_{2u} = \frac{\partial \tau_2}{\partial u} = -2\cos\phi_\infty\left(-G_{21}V_1 - G_{22}V_2 + G_{23}V_3 + G_{24}V_4\right)$$
$$K_{2v} = \frac{\partial \tau_2}{\partial v} = -2\cos\phi_\infty\left(G_{21}V_1 - G_{22}V_2 - G_{23}V_3 + G_{24}V_4\right)$$

Thus, we have developed the partial derivatives for the two actuating voltages, (u, v), in terms of the generalized "G" function which is appropriate for software implementation utilizing the symmetry in the τ_1 and τ_2 torque expressions originally developed in Chapter 3. Typical values of K are given in Table 5.1. for K_{1u}, K_{1v}, K_{2u}, and K_{2v}.

Table 5.1 Typical Values of K_{1u}, K_{1v}, K_{2u}, and K_{2v}

Parameter $K_{\text{i,volt}}$	K_{1u} $(\mu N \cdot \mu m / V)$	K_{1v} $(\mu N \cdot \mu m / V)$	K_{2u} $(\mu N \cdot \mu m / V)$	K_{2v} $(\mu N \cdot \mu m / V)$
u=0, v=0 $(\phi, \theta) = (0^0, 0^0)$	12.68e+000	0.00e+000	12.91e+000	0.00e+000
u=5, v=5 $(\phi, \theta) = (0.20^0, 0.23^0)$	12.79e+000	9.91e-002	12.96e+000	1.02e-002
u=15, v=15 $(\phi, \theta) = (0.69^0, 0.83^0)$	13.76e+000	1.02e+000	13.95e+003	1.05e+000
u=25, v=25 $(\phi, \theta) = (1.6^0, 2.2^0)$	18.43e+000	5.45e+000	18.75e+000	5.62e+000

Partial Derivatives of Angles ϕ and θ:

Since the partial derivatives of the G functions enter through the variables "p" and "q", it is expedient to first determine derivative expressions as follows from equation (3.3):

$$G(r_1, r_2, s_1, s_2, p, q) = \frac{\varepsilon}{2p^2 q}\Big\{(h_0 - qs_2)\big[\ln(h_0 - pr_2 - qs_2) - \ln(h_0 - pr_1 - qs_2)\big] - (h_0 - qs_1)\big[\ln(h_0 - pr_2 - qs_1) - \ln(h_0 - pr_1 - qs_1)\big]\Big\}$$

The partial derivative with respect to "p" is straight forward and is given by:

$$G_p(r_1,r_2,s_1,s_2,p,q)=\frac{\partial G}{\partial p}=-\frac{\varepsilon}{p^3q}\Big\{(h_0-qs_2)\Big[\ln(h_0-pr_2-qs_2)-\ln(h_0-pr_1-qs_2)\Big]$$
$$-(h_0-qs_1)\Big[\ln(h_0-pr_2-qs_1)-\ln(h_0-pr_1-qs_1)\Big]\Big\}$$
$$+\frac{\varepsilon}{2p^2q}\left\{(h_0-qs_2)\left[\frac{-r_2}{h_0-pr_2-qs_2}+\frac{r_1}{h_0-pr_1-qs_2}\right]\right.$$
$$\left.-(h_0-qs_1)\left[\frac{-r_2}{h_0-pr_2-qs_1}+\frac{r_1}{h_0-pr_1-qs_1}\right]\right\}$$

The partial derivatives with respect to "q" are now found.

Since $\frac{\partial}{\partial q}\left(\frac{h_0-qs_2}{q}\right)=\frac{-h_0}{q^2}$ and $\frac{\partial}{\partial q}\left(\frac{h_0-qs_1}{q}\right)=\frac{-h_0}{q^2}$ then

$$\frac{\partial G}{\partial q}=G_q(r_1,r_2,s_1,s_2,p,q)=\frac{\varepsilon}{2p^2}\left\{\frac{-h_0}{q^2}\Big[\ln(h_0-pr_2-qs_2)-\ln(h_0-pr_1-qs_2)\Big]\right.$$
$$\left.-\frac{-h_0}{q^2}\Big[\ln(h_0-pr_2-qs_1)-\ln(h_0-pr_1-qs_1)\Big]\right\}$$
$$+\frac{\varepsilon}{2p^2}\left\{\frac{h_0-qs_2}{q}\left[\frac{-s_2}{h_0-pr_2-qs_2}+\frac{s_2}{h_0-pr_1-qs_2}\right]\right.$$
$$\left.-\frac{h_0-qs_1}{q}\left[\frac{-s_1}{h_0-pr_2-qs_1}+\frac{s_1}{h_0-pr_1-qs_1}\right]\right\}$$

We can represent the partial derivatives for “p” and “q” for all the voltage pads by defining the following functions:

$$G_{11A}=\left.\frac{\partial G}{\partial p}\right|_{r_1=x_1,r_2=x_2,s_1=z_1,s_2=z_2,p=A,q=C} \qquad G_{11C}=\left.\frac{\partial G}{\partial q}\right|_{r_1=x_1,r_2=x_2,s_1=z_1,s_2=z_2,p=A,q=C}$$

$$G_{12A}=\left.\frac{\partial G}{\partial p}\right|_{r_1=x_1,r_2=x_2,s_1=-z_2,s_2=-z_1,p=A,q=C} \qquad G_{12C}=\left.\frac{\partial G}{\partial q}\right|_{r_1=x_1,r_2=x_2,s_1=-z_2,s_2=-z_1,p=A,q=C}$$

$$G_{13A}=\left.\frac{\partial G}{\partial p}\right|_{r_1=-x_2,r_2=-x_1,s_1=-z_2,s_2=-z_1,p=A,q=C} \qquad G_{13C}=\left.\frac{\partial G}{\partial q}\right|_{r_1=-x_2,r_2=-x_1,s_1=-z_2,s_2=-z_1,p=A,q=C}$$

$$G_{14A}=\left.\frac{\partial G}{\partial p}\right|_{r_1=-x_2,r_2=-x_1,s_1=z_1,s_2=z_2,p=A,q=C} \qquad G_{14C}=\left.\frac{\partial G}{\partial q}\right|_{r_1=-x_2,r_2=-x_1,s_1=z_1,s_2=z_2,p=A,q=C}$$

$$G_{21A}=\left.\frac{\partial G}{\partial q}\right|_{r_1=z_1,r_2=z_2,s_1=x_1,s_2=x_2,p=C,q=A} \qquad G_{21C}=\left.\frac{\partial G}{\partial p}\right|_{r_1=z_1,r_2=z_2,s_1=x_1,s_2=x_2,p=C,q=A}$$

$$G_{22A}=\left.\frac{\partial G}{\partial q}\right|_{r_1=-z_2,r_2=-z_1,s_1=x_1,s_2=x_2,p=C,q=A} \qquad G_{22C}=\left.\frac{\partial G}{\partial p}\right|_{r_1=-z_2,r_2=-z_1,s_1=x_1,s_2=x_2,p=C,q=A}$$

$$G_{23A}=\left.\frac{\partial G}{\partial q}\right|_{r_1=-z_2,r_2=-z_1,s_1=-x_2,s_2=-x_1,p=C,q=A} \qquad G_{23C}=\left.\frac{\partial G}{\partial p}\right|_{r_1=-z_2,r_2=-z_1,s_1=-x_2,s_2=-x_1,p=C,q=A}$$

$$G_{24A}=\left.\frac{\partial G}{\partial q}\right|_{r_1=z_1,r_2=z_2,s_1=-x_2,s_2=-x_1,p=C,q=A} \qquad G_{24C}=\left.\frac{\partial G}{\partial p}\right|_{r_1=z_1,r_2=z_2,s_1=-x_2,s_2=-x_1,p=C,q=A}$$

Remembering that

$$\tau_1 = G_{11}V_1^2 + G_{12}V_2^2 + G_{13}V_3^2 + G_{14}V_4^2$$

The new functional expression leads to

$$K_{1\phi} = \frac{\partial \tau_1}{\partial \phi} = V_1^2\left(G_{11A}\frac{\partial A}{\partial \phi} + G_{11C}\frac{\partial C}{\partial \phi}\right) + V_2^2\left(G_{12A}\frac{\partial A}{\partial \phi} + G_{12C}\frac{\partial C}{\partial \phi}\right) + V_3^2\left(G_{13A}\frac{\partial A}{\partial \phi} + G_{13C}\frac{\partial C}{\partial \phi}\right) + V_4^2\left(G_{14A}\frac{\partial A}{\partial \phi} + G_{14C}\frac{\partial C}{\partial \phi}\right)$$

Since $\frac{\partial A}{\partial \phi} = -\sec^2\phi$ and $\frac{\partial C}{\partial \phi} = \tan\theta\tan\phi\sec\phi$ then

$$K_{1\phi} = \frac{\partial \tau_1}{\partial \phi} = V_1^2\left(-G_{11A}\sec^2\phi + G_{11C}\tan\theta\tan\phi\sec\phi\right) + V_2^2\left(-G_{12A}\sec^2\phi + G_{12C}\tan\theta\tan\phi\sec\phi\right) + V_3^2\left(-G_{13A}\sec^2\phi + G_{13C}\tan\theta\tan\phi\sec\phi\right) + V_4^2\left(-G_{14A}\sec^2\phi + G_{14C}\tan\theta\tan\phi\sec\phi\right)$$

or

$$K_{1\phi} = \left(-G_{11A}V_1^2 - G_{12A}V_2^2 - G_{13A}V_3^2 - G_{14A}V_4^2\right)\sec^2\phi + \left(G_{11C}V_1^2 + G_{12C}V_2^2 + G_{13C}V_3^2 + G_{14C}V_4^2\right)\tan\theta\tan\phi\sec\phi$$

$$K_{1\theta} = \frac{\partial \tau_1}{\partial \theta} = V_1^2\left(G_{11A}\frac{\partial A}{\partial \theta} + G_{11C}\frac{\partial C}{\partial \theta}\right) + V_2^2\left(G_{12A}\frac{\partial A}{\partial \theta} + G_{12C}\frac{\partial C}{\partial \theta}\right) + V_3^2\left(G_{13A}\frac{\partial A}{\partial \theta} + G_{13C}\frac{\partial C}{\partial \theta}\right) + V_4^2\left(G_{14A}\frac{\partial A}{\partial \theta} + G_{14C}\frac{\partial C}{\partial \theta}\right)$$

Since $\frac{\partial A}{\partial \theta} = 0$ and $\frac{\partial C}{\partial \theta} = \sec^2\theta\sec\phi$ then

$$K_{1\theta} = \left(G_{11C}V_1^2 + G_{12C}V_2^2 + G_{13C}V_3^2 + G_{14C}V_4^2\right)\sec^2\theta\sec\phi$$

$$\tau_2 = -\cos\phi\left(G_{21}V_1^{\,2} + G_{22}V_2^{\,2} + G_{23}V_3^{\,2} + G_{24}V_4^{\,2}\right)$$

$$K_{2\phi} = \frac{\partial\tau_2}{\partial\phi} = -\cos\phi\left\{V_1^{\,2}\left(G_{21A}\frac{\partial A}{\partial\phi} + G_{21C}\frac{\partial C}{\partial\phi}\right) + V_2^{\,2}\left(G_{22A}\frac{\partial A}{\partial\phi} + G_{22C}\frac{\partial C}{\partial\phi}\right)\right.$$
$$\left. + V_3^{\,2}\left(G_{23A}\frac{\partial A}{\partial\phi} + G_{23C}\frac{\partial C}{\partial\phi}\right) + V_4^{\,2}\left(G_{24A}\frac{\partial A}{\partial\phi} + G_{24C}\frac{\partial C}{\partial\phi}\right)\right\}$$
$$+ \sin\phi\left(G_{21}V_1^{\,2} + G_{22}V_2^{\,2} + G_{23}V_3^{\,2} + G_{24}V_4^{\,2}\right)$$

$$K_{2\phi} = -\cos\phi\left[G_{21A}V_1^{\,2} + G_{22A}V_2^{\,2} + G_{23A}V_3^{\,2} + G_{24A}V_4^{\,2}\right]\frac{\partial A}{\partial\phi}$$
$$-\cos\phi\left[G_{21C}V_1^{\,2} + G_{22C}V_2^{\,2} + G_{23C}V_3^{\,2} + G_{24C}V_4^{\,2}\right]\frac{\partial C}{\partial\phi}$$
$$+ \sin\phi\left(V_1^{\,2}G_{21} + V_2^{\,2}G_{22} + V_3^{\,2}G_{23} + V_4^{\,2}G_{24}\right)$$

$$K_{2\phi} = \sec\phi\left[G_{21A}V_1^{\,2} + G_{22A}V_2^{\,2} + G_{23A}V_3^{\,2} + G_{24A}V_4^{\,2}\right]$$
$$-\tan\theta\tan\phi\left[G_{21C}V_1^{\,2} + G_{22C}V_2^{\,2} + G_{23C}V_3^{\,2} + G_{24C}V_4^{\,2}\right]$$
$$+ \sin\phi\left(G_{21}V_1^{\,2} + G_{22}V_2^{\,2} + G_{23}V_3^{\,2} + G_{24}V_4^{\,2}\right)$$

$$K_{2\theta} = \frac{\partial\tau_2}{\partial\theta} = -\cos\phi\left\{V_1^{\,2}\left(G_{21A}\frac{\partial A}{\partial\theta} + G_{21C}\frac{\partial C}{\partial\theta}\right) + V_2^{\,2}\left(G_{22A}\frac{\partial A}{\partial\theta} + G_{22C}\frac{\partial C}{\partial\theta}\right)\right.$$
$$\left. + V_3^{\,2}\left(G_{23A}\frac{\partial A}{\partial\theta} + G_{23C}\frac{\partial C}{\partial\theta}\right) + V_4^{\,2}\left(G_{24A}\frac{\partial A}{\partial\theta} + G_{24C}\frac{\partial C}{\partial\theta}\right)\right\}$$

$$K_{2\theta} = -\cos\phi\left[G_{21C}V_1^{\,2} + G_{22C}V_2^{\,2} + G_{23C}V_3^{\,2} + G_{24C}V_4^{\,2}\right]\frac{\partial C}{\partial\theta}$$

$$K_{2\theta} = -\sec^2\theta\left[G_{21C}V_1^2 + G_{22C}V_2^2 + G_{23C}V_3^2 + G_{24C}V_4^2\right]$$

Typical values for the partial derivatives of $K_{1\phi}$, $K_{2\theta}$, $K_{1\theta}$, and $K_{2\phi}$ are given in Table 5.2.

Table 5.2 Values of $K_{1\phi}$, $K_{2\theta}$, $K_{1\theta}$, and $K_{2\phi}$

Parameter $K_{i\angle}$	$K_{1\phi}$ $(\mu N\cdot\mu m/rad)$	$K_{1\theta}$ $(\mu N\cdot\mu m/rad)$	$K_{2\phi}$ $(\mu N\cdot\mu m/rad)$	$K_{2\theta}$ $(\mu N\cdot\mu m/rad)$
u=0, v=0 $(\phi,\theta)=(0^0,0^0)$	3.05e+003	0.00	0.00	3.14e+003
u=5, v=5 $(\phi,\theta)=(0.20^0,0.23^0)$	3.19e+003	1.08e+002	1.07e+002	3.29e+003
u=15, v=15 $(\phi,\theta)=(0.69^0,0.83^0)$	4.44e+003	1.11e+003	1.11e+003	4.61e+003
u=25, v=25 $(\phi,\theta)=(1.6^0,2.2^0)$	1.02e+004	6.06e+003	6.04e+003	1.09e+004

Equation (5.5) can be simplified by letting:

$$F_1 = \tau_{1vib} + \tau_{1bm} + K_{1u}u_n + K_{1v}v_n \tag{5.7}$$

and it becomes

$$I_1\ddot{\phi}_\Delta + D_1\dot{\phi}_\Delta + K_{1E}\phi_\Delta = K_{1\theta}\theta_\Delta + F_1$$

A similar derivation for θ and $\tau_2(\phi,\theta,u,v)$ leads to

$$I_1\ddot{\theta}_\Delta + D_1\dot{\theta}_\Delta + K_{2E}\theta_\Delta = K_{2\phi}\phi_\Delta + F_2$$

where

$$K_{2E} = K_{2m} - K_{2\theta}$$

and

$$F_2 = \tau_{2vib} + \tau_{2bm} + K_{2u}u_n + K_{2v}v_n$$

Let L_1 and L_2 be transformations defined by:

$$L_1 = I_1\frac{d^2}{dt^2} + D_1\frac{d}{dt} + K_{1E} \text{ and } L_2 = I_2\frac{d^2}{dt^2} + D_2\frac{d}{dt} + K_{2E}$$

Equation (5.7) can then be expressed in terms of ϕ and θ as

$$L_1\phi_\Delta = K_{1\theta}\theta_\Delta + F_1 \tag{5.8}$$

$$L_2\theta_\Delta = K_{2\phi}\phi_\Delta + F_2 \tag{5.9}$$

this is easily rewritten as

$$\begin{aligned} L_2L_1\phi_\Delta &= K_{1\theta}L_2\theta_\Delta + L_2F_1 \\ L_1L_2\theta_\Delta &= K_{2\phi}L_1\phi_\Delta + L_1F_2 \end{aligned} \tag{5.10}$$

Substituting equation (5.8) and equation (5.9) into equation (5.10) yields

$$\begin{aligned} L_2L_1\phi_\Delta &= K_{1\theta}\left(K_{2\phi}\phi_\Delta + F_2\right) + L_2F_1 \\ L_1L_2\theta_\Delta &= K_{2\phi}\left(K_{1\theta}\theta_\Delta + F_1\right) + L_1F_2 \end{aligned}$$

After collecting all of the terms involving ϕ_Δ and θ_Δ on the LHS and noting that the operator commutes, then one obtains

$$\begin{aligned} L\phi_\Delta &= K_{1\theta}F_2 + L_2F_1 \\ L\theta_\Delta &= K_{2\phi}F_1 + L_1F_2 \end{aligned}$$

where $L = L_1L_2 - K_{1\theta}K_{2\phi}$. Denoting the RHS terms as one variable gives

$$\begin{aligned} L\phi_{\Delta} &= w_1(t) \\ L\theta_{\Delta} &= w_2(t) \end{aligned} \tag{5.11}$$

where the RHS of the pair of equations in (5.11) is denoted by $w_i(t)$ and

$$\begin{aligned} w_1(t) &= K_{1\theta}F_2(t) + L_2F_1(t) \\ w_2(t) &= K_{2\phi}F_1(t) + L_1F_2(t) \end{aligned}$$

The transfer function relationships for both expressions in equation (5.11) are defined by

$$\begin{aligned} \Phi_{\Delta}(j\omega) &= H(j\omega)W_1(j\omega) \\ \Theta_{\Delta}(j\omega) &= H(j\omega)W_2(j\omega) \end{aligned} \tag{5.12}$$

with

$$H(j\omega) = \frac{1}{P_1(j\omega)P_2(j\omega) - K_{1\theta}K_{2\phi}}$$

where $P_i(j\omega)$ is the Fourier transform equivalent of the operator L_i or

$$P_1(j\omega) = -\omega^2 I_1 + j\omega D_1 + (K_{1m} - K_{1\phi})$$

$$P_2(j\omega) = -\omega^2 I_2 + j\omega D_2 + (K_{2m} - K_{2\theta})$$

The $w_i(t)$ functions have associated autocorrelation and cross-correlation functions as well as equivalent power spectral densities (PSD) and cross-power spectral densities. One denotes these as follows:

$$R_{w_iw_j}(\tau) = E\left[w_i(t)\,w_j(t+\tau)\right], \quad \Leftrightarrow \quad S_{w_iw_j}(j\omega) = FT\left\{R_{w_iw_j}(\tau)\right\}$$

where i=1,2, j=1,2 and $FT\{\cdot\}$ denotes the Fourier transform. Given these functions the output fluctuations can then be characterized in terms of their correlation and spectral functions using equation (5.12) as follows.

$$S_{\phi_\Delta\phi_\Delta}(j\omega) = |H(j\omega)|^2 S_{w_1w_1}(j\omega)$$
$$S_{\theta_\Delta\theta_\Delta}(j\omega) = |H(j\omega)|^2 S_{w_2w_2}(j\omega)$$
$$S_{\theta_\Delta\phi_\Delta}(j\omega) = |H(j\omega)|^2 S_{w_2w_1}(j\omega)$$
$$S_{\phi_\Delta\theta_\Delta}(j\omega) = |H(j\omega)|^2 S_{w_1w_2}(j\omega)$$

Now the input power spectral densities are found, beginning with $S_{w_1w_1}$.

Development of $S_{w_1w_1}(j\omega)$ or $S_{w_1w_1}$

Beginning with the definition of the autocorrelation function:

$$R_{w_1w_1}(\tau) = E\left[w_1(t)\,w_1(t+\tau)\right] = E\left[\left(K_{1\theta}F_2(t) + L_2F_1(t)\right)\left(K_{1\theta}F_2(t+\tau) + L_2F_1(t+\tau)\right)\right]$$

We can pull the constants out of the expectation resulting in:

$$R_{w_1w_1}(\tau) = K_{1\theta}^2 E\left[F_2(t)F_2(t+\tau)\right] + K_{1\theta}E\left[L_2F_1(t)F_2(t+\tau)\right] + K_{1\theta}E\left[F_2(t)L_2F_1(t+\tau)\right] + E\left[L_2F_1(t)L_2F_1(t+\tau)\right]$$

$$R_{w_1w_1}(\tau) = K_{1\theta}^2 R_{F_2F_2}(\tau) + K_{1\theta}R_{L_2F_1F_2}(\tau) + K_{1\theta}R_{F_2L_2F_1}(\tau) + R_{L_2F_1L_2F_1}(\tau)$$

Taking the Fourier transform yields

$$S_{w_1w_1}(j\omega) = K_{1\theta}^2 S_{F_2F_2}(j\omega) + K_{1\theta}S_{L_2F_1F_2}(j\omega) + K_{1\theta}S_{F_2L_2F_1}(j\omega) + S_{L_2F_1L_2F_1}(j\omega) \qquad (5.13)$$

At this point to simplify notation we can omit the explicit $j\omega$ argument in the spectral density and polynomial functions and show it only where necessary. Using the derivative property for spectral density functions Equation (5.13) becomes

$$S_{w_1w_1} = K_{1\theta}^2 S_{F_2F_2} + K_{1\theta} P_2 S_{F_1F_2} + P_2^* K_{1\theta} S_{F_2F_1} + P_2 P_2^* S_{F_1F_1}$$

or

$$S_{w_1w_1} = K_{1\theta}^2 S_{F_2F_2} + K_{1\theta} P_2 S_{F_1F_2} + P_2^* K_{1\theta} S_{F_1F_2}^* + \left|P_2\right|^2 S_{F_1F_1}$$

$$S_{w_1w_1} = K_{1\theta}^2 S_{F_2F_2} + 2K_{1\theta}\,\mathrm{Re}\left\{P_2 S_{F_1F_2}\right\} + \left|P_2\right|^2 S_{F_1F_1} \tag{5.14}$$

To now get the spectral density functions we only need to consider the F_1 functions further. They can be written as:

$$F_1 = F_{1vib} + F_{1bm} + F_{1n}$$

$$F_{1vib} = \textit{vibration fluctuations affecting } \phi$$
$$F_{1bm} = \textit{Brownian motion fluctuations affecting } \phi$$
$$F_{1n} = \textit{control voltage noise fluctuations affecting } \phi$$

They are determined by

$$F_{1vib} = \tau_{1vib}$$
$$F_{1bm} = \tau_{1bm}$$
$$F_{1n} = K_{1u} u_n + K_{1v} v_n$$

The second F function can be similarly considered to give:

$$F_2 = F_{2vib} + F_{2bm} + F_{2n}$$

$$F_{2vib} = \textit{vibration fluctuations affecting } \theta$$
$$F_{2bm} = \textit{Brownian motion fluctuations affecting } \theta$$
$$F_{2n} = \textit{control voltage noise fluctuations affecting } \theta$$

They are determined by

$$F_{2vib} = \tau_{2vib}$$
$$F_{2bm} = \tau_{2bm}$$
$$F_{2n} = K_{2u} u_n + K_{2v} v_n$$

There are three types of noise sources: (1) control voltage fluctuations, (2) platform vibration, and (3) Brownian motion noise. Each type is uncorrelated with the other types of noise. Thus, the input noise expressions simplify to

$$S_{F_1F_1} = S_{F_{1vib}F_{1vib}} + S_{F_{1bm}F_{1bm}} + S_{F_{1n}F_{1n}} \tag{5.15a}$$

$$S_{F_2F_2} = S_{F_{2vib}F_{2vib}} + S_{F_{2bm}F_{2bm}} + S_{F_{2n}F_{2n}} \tag{5.15b}$$

$$S_{F_1F_2} = S_{F_{1vib}F_{2vib}} + S_{F_{1bm}F_{2bm}} + S_{F_{1n}F_{2n}} \tag{5.15c}$$

Note that the model permits cross-correlations and cross spectral densities within each type of noise (for example, actuating voltages u and v can be correlated.)

$$S_{F_{1vib}F_{1vib}} = S_{\tau_{1vib}\tau_{1vib}}$$

$$S_{F_{2vib}F_{2vib}} = S_{\tau_{2vib}\tau_{2vib}}$$

$$S_{F_{1vib}F_{2vib}} = S_{\tau_{1vib}\tau_{2vib}}$$

$$S_{F_{1n}F_{1n}} = K_{1u}^2 S_{u_nu_n} + K_{1u}K_{1v}S_{u_nv_n} + K_{1v}K_{1u}S_{v_nu_n} + K_{1v}K_{1v}S_{v_nv_n}$$

$$S_{F_{2n}F_{2n}} = K_{2u}^2 S_{u_nu_n} + K_{2u}K_{2v}S_{u_nv_n} + K_{2v}K_{2u}S_{v_nu_n} + K_{2v}^2 S_{v_nv_n}$$

$$S_{F_{1n}F_{2n}} = K_{1u}K_{2u}S_{u_nu_n} + K_{1u}K_{2v}S_{u_nv_n} + K_{1v}K_{2u}S_{v_nu_n} + K_{1v}K_{2v}S_{v_nv_n}$$

Inserting equations (5.15) (a)-(c) into equation (5.14) results in

$$\begin{aligned} S_{w_1w_1} = {} & K_{1\theta}^2\left[S_{F_{2vib}F_{2vib}} + S_{F_{2bm}F_{2bm}} + S_{F_{2n}F_{2n}}\right] + 2K_{1\theta}\,\mathrm{Re}\left\{P_2\left[S_{F_{1vib}F_{2vib}} + S_{F_{1bm}F_{2bm}} + S_{F_{1n}F_{2n}}\right]\right\} \\ & + \left|P_2\right|^2\left[S_{F_{1vib}F_{1vib}} + S_{F_{1bm}F_{1bm}} + S_{F_{1n}F_{1n}}\right] \end{aligned}$$

$$\begin{aligned} S_{w_1w_1} = {} & K_{1\theta}^2 S_{F_{2vib}F_{2vib}} + 2K_{1\theta}\,\mathrm{Re}\left\{P_2 S_{F_{1vib}F_{2vib}}\right\} + \left|P_2\right|^2 S_{F_{1vib}F_{1vib}} + K_{1\theta}^2 S_{F_{2bm}F_{2bm}} + 2K_{1\theta}\,\mathrm{Re}\left\{P_2 S_{F_{1bm}F_{2bm}}\right\} \\ & + \left|P_2\right|^2 S_{F_{1bm}F_{1bm}} + K_{1\theta}^2 S_{F_{2n}F_{2n}} + 2K_{1\theta}\,\mathrm{Re}\left\{P_2 S_{F_{1n}F_{2n}}\right\} + \left|P_2\right|^2 S_{F_{1n}F_{1n}} \end{aligned}$$

$$
\begin{aligned}
S_{w_1w_1} = {} & K_{1\theta}^2 S_{\tau_{2vib}\tau_{2vib}} + 2K_{1\theta}\,\mathrm{Re}\left\{P_2\left[S_{\tau_{1vib}\tau_{2vib}}\right]\right\} + \left|P_2\right|^2\left[S_{\tau_{1vib}\tau_{1vib}}\right] \\
& + K_{1\theta}^2\left[K_{2u}^2 S_{u_nu_n} + K_{2u}K_{2v}S_{u_nv_n} + K_{2v}K_{2u}S_{v_nu_n} + K_{2v}^2 S_{v_nv_n}\right] \\
& + 2K_{1\theta}\,\mathrm{Re}\left\{P_2\left[K_{1u}K_{2u}S_{u_nu_n} + K_{1u}K_{2v}S_{u_nv_n} + K_{1v}K_{2u}S_{v_nu_n} + K_{1v}K_{2v}S_{v_nv_n}\right]\right\} \\
& + \left|P_2\right|^2\left[K_{1u}^2 S_{u_nu_n} + K_{1u}K_{1v}S_{u_nv_n} + K_{1v}K_{1u}S_{v_nu_n} + K_{1v}K_{1v}S_{v_nv_n}\right] \\
& + S_{1BM}
\end{aligned}
$$

where the Brownian motion power spectral density is given by

$$
S_{1BM} = K_{1\theta}^2 S_{\tau_{2bm}\tau_{2bm}} + 2K_{1\theta}\,\mathrm{Re}\left\{P_2 S_{\tau_{1bm}\tau_{2bm}}\right\} + \left|P_2\right|^2 S_{\tau_{1bm}\tau_{1bm}}
$$

Simplification yields

$$
\begin{aligned}
S_{w_1w_1} = {} & K_{1\theta}^2 S_{\tau_{2vib}\tau_{2vib}} + K_{1\theta}\left[P_2\left[S_{\tau_{1vib}\tau_{2vib}}\right] + P_2^*\left[S^*{}_{\tau_{1vib}\tau_{2vib}}\right]\right] + \left|P_2\right|^2\left[S_{\tau_{1vib}\tau_{1vib}}\right] \\
& + K_{1\theta}^2\left[K_{2u}^2 S_{u_nu_n} + K_{2u}K_{2v}S_{u_nv_n} + K_{2v}K_{2u}S_{v_nu_n} + K_{2v}^2 S_{v_nv_n}\right] \\
& + K_{1\theta}P_2\left[K_{1u}K_{2u}S_{u_nu_n} + K_{1u}K_{2v}S_{u_nv_n} + K_{1v}K_{2u}S_{v_nu_n} + K_{1v}K_{2v}S_{v_nv_n}\right] \\
& + K_{1\theta}P_2^*\left[K_{1u}K_{2u}S^*{}_{u_nu_n} + K_{1u}K_{2v}S^*{}_{u_nv_n} + K_{1v}K_{2u}S^*{}_{v_nu_n} + K_{1v}K_{2v}S^*{}_{v_nv_n}\right] \\
& + \left|P_2\right|^2\left[K_{1u}^2 S_{u_nu_n} + K_{1u}K_{1v}S_{u_nv_n} + K_{1v}K_{1u}S_{v_nu_n} + K_{1v}K_{1v}S_{v_nv_n}\right] + S_{1BM}
\end{aligned}
$$

$$
\begin{aligned}
S_{w_1w_1} = {} & K_{1\theta}^2 S_{\tau_{2vib}\tau_{2vib}} + \left|P_2\right|^2 S_{\tau_{1vib}\tau_{1vib}} + P_2K_{1\theta}S_{\tau_{1vib}\tau_{2vib}} + P_2^*K_{1\theta}S^*{}_{\tau_{1vib}\tau_{2vib}} \\
& + \left[K_{1\theta}^2K_{2u}^2 + P_2K_{1\theta}K_{1u}K_{2u} + P_2^*K_{1\theta}K_{1u}K_{2u} + \left|P_2\right|^2 K_{1u}^2\right]S_{u_nu_n} \\
& + \left[K_{1\theta}^2K_{2v}^2 + P_2K_{1\theta}K_{1v}K_{2v} + P_2^*K_{1\theta}K_{1v}K_{2v} + \left|P_2\right|^2 K^2{}_{1v}\right]S_{v_nv_n} \\
& + \left[\begin{array}{l} 2K_{1\theta}^2K_{2u}K_{2v} + P_2K_{1\theta}K_{1u}K_{2v} + P_2K_{1\theta}K_{1v}K_{2u} + P_2^*K_{1\theta}K_{1u}K_{2v} \\ + K_{1v}\left(P_2^*K_{1\theta}K_{2u} + 2\left|P_2\right|^2 K_{1u}\right) \end{array}\right]S_{u_nv_n} + S_{1BM}
\end{aligned}
$$

Development of $S_{w_2w_2}(j\omega)$ or $S_{w_2w_2}$

One begins with

$$
R_{w_2w_2}(\tau) = E\left[w_2(t)\,w_2(t+\tau)\right] = E\left[\left(K_{2\phi}F_1(t) + L_1F_2(t)\right)\left(K_{2\phi}F_1(t+\tau) + L_1F_2(t+\tau)\right)\right]
$$

$$R_{w_2w_2}(\tau) = K_{2\phi}^2 E\left[F_1(t)F_1(t+\tau)\right] + K_{2\phi}E\left[L_1F_2(t)F_1(t+\tau)\right] + K_{2\phi}E\left[F_1(t)L_1F_2(t+\tau)\right] + E\left[L_1F_2(t)L_1F_2(t+\tau)\right]$$

$$R_{w_2w_2}(\tau) = K_{2\phi}^2 R_{F_1F_1}(\tau) + K_{2\phi}R_{L_1F_2\,F_1}(\tau) + K_{2\phi}R_{F_1\,L_1F_2}(\tau) + R_{L_1F_2\,L_1F_2}(\tau)$$

Taking the Fourier transform yields

$$S_{w_2w_2} = K_{2\phi}^2 S_{F_1F_1} + K_{2\phi}S_{L_1F_2\,F_1} + K_{2\phi}S_{F_1\,L_1F_2} + S_{L_1F_2\,L_1F_2}$$

Using the derivative property for spectral density functions yields

$$S_{w_2w_2} = K_{2\phi}^2 S_{F_1F_1} + K_{2\phi}P_1S_{F_2F_1} + P_1^*K_{2\phi}S_{F_1F_2} + P_1P_1^*S_{F_2F_2}$$

or

$$S_{w_2w_2} = K_{2\phi}^2 S_{F_1F_1} + K_{2\phi}P_1S_{F_2F_1} + P_1^*K_{2\phi}S_{F_2F_1}^* + |P_1|^2 S_{F_2F_2}$$

$$S_{w_2w_2} = K_{2\phi}^2 S_{F_1F_1} + 2K_{2\phi}\,\mathrm{Re}\left\{P_1S_{F_2F_1}\right\} + |P_1|^2 S_{F_2F_2}$$

Following a similar development as in the derivation of $S_{w_1w_1}$ yields:

$$\begin{aligned} S_{w_2w_2} = {} & K_{2\phi}^2 S_{\tau_{1vib}\tau_{1vib}} + 2K_{2\phi}\,\mathrm{Re}\left\{P_1\left[S_{\tau_{2vib}\tau_{1vib}}\right]\right\} + |P_2|^2\left[S_{\tau_{2vib}\tau_{2vib}}\right] \\ & + K_{2\phi}^2\left[K_{1v_n}^2 S_{v_nv_n} + K_{1v}K_{1u}S_{v_nu_n} + K_{1u}K_{1v}S_{u_nv_n} + K_{1u}^2 S_{u_nu_n}\right] \\ & + 2K_{2\phi}\,\mathrm{Re}\left\{P_1\left[K_{2v}K_{1v}S_{v_nv_n} + K_{2v}K_{1u}S_{v_nu_n} + K_{2u}K_{1v}S_{u_nv_n} + K_{2u}K_{1u}S_{u_nu_n}\right]\right\} \\ & + |P_1|^2\left[K_{2v}^2 S_{v_nv_n} + K_{2v}K_{2u}S_{v_nu_n} + K_{2u}K_{2v}S_{u_nv_n} + K_{2u}K_{2u}S_{u_nu_n}\right] \\ & + S_{2BM} \end{aligned}$$

where the Brownian motion power spectral density is given by

$$S_{2BM} = K_{2\phi}^2 S_{\tau_{1vib}\tau_{1vib}} + 2K_{2\phi}\,\mathrm{Re}\left\{P_1S_{\tau_{2vib}\tau_{1vib}}S_{\tau_{2vib}\tau_{1vib}}\right\} + |P_1|^2 S_{\tau_{2vib}\tau_{2vib}}$$

Simplification yields

$$
\begin{aligned}
S_{w_2w_2} &= K_{2\phi}^2 S_{\tau_{1vib}\tau_{1vib}} + K_{2\phi}\left[P_1\left[S_{\tau_{2vib}\tau_{1vib}}\right] + P_1^*\left[S^*_{\tau_{2vib}\tau_{1vib}}\right]\right] + \left|P_1\right|^2\left[S_{\tau_{2vib}\tau_{2vib}}\right] \\
&\quad + K_{2\phi}^2\left[K_{1v}^2 S_{v_nv_n} + K_{1v}K_{1u}S_{v_nu_n} + K_{1u}K_{1v}S_{u_nv_n} + K_{1u}^2 S_{u_nu_n}\right] \\
&\quad + K_{2\phi}P_1\left[K_{2v}K_{1v}S_{v_nv_n} + K_{2v}K_{1u}S_{v_nu_n} + K_{2u}K_{1v}S_{u_nv_n} + K_{2u}K_{1u}S_{u_nu_n}\right] \\
&\quad + K_{2\phi}P_1^*\left[K_{2v}K_{1v}S^*_{v_nv_n} + K_{2v}K_{1u}S^*_{v_nu_n} + K_{2u}K_{1v}S^*_{u_nv_n} + K_{2u}K_{1u}S^*_{u_nu_n}\right] \\
&\quad + \left|P_1\right|^2\left[K_{2v}^2 S_{v_nv_n} + K_{2v}K_{2u}S_{v_nu_n} + K_{2u}K_{2v}S_{u_nv_n} + K_{2u}K_{2u}S_{u_nu_n}\right] + S_{2BM} \\
S_{w_2w_2} &= K_{2\phi}^2 S_{\tau_{1vib}\tau_{1vib}} + \left|P_1\right|^2 S_{\tau_{2vib}\tau_{2vib}} + P_1K_{2\phi}S_{\tau_{2vib}\tau_{1vib}} + P_1^*K_{2\phi}S^*_{\tau_{2vib}\tau_{1vib}} \\
&\quad + \left[K_{2\phi}^2K_{1v}^2 + P_1K_{2\phi}K_{2v}K_{1v} + P_1^*K_{2\phi}K_{2v}K_{1v} + \left|P_1\right|^2 K_{2v}^2\right]S_{v_nv_n} \\
&\quad + \left[K_{2\phi}^2K_{1u}^2 + P_1K_{2\phi}K_{2u}K_{1u} + P_1^*K_{2\phi}K_{2u}K_{1u} + \left|P_1\right|^2 K^2{}_{2u}\right]S_{u_nu_n} \\
&\quad + \left[\begin{array}{l} 2K_{2\phi}^2K_{1v}K_{1u} + P_1K_{2\phi}K_{2v}K_{1u} + P_1K_{2\phi}K_{2u}K_{1v} + P_1^*K_{2\phi}K_{2v}K_{1u} \\ + K_{2u}\left(P_1^*K_{2\phi}K_{2v} + 2\left|P_1\right|^2 K_{2v}\right) \end{array}\right]S_{v_nu_n} + S_{2BM}
\end{aligned}
$$

Development of $S_{w_1w_2}\left(j\omega\right)$ or $S_{w_1w_2}$

We begin with

$$
R_{w_1w_2}\left(\tau\right) = E\left[w_1\left(t\right)w_2\left(t+\tau\right)\right] = E\left[\left(K_{1\theta}F_2\left(t\right) + L_2F_1\left(t\right)\right)\left(K_{2\phi}F_1\left(t+\tau\right) + L_1F_2\left(t+\tau\right)\right)\right]
$$

$$
\begin{aligned}
R_{w_1w_2}\left(\tau\right) &= K_{1\theta}K_{2\phi}E\left[F_2\left(t\right)F_1\left(t+\tau\right)\right] + K_{1\theta}E\left[F_2\left(t\right)L_1F_2\left(t+\tau\right)\right] \\
&\quad + K_{2\phi}E\left[L_2F_1\left(t\right)F_1\left(t+\tau\right)\right] + E\left[L_2F_1\left(t\right)L_1F_2\left(t+\tau\right)\right]
\end{aligned}
$$

$$
R_{w_1w_2}\left(\tau\right) = K_{1\theta}K_{2\phi}R_{F_2F_1} + K_{1\theta}R_{F_2L_1F_2} + K_{2\phi}R_{L_2F_1F_1} + R_{L_2F_1L_1F_2}
$$

Taking the Fourier transform yields

$$
S_{w_1w_2}\left(j\omega\right) = K_{1\theta}K_{2\phi}S_{F_2F_1}\left(j\omega\right) + K_{1\theta}S_{F_2L_1F_2}\left(j\omega\right) + K_{2\phi}S_{L_2F_1F_1}\left(j\omega\right) + S_{L_2F_1L_1F_2}\left(j\omega\right)
$$

Using the derivative property for spectral density functions yields

$$
S_{w_1w_2} = K_{1\theta}K_{2\phi}S_{F_2F_1} + K_{1\theta}P_1^*S_{F_2F_2} + K_{2\phi}P_2S_{F_1F_1} + P_2P_1^*S_{F_1F_2}
$$

$$
\begin{aligned}
S_{w_1w_2} = {} & K_{1\theta}K_{2\phi}\left(S^{*}_{\tau_{1vib}\tau_{2vib}} + S^{*}_{F_{1bm}F_{2bm}} + K_{1u}K_{2u}S^{*}_{u_nu_n} + K_{1u}K_{2v}S^{*}_{u_nv_n} + K_{1v}K_{2u}S^{*}_{v_nu_n} + K_{1v}K_{2v}S^{*}_{v_nv_n}\right) \\
& + K_{1\theta}P_1^{*}\left(S_{\tau_{2vib}\tau_{2vib}} + S_{F_{2bm}F_{2bm}} + K_{2u}^2S_{u_nu_n} + K_{2u}K_{2v}S_{u_nv_n} + K_{2v}K_{2u}S_{v_nu_n} + K_{2v}^2S_{v_nv_n}\right) \\
& + K_{2\phi}P_2\left(S_{\tau_{1vib}\tau_{1vib}} + S_{F_{1bm}F_{1bm}} + K_{1u}^2S_{u_nu_n} + K_{1u}K_{1v}S_{u_nv_n} + K_{1v}K_{1u}S_{v_nu_n} + K_{1v}K_{1v}S_{v_nv_n}\right) \\
& + P_2P_1^{*}\left(S_{\tau_{1vib}\tau_{2vib}} + S_{F_{1bm}F_{2bm}} + K_{1u}K_{2u}S_{u_nu_n} + K_{1u}K_{2v}S_{u_nv_n} + K_{1v}K_{2u}S_{v_nu_n} + K_{1v}K_{2v}S_{v_nv_n}\right)
\end{aligned}
$$

$$
\begin{aligned}
S_{w_1w_2} = {} & K_{1\theta}K_{2\phi}\left(S^{*}_{\tau_{1vib}\tau_{2vib}} + K_{1u}K_{2u}S^{*}_{u_nu_n} + K_{1u}K_{2v}S^{*}_{u_nv_n} + K_{1v}K_{2u}S^{*}_{v_nu_n} + K_{1v}K_{2v}S^{*}_{v_nv_n}\right) \\
& + K_{1\theta}P_1^{*}\left(S_{\tau_{2vib}\tau_{2vib}} + K_{2u}^2S_{u_nu_n} + K_{2u}K_{2v}S_{u_nv_n} + K_{2v}K_{2u}S_{v_nu_n} + K_{2v}^2S_{v_nv_n}\right) \\
& + K_{2\phi}P_2\left(S_{\tau_{1vib}\tau_{1vib}} + K_{1u}^2S_{u_nu_n} + K_{1u}K_{1v}S_{u_nv_n} + K_{1v}K_{1u}S_{v_nu_n} + K_{1v}K_{1v}S_{v_nv_n}\right) \\
& + P_2P_1^{*}\left(S_{\tau_{1vib}\tau_{2vib}} + K_{1u}K_{2u}S_{u_nu_n} + K_{1u}K_{2v}S_{u_nv_n} + K_{1v}K_{2u}S_{v_nu_n} + K_{1v}K_{2v}S_{v_nv_n}\right) \\
& + S_{12bm}
\end{aligned}
$$

$$
\begin{aligned}
S_{w_1w_2} = {} & K_{1\theta}K_{2\phi}\left(S_{\tau_{2vib}\tau_{1vib}} + K_{1u}K_{2u}S_{u_nu_n} + K_{1u}K_{2v}S_{v_nu_n} + K_{1v}K_{2u}S_{u_nv_n} + K_{1v}K_{2v}S_{v_nv_n}\right) \\
& + K_{1\theta}P_1^{*}\left(S_{\tau_{2vib}\tau_{2vib}} + K_{2u}^2S_{u_nu_n} + K_{2u}K_{2v}S_{u_nv_n} + K_{2v}K_{2u}S_{v_nu_n} + K_{2v}^2S_{v_nv_n}\right) \\
& + K_{2\phi}P_2\left(S_{\tau_{1vib}\tau_{1vib}} + K_{1u}^2S_{u_nu_n} + K_{1u}K_{1v}S_{u_nv_n} + K_{1v}K_{1u}S_{v_nu_n} + K_{1v}K_{1v}S_{v_nv_n}\right) \\
& + P_2P_1^{*}\left(S_{\tau_{1vib}\tau_{2vib}} + K_{1u}K_{2u}S_{u_nu_n} + K_{1u}K_{2v}S_{u_nv_n} + K_{1v}K_{2u}S_{v_nu_n} + K_{1v}K_{2v}S_{v_nv_n}\right) \\
& + S_{12bm}
\end{aligned}
$$

$$
\begin{aligned}
S_{w_1w_2} = {} & K_{1\theta}K_{2\phi}K_{1u}K_{2u}S_{u_nu_n} + K_{1\theta}P_1^{*}K_{2u}^2S_{u_nu_n} + K_{2\phi}P_2K_{1u}^2S_{u_nu_n} + P_2P_1^{*}K_{1u}K_{2u}S_{u_nu_n} \\
& + K_{1\theta}K_{2\phi}K_{1v}K_{2v}S_{v_nv_n} + K_{1\theta}P_1^{*}K_{2v}^2S_{v_nv_n} + K_{2\phi}P_2K_{1v}K_{1v}S_{v_nv_n} + P_2P_1^{*}K_{1v}K_{2v}S_{v_nv_n} \\
& + K_{1\theta}K_{2\phi}K_{1v}K_{2u}S_{u_nv_n} + K_{1\theta}P_1^{*}K_{2u}K_{2v}S_{u_nv_n} + K_{2\phi}P_2K_{1u}K_{1v}S_{u_nv_n} + P_2P_1^{*}K_{1u}K_{2v}S_{u_nv_n} \\
& + K_{1\theta}K_{2\phi}K_{1u}K_{2v}S_{v_nu_n} + K_{1\theta}P_1^{*}K_{2v}K_{2u}S_{v_nu_n} + K_{2\phi}P_2K_{1v}K_{1u}S_{v_nu_n} + P_2P_1^{*}K_{1v}K_{2u}S_{v_nu_n} \\
& + K_{2\phi}P_2S_{\tau_{1vib}\tau_{1vib}} \\
& + K_{1\theta}P_1^{*}S_{\tau_{2vib}\tau_{2vib}} \\
& + P_2P_1^{*}S_{\tau_{1vib}\tau_{1vib}} \\
& + K_{1\theta}K_{2\phi}S_{\tau_{2vib}\tau_{1vib}} \\
& + S_{12bm}
\end{aligned}
$$

$$
\begin{aligned}
S_{w_1w_2} = &\left(K_{1\theta}K_{2\phi}K_{1u}K_{2u} + K_{1\theta}P_1^*K_{2u}^2 + K_{2\phi}P_2K_{1u}^2 + P_2P_1^*K_{1u}K_{2u}\right)S_{u_nu_n} \\
&+\left(K_{1\theta}K_{2\phi}K_{1v}K_{2v} + K_{1\theta}P_1^*K_{2v}^2 + K_{2\phi}P_2K_{1v}K_{1v} + P_2P_1^*K_{1v}K_{2v}\right)S_{v_nv_n} \\
&+\left(K_{1\theta}K_{2\phi}K_{1v}K_{2u} + K_{1\theta}P_1^*K_{2u}K_{2v} + K_{2\phi}P_2K_{1u}K_{1v} + P_2P_1^*K_{1u}K_{2v}\right)S_{u_nv_n} \\
&+\left(K_{1\theta}K_{2\phi}K_{1u}K_{2v} + K_{1\theta}P_1^*K_{2v}K_{2u} + K_{2\phi}P_2K_{1v}K_{1u} + P_2P_1^*K_{1v}K_{2u}\right)S_{v_nu_n} \\
&+K_{2\phi}P_2S_{\tau_{1vib}\tau_{1vib}} \\
&+K_{1\theta}P_1^*S_{\tau_{2vib}\tau_{2vib}} \\
&+P_2P_1^*S_{\tau_{1vib}\tau_{2vib}} \\
&+K_{1\theta}K_{2\phi}S_{\tau_{2vib}\tau_{1vib}} \\
&+S_{12bm}
\end{aligned}
$$

where

$$
S_{12BM} = K_{1\theta}P_1^*S_{\tau_{2vib}\tau_{2vib}} + K_{2\phi}P_2S_{\tau_{1vib}\tau_{1vib}} + 2\left(K_{1\theta}K_{2\phi} + P_2P_1^*\right)\mathrm{Re}\left\{S_{\tau_{1vib}\tau_{2vib}}\right\}
$$

Since the cross-power spectral densities of the Brownian motion noise is zero (uncorrelated), this simplifies to

$$
S_{12BM} = K_{1\theta}P_1^*S_{\tau_{2vib}\tau_{2vib}} + K_{2\phi}P_2S_{\tau_{1vib}\tau_{1vib}}
$$

Development of $S_{w_2w_1}(j\omega)$ or $S_{w_2w_1}$

$$
R_{w_2w_1}(\tau) = E\left[w_2(t)w_1(t+\tau)\right] = E\left[\left(K_{2\phi}F_1(t) + L_1F_2(t)\right)\left(K_{1\theta}F_2(t+\tau) + L_2F_1(t+\tau)\right)\right]
$$

$$
\begin{aligned}
R_{w_2w_1}(\tau) = &K_{2\phi}K_{1\theta}E\left[F_1(t)F_2(t+\tau)\right] + K_{2\phi}E\left[F_1(t)L_2F_1(t+\tau)\right] \\
&+K_{1\theta}E\left[L_1F_2(t)F_2(t+\tau)\right] + E\left[L_1F_2(t)L_2F_1(t+\tau)\right]
\end{aligned}
$$

$$
R_{w_2w_1}(\tau) = K_{2\phi}K_{1\theta}R_{F_2F_1} + K_{2\phi}R_{F_1L_2F_1} + K_{1\theta}R_{L_1F_2F_2} + R_{L_1F_2L_2F_1}
$$

$$
S_{w_2w_1}(j\omega) = K_{2\phi}K_{1\theta}S_{F_2F_1}(j\omega) + K_{2\phi}P_2^*S_{F_1F_1}(j\omega) + K_{1\theta}P_1S_{F_2F_2}(j\omega) + P_1P_2^*S_{F_2F_1}(j\omega)
$$

$$
S_{w_2w_1} = K_{2\phi}K_{1\theta}S_{F_2F_1} + K_{2\phi}P_2^*S_{F_1F_1} + K_{1\theta}P_1S_{F_2F_2} + P_1P_2^*S_{F_2F_1}
$$

Assuming that $S_{F_1F_2} = S^*_{F_2F_1}$

$$
\begin{aligned}
S_{w_2w_1} &= K_{2\phi}K_{1\theta}\left(S^*_{\tau_{1vib}\tau_{2vib}} + S^*_{F_{1bm}F_{2bm}} + K_{1u}K_{2u}S^*_{u_nu_n} + K_{1u}K_{2v}S^*_{u_nv_n} + K_{1v}K_{2u}S^*_{v_nu_n} + K_{1v}K_{2v}S^*_{v_nv_n}\right) \\
&\quad + K_{2\phi}P_2^*\left(S_{\tau_{1vib}\tau_{1vib}} + S_{F_{1bm}F_{1bm}} + K_{1u}^2S_{u_nu_n} + K_{1u}K_{1v}S_{u_nv_n} + K_{1v}K_{1u}S_{v_nu_n} + K_{1v}K_{1v}S_{v_nv_n}\right) \\
&\quad + K_{1\theta}P_1\left(S_{\tau_{2vib}\tau_{2vib}} + S_{F_{2bm}F_{2bm}} + K_{2u}^2S_{u_nu_n} + K_{2u}K_{2v}S_{u_nv_n} + K_{2v}K_{2u}S_{v_nu_n} + K_{2v}^2S_{v_nv_n}\right) \\
&\quad + P_1P_2^*\left(S_{\tau_{1vib}\tau_{2vib}} + S_{F_{1bm}F_{2bm}} + K_{1u}K_{2u}S_{u_nu_n} + K_{1u}K_{2v}S_{u_nv_n} + K_{1v}K_{2u}S_{v_nu_n} + K_{1v}K_{2v}S_{v_nv_n}\right)
\end{aligned}
$$

$$
\begin{aligned}
S_{w_2w_1} &= K_{2\phi}K_{1\theta}\left(S^*_{\tau_{1vib}\tau_{2vib}} + K_{1u}K_{2u}S^*_{u_nu_n} + K_{1u}K_{2v}S^*_{u_nv_n} + K_{1v}K_{2u}S^*_{v_nu_n} + K_{1v}K_{2v}S^*_{v_nv_n}\right) \\
&\quad + K_{2\phi}P_2^*\left(S_{\tau_{1vib}\tau_{1vib}} + K_{1u}^2S_{u_nu_n} + K_{1u}K_{1v}S_{u_nv_n} + K_{1v}K_{1u}S_{v_nu_n} + K_{1v}K_{1v}S_{v_nv_n}\right) \\
&\quad + K_{1\theta}P_1\left(S_{\tau_{2vib}\tau_{2vib}} + K_{2u}^2S_{u_nu_n} + K_{2u}K_{2v}S_{u_nv_n} + K_{2v}K_{2u}S_{v_nu_n} + K_{2v}^2S_{v_nv_n}\right) \\
&\quad + P_1P_2^*\left(S_{\tau_{1vib}\tau_{2vib}} + K_{1u}K_{2u}S_{u_nu_n} + K_{1u}K_{2v}S_{u_nv_n} + K_{1v}K_{2u}S_{v_nu_n} + K_{1v}K_{2v}S_{v_nv_n}\right) \\
&\quad + K_{2\phi}K_{1\theta}S^*_{F_{1bm}F_{2bm}} + K_{2\phi}P_2^*S_{F_{1bm}F_{1bm}} + K_{1\theta}P_1S_{F_{2bm}F_{2bm}} + P_1P_2^*S_{F_{1bm}F_{2bm}}
\end{aligned}
$$

$$
\begin{aligned}
S_{w_2w_1} &= K_{2\phi}K_{1\theta}\left(S^*_{\tau_{1vib}\tau_{2vib}} + K_{1u}K_{2u}S^*_{u_nu_n} + K_{1u}K_{2v}S^*_{u_nv_n} + K_{1v}K_{2u}S^*_{v_nu_n} + K_{1v}K_{2v}S^*_{v_nv_n}\right) \\
&\quad + K_{2\phi}P_2^*\left(S_{\tau_{1vib}\tau_{1vib}} + K_{1u}^2S_{u_nu_n} + K_{1u}K_{1v}S_{u_nv_n} + K_{1v}K_{1u}S_{v_nu_n} + K_{1v}K_{1v}S_{v_nv_n}\right) \\
&\quad + K_{1\theta}P_1\left(S_{\tau_{2vib}\tau_{2vib}} + K_{2u}^2S_{u_nu_n} + K_{2u}K_{2v}S_{u_nv_n} + K_{2v}K_{2u}S_{v_nu_n} + K_{2v}^2S_{v_nv_n}\right) \\
&\quad + P_1P_2^*\left(S_{\tau_{1vib}\tau_{2vib}} + K_{1u}K_{2u}S_{u_nu_n} + K_{1u}K_{2v}S_{u_nv_n} + K_{1v}K_{2u}S_{v_nu_n} + K_{1v}K_{2v}S_{v_nv_n}\right) \\
&\quad + S_{21BM}
\end{aligned}
$$

where

$$
\begin{aligned}
S_{21BM} &= K_{2\phi}K_{1\theta}S^*_{\tau_{1vib}\tau_{2vib}} + K_{2\phi}P_2^*S_{\tau_{1vib}\tau_{1vib}} \\
&\quad + K_{1\theta}P_1S_{\tau_{2vib}\tau_{2vib}} + P_1P_2^*S_{\tau_{1vib}\tau_{2vib}}
\end{aligned}
$$

this simplifies to:

$$
S_{21BM} = K_{2\phi}P_2^*S_{\tau_{1vib}\tau_{1vib}} + K_{1\theta}P_1S_{\tau_{2vib}\tau_{2vib}}
$$

$$
\begin{aligned}
S_{w_2w_1} &= K_{2\phi}K_{1\theta}S^{*}_{\tau_{1vib}\tau_{2vib}} + K_{2\phi}K_{1\theta}K_{1u}K_{2u}S^{*}_{u_nu_n} + K_{2\phi}K_{1\theta}K_{1u}K_{2v}S^{*}_{u_nv_n} \\
&+ K_{2\phi}K_{1\theta}K_{1v}K_{2u}S^{*}_{v_nu_n} + K_{2\phi}K_{1\theta}K_{1v}K_{2v}S^{*}_{v_nv_n} + K_{2\phi}P_2^{*}S_{\tau_{1vib}\tau_{1vib}} \\
&+ K_{2\phi}P_2^{*}K_{1u}^2S_{u_nu_n} + K_{2\phi}P_2^{*}K_{1u}K_{1v}S_{u_nv_n} + K_{2\phi}P_2^{*}K_{1v}K_{1u}S_{v_nu_n} \\
&+ K_{2\phi}P_2^{*}K_{1v}K_{1v}S_{v_nv_n} + K_{1\theta}P_1S_{\tau_{2vib}\tau_{1vib}} + K_{1\theta}P_1K_{2u}^2S_{u_nu_n} \\
&+ K_{1\theta}P_1K_{2u}K_{2v}S_{u_nv_n} + K_{1\theta}P_1K_{2v}K_{2u}S_{v_nu_n} + K_{1\theta}P_1K_{2v}^2S_{v_nv_n} \\
&+ P_1P_2^{*}S_{\tau_{1vib}\tau_{2vib}} + P_1P_2^{*}K_{1u}K_{2u}S_{u_nu_n} + P_1P_2^{*}K_{1u}K_{2v}S_{u_nv_n} \\
&+ P_1P_2^{*}K_{1v}K_{2u}S_{v_nu_n} + P_1P_2^{*}K_{1v}K_{2v}S_{v_nv_n} + S_{21BM}
\end{aligned}
$$

$$
\begin{aligned}
S_{w_2w_1} &= +K_{2\phi}K_{1\theta}K_{1u}K_{2u}S_{u_nu_n} + P_1P_2^{*}K_{1u}K_{2u}S_{u_nu_n} + K_{2\phi}P_2^{*}K_{1u}^2S_{u_nu_n} + K_{1\theta}P_1K_{2u}^2S_{u_nu_n} \\
&+ K_{2\phi}P_2^{*}K_{1v}K_{1v}S_{v_nv_n} + K_{2\phi}K_{1\theta}K_{1v}K_{2v}S_{v_nv_n} + K_{1\theta}P_1K_{2v}^2S_{v_nv_n} + P_1P_2^{*}K_{1v}K_{2v}S_{v_nv_n} \\
&+ K_{2\phi}K_{1\theta}K_{1v}K_{2u}S_{u_nv_n} + K_{2\phi}P_2^{*}K_{1u}K_{1v}S_{u_nv_n} + K_{1\theta}P_1K_{2u}K_{2v}S_{u_nv_n} + P_1P_2^{*}K_{1u}K_{2v}S_{u_nv_n} \\
&+ K_{2\phi}P_2^{*}K_{1v}K_{1u}S_{v_nu_n} + K_{1\theta}P_1K_{2v}K_{2u}S_{v_nu_n} + K_{2\phi}K_{1\theta}K_{1u}K_{2v}S_{v_nu_n} + P_1P_2^{*}K_{1v}K_{2u}S_{v_nu_n} \\
&+ K_{2\phi}P_2^{*}S_{\tau_{1vib}\tau_{1vib}} \\
&+ K_{1\theta}P_1S_{\tau_{2vib}\tau_{2vib}} \\
&+ P_1P_2^{*}S_{\tau_{1vib}\tau_{2vib}} \\
&+ K_{2\phi}K_{1\theta}S_{\tau_{2vib}\tau_{1vib}} \\
&+ S_{21BM}
\end{aligned}
$$

$$
\begin{aligned}
S_{w_2w_1} &= \left(K_{2\phi}K_{1\theta}K_{1u}K_{2u} + P_1P_2^{*}K_{1u}K_{2u} + K_{2\phi}P_2^{*}K_{1u}^2 + K_{1\theta}P_1K_{2u}^2\right)S_{u_nu_n} \\
&+ \left(K_{2\phi}P_2^{*}K_{1v}K_{1v} + K_{2\phi}K_{1\theta}K_{1v}K_{2v} + K_{1\theta}P_1K_{2v}^2 + P_1P_2^{*}K_{1v}K_{2v}\right)S_{v_nv_n} \\
&+ \left(K_{2\phi}K_{1\theta}K_{1v}K_{2u} + K_{2\phi}P_2^{*}K_{1u}K_{1v} + K_{1\theta}P_1K_{2u}K_{2v} + P_1P_2^{*}K_{1u}K_{2v}\right)S_{u_nv_n} \\
&+ \left(K_{2\phi}P_2^{*}K_{1v}K_{1u} + K_{1\theta}P_1K_{2v}K_{2u} + K_{2\phi}K_{1\theta}K_{1u}K_{2v} + P_1P_2^{*}K_{1v}K_{2u}\right)S_{v_nu_n} \\
&+ K_{2\phi}P_2^{*}S_{\tau_{1vib}\tau_{1vib}} \\
&+ K_{1\theta}P_1S_{\tau_{2vib}\tau_{2vib}} \\
&+ P_1P_2^{*}S_{\tau_{1vib}\tau_{2vib}} \\
&+ K_{2\phi}K_{1\theta}S_{\tau_{2vib}\tau_{1vib}} \\
&+ S_{21BM}
\end{aligned}
$$

Note that cross-axial correlations of noise as denoted by the cross-power spectral densities are represented in the expression of $S_{w_1w_1}$, $S_{w_2w_2}$ $S_{w_2w_1}$ and $S_{w_1w_2}$. These expressions can be written in terms of constants scaling each PSD and C-PSD. For $S_{w_1w_1}$ this expression is

$$\begin{aligned} S_{w_1w_1} = {} & \alpha_1 S_{\tau_{1vib}\tau_{1vib}} + \alpha_2 S_{\tau_{1vib}\tau_{2vib}} + \alpha_3 S_{\tau_{2vib}\tau_{1vib}} + \alpha_4 S_{\tau_{2vib}\tau_{2vib}} \\ & + \alpha_5 S_{u_nu_n} + \alpha_6 S_{u_nv_n} + \alpha_7 S_{v_nu_n} + \alpha_8 S_{v_nv_n} \\ & + \alpha_9 S_{\tau_{1bm}\tau_{1bm}} + \alpha_{10} S_{\tau_{2bm}\tau_{2bm}} \end{aligned} \tag{5.16}$$

where

$$\begin{aligned} \alpha_1 &= \left|P_2\right|^2 \\ \alpha_2 &= P_2 K_{1\theta} \\ \alpha_3 &= P_2^* K_{1\theta} \\ \alpha_4 &= K_{1\theta}^2 \\ \alpha_5 &= K_{1\theta}^2 K_{2u}^2 + P_2 K_{1\theta} K_{1u} K_{2u} + P_2^* K_{1\theta} K_{1u} K_{2u} + \left|P_2\right|^2 K_{1u}^2 \\ \alpha_6 &= K_{1\theta}^2 K_{2u} K_{2v} + P_2 K_{1\theta} K_{1u} K_{2v} + P_2 K_{1\theta} K_{1v} K_{2u} + \left|P_2\right|^2 K_{1u} K_{1v} \\ \alpha_7 &= K_{1\theta}^2 K_{2u} K_{2v} + P_2^* K_{1\theta} K_{1u} K_{2v} + P_2^* K_{1\theta} K_{2u} K_{1v} + \left|P_2\right|^2 K_{1u} K_{1v} \\ \alpha_8 &= K_{1\theta}^2 K_{2v}^2 + P_2 K_{1\theta} K_{1v} K_{2v} + P_2^* K_{1\theta} K_{1v} K_{2v} + \left|P_2\right|^2 K^2{}_{1v} \\ \alpha_9 &= K_{1\theta}^2 \\ \alpha_{10} &= \left|P_2\right|^2 \end{aligned} \tag{5.17}$$

Similarly, $S_{w_2w_2}(j\omega)$ is

$$\begin{aligned} S_{w_2w_2} = {} & \beta_1 S_{\tau_{1vib}\tau_{1vib}} + \beta_2 S_{\tau_{1vib}\tau_{2vib}} + \beta_3 S_{\tau_{2vib}\tau_{1vib}} + \beta_4 S_{\tau_{2vib}\tau_{1vib}} \\ & + \beta_5 S_{u_nu_n} + \beta_6 S_{u_nv_n} + \beta_7 S_{v_nu_n} + \beta_8 S_{v_nv_n} \\ & + \beta_9 S_{\tau_{1bm}\tau_{1bm}} + \beta_{10} S_{\tau_{2bm}\tau_{2bm}} \end{aligned}$$

where

$$\beta_1 = K_{2\phi}^2$$
$$\beta_2 = P_1^* K_{2\phi}$$
$$\beta_3 = P_1 K_{2\phi}$$
$$\beta_4 = \left|P_1\right|^2$$
$$\beta_5 = K_{2\phi}^2 K_{1u}^2 + P_1 K_{2\phi} K_{2u} K_{1u} + P_1^* K_{2\phi} K_{2u} K_{1u} + \left|P_1\right|^2 K^2{}_{2u}$$
$$\beta_6 = K_{2\phi}^2 K_{1v} K_{1u} + P_1^* K_{2\phi} K_{2v} K_{1u} + P_1^* K_{2\phi} K_{2u} K_{2v} + \left|P_1\right|^2 K_{2u} K_{2v}$$
$$\beta_7 = K_{2\phi}^2 K_{1v} K_{1u} + P_1 K_{2\phi} K_{2v} K_{1u} + P_1 K_{2\phi} K_{2u} K_{1v} + \left|P_1\right|^2 K_{2u} K_{2v}$$
$$\beta_8 = K_{2\phi}^2 K_{1v}^2 + P_1 K_{2\phi} K_{2v} K_{1v} + P_1^* K_{2\phi} K_{2v} K_{1v} + \left|P_1\right|^2 K_{2v}^2$$
$$\beta_9 = K_{2\phi}^2$$
$$\beta_{10} = \left|P_1\right|^2$$

Similarly $S_{w_1 w_2}(j\omega)$ is

$$\begin{aligned} S_{w_1 w_2} = {} & \gamma_1 S_{\tau_{1vib}\tau_{1vib}} + \gamma_2 S_{\tau_{1vib}\tau_{2vib}} + \gamma_3 S_{\tau_{2vib}\tau_{1vib}} + \gamma_4 S_{\tau_{2vib}\tau_{2vib}} \\ & + \gamma_5 S_{u_n u_n} + \gamma_6 S_{u_n v_n} + \gamma_7 S_{v_n u_n} + \gamma_8 S_{v_n v_n} \\ & + \gamma_9 S_{\tau_{1bm}\tau_{1bm}} + \gamma_{10} S_{\tau_{2bm}\tau_{2bm}} \end{aligned}$$

where

$$\gamma_1 = K_{2\phi} P_2$$
$$\gamma_2 = P_2 P_1^*$$
$$\gamma_3 = K_{1\theta} K_{2\phi}$$
$$\gamma_4 = K_{1\theta} P_1^*$$
$$\gamma_5 = K_{1\theta} K_{2\phi} K_{1u} K_{2u} + K_{1\theta} P_1^* K_{2u}^2 + K_{2\phi} P_2 K_{1u}^2 + P_2 P_1^* K_{1u} K_{2u}$$
$$\gamma_6 = K_{1\theta} K_{2\phi} K_{1v} K_{2u} + K_{1\theta} P_1^* K_{2u} K_{2v} + K_{2\phi} P_2 K_{1u} K_{1v} + P_2 P_1^* K_{1u} K_{2v}$$
$$\gamma_7 = K_{1\theta} K_{2\phi} K_{1u} K_{2v} + K_{1\theta} P_1^* K_{2v} K_{2u} + K_{2\phi} P_2 K_{1v} K_{1u} + P_2 P_1^* K_{1v} K_{2u}$$
$$\gamma_8 = K_{1\theta} K_{2\phi} K_{1v} K_{2v} + K_{1\theta} P_1^* K_{2v}^2 + K_{2\phi} P_2 K_{1v} K_{1v} + P_2 P_1^* K_{1v} K_{2v}$$
$$\gamma_9 = K_{2\phi} P_2$$
$$\gamma_{10} = K_{1\theta} P_1^*$$

Similarly $S_{w_2 w_1}(j\omega)$ is

$$
\begin{aligned}
S_{w_2 w_1} = {} & \chi_1 S_{\tau_{1vib}\tau_{1vib}} + \chi_2 S_{\tau_{1vib}\tau_{2vib}} + \chi_3 S_{\tau_{2vib}\tau_{1vib}} + \chi_4 S_{\tau_{2vib}\tau_{2vib}} \\
& + \chi_5 S_{u_n u_n} + \chi_6 S_{u_n v_n} + \chi_7 S_{v_n u_n} + \chi_8 S_{v_n v_n} \\
& + \chi_9 S_{\tau_{1bm}\tau_{1bm}} + \chi_{10} S_{\tau_{2bm}\tau_{2bm}}
\end{aligned}
$$

where

$$
\begin{aligned}
\chi_1 &= K_{2\phi} P_2 \\
\chi_2 &= P_2 P_1^* \\
\chi_3 &= K_{1\theta} K_{2\phi} \\
\chi_4 &= K_{1\theta} P_1^* \\
\chi_5 &= K_{1\theta} K_{2\phi} K_{1u} K_{2u} + K_{1\theta} P_1^* K_{2u}^2 + K_{2\phi} P_2 K_{1u}^2 + P_2 P_1^* K_{1u} K_{2u} \\
\chi_6 &= K_{1\theta} K_{2\phi} K_{1v} K_{2u} + K_{1\theta} P_1^* K_{2u} K_{2v} + K_{2\phi} P_2 K_{1u} K_{1v} + P_2 P_1^* K_{1u} K_{2v} \\
\chi_7 &= K_{1\theta} K_{2\phi} K_{1u} K_{2v} + K_{1\theta} P_1^* K_{2v} K_{2u} + K_{2\phi} P_2 K_{1v} K_{1u} + P_2 P_1^* K_{1v} K_{2u} \\
\chi_8 &= K_{1\theta} K_{2\phi} K_{1v} K_{2v} + K_{1\theta} P_1^* K_{2v}^2 + K_{2\phi} P_2 K_{1v} K_{1v} + P_2 P_1^* K_{1v} K_{2v} \\
\chi_9 &= K_{2\phi} P_2 \\
\chi_{10} &= K_{1\theta} P_1^*
\end{aligned}
$$

Although the model assumes that the physical types of noise are uncorrelated (such as the platform vibration not being correlated with Brownian motion noise, or voltage noise being uncorrelated with Brownian motion noise) it does allow for cross-axial correlations within noise sources such as vibration noise or voltage noise. Customarily, the jitter in each of the rotational angles is calculated, i.e., $S_{\phi_\Delta \phi_\Delta}(j\omega)$, $S_{\theta_\Delta \theta_\Delta}(j\omega)$, but in addition, the coupling of tip and tilt rotational motions is accounted for in the model cross-correlations, as denoted by the cross-power spectral densities, $S_{\phi_\Delta \theta_\Delta}(j\omega)$.

The MJM model allows for the resulting jitter output to be considered as the superposition of various power spectral densities and cross-power spectral densities scaled by the transfer function, $\left|H\left(j\omega\right)\right|^2$ and constants, as shown in equations (5.16) and (5.17). Table 5.3(a)-(c) gives weighting factors, α_i for the mirror used for the measurements in Chapter 4 for multiple mirror angular positions. An example transfer function (which itself is angle dependent) associated with this mirror is shown in Figure 5.2. The α_i factors scale the PSD noise source components to yield the total jitter.

Table 5.3 (a)-(c) α_i Values

Parameter α_i	u=0 v=0 $(\phi,\theta)=(0^0,0^0)$	u=10 v=0 $(\phi,\theta)=(0.39^0,0^0)$	u=20 v=0 $(\phi,\theta)=(1.10^0,0^0)$	u=35 v=0 $(\phi,\theta)=(2.8^0,0^0)$
α_1	3.217e+31	3.044e+31	2.530e+31	1.2354e+31
α_2	7.651e+29	1.568e+29	3.176e+29	5.0538e+29
α_3	7.651e+29	1.568e+29	3.176e+29	5.0538e+29
α_4	1.819e+26	8.078e+26	3.988e+27	2.0674e+28
α_5	1.048e+10	1.024e+10	9.443e+9	6.3737e+9
α_6	2.438e+8	7.959e+7	1.671e+8	3.0785e+8
α_7	2.438e+8	7.959e+7	1.671e+8	3.0785e+8
α_8	5.673e+6	6.184e+5	2.955e+6	1.4868e+7
α_9	3.217e+31	3.044e+31	2.530e+31	1.2354e+31
α_{10}	1.819e+26	8.078e+26	3.988e+27	2.0674e+28

(a)

Parameter α_i	u=0 v=10 $(\phi,\theta)=(0^0,0.5^0)$	u=0 v=20 $(\phi,\theta)=(0^0,1.2^0)$	u=0 v=30 $(\phi,\theta)=(0^0,2.1^0)$	u=0 v=40 $(\phi,\theta)=(0^0,3.2^0)$
α_1	2.998e+31	2.352e+31	1.359e+31	3.768e+30
α_2	8.703e+28	3.611e+29	4.249e+29	3.854e+29
α_3	8.703e+28	3.611e+29	4.249e+29	3.854e+29
α_4	2.526e+26	5.541e+27	1.328e+28	3.942e+29
α_5	1.000e+10	8.486e+9	5.654e+9	2.007e+9
α_6	5.617e+7	1.784e+8	2.316e+8	2.345e+8
α_7	5.617e+7	1.784e+8	2.316e+8	2.345e+8
α_8	3.153e+5	3.753e+6	9.491e+6	2.739e+7
α_9	2.998e+31	2.352e+31	1.359e+31	3.768e+30
α_{10}	2.526e+26	5.541e+27	1.328e+28	3.942e+28

(b)

Parameter α_i	u=5 v=5 $(\phi,\theta)=(0.20^0,0.23^0)$	u=15 v=15 $(\phi,\theta)=(0.69^0,0.83^0)$	u=25 v=25 $(\phi,\theta)=(1.6^0,2.2^0)$
α_1	3.121e+31	2.310e+31	6.265e+31
α_2	4.222e+29	3.469e+30	7.238e+30
α_3	4.222e+29	3.469e+30	7.238e+30
α_4	5.711e+27	5.208e+29	8.363e+30
α_5	1.030e+10	8.828e+9	4.890e+9
α_6	2.072e+8	1.848e+9	5.512e+9
α_7	2.072e+8	1.848e+9	5.512e+9
α_8	4.166e+6	3.864e+8	5.901e+9
α_9	3.121e+31	2.310e+31	6.265e+30
α_{10}	5.711e+27	5.208e+29	8.363e+30

(c)

The parameters for the model are determined through measurements and calculations as specified in Chapter 4. *A mirror selected for simulation and eventual testing has constants summarized in Table 5.4.* The units μN, μm and seconds form a consistent set of units as verified through dimensional analysis.

Table 5.4 Mirror Dynamic Constants and Effective Length

$I_1\left(\mu N \mu m\, s^2\right)$	1.84e(-4)
$I_2\left(\mu N \mu m\, s^2\right)$	1.14e(-4)
$D_1\ \left(\mu N\ \mu m\ \mathrm{s}\right)$	0.615
$D_2\ \left(\mu N\ \mu m\ \mathrm{s}\right)$	0.285
$K_{1m}\left(\mu N \mu m\right)$	22605
$K_{2m}\left(\mu N \mu m\right)$	18801
$\ell_{z_{eff}}\left(\mu m\right)$	525
$\ell_{x_{eff}}\left(\mu m\right)$	522

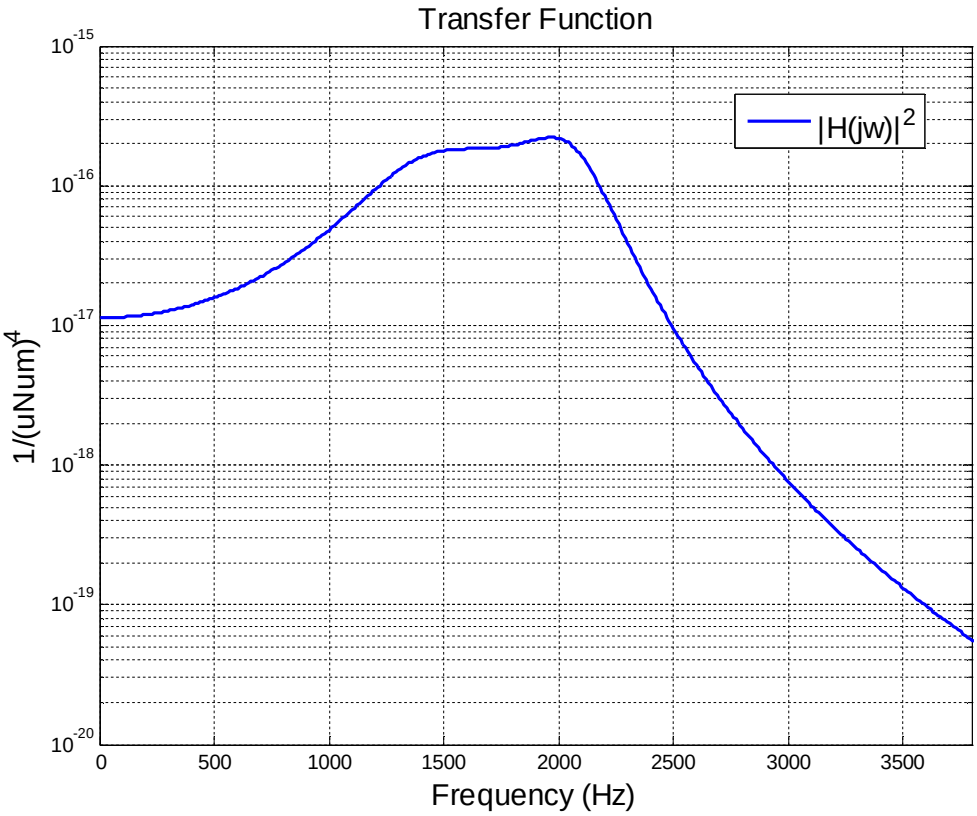


Figure 5.2 Plot of $\left|H\left(jw\right)\right|^2$

It is clear from the values in Table 5.3 that vibration could play a significant role in jitter but because of its low frequency characteristics, as shown in the transfer function of Figure 5.2, it has minimal effect in these circumstances. The voltage fluctuations also are power supply dependent, and Table 5.3 would allow the designer to specify system jitter performance tolerances and then define specifications for the power supplies in terms of control voltage noise.

Simulations

It is important to show that the MJM is a close approximation to the full nonlinear model for operationally relevant input noise source levels. Just as in Chapter 3, the model is programmed into SIMULINK and MATLAB as shown in Figure 5.1 but with zero-mean second order stochastic input processes added to account for different sources of jitter. This is done by specifying the total power or variance of a sampled second-order random signal. Furthermore, the signal can be filtered to account for appropriate power spectral density shape if necessary. Control voltage fluctuation and the Brownian motion of the air (electrical thermal noise and Brownian motion noise) are assumed to be uncorrelated (white) noise. However, since platform vibration generally only manifests itself at low frequencies, a low-pass filter is used to shape the input platform vibration appropriately.

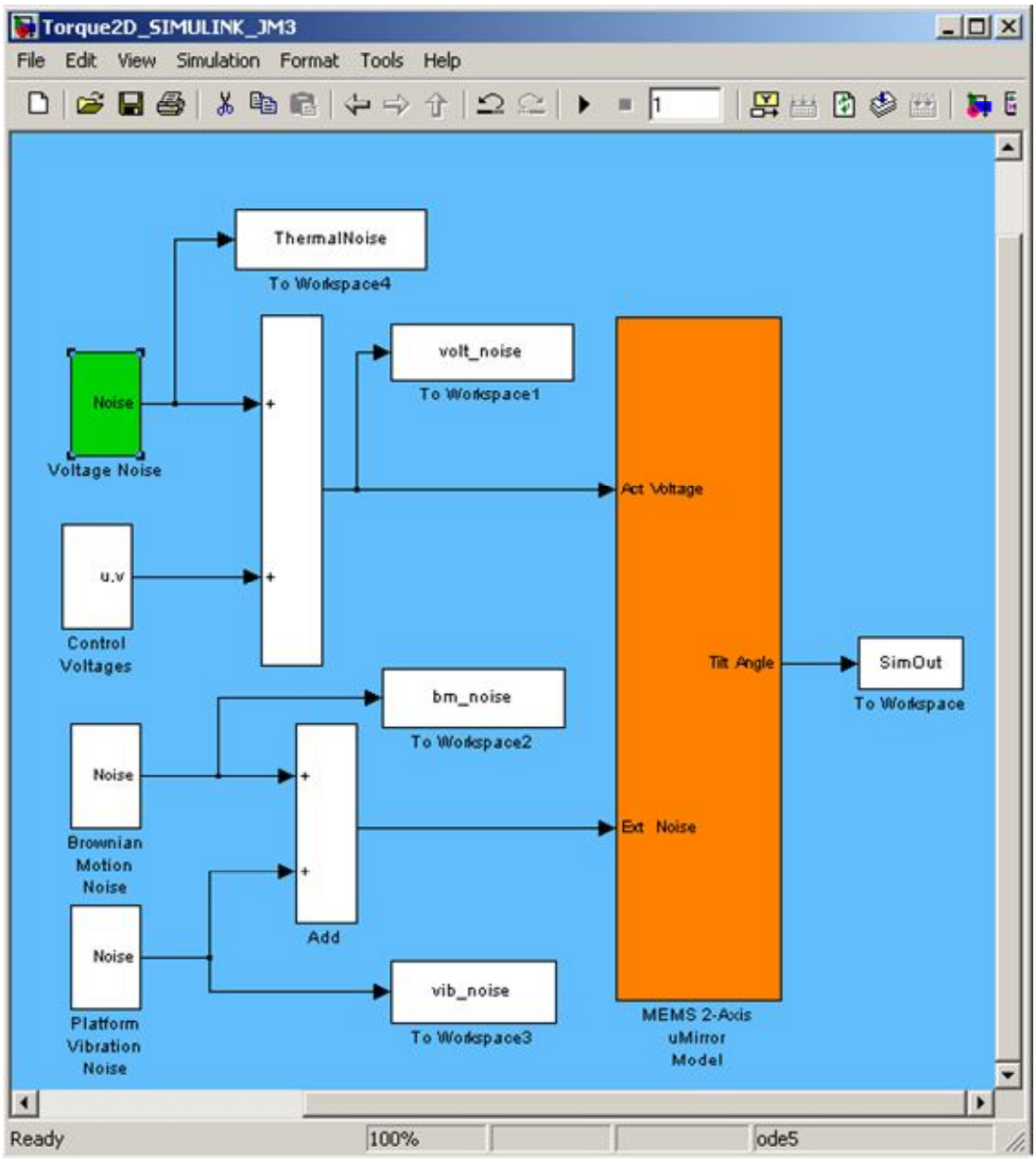


Figure 5.3 Illustration of SIMULINK with Stochastic Inputs

MATLAB's random number generator will produce a random input signal with variance specified by the user. The variance of a zero-mean process is proportional to the integrated area of the power spectral density and sample period. The variance of

the jitter can be determined either from the time-series data or the integrated area of the measured power spectral density.

Because the MJM model is developed through linearization it is important to evaluate how close the linearized MJM model is to the nonlinear solution. A number of metrics were utilized to evaluate the "goodness" of the MJM as compared to the nonlinear model. Four metrics were used

1. Visual inspection of the power spectral densities,
2. Root mean square (RMS) deviation of the PSD linearization vs. nonlinear,
3. The relative difference between the variance of the non-linear model and the linearized model (shown as a percent error), and
4. Single Frequency Deviation at 1610 Hz (near resonance).

The last metric employs a sine wave input to the system. The sine wave produces a delta function in the frequency domain. One can increase the sine wave's amplitude to determine when the linearized model deviates significantly from the nonlinear model. This provided a probing feature for comparison at different frequencies. Table 5.5 gives corresponding mirror angles and actuating voltage relationships.

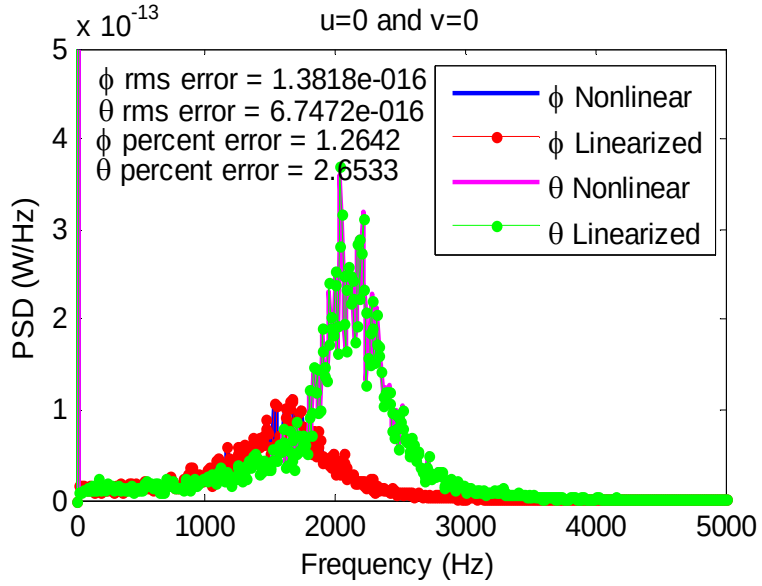
Table 5.5 Control Voltage Vs. Tip-Tilt Angle

u (volts)	v (volts)	ϕ (deg.)	θ (deg.)
0	0	0	0
10	10	0.59	0.75
25	25	1.6	2.2
35	0	2.5	0
0	40	0	3.0

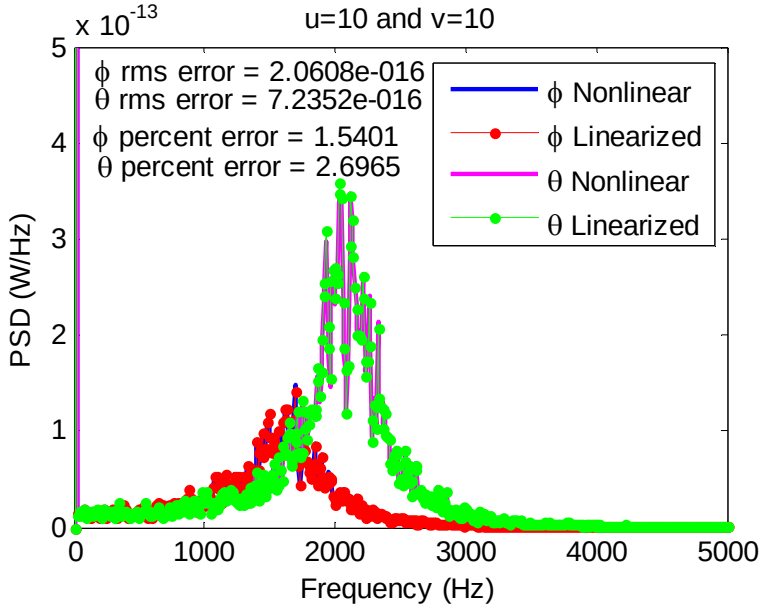
Figures 5.4(a)-(e) are plots of the nonlinear solution compared to the linear solution for varying actuating voltages. Figure 5.4(a) has the mirror biased $V_b = 50\text{v}$ and with both actuating voltages set to 0 volts $(\text{i.e. } u = 0 \text{ and } v = 0)$. Figure 5.4(b) has the mirror with actuating voltages set to 10 volts $(\text{i.e. } u = 10 \text{ and } v = 10)$. Figure 5.4(c) has the mirror with actuating voltages set to 25 volts $(\text{i.e. } u = 25 \text{ and } v = 25)$. Figure 5.4(d) has the mirror with actuating voltages set to 35 and 0 volts respectively $(\text{i.e. } u = 35 \text{ and } v = 0)$ causing the mirror to be close to pull-in for the ϕ axis. Figure 5.4(d) has the mirror with actuating voltages set to 0 and 40 volts respectively $(\text{i.e. } u = 0 \text{ and } v = 40)$ causing the mirror to be close to pull-in for the θ axis.

The MJM is nearly identical to the nonlinear model. As the applied voltages position the mirror closer to pull-in $(u = 27, v = 29)$, the MJM begins to deviate from the

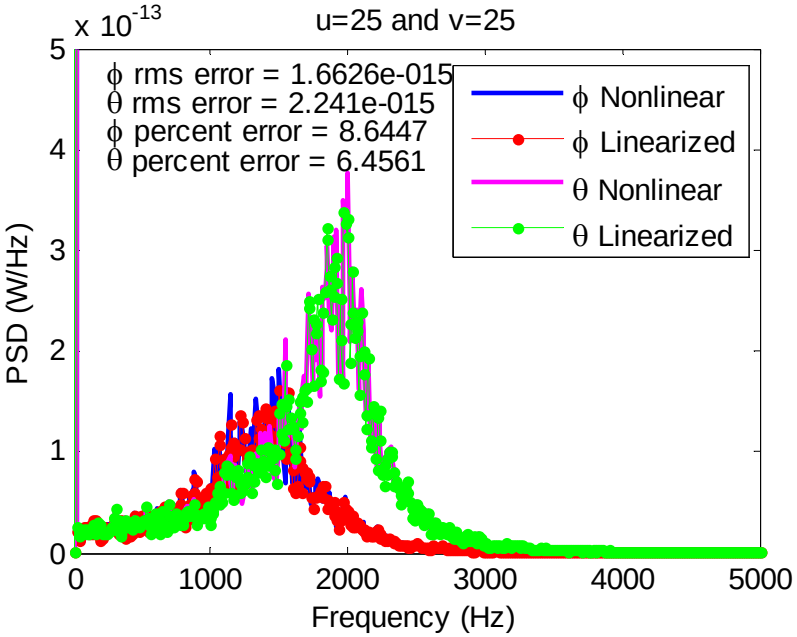
nonlinear model slightly but has smaller errors when operating in single-axis mode close to pull-in. Different noise sources values were simulated with different sources dominating the output jitter.



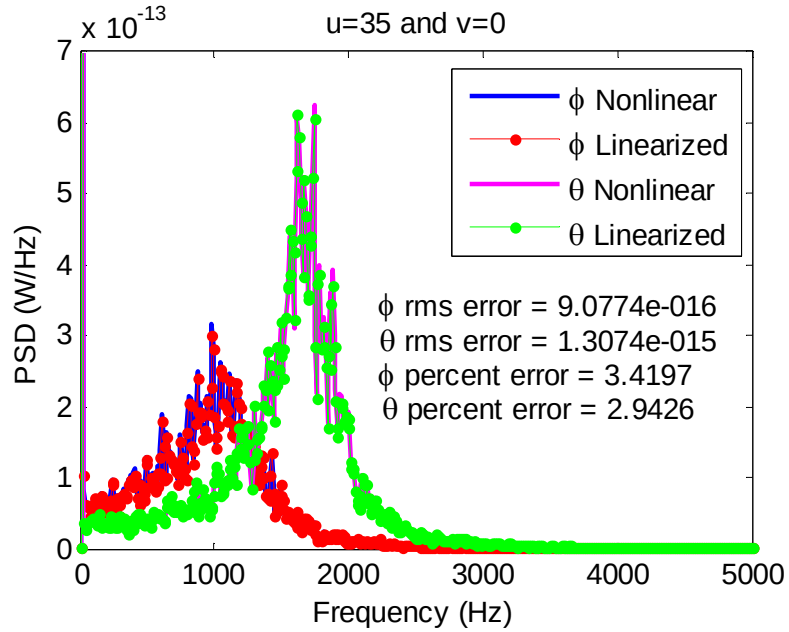
(a)



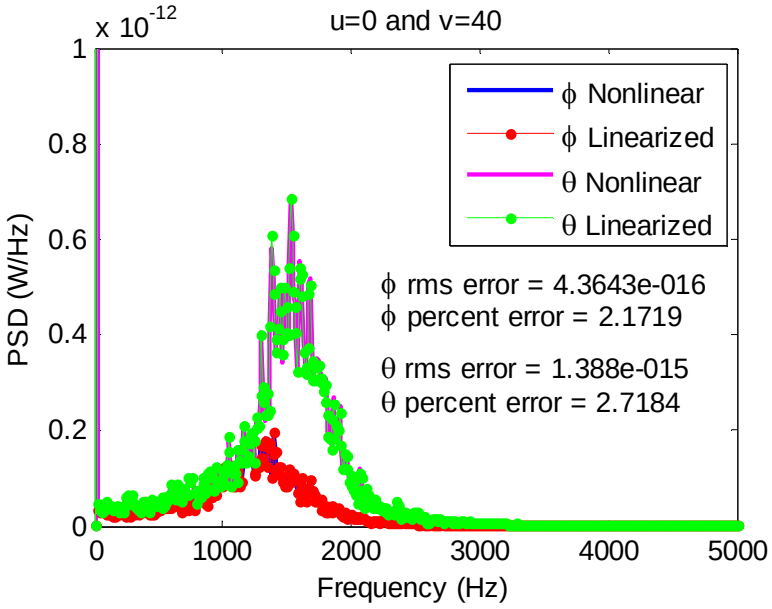
(b)


x 10^-13
u=25 and v=25
ϕ rms error = 1.6626e-015
θ rms error = 2.241e-015
ϕ percent error = 8.6447
θ percent error = 6.4561
ϕ Nonlinear
ϕ Linearized
θ Nonlinear
θ Linearized
PSD (W/Hz)
Frequency (Hz)
0
1
2
3
4
5
0
1000
2000
3000
4000
5000

(c)


x 10^-13
u=35 and v=0
ϕ Nonlinear
ϕ Linearized
θ Nonlinear
θ Linearized
ϕ rms error = 9.0774e-016
θ rms error = 1.3074e-015
ϕ percent error = 3.4197
θ percent error = 2.9426
PSD (W/Hz)
Frequency (Hz)
0
1
2
3
4
5
6
7
0
1000
2000
3000
4000
5000

(d)



(e)

Figure 5.4 (a)-(e) Comparisons of the PSD for the Micro-mirror Jitter Model. Units of θ and ϕ are Radians

The previous plots showed the effects of the mirror position as determined by actuating voltages u and v, given constant values for control voltage noise, Brownian motion noise, and platform vibration. The next step is to hold the mirror position constant and to vary the magnitude of each noise source to determine when the linearized solution begins to deviate significantly from the nonlinear simulation.

Figure 5.5 is an enhanced view of the sinusoidal input enabling a view of the difference between the MJM and nonlinear model. The resulting PSD is the sampled equivalent of a delta function (power in one frequency bin) and allows for a single

frequency to be probed and compared to the nonlinear solution for different sine wave amplitudes.

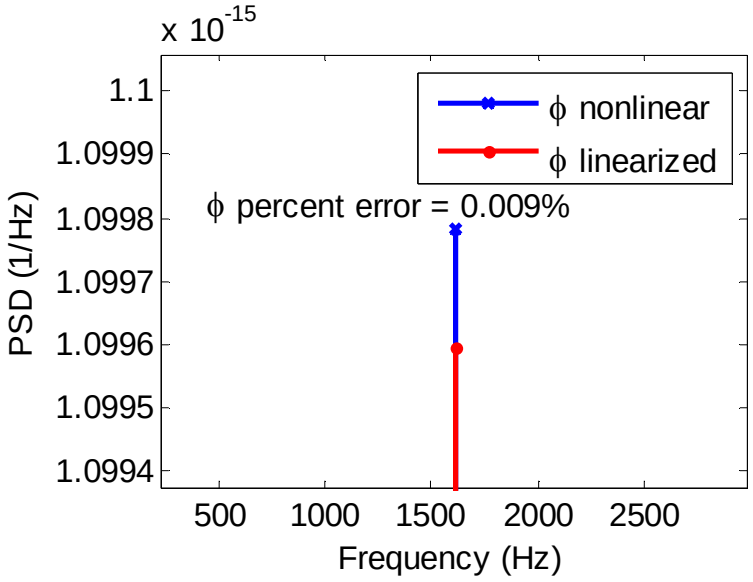


Figure 5.5 Illustration of Linearized Vs. Nonlinear Model for Sine Wave Input.

Table 5.6 gives the relative percent error for a sinusoidal stimulus with varying amplitude at 1610 Hz for different actuating voltages of u showing the sensitivity to the pull-in region. Table 5.7 repeats the process for platform vibration τ_{1vib} as sinusoidal stimulus. This was repeated for the other axis with similar results.

Table 5.6 Relative Percent Error Operating Voltage 'u' and v=0

Sine Wave Amplitude	Voltage u=0 v=0	Voltage u=25 v=0	Voltage u=35 v=0
1e-4	0.005%	0.10%	0.78%
1e-2	0.01%	1.10%	1.08%
1	0.03%	2.10%	3.10%

Table 5.7 Relative Percent Error Platform Vibration τ_{1vib} and 'u' and v=0

Sine Wave Amplitude	Voltage u=0 v=0	Voltage u=25 v=0	Voltage u=35 v=0
1e-12	0.01%	1.00%	1.30%
1e-10	1.00%	1.60%	21.1%

From the simulation results shown in the plots such as in Figure 5.4, and from the values in Tables 5.4 and 5.5, it is clear that if the inputs to the simulation are based on operationally realistic values, then the MJM model is an excellent approximation for the nonlinear solution.

Conclusions

This chapter presents the MJM model development that is based in part on earlier modeling work (2D Torque Model and MPM model) to account for mirror facet jitter due to the three originating noise sources. These three sources are: (1) control voltage fluctuations, (2) platform vibration, and (3) Brownian motion noise (or mechanical-thermal noise.) These sources of noise have been studied and published for other MEMS devices but not for MEMS mirrors [58], [59], [66]. This chapter derives the Micro-mirror Jitter Model (MJM) through a multidimensional first-order Taylor expansion about some deterministic point. The MJM model will enable system engineer to view the mirror facet jitter as the scaled superposition of power spectral densities and cross-power spectral densities of the noise sources. Scaled PSD is multiplied by an angle dependent transfer function. With these explicit relationships, one can predict mirror facet jitter when the noise characteristics of components are specified. Also if a system engineer is given jitter tolerance, he or she can specify the sub-system components' requirements. The MJM model is compared to the nonlinear solution using MATLAB and SIMULINK simulations. The MJM model is an excellent approximation for operationally realistic noise source magnitudes. Furthermore, the MJM model and nonlinear solution only begin to deviate close to pull-in.

Chapter 6: Experimental Measurements and the MJM Model

Overview

This chapter presents the experimental apparatus and measurements for the Micro-mirror Jitter Model (MJM) developed in Chapter 5. The MJM model relates the input power spectral densities (PSD) of the noise sources to the resulting jitter of the micro-mirror facet. The validation will concentrate on the case where the jitter sources are largely dominated by Brownian motion noise as compared to control voltage fluctuations and platform vibration. This case is of particular interest because this situation represents the best-case jitter performance for a MEMS laser pointing system because voltage and vibration can be reduced by design choices.

These are the first direct observations of Brownian motion generated jitter on a MEMS tip-tilt mirror, as far as is known. Brownian motion noise can only be estimated or inferred indirectly through a series of preliminary measurements while the *other* input noise sources can be measured directly. An accurate model that has been validated through experimental measurements is essential so one can tell when the jitter is primarily generated by the Brownian motion or one of the other sources of mirror jitter. An experimental set-up was configured that minimized the other sources of jitter while still monitoring them with the data acquisition system. The data acquisition (DAQ) system adopted to experimentally observe and quantify the effects of the Brownian motion noise consisted of a National Instruments (NI)

CompactDaq™ chassis connected to a laptop via a USB interface, shown in Figure 6.1. This system offered state-of-the-art inter-channel isolation and 24 bits of precision for a ± 10 volt range of inputs. This resulted in the DAQ system having a voltage resolution of $1.2\mu V$. The CompactDaq also enabled the use of modules customized for the sensor being monitored. The four voltages controlling the MEMS device were monitored as well as the three voltages from the quad cell using two NI 9239 analog input modules. These also had anti-aliasing filters and were sampled at a rate of 50K samples-per-second. A single NI 9233 signal conditioning module was used to record accelerometer measurements.

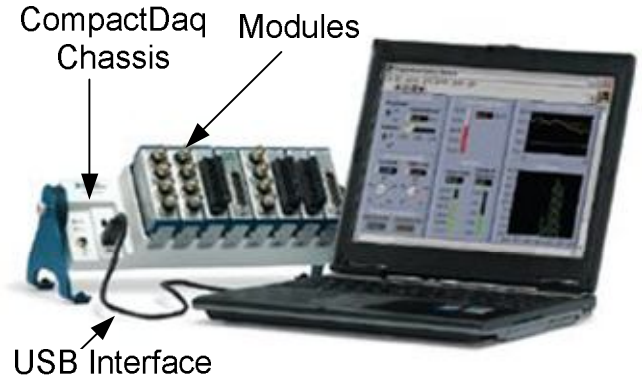


Figure 6.1 National Instruments CompactDaq Chassis with Laptop and Modules

Measurement Requirements for Brownian Motion Induced Jitter

The purpose of this section is to establish a minimum resolution or precision requirement for the experimental apparatus to measure the jitter caused by Brownian motion noise.

One can begin with the equipartition theorem, which states that at thermal equilibrium, the kinetic energy of the air gases must equal the potential energy stored in the mirror's torsional spring. This implies that the energy per degree-of-freedom is $\frac{1}{2}K_B T$, where K_B is Boltzmann's constant and T is absolute temperature. For the two-axis tip-tilt MEMS mirror, the Brownian motion noise will be determined by stochastic processes, $\tau_{1bm}(t)$ and $\tau_{2bm}(t)$ as represented in equation (5.3) where:

$$\frac{1}{2}K_{1m}\left\langle \phi_{bm}{}^2 \right\rangle = \frac{1}{2K_{1m}}\left\langle \tau_{1bm}{}^2 \right\rangle = \frac{1}{2}K_B T$$

and

$$\frac{1}{2}K_{2m}\left\langle \theta_{bm}{}^2 \right\rangle = \frac{1}{2K_{2m}}\left\langle \tau_{2bm}{}^2 \right\rangle = \frac{1}{2}K_B T$$

or

$$\left\langle \tau_{1bm}{}^2 \right\rangle = K_{1m}K_B T$$

and

$$\left\langle \tau_{2bm}{}^2 \right\rangle = K_{2m}K_B T$$

Because τ_{1bm} and τ_{2bm} are zero mean random processes (as result of the input sources of jitter being zero mean random processes), $\left\langle \tau_{1bm}{}^2 \right\rangle$ and $\left\langle \tau_{2bm}{}^2 \right\rangle$ are also the variance of τ_{1bm} and τ_{2bm} with associated power spectral densities (PSDs), $S_{\tau_{1bm}\tau_{1bm}}(j\omega)$ and $S_{\tau_{2bm}\tau_{2bm}}(j\omega)$. These power specral density (PSD) is related to the variance by the expression

$$R_{\tau_{1bm}}[0] = \left\langle \tau_{1bm}{}^2 \right\rangle = \frac{1}{2\pi}\int_{2\pi} S_{\tau_{1bm}\tau_{1bm}}(j\omega)d\omega. \tag{6.1}$$

and

$$R_{\tau_{2bm}}[0] = \left\langle \tau_{2bm}{}^2 \right\rangle = \frac{1}{2\pi}\int_{2\pi} S_{\tau_{2bm}\tau_{2bm}}(j\omega)d\omega \tag{6.2}$$

The ω represents a normalized frequency. Assuming that the amplitude of $S_{\tau_{1bm}\tau_{1bm}}$ and $S_{\tau_{2bm}\tau_{2bm}}$ are flat denoting a white process, equations (6.1) and (6.2) can be used to solve for the amplitude values which for time sampled data reduce to

$$\left\langle \tau_{1bm}{}^2 \right\rangle = S_{\tau_{1bm}\tau_{1bm}} \cdot F_s$$

And

$$\left\langle \tau_{2bm}{}^2 \right\rangle = S_{\tau_{2bm}\tau_{2bm}} \cdot F_s$$

or

$$S_{\tau_{1bm}\tau_{1bm}} = \frac{\left\langle \tau_{1bm}{}^2 \right\rangle}{F_s}$$

and

$$S_{\tau_{2bm}\tau_{2bm}} = \frac{\left\langle \tau_{2bm}{}^2 \right\rangle}{F_s}$$

where F_s is the sampling frequency of the data acquisition system.

This is the amplitude of the input PSD. As shown in Chapter 5, the input and output PSDs are related to each other through the MJM model. Assuming that one has sufficiently minimized the other noise sources, the only noise input affecting the mirror is the Brownian motion. Then all of the PSDs multiplying the α_i's are minimized (close to zero) except those multiplying α_9 and α_{10},which scale the Brownian motion noise PSDs. Equation (5.16) reduces to:

$$S_{\phi\phi} = \left|H(j\omega)\right|^2 \left(\alpha_9 S_{\tau_{1bm}\tau_{1bm}} + \alpha_{10} S_{\tau_{2bm}\tau_{2bm}}\right)$$

The equivalent expression for the θ PSD is:

$$S_{\theta\theta} = \left|H(j\omega)\right|^2 \left(\beta_9 S_{\tau_{1bm}\tau_{1bm}} + \beta_{10} S_{\tau_{2bm}\tau_{2bm}}\right)$$

Again, using the same relationship between the PSD and the variance of a zero-mean process in equations (6.1) and (6.2), the variance of the mirror jitter is $\left\langle \phi^2 \right\rangle$ and $\left\langle \theta^2 \right\rangle$ is calculated from:

$$\left\langle \phi^2 \right\rangle = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S_{\phi\phi}(j\omega)d\omega . \qquad (6.3)$$

and

$$\left\langle \theta^2 \right\rangle = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S_{\theta\theta}(j\omega)d\omega \qquad (6.4)$$

The minimum precision required by the experimental set-up is the square root of the variance or

$$\phi_{\min} = \sqrt{\langle \phi^2 \rangle}. \tag{6.5}$$

and

$$\theta_{\min} = \sqrt{\langle \theta^2 \rangle} \tag{6.6}$$

Using Tables 5.3 and the appropriate transfer functions, Table 6.1 gives the $\phi_{\min}$ values for an experimental set-up that will detect Brownian motion noise on a MEMS Two-Axis (Tip-Tilt) Mirror.

Table 6.1 Required Resolution, $\phi_{\min}$, of Experimental Set-Up

Parameter	$\phi_{\min}$ (nano-radians)
u=0, v=0 $(\phi,\theta)=(0^0,0^0)$	425
u=5, v=5 $(\phi,\theta)=(0.20^0,0.23^0)$	451
u=15, v=15 $(\phi,\theta)=(0.69^0,0.83^0)$	628
u=25, v=25 $(\phi,\theta)=(1.6^0,2.2^0)$	901

Control Voltage Noise Measurements

If the control voltage set-up from the MPM validation had remained unchanged, the jitter would have been dominated by the control voltage noise (approximately 5mV) contribution and the effects of Brownian motion noise would have been difficult to discern. Thus, in order to reduce the effects of the control voltage noise, the quad cell and the MEMS mirror were powered by battery banks rather than switching or 60Hz based power supplies. A picture of one such battery bank is shown in Figure 6.2. Batteries are the best source of power if a low-ripple, flat-spectrum, is desired. Batteries are also as close to a thermal noise limited power source as possible because the voltages are produced from chemical reactions. The batteries were attached to the spectrum analyzer through a DC block to measure the ripple. The voltage noise variance or ripple was smaller than $10^{-9}V$. However, when the MEMS device was attached to the batteries, the voltage noise was recorded and measured to be $50\mu V$ with the NI 9239 modules. Again the precision of the NI 9239 modules was $1.2\mu V$. The mirror was biased with batteries that could range in series from 1.5 to 100 volts.

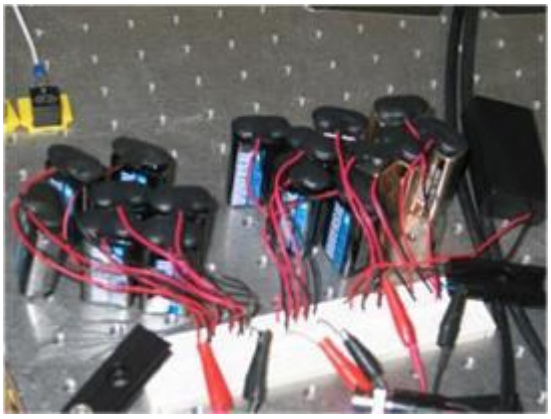


Figure 6.2 Batter Banks Used as Power Supplies to Quad Cell and Mirror

Platform Vibration Measurements

Platform vibration was monitored using an orthogonal triad configuration of Endevco 61C12 accelerometers. Additionally, three Endevco IEPE (Internal Electronics Piezo Electric) 61C12 accelerometers were employed in a triad configuration to measure angular acceleration.

The Endevco 61C12 Accelerometers output a time-series voltage waveform that is proportional to the linear acceleration experienced by the sensor device at its given location. The voltage output is related to acceleration by a scale factor of 100mV/g. Multiplying by the gravitational constant "g" gives acceleration in SI units.

The accelerometers are designed to measure linear accelerations but their outputs can be differenced to infer angular accelerations. To do this, the difference must be taken between adjacent accelerometers as shown in Figure 6.3. The residual time-series

acceleration is twice integrated to determine a displacement. Once a time-series displacement is known, then the statistical properties can be analyzed including the PSD of τ_{1vib} and τ_{2vib} which are proportional to ϕ_{vib} and θ_{vib} and enter into equation (5.3).

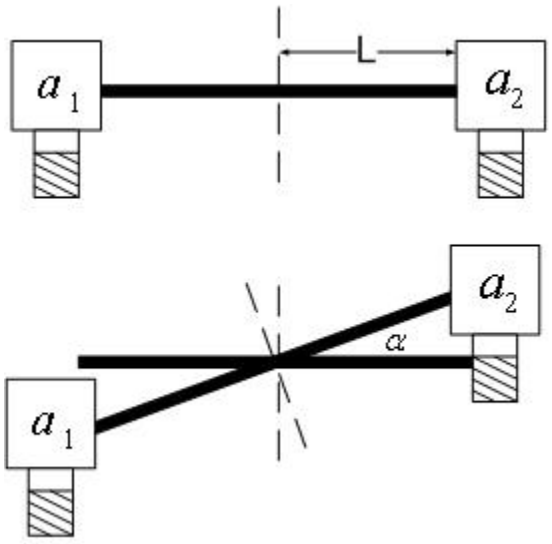


Figure 6.3 Angular Acceleration from Accelerometers

When the optical table is floating, it is expected that the platform vibration will be close to zero for frequencies higher than 2Hz. The optical table and vibration isolation profile are shown in Figure 6.4.

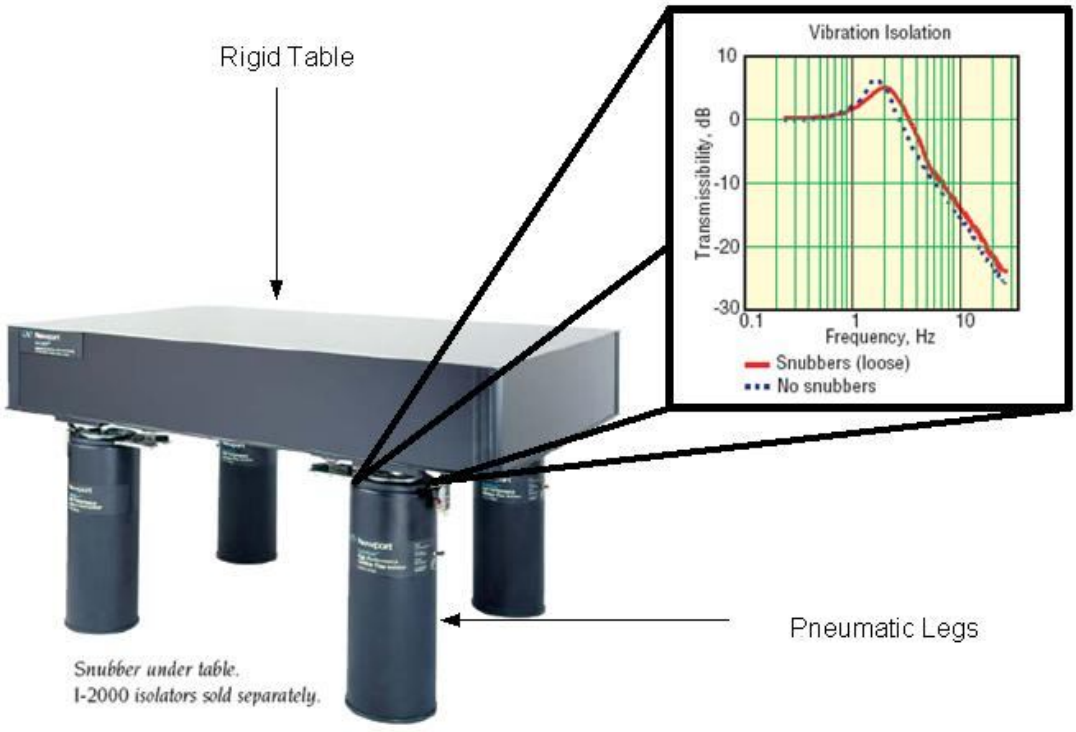


Figure 6.4 Newport Optical Table and Vibration Isolation Spectrum

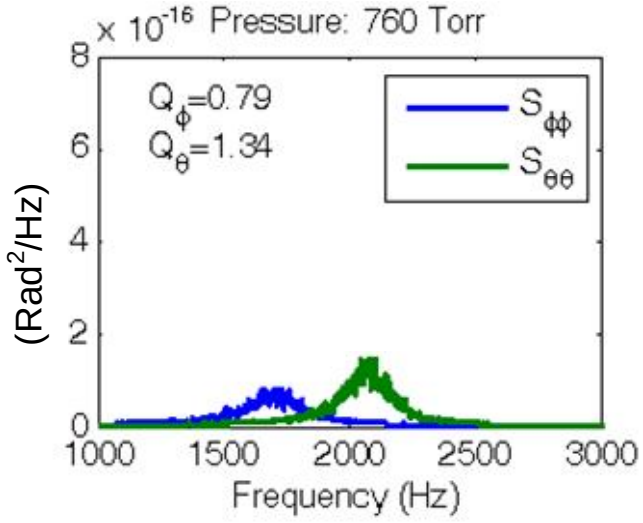
Pressure Variation

A vacuum collar was used to house the MEMS device, as shown in Figure 4.2, which allows the ambient pressure on the mirror to be varied so that for the damping coefficient and the "Q" of the mirror to be determined as a function of pressure. Both the MPM model and MJM model include damping terms which are functions of the ambient pressure surrounding the mirrors. Depending on the environment resulting from land, air, or space operations, the pressure may vary. As the pressure is reduced, the damping factor decreases. As the mirror damping decreases, the mirror's "Q" will increase. By varying the pressure and looking at the Q of the mirror, one can

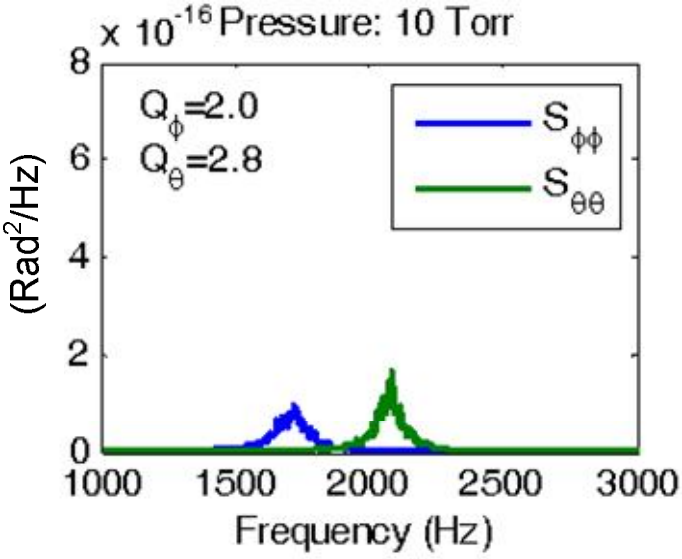
determine the damping coefficients for varying pressures. The pressure variation is accomplished by Pfeiffer mechanical roughing and molecular turbo pumps, which together achieve low pressures of 10^{-6} Torr (760Torr = 1 Atmosphere).

Measurements

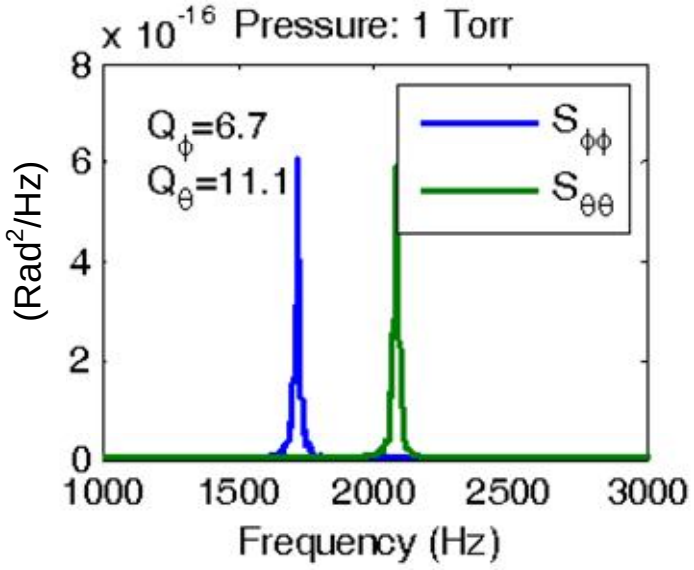
The Brownian motion noise jitter remained constant as a function of pressure as the equipartition theorem requires, but the mirror frequency response changed dramatically as the damping was reduced. The vacuum system allows one to relate the damping coefficients D_1 and D_2 to the in-situ pressure around the mirror in the chamber. For the controls engineer, understanding the viscous damping is paramount to critically damping the mirror as the mirror is pointed in different directions. The vacuum pressure was varied from 760 Torr to 6.3×10^{-6} Torr. Figure 6.5 (a)-(d) shows the mirror's Q increasing for decreasing pressures.


x 10^-16 Pressure: 760 Torr
Q_φ=0.79
Q_θ=1.34
S_φφ
S_θθ
(Rad^2/Hz)
Frequency (Hz)

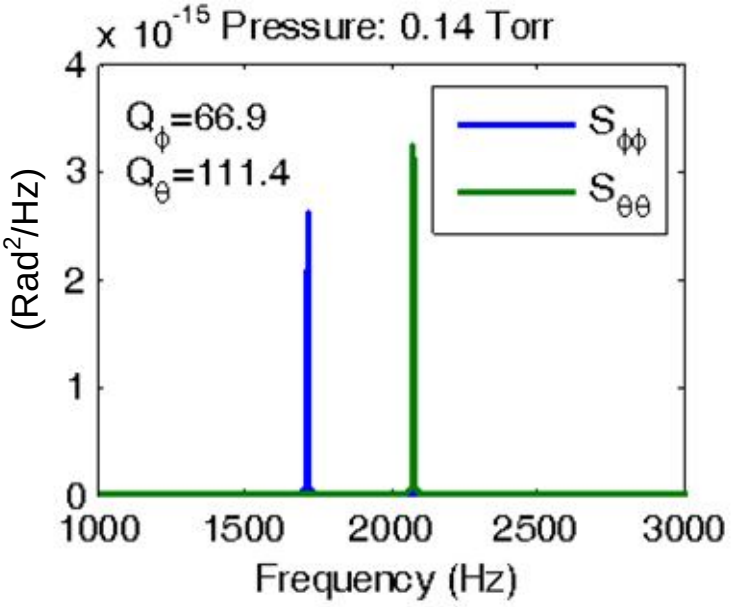
(a)


x 10^-16 Pressure: 10 Torr
Q_φ=2.0
Q_θ=2.8
S_φφ
S_θθ
(Rad^2/Hz)
Frequency (Hz)

(b)



(c)



(d)

Figure 6.5 (a)-(d) Illustrates the Sharpening "Q" as Pressures Decreases.

The “Q” can be related to damping through the relationship

$$Q_i = \frac{\omega_{0i}}{\Delta\omega_i} = \frac{\omega_{0i} I_i}{D_i}$$

where ω_{0i} is the resonant frequency, and $\Delta\omega_i$ is the full-width-half-max value around the resonant frequency. Figure 6.6 relates pressure to Q.

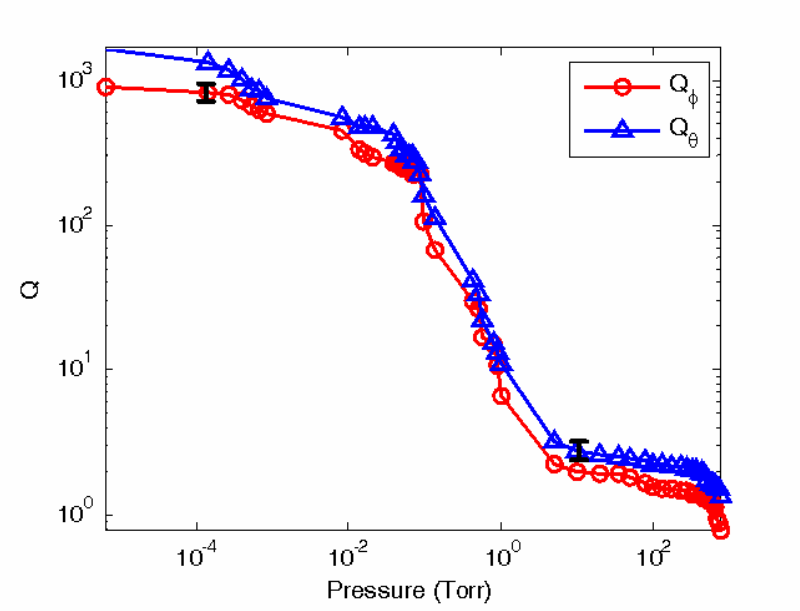


Figure 6.6 Mirror Q as a Function of Pressure

When estimating the effects of the Brownian motion noise, the mirror was not biased with any voltage and the optical table was floated so that the only source of jitter was the molecular bombardment of the air on the mirror.

Brownian Motion Induced Jitter: Invariant with Pressure

Figure 6.7 is a plot of the jitter variance as a function of pressure for the ϕ and θ angles for the mirror. The variance remained constant over nine orders of magnitude of pressure as required by the equipartition theorem. The mean value of the variance for the angles is found in Table 6.2

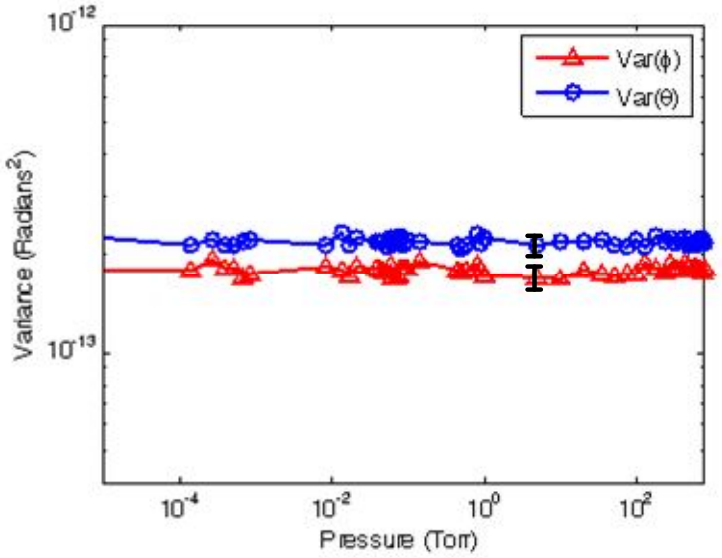


Figure 6.7 Illustration of the Variance in the Angular Jitter as a Function of Ambient Pressure Around the Mirror.

Table 6.2 Mean Value (RMS) of Jitter Variance

$E\{Var(\phi_{bm})\}$ (Radians^2)	1.79e-13 $\pm$ 5e-15
$E\{Var(\theta_{bm})\}$ (Radians^2)	2.16e-13 $\pm$ 4e-15

Validating the Micro-Mirror Jitter Model (MJM) (for Brownian-Motion Noise Dominated Environment)

The following was the methodology for evaluating the MJM model with time-series data. Note that this is in addition to the methodology from Chapter 4 for determining the constant coefficients for the pair of damped harmonic oscillators. Some of the additional steps are:

1. The mirror was held at a constant deterministic voltage, and time-series data were recorded for the mirror jitter as well as the control voltage noise. All signals (three from the quad cell, four from the voltage pads, and three from the accelerometers) were recorded with National Instruments CompactDAQ chassis and NI modules.
2. The component PSDs of the MJM model were summed, yielding the MJM model estimate for the mirror jitter.
3. The pressure was varied, and the process is repeated for different pressures and angles, illustrating mirror sensitivities under different conditions.

Micro-Mirror Jitter Model Vs. Measurements

The platform vibration curves all appear to be small. The PSD plot for the platform vibration is given in Figure 6.8. It is clear that with the optical table floating the vibration is very small, with the PSD amplitude approximately 1×10^{-30} 1/Hz at the mirror's resonance frequency.

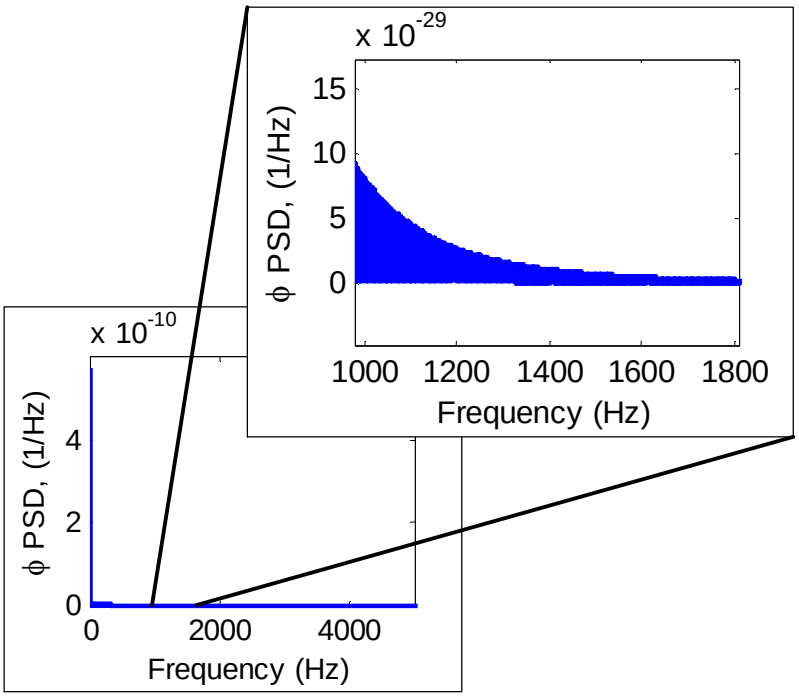


Figure 6.8 Platform Vibration Power Spectral Density

A sampling of measurements, which were taken for pressures of 760 Torr, 10 Torr, and 1 Torr respectively, is shown in Figure 6.9-6.11. The mirror also had different actuation voltages u and v applied to demonstrate the mirror's sensitivity to angle. The PSD of the angular jitter was measured. The platform vibration and control voltage noise were recorded as time-series data. Finally the MJM model parameters were determined from the recorded control voltage and platform vibration noise measurements and the estimated Brownian motion noise. Because the mirror and quad cell were biased with batteries and the massive optical table was floating, the voltage noise and platform vibration were small with respect to the Brownian motion noise as expected. Thus in this case the dominant source of noise was the Brownian

motion noise. All of the noise components were scaled as prescribed by the MJM model, and the sum of the 10 scaled component PSDs in equation (5.16) is equal to the "$S_{\phi\phi}$ Model" curve. The following plots show the components of equation (5.16) and (5.17) for the angle ϕ (and corresponding equations for θ) multiplied by the transfer function $\left|H\left(j\omega\right)\right|^2$ or:

$$S_{\phi\phi}\left(j\omega\right)=\left|H\left(j\omega\right)\right|^2 S_{w_1w_1}$$

or

$$S_{\phi\phi}\left(j\omega\right)=\left|H\left(j\omega\right)\right|^2\begin{Bmatrix}\alpha_1 S_{\tau_{1vib}\tau_{1vib}}+\alpha_2 S_{\tau_{1vib}\tau_{2vib}}+\alpha_3 S_{\tau_{2vib}\tau_{1vib}}+\alpha_4 S_{\tau_{2vib}\tau_{2vib}}\\+\alpha_5 S_{u_nu_n}+\alpha_6 S_{u_nv_n}+\alpha_7 S_{v_nu_n}+\alpha_8 S_{v_nv_n}\\+\alpha_9 S_{\tau_{1bm}\tau_{1bm}}+\alpha_{10} S_{\tau_{2bm}\tau_{2bm}}\end{Bmatrix}$$

This total angular jitter equation can be viewed as being composed of components from voltage noise, platform vibration, and Brownian motion noise or:

$$S_{\phi\phi}\left(j\omega\right)=S_{\phi\phi}\left(BM\right)+S_{\phi\phi}\left(Volt\right)+S_{\phi\phi}\left(Vib\right)$$

where

$$S_{\phi\phi}\left(BM\right)=\left|H\left(j\omega\right)\right|^2\left\{\alpha_9 S_{\tau_{1bm}\tau_{1bm}}+\alpha_{10} S_{\tau_{2bm}\tau_{2bm}}\right\},$$

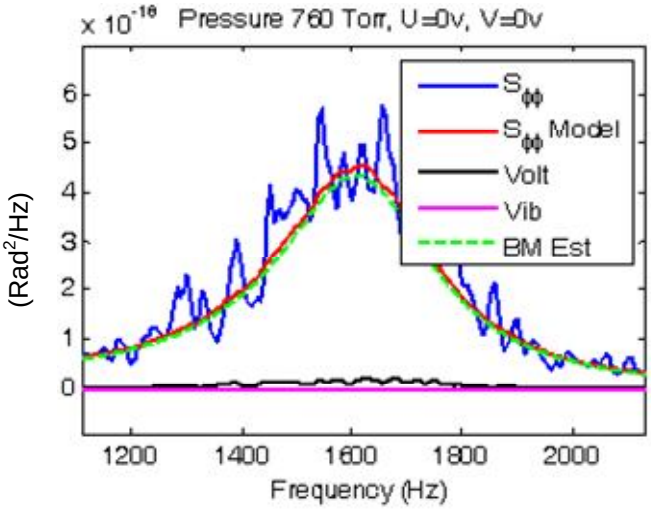
$$S_{\phi\phi}\left(Volt\right)=\left|H\left(j\omega\right)\right|^2\left\{\alpha_5 S_{u_nu_n}+\alpha_6 S_{u_nv_n}+\alpha_7 S_{v_nu_n}+\alpha_8 S_{v_nv_n}\right\},$$

and

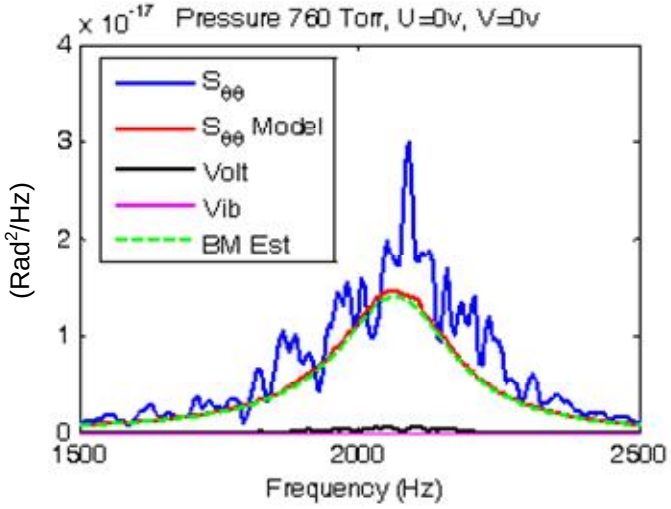
$$S_{\phi\phi}\left(Vib\right)=\left|H\left(j\omega\right)\right|^2\left\{\alpha_1 S_{\tau_{1vib}\tau_{1vib}}+\alpha_2 S_{\tau_{1vib}\tau_{2vib}}+\alpha_3 S_{\tau_{2vib}\tau_{1vib}}+\alpha_4 S_{\tau_{2vib}\tau_{2vib}}\right\}$$

Each plot displays five curves. The blue curve is the experimental jitter measurements. The red curve is the Micro-mirror Jitter Model. The next three

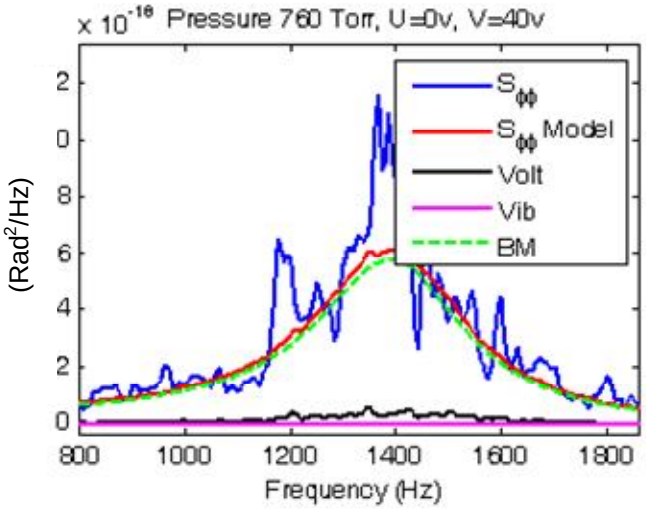
curves, black, magenta and green, are the components $S_{\phi\phi}(Volt)$, $S_{\phi\phi}(Vib)$, and $S_{\phi\phi}(BM)$ respectively, and sum to the MJM model.


x 10^-16 Pressure 760 Torr, U=0v, V=0v
(Rad²/Hz)
Frequency (Hz)
S φφ
S φφ Model
Volt
Vib
BM Est

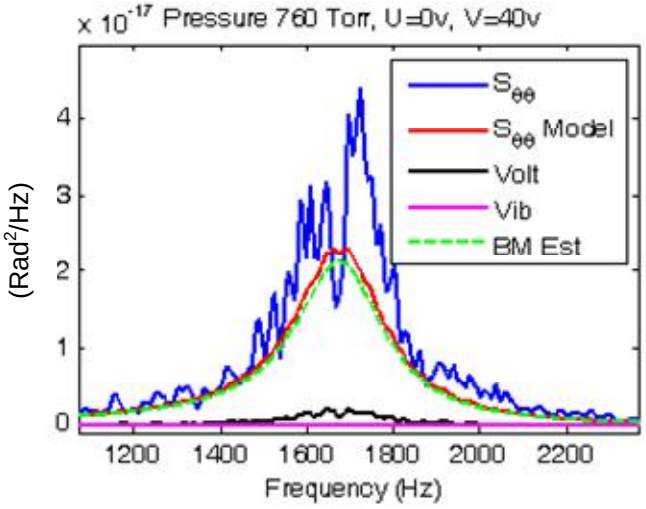
(a)


x 10^-17 Pressure 760 Torr, U=0v, V=0v
(Rad²/Hz)
Frequency (Hz)
S θθ
S θθ Model
Volt
Vib
BM Est

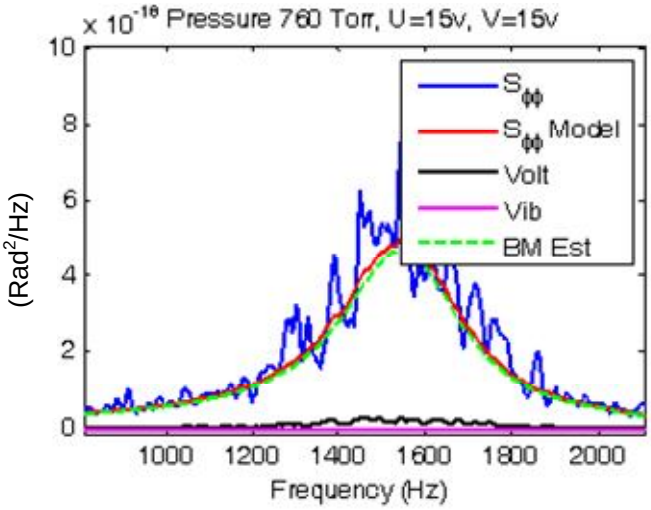
(b)


x 10^-16 Pressure 760 Torr, U=0v, V=40v
(Rad²/Hz)
Frequency (Hz)
S_φφ
S_φφ Model
Volt
Vib
BM

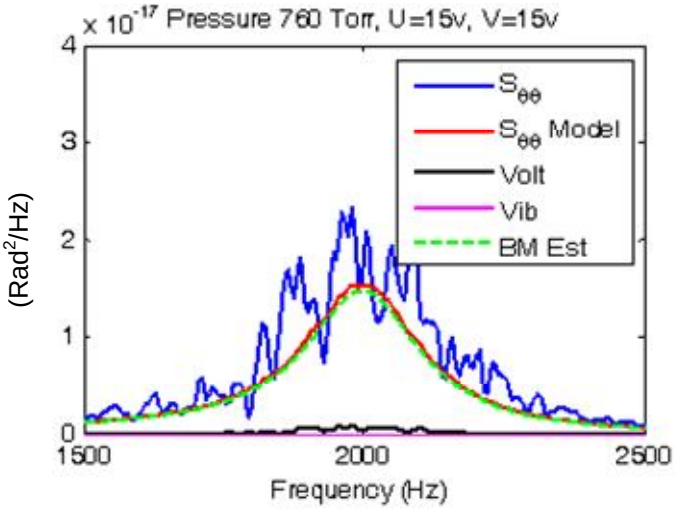
(c)


x 10^-17 Pressure 760 Torr, U=0v, V=40v
(Rad²/Hz)
Frequency (Hz)
S_θθ
S_θθ Model
Volt
Vib
BM Est

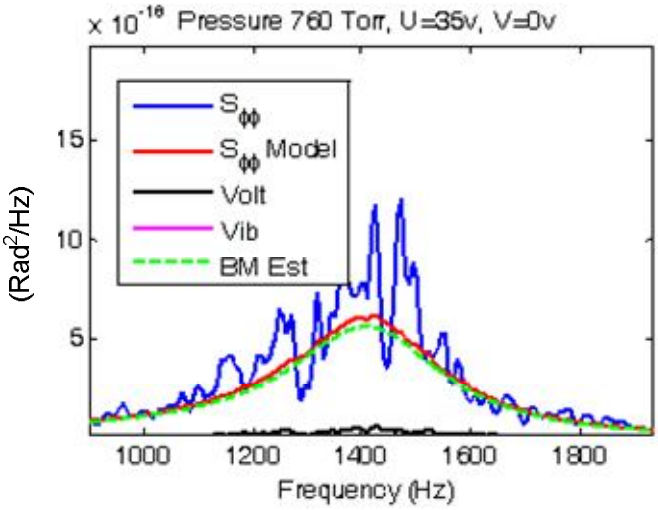
(d)


x 10^-16 Pressure 760 Torr, U=15v, V=15v
(Rad²/Hz)
Frequency (Hz)
Sϕϕ
Sϕϕ Model
Volt
Vib
BM Est

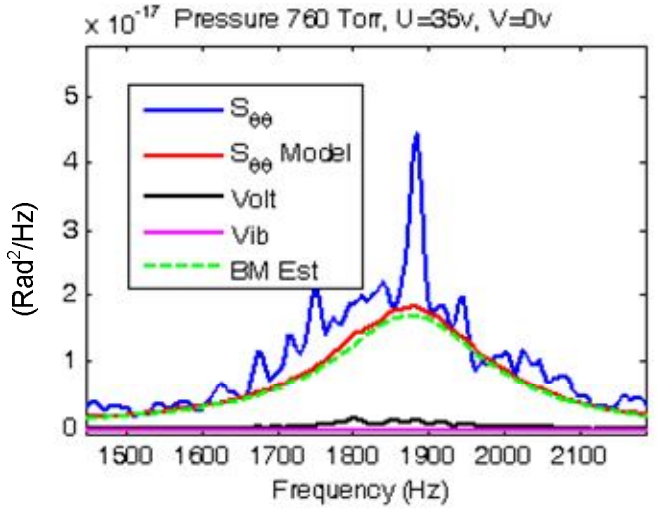
(e)


x 10^-17 Pressure 760 Torr, U=15v, V=15v
(Rad²/Hz)
Frequency (Hz)
Sθθ
Sθθ Model
Volt
Vib
BM Est

(f)

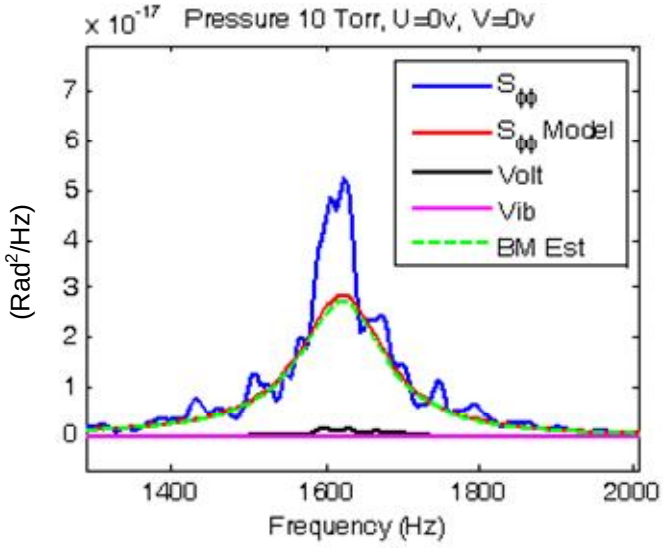


(g)

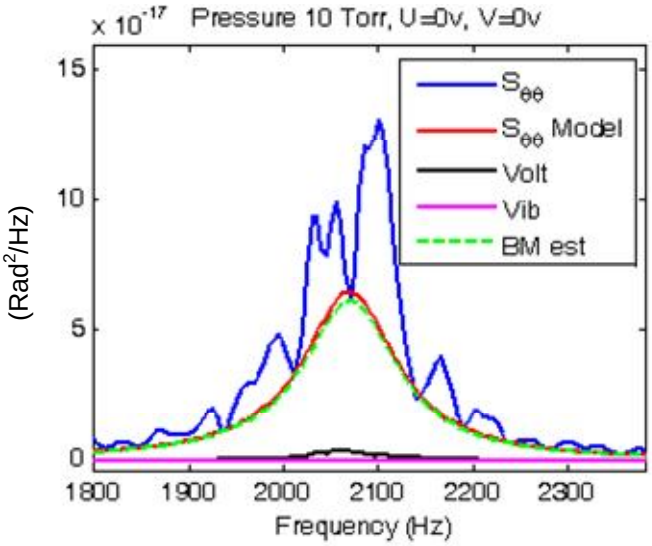


(h)

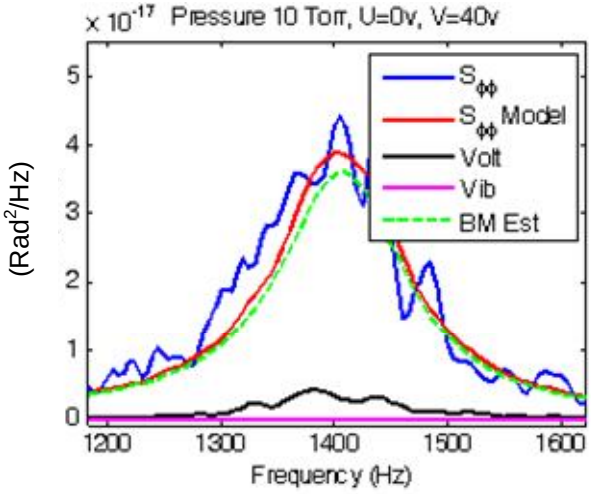
Figure 6.9 (a)-(h) Model Vs. Measurements at 760 Torr (1 Atmosphere)


x 10^-17 Pressure 10 Torr, U=0v, V=0v
(Rad²/Hz)
Frequency (Hz)
S_φφ
S_φφ Model
Volt
Vib
BM Est

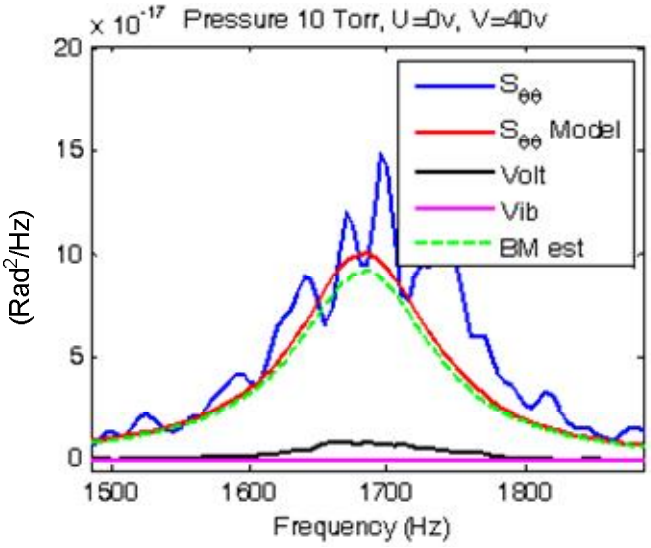
(a)


x 10^-17 Pressure 10 Torr, U=0v, V=0v
(Rad²/Hz)
Frequency (Hz)
S_θθ
S_θθ Model
Volt
Vib
BM est

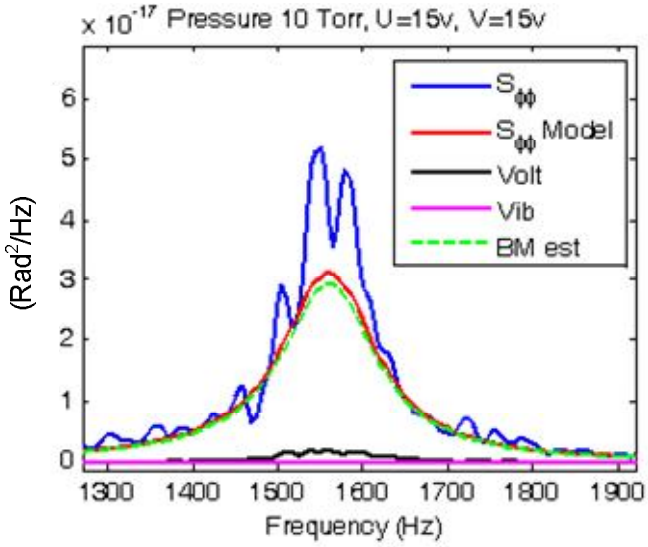
(b)


x 10^-17 Pressure 10 Torr, U=0v, V=40v
S_φφ
S_φφ Model
Volt
Vib
BM Est
(Rad²/Hz)
Frequency (Hz)

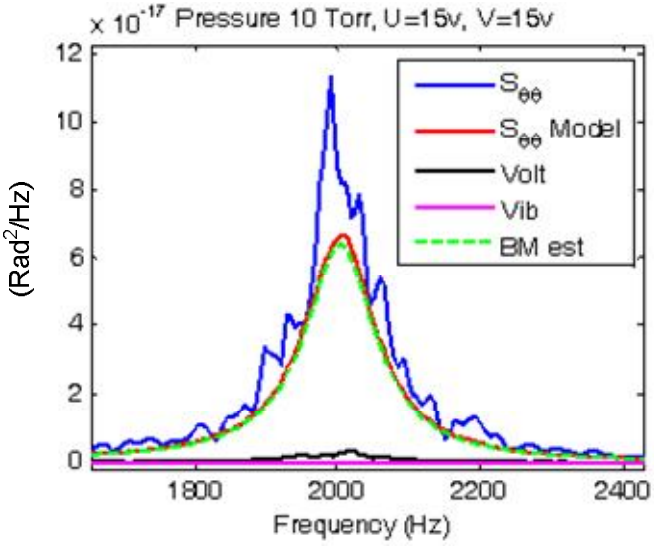
(c)


x 10^-17 Pressure 10 Torr, U=0v, V=40v
S_θθ
S_θθ Model
Volt
Vib
BM est
(Rad²/Hz)
Frequency (Hz)

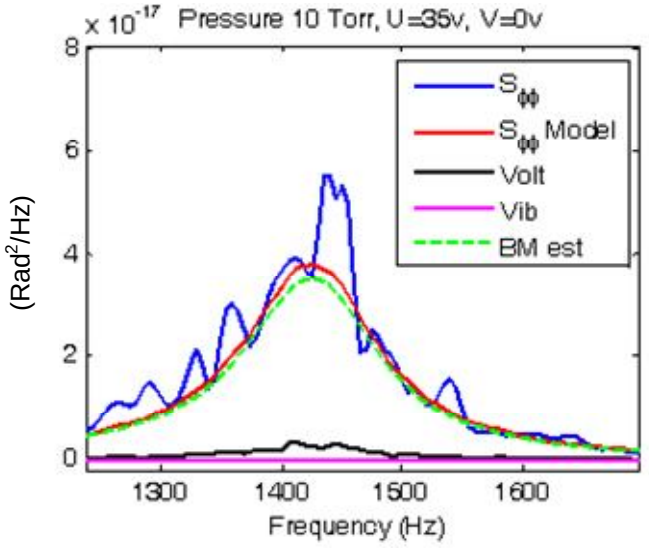
(d)


x 10^-17 Pressure 10 Torr, U=15v, V=15v
S_φφ
S_φφ Model
Volt
Vib
BM est
(Rad^2/Hz)
Frequency (Hz)

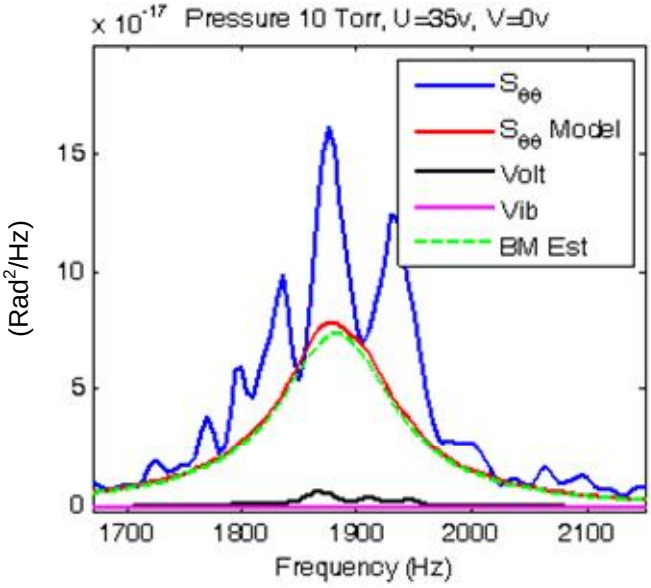
(e)


x 10^-17 Pressure 10 Torr, U=15v, V=15v
S_θθ
S_θθ Model
Volt
Vib
BM est
(Rad^2/Hz)
Frequency (Hz)

(f)

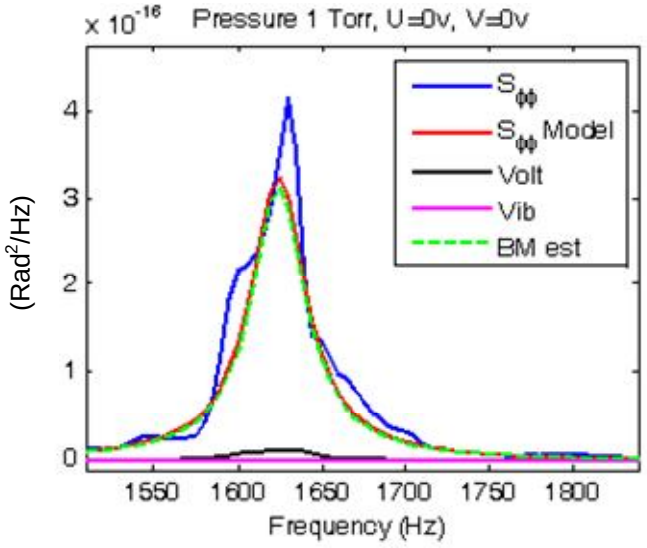


(g)

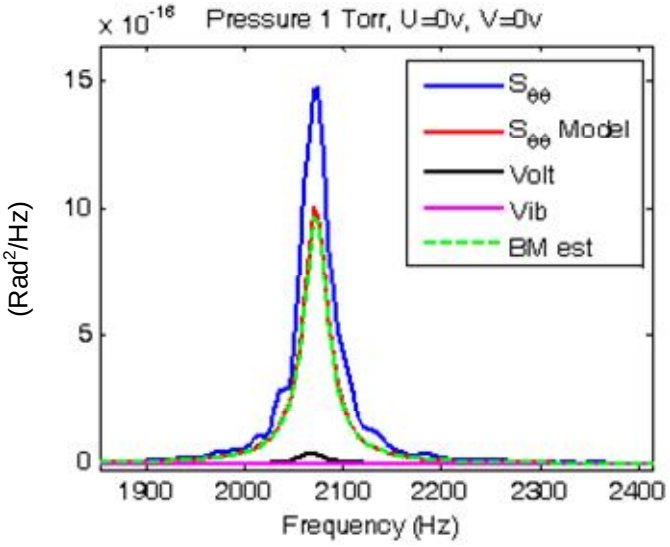


(h)

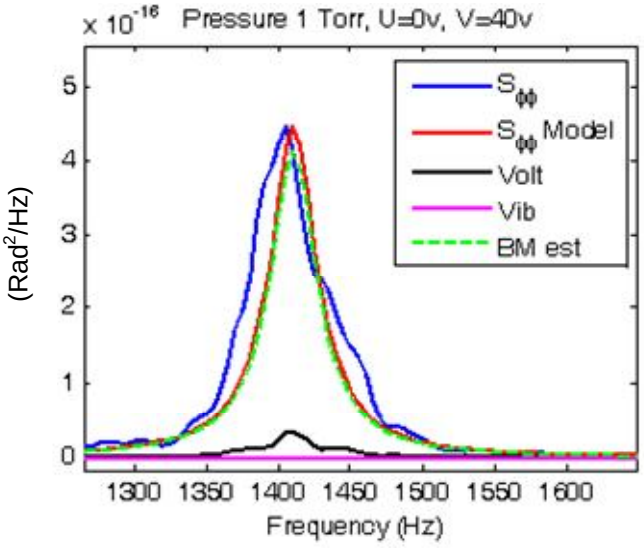
Figure 6.10 (a)-(h) Model Vs. Measurements at 10 Torr


x 10^-16
Pressure 1 Torr, U=0v, V=0v
S_φφ
S_φφ Model
Volt
Vib
BM est
(Rad²/Hz)
Frequency (Hz)

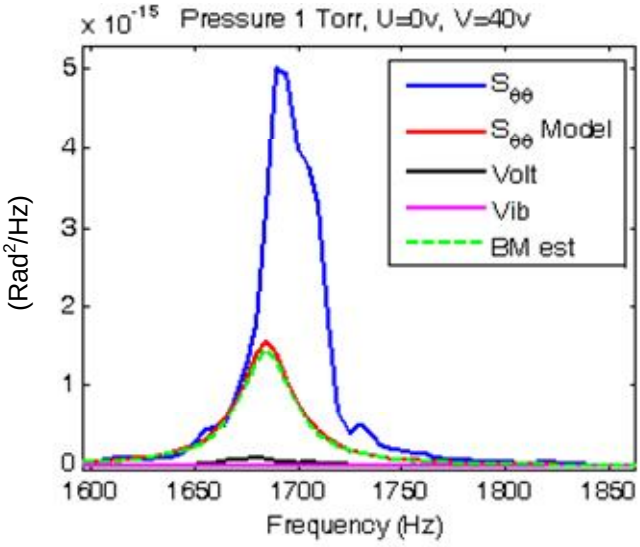
(a)


x 10^-16
Pressure 1 Torr, U=0v, V=0v
S_θθ
S_θθ Model
Volt
Vib
BM est
(Rad²/Hz)
Frequency (Hz)

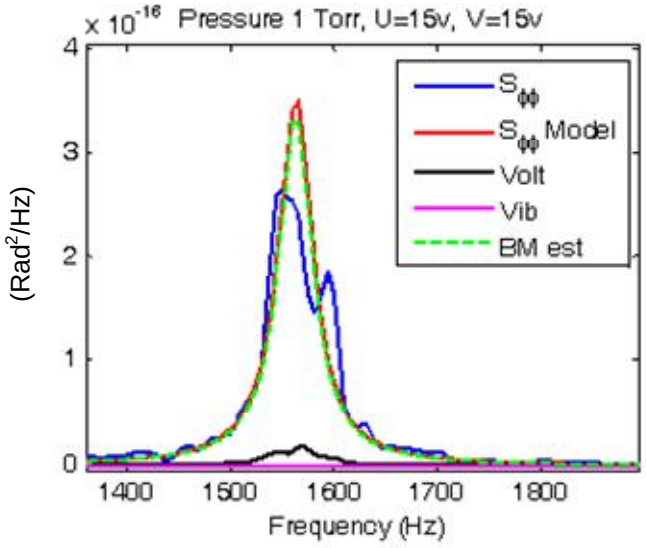
(b)


x 10^-16 Pressure 1 Torr, U=0v, V=40v
S_φφ
S_φφ Model
Volt
Vib
BM est
(Rad²/Hz)
Frequency (Hz)
0
1
2
3
4
5
1300 1350 1400 1450 1500 1550 1600

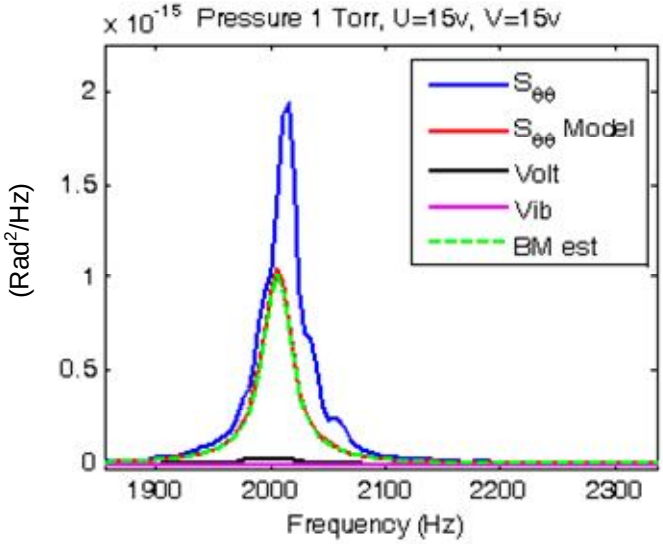
(c)


x 10^-15 Pressure 1 Torr, U=0v, V=40v
S_θθ
S_θθ Model
Volt
Vib
BM est
(Rad²/Hz)
Frequency (Hz)
0
1
2
3
4
5
1600 1650 1700 1750 1800 1850

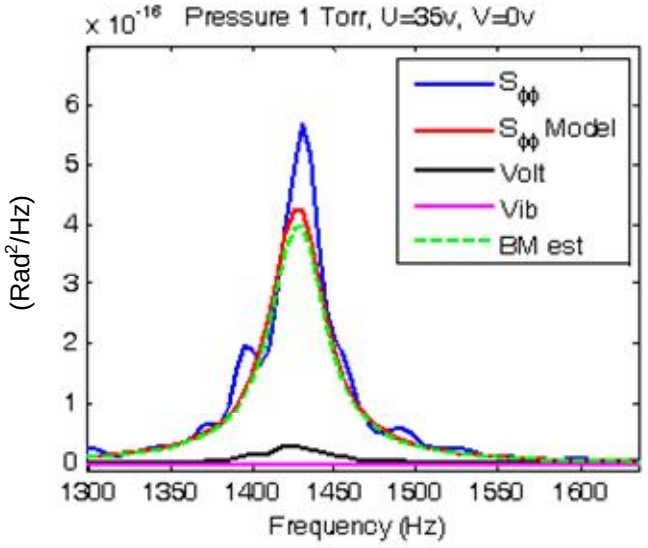
(d)


x 10^-16 Pressure 1 Torr, U=15v, V=15v
(Rad²/Hz)
Frequency (Hz)
S_φφ
S_φφ Model
Volt
Vib
BM est

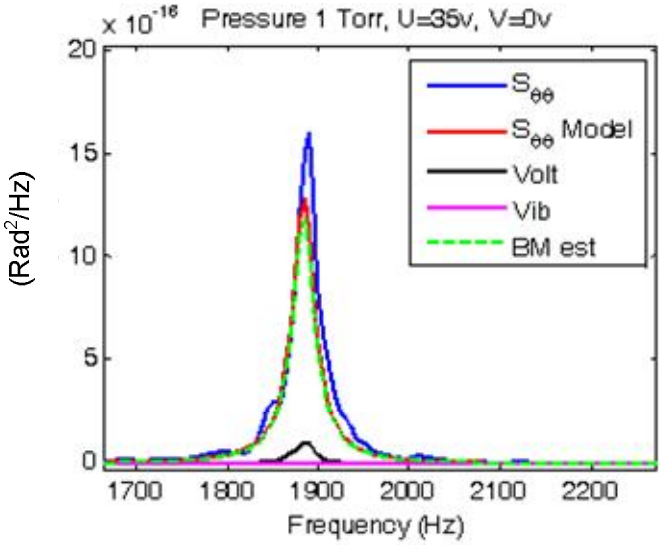
(e)


x 10^-15 Pressure 1 Torr, U=15v, V=15v
(Rad²/Hz)
Frequency (Hz)
S_θθ
S_θθ Model
Volt
Vib
BM est

(f)



(g)



(h)

Figure 6.11 (a)-(h) Model Vs. Measurements at 1 Torr

Conclusions

As expected, the floating optical table isolated the mirror from external vibration as shown through the angular accelerometer measurement curve. The batteries actuating the mirror minimized the control voltage noise, as well as the noise from the quad cell detector.

The MJM model remains accurate when varied in position and pressure, yet performs slightly better at higher pressures. The model was in agreement for all the pressures and angles tested with the exception of Figure 6.11 (d) (u=0, v=40 at 1 Torr).

It was planned to re-take this set of measurements to determine if this case, u=0, v=40 at 1 Torr, would agree with the MJM model and that some anomaly caused the disagreement. Unfortunately, during the process of repeating earlier measurements from Chapter 4, a voltage spike or electrostatic discharge occurred while the mirror was a 0.5 Torr and damaged the mirror. Further measurements were not possible after this point.

Chapter 7: Research Conclusions

Prior to this dissertation, the standard model for the MEMS Two-Axis (Tip-Tilt) Mirror (2D Mirror) was based on the model published by Dr. Ming C. Wu in 2001 [57]. It required numerical integration and the computational burden made it *unwieldy* for use in studying dynamic motion of MEMS Two-Axis (Tip-Tilt) Mirrors. Consequently there are no known publications reporting on dynamic modeling with or without measurement comparisons for the 2D MEMS Mirror. The Wu Model did account for translational movement and warping of the mirror, two characteristics not modeled by the current work. However, these characteristics are less important since they are currently negligible because of Bulk MEMS Micro-fabrication-processing has produced sufficiently thick mirror facets.

This dissertation presents a first-principles analytic, closed-form 2D Torque Model, Micro-mirror Pointing Model (MPM), and Micro-mirror Jitter Model (MJM) for the MEMS Two-Axis (Tip-Tilt) Mirror that are supported by experimental measurements. The 2D Torque Model is analytic and includes all of the fundamental electrodynamics in modeling the MEMS Two-Axis Tip-Tilt Mirror.

The relationship between these models is shown in Figure 7.1. These three models provide explicit analytical relationships between the MEMS mirror physical,

electrical, and environmental design parameters, and mirror transient and steady-state pointing performance. The 2D Torque Model is utilized to develop the Micro-mirror Pointing Model (MPM) and the Micro-mirror Jitter Model (MJM). The MPM model is used to analyze the dynamics of the mirror with the 2D Torque Model as the inputs to a pair of damped simple harmonic oscillators that are coupled through the 2D Torque Model. This formulation is imposed by Euler dynamic equations and the rigid mirror structure. The MJM model provides an explicit relationship between noise sources and the resulting mirror jitter.

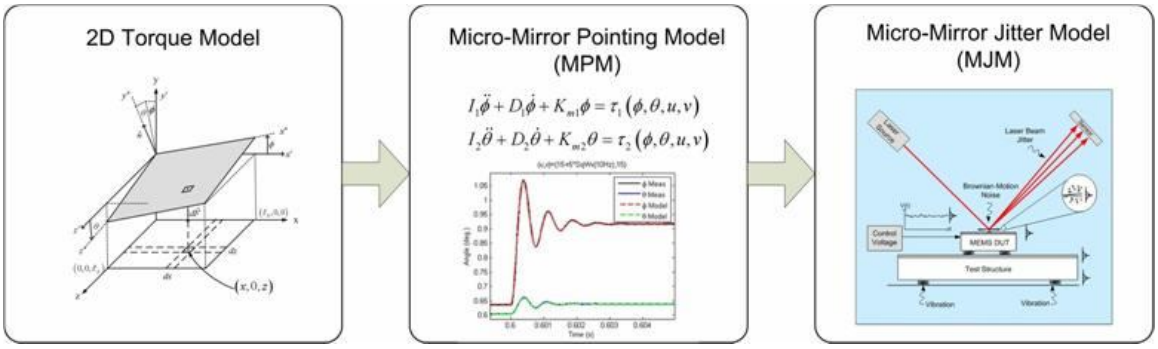


Figure 7.1 Technical Contribution of Dissertation

Overview of Three Models

The first-principles analytic 2D Torque Model provides a physical connection between the physical size of the mirror and the torques, as well as higher-order effects relating to the dynamics of the micro-mirror. A generalized function, "G", has also been presented that utilizes the symmetry in the torsional expressions of the two axes of rotation and is suitable for inclusion into software simulations. Removable

singularities in the G function have been identified and evaluated analytically to facilitate simulation.

Micro-Mirror Pointing Model (MPM)

It has been shown that a pair of damped simple harmonic oscillator equations is imposed by the Euler dynamic equations describing rotational motion of the 2D mirror structure. The axially-coupled 2D Torque Model is a critical component in the dynamic equations. Pull-in and stability were also examined in terms of the poles of the MPM model.

A methodology has been also presented determining the dynamic constants for the mirror as well as an "effective length" which accounts for the fringing electric field effects. The model calibration procedure and estimation of physical parameters such as torsional restoring constants and effective lengths using limited steady-state measurements. The calibration represents a strength of the analytical model: While MEMS fabrication does lead to an inherent variation in performance, the MPM model can be calibrated for each mirror to account for this variation. In fitting to these physical parameters, knowledge is gained about fabrication processes.

An experimental apparatus has been described that validated the MPM model for steady-state and transient measurements relating angles to actuating voltages as well

as measurements of the mirror being operated in a dynamic switching manner. The simulations and measurements of the MPM model confirm that the model provides an excellent approximation for the two-axis (tip-tilt) micro-mirror, and the coupled damped harmonic oscillator model accurately describes the dynamics.

Micro-Mirror Jitter Model (MJM))

Illustrating the value of analytic models, the 2D Torque Model and MPM model were used in developing the Micro-mirror Jitter Model (MJM) to address the effects of mirror facet jitter. The MJM model can be used to simulate, predict and model the effects of mirror jitter as a function of the originating noise sources including: control voltage fluctuations, platform vibration, and Brownian motion (or mechanical-thermal) noise.

A methodology has also been developed for validating the MJM model. Measurements from the experimental set-up support the model. Additionally, a damping versus pressure curve for both axes is provided. The mirror jitter was recorded and analyzed for varying pressures and tip-tilt angles. The variance of the Brownian motion generated jitter is observed to be pressure invariant, as the theoretical development and equipartition theorem suggest. The experimental measurements validate the MJM model for the Brownian motion noise dominated

environment. The case is important because it represents the best case for mirror jitter for a MEMS micro-mirror system.

The models produced by this work will enable designers to optimize mirror operations for applications such as optical cross-connect switching, advanced LIDAR scanning, or free-space optical communication where fast and precise pointing are critical to overall system performance.

Significant Accomplishments of this Research

The significant contributions and accomplishments of this research include:

- 2D Torque Model and MPM model
    - Analytic and Closed-form
    - Developed from physical, electrical and environmental design parameters
    - The 2D Pull-in is expressed in terms of poles in the S-plane.
    - Mirror dynamics based on Euler equation formulation
    - The MPM model agree with experimental results
- MJM model
    - Developed using first-order multi-dimensional Taylor expansion.
    - Estimates of jitter sources based on fundamental physical relationships

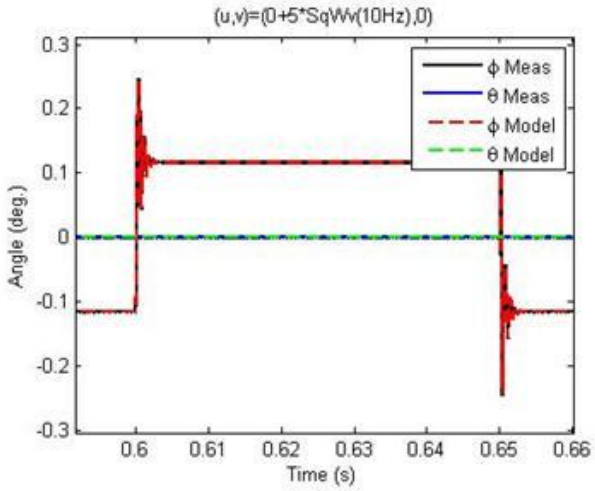
- Jitter related by transfer functions and source power spectral densities (PSD)
- Damping coefficients or mirror “Q” is measured.
- Jitter variance for both axes of rotation invariant with pressure.
- Model and measurements agree for varying 2D tip-tilt mirror angles
- Experiment quantified jitter sources including Brownian motion noise.

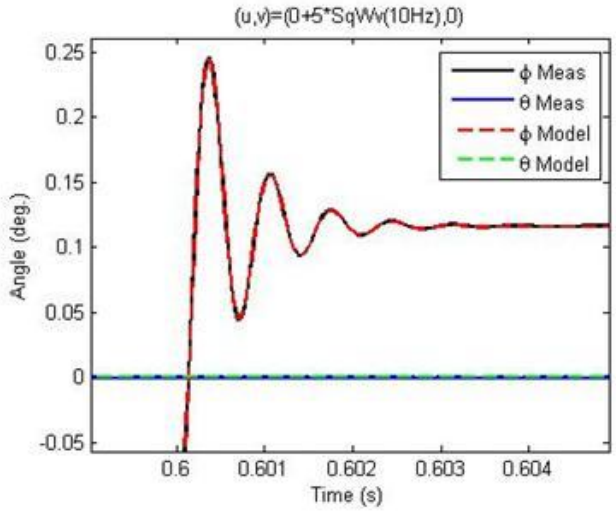
- Systems Applications
  - Simulations of 2D Torque, MPM, and MJM models enable design engineers to evaluate system tolerances in terms of expected mirror dynamics and jitter.
  - System engineers can determine components and device noise requirements to ensure that the entire system stays within specification.
  - Given a set of components in a system, system engineers can predict jitter of mirror facet.

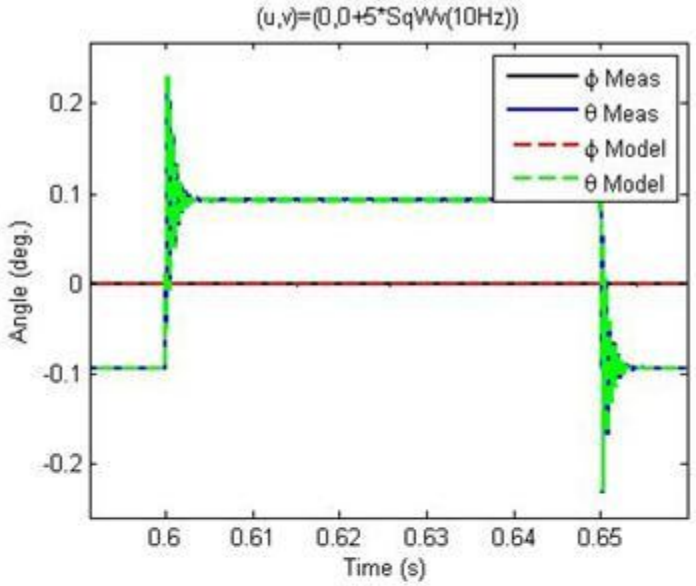
Appendix A

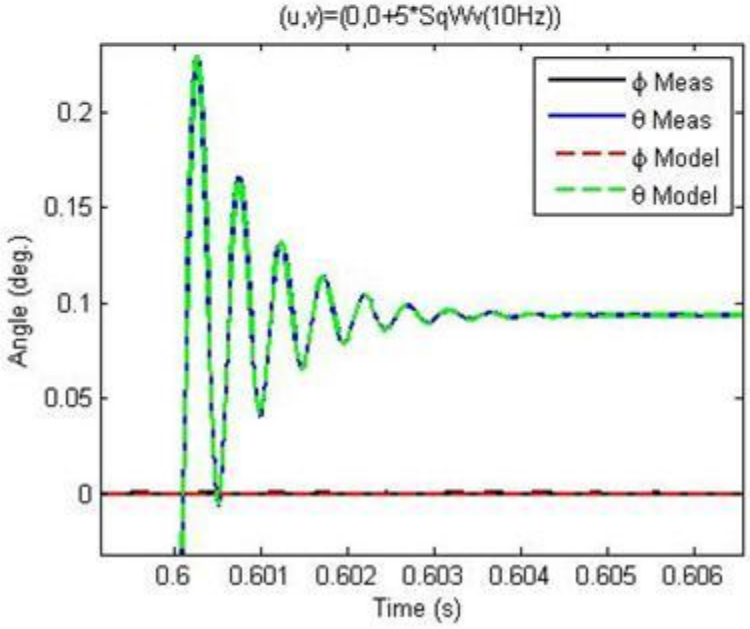
Complete Set of MPM Model Vs. Measurement Plots

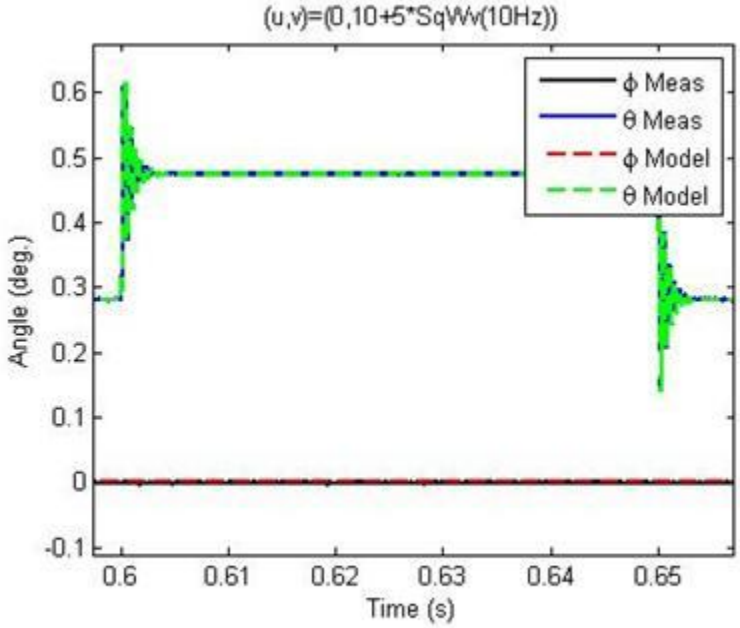
Because of the large number of plots only are few are included in Chapter 3

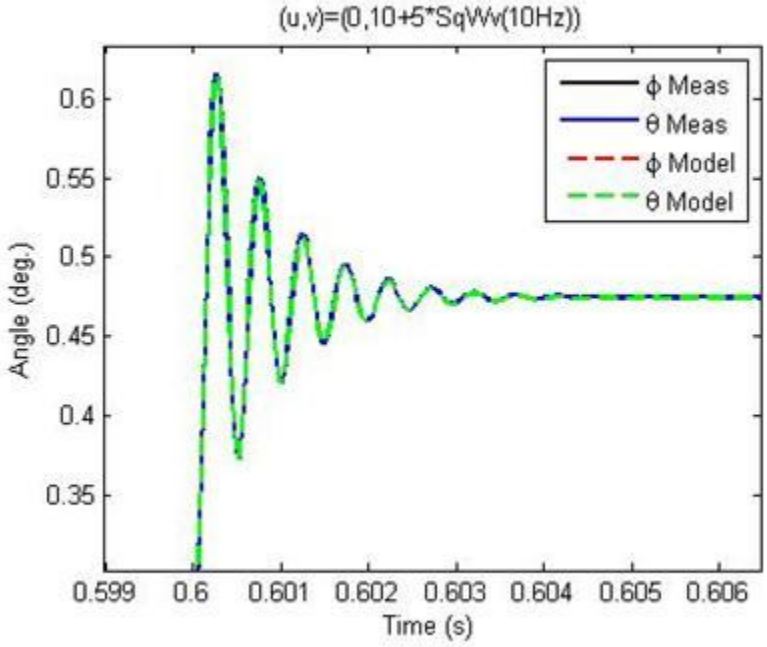


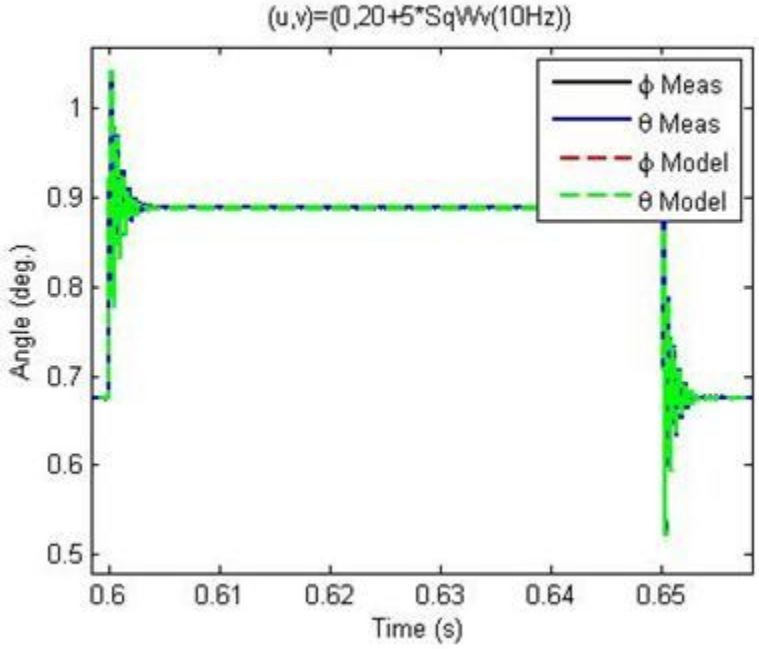


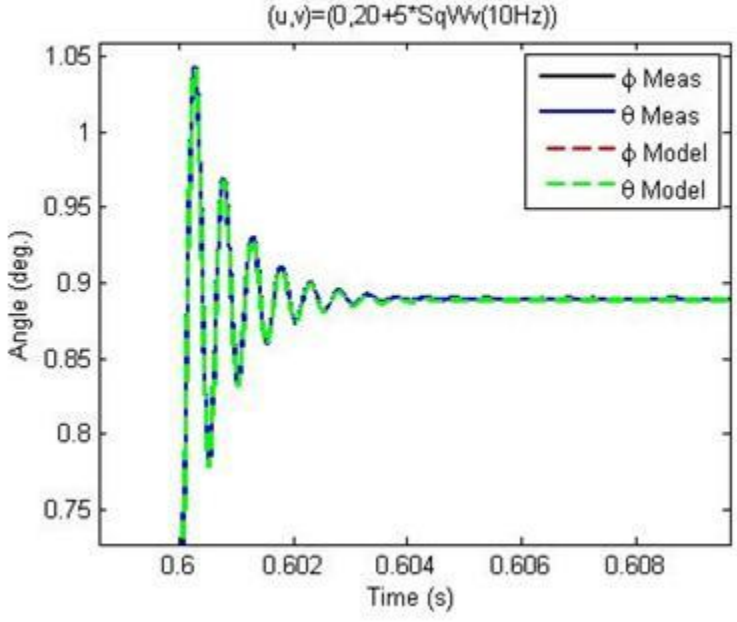

(u,v)=(0,0+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
0.2
0.1
0
-0.1
-0.2
0.6
0.61
0.62
0.63
0.64
0.65
Time (s)

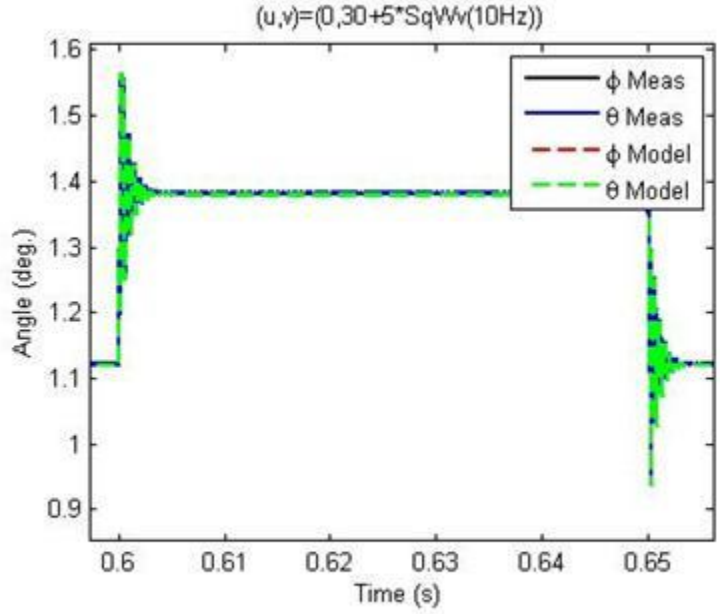

(u,v)=(0,0+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
0.2
0.15
0.1
0.05
0
0.6
0.601
0.602
0.603
0.604
0.605
0.606
Time (s)

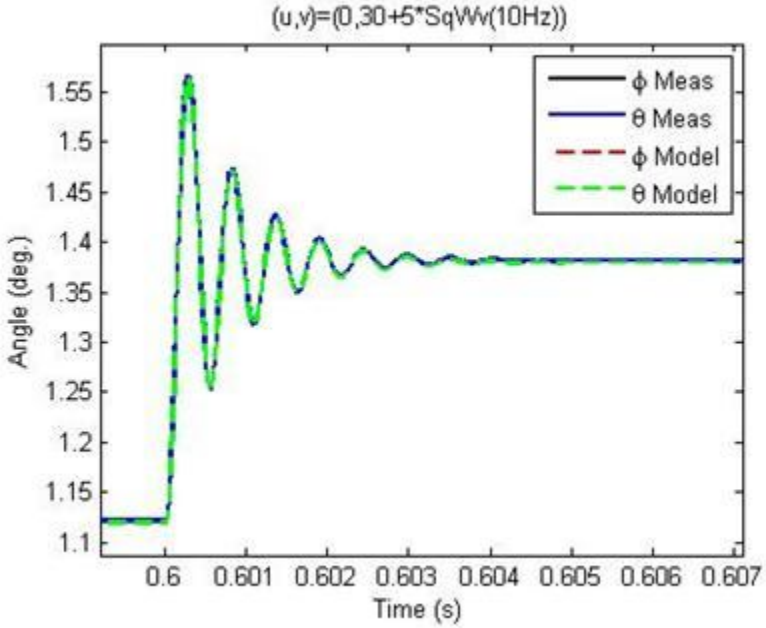

(u,v)=(0,10+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0.6
0.61
0.62
0.63
0.64
0.65

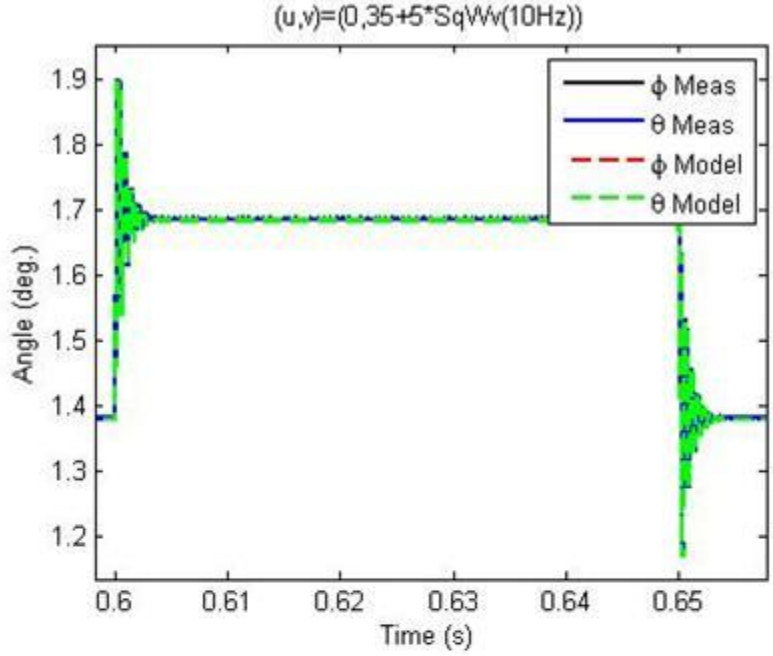

(u,v)=(0,10+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.6
0.55
0.5
0.45
0.4
0.35
0.599
0.6
0.601
0.602
0.603
0.604
0.605
0.606

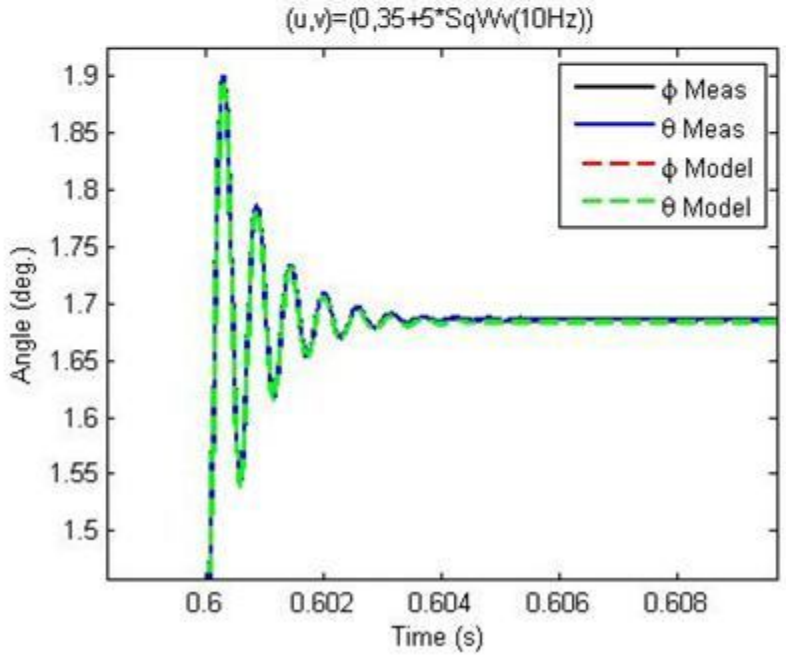

(u,v)=(0,20+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1
0.9
0.8
0.7
0.6
0.5
0.6
0.61
0.62
0.63
0.64
0.65

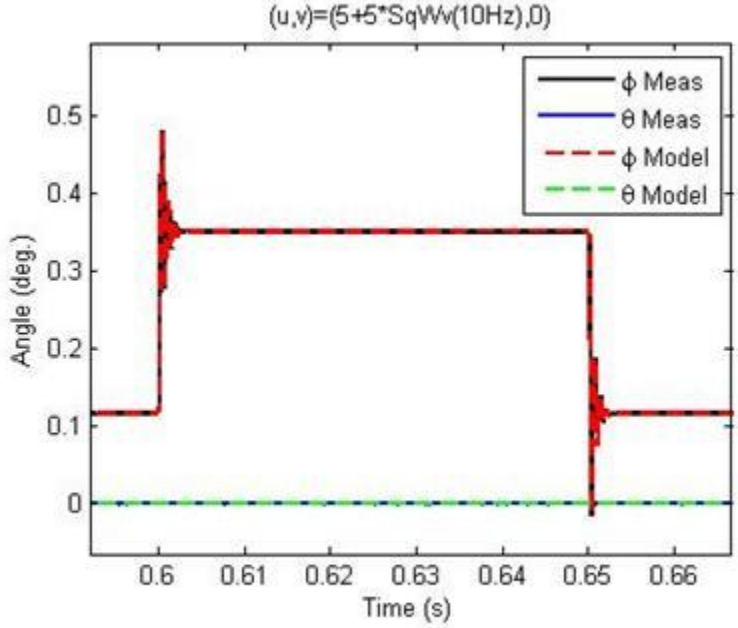

(u,v)=(0,20+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.05
1
0.95
0.9
0.85
0.8
0.75
0.6
0.602
0.604
0.606
0.608

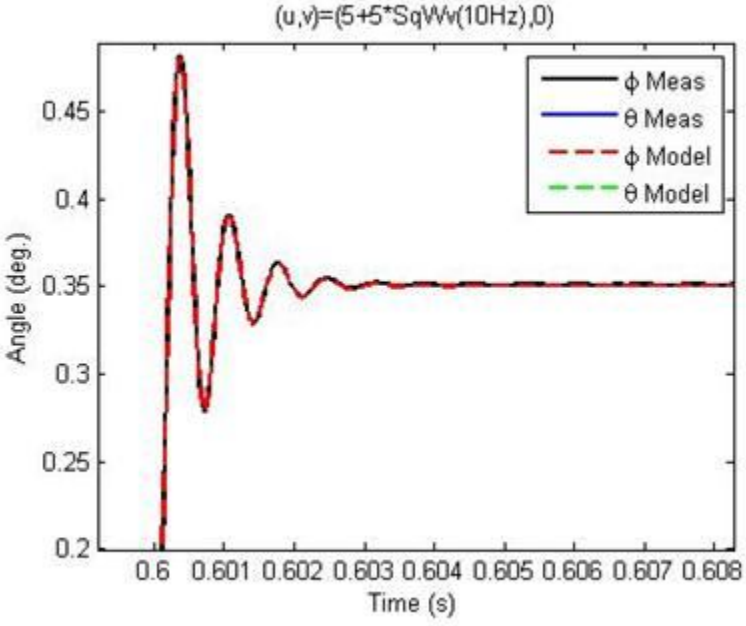

(u,v)=(0,30+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.6
0.61
0.62
0.63
0.64
0.65

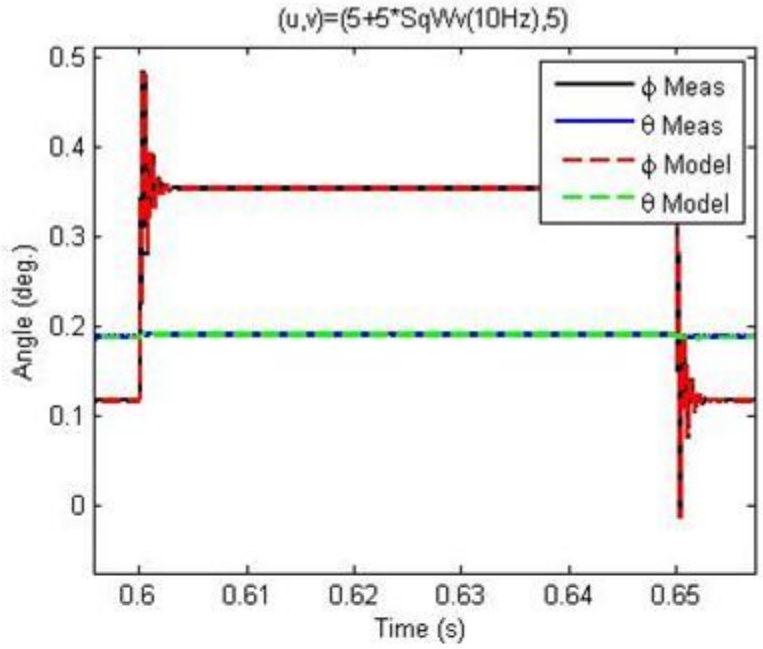

(u,v)=(0,30+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.55
1.5
1.45
1.4
1.35
1.3
1.25
1.2
1.15
1.1
0.6
0.601
0.602
0.603
0.604
0.605
0.606
0.607

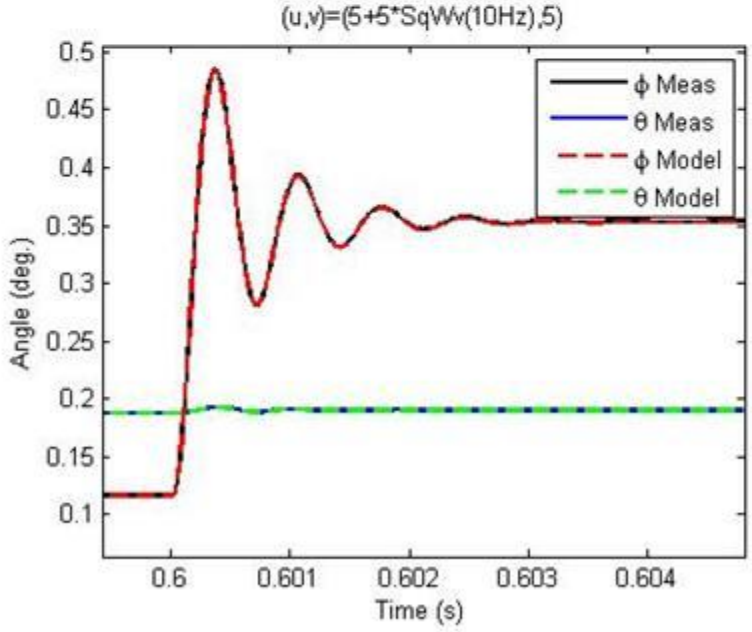

(u,v)=(0,35+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
0.6
0.61
0.62
0.63
0.64
0.65

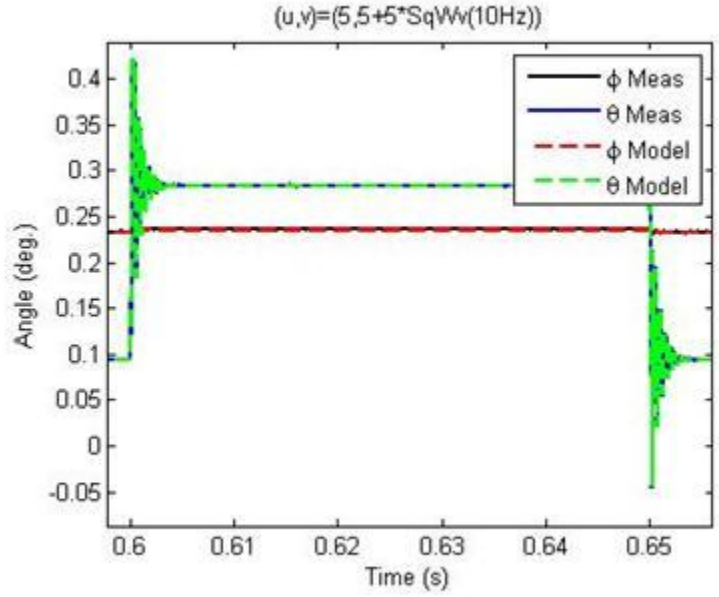

(u,v)=(0,35+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.9
1.85
1.8
1.75
1.7
1.65
1.6
1.55
1.5
0.6
0.602
0.604
0.606
0.608

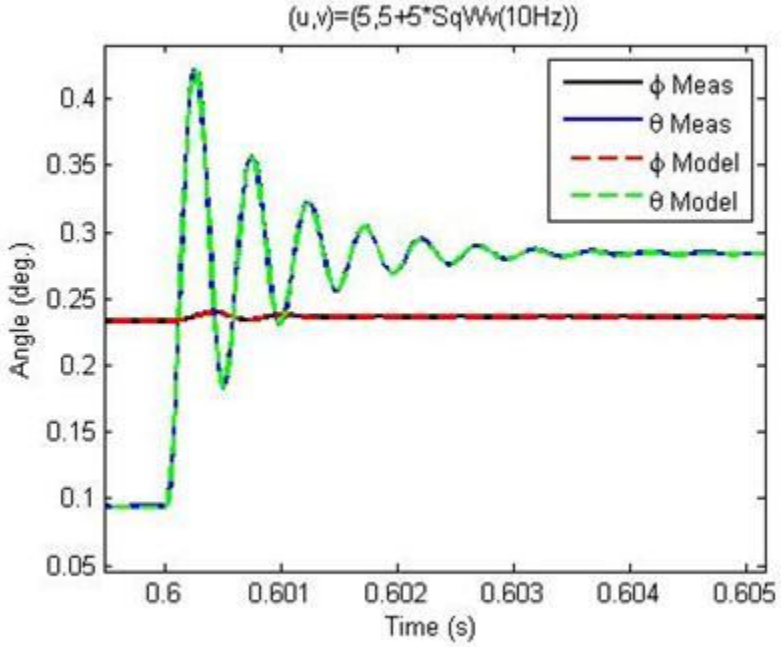

(u,v)=(5+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
0.5
0.4
0.3
0.2
0.1
0
0.6
0.61
0.62
0.63
0.64
0.65
0.66
Time (s)

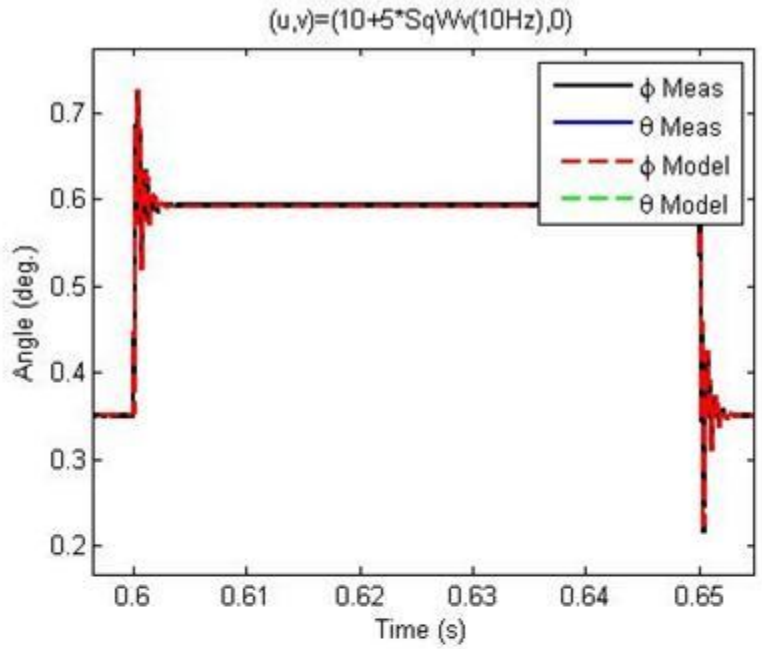

(u,v)=(5+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
0.45
0.4
0.35
0.3
0.25
0.2
0.6
0.601
0.602
0.603
0.604
0.605
0.606
0.607
0.608
Time (s)

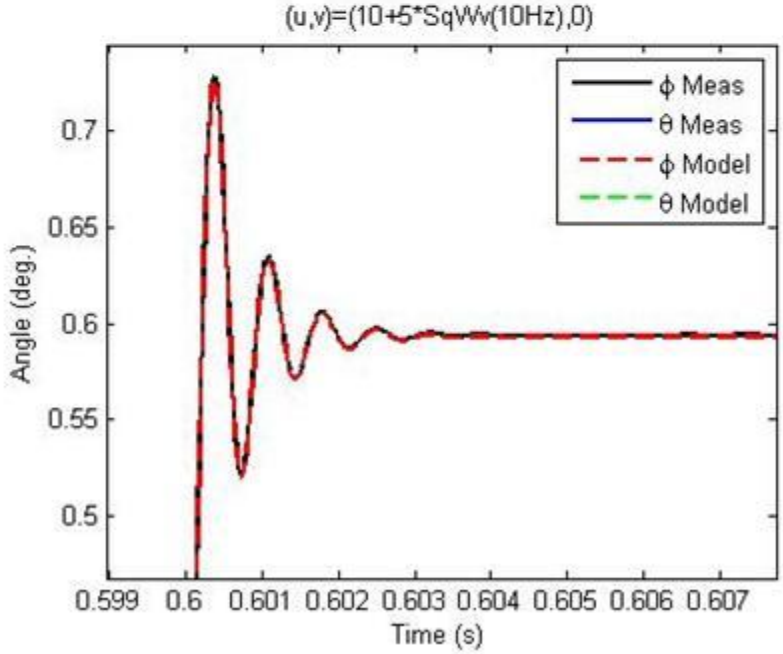

(u,v)=(5+5*SqWv(10Hz),5)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.5
0.4
0.3
0.2
0.1
0
0.6
0.61
0.62
0.63
0.64
0.65

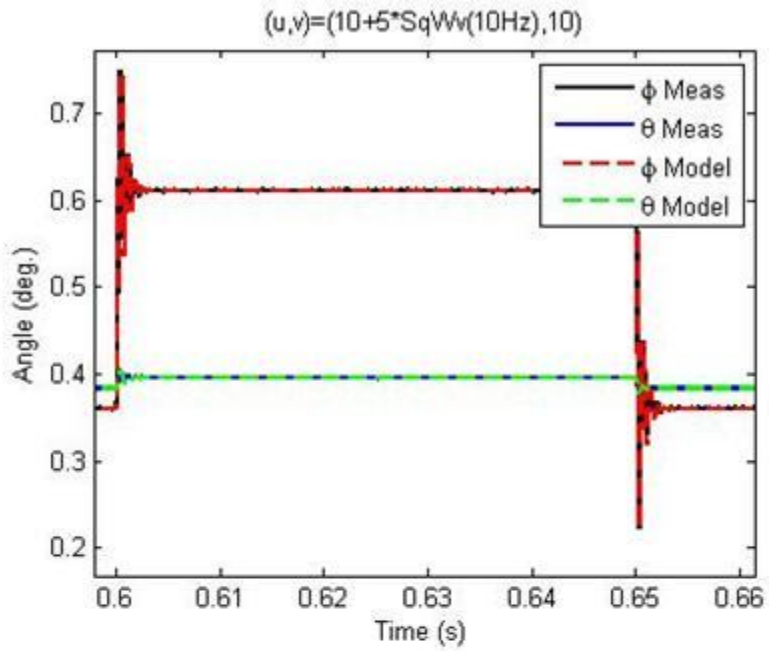

(u,v)=(5+5*SqWv(10Hz),5)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.6
0.601
0.602
0.603
0.604

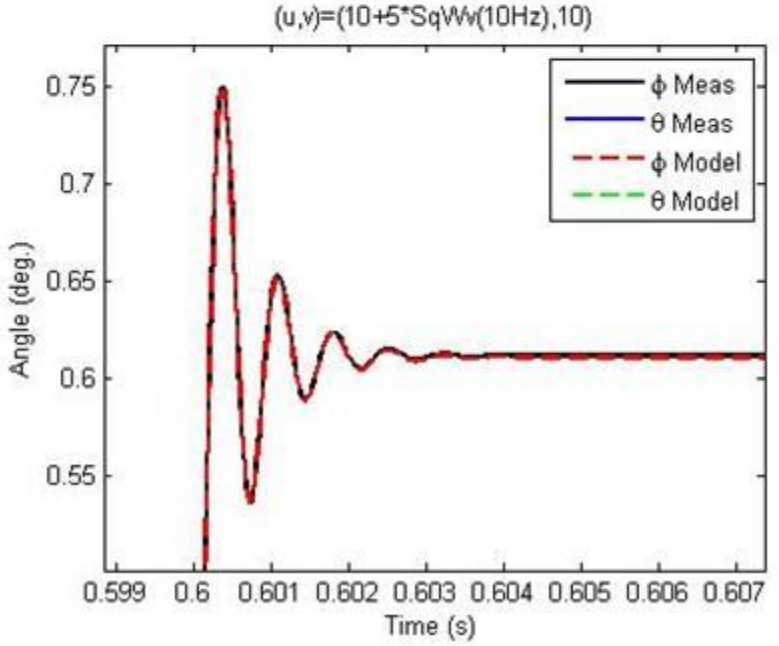

(u,v)=(5,5+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
0.6
0.61
0.62
0.63
0.64
0.65

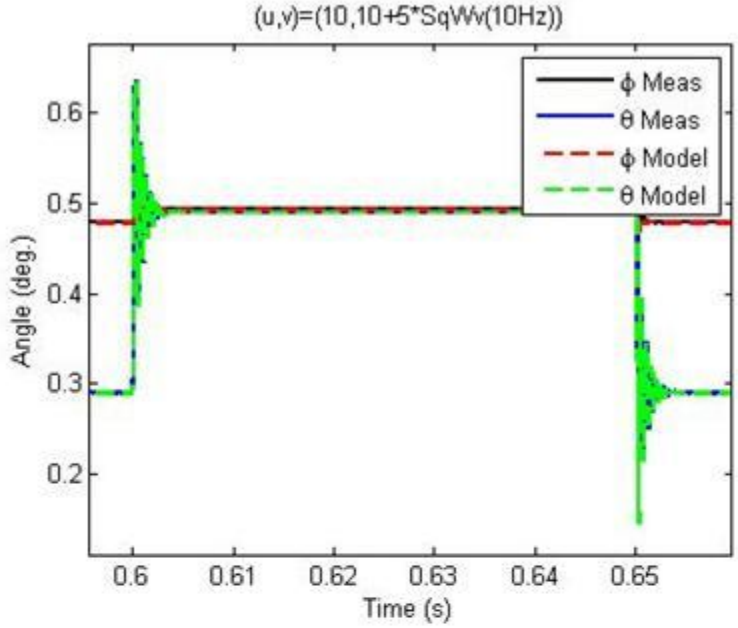

(u,v)=(5,5+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0.6
0.601
0.602
0.603
0.604
0.605

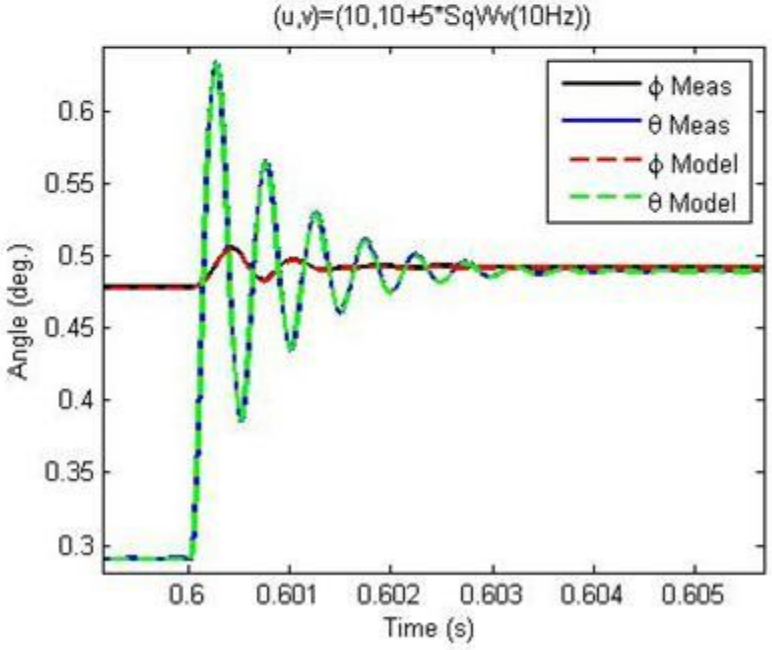

(u,v)=(10+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
0.7
0.6
0.5
0.4
0.3
0.2
0.6
0.61
0.62
0.63
0.64
0.65
Time (s)

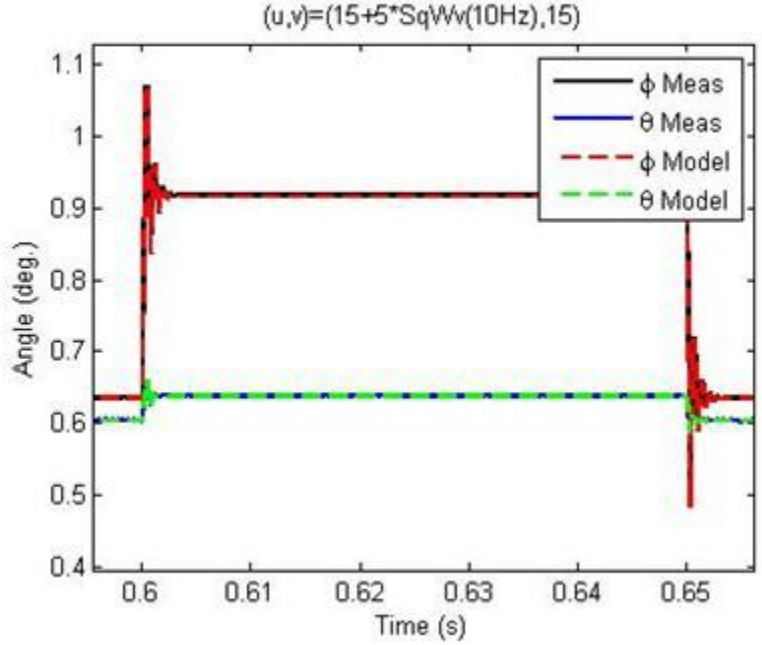

(u,v)=(10+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
0.7
0.65
0.6
0.55
0.5
0.599
0.6
0.601
0.602
0.603
0.604
0.605
0.606
0.607
Time (s)

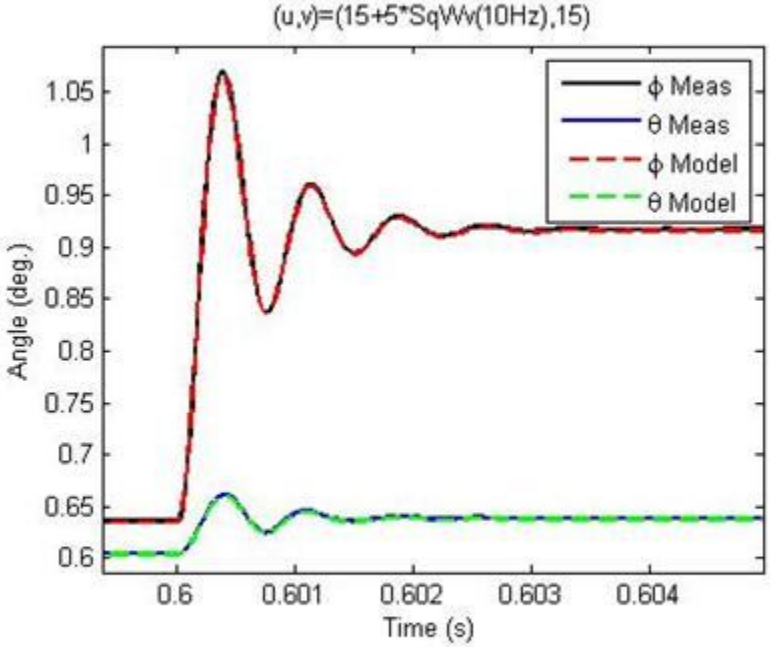

(u,v)=(10+5*SqWv(10Hz),10)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.2
0.3
0.4
0.5
0.6
0.7
0.6
0.61
0.62
0.63
0.64
0.65
0.66

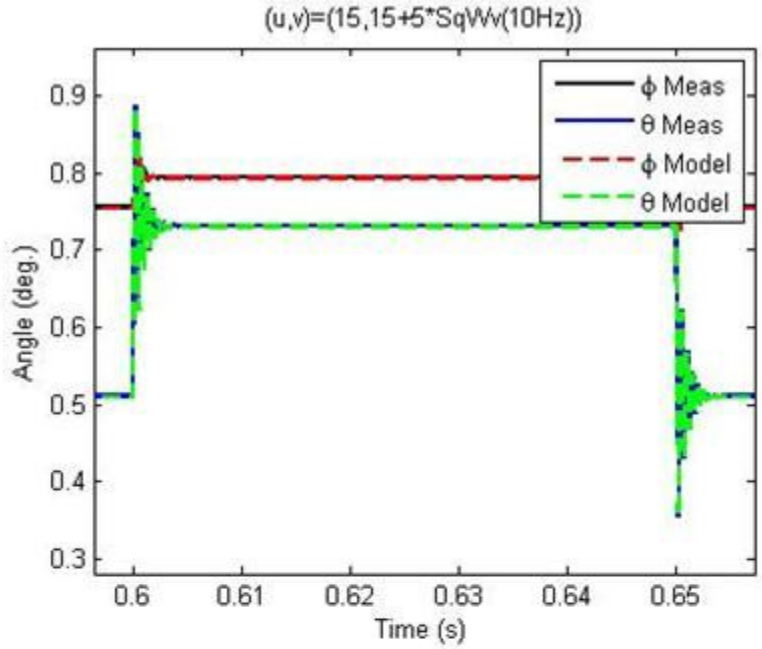

(u,v)=(10+5*SqWv(10Hz),10)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.55
0.6
0.65
0.7
0.75
0.599
0.6
0.601
0.602
0.603
0.604
0.605
0.606
0.607

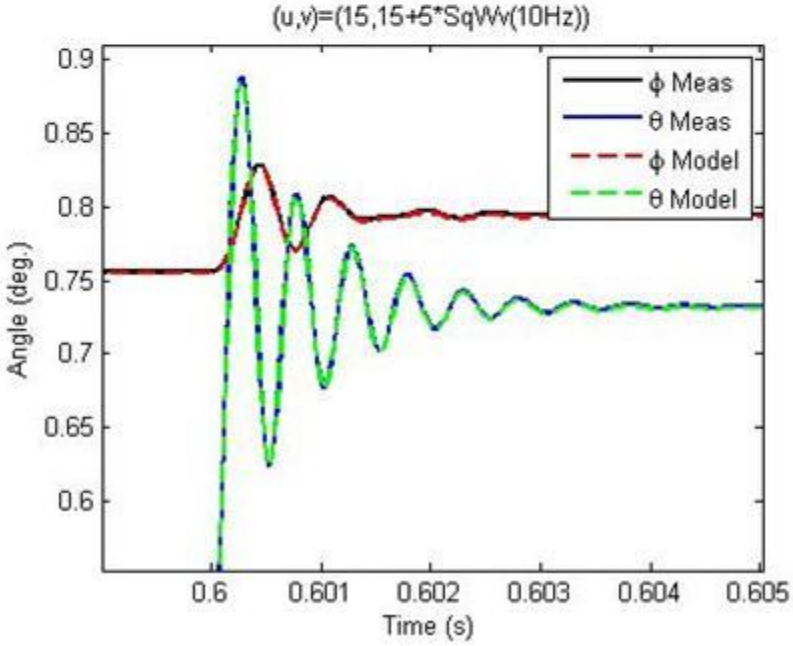

(u,v)=(10,10+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
0.6
0.5
0.4
0.3
0.2
0.6
0.61
0.62
0.63
0.64
0.65
Time (s)

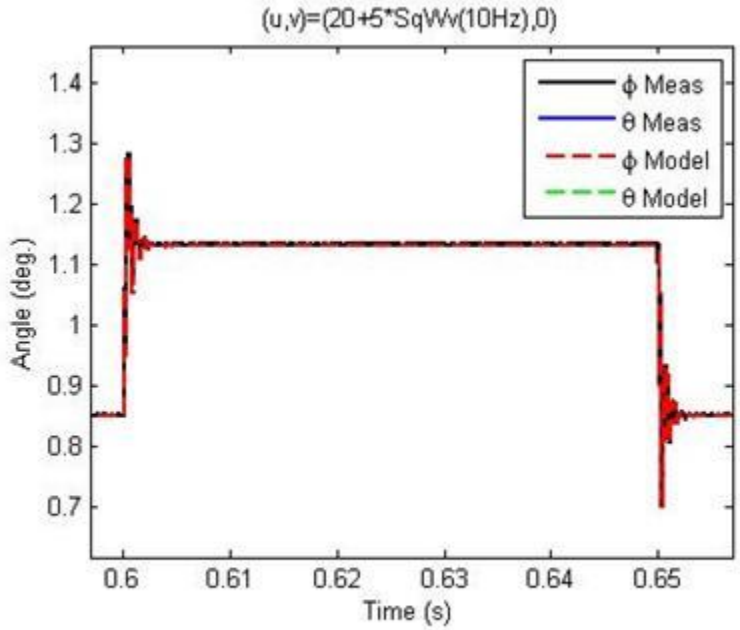

(u,v)=(10,10+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.6
0.601
0.602
0.603
0.604
0.605
Time (s)

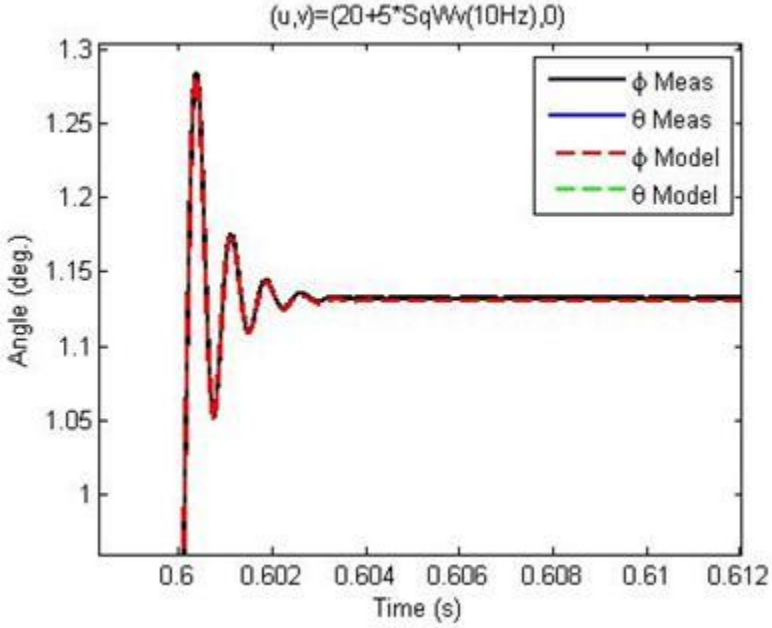

(u,v)=(15+5*SqWv(10Hz),15)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.6
0.61
0.62
0.63
0.64
0.65

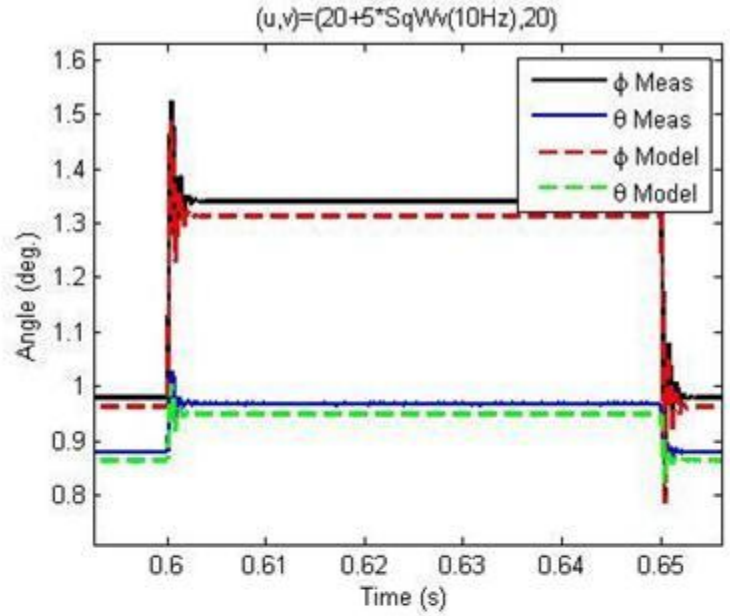

(u,v)=(15+5*SqWv(10Hz),15)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.6
0.601
0.602
0.603
0.604

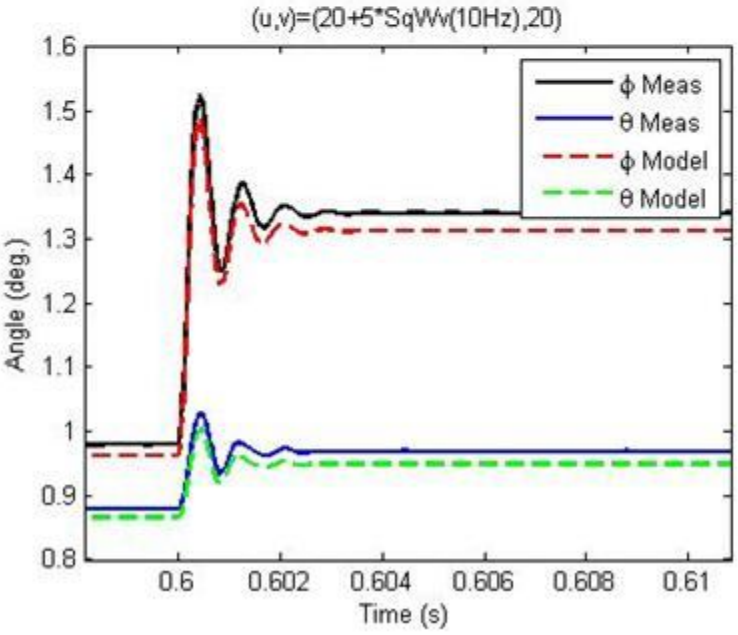

(u,v)=(15,15+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.6
0.61
0.62
0.63
0.64
0.65

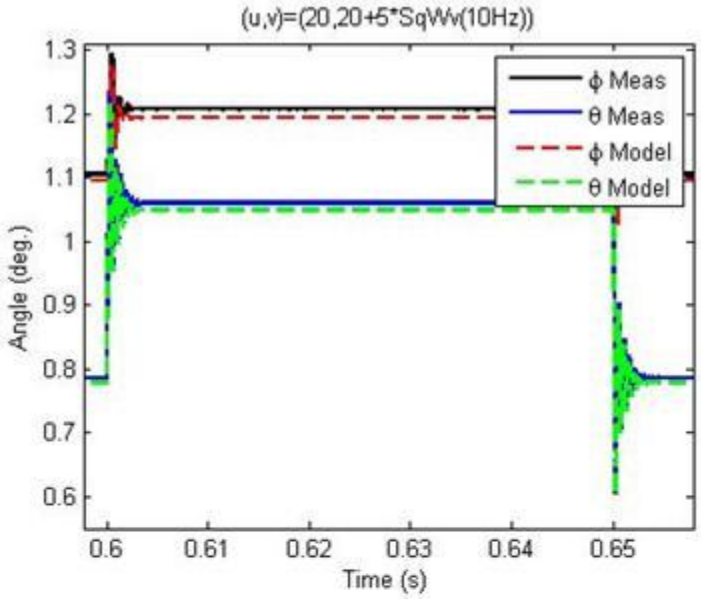

(u,v)=(15,15+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.6
0.601
0.602
0.603
0.604
0.605

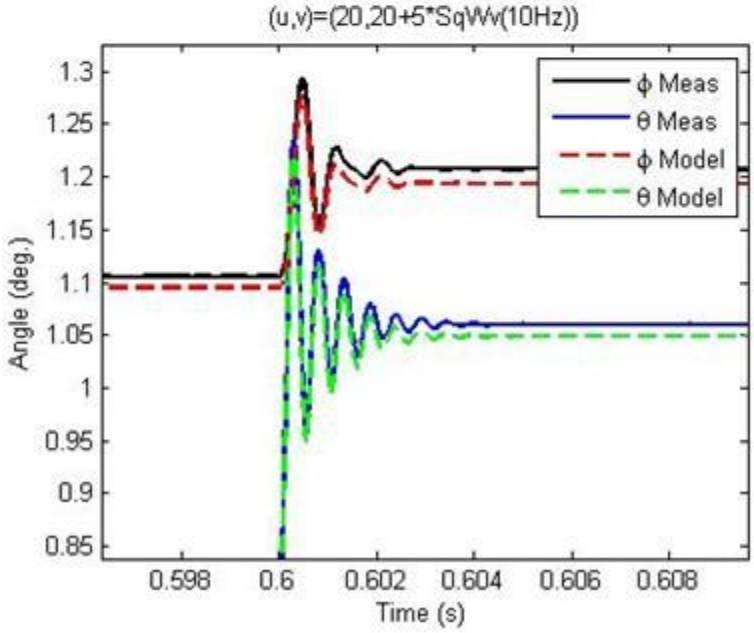

(u,v)=(20+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.61
0.62
0.63
0.64
0.65

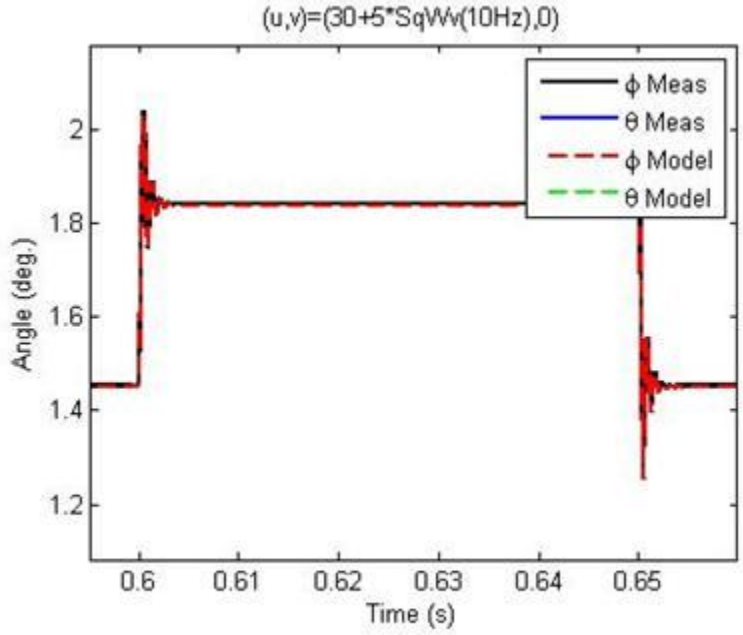

(u,v)=(20+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.3
1.25
1.2
1.15
1.1
1.05
1
0.6
0.602
0.604
0.606
0.608
0.61
0.612

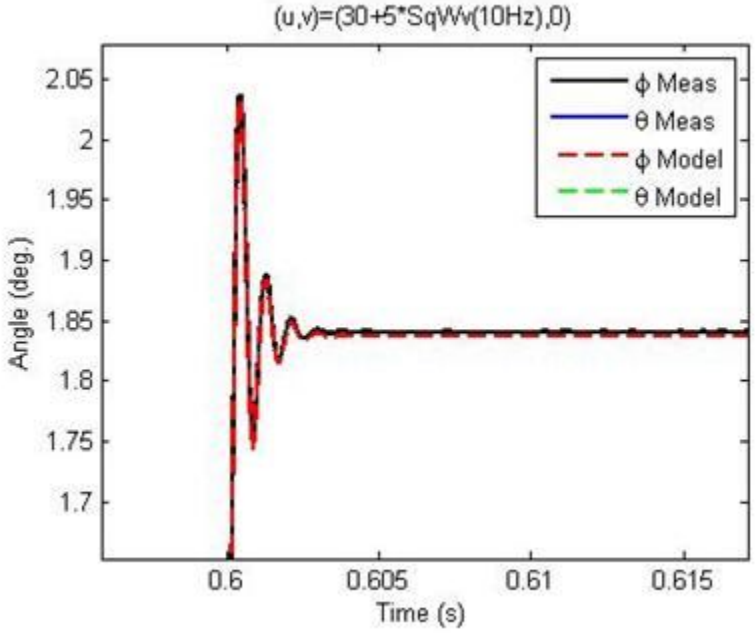

(u,v)=(20+5*SqWv(10Hz),20)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.6
0.61
0.62
0.63
0.64
0.65

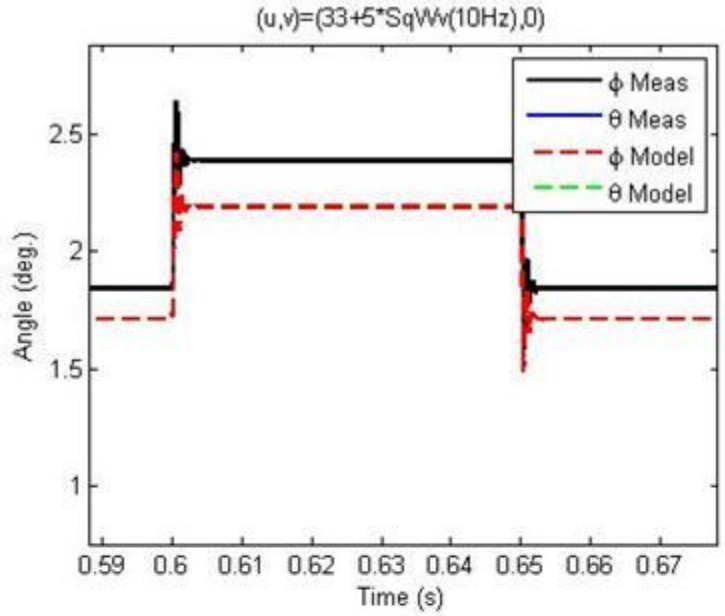

(u,v)=(20+5*SqWv(10Hz),20)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.6
0.602
0.604
0.606
0.608
0.61

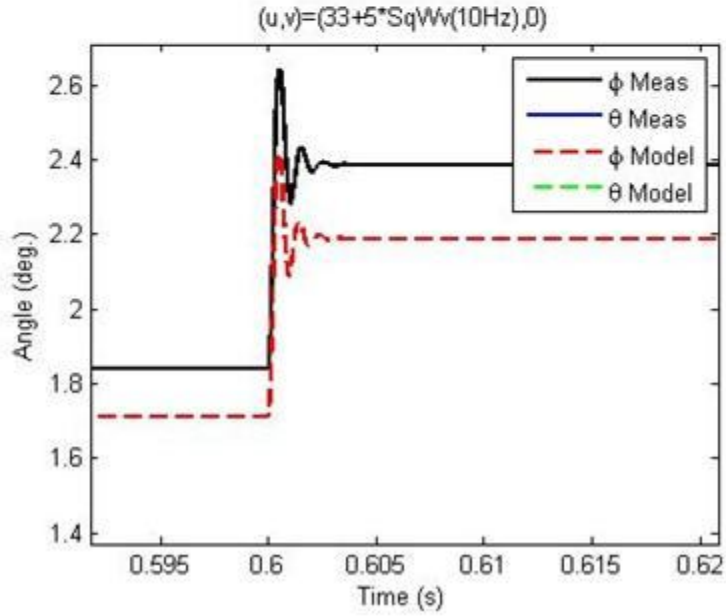

(u,v)=(20,20+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.6
0.61
0.62
0.63
0.64
0.65


(u,v)=(20,20+5*SqWv(10Hz))
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
1.3
1.25
1.2
1.15
1.1
1.05
1
0.95
0.9
0.85
0.598
0.6
0.602
0.604
0.606
0.608


(u,v)=(30+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
2
1.8
1.6
1.4
1.2
0.6
0.61
0.62
0.63
0.64
0.65
Time (s)


(u,v)=(30+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
2.05
2
1.95
1.9
1.85
1.8
1.75
1.7
0.6
0.605
0.61
0.615
Time (s)


(u,v)=(33+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
2.5
2
1.5
1
0.59
0.6
0.61
0.62
0.63
0.64
0.65
0.66
0.67


(u,v)=(33+5*SqWv(10Hz),0)
ϕ Meas
θ Meas
ϕ Model
θ Model
Angle (deg.)
Time (s)
2.6
2.4
2.2
2
1.8
1.6
1.4
0.595
0.6
0.605
0.61
0.615
0.62

Spring Oscillator Formulation Including Noise Coupling

The research into the angular rotation of MEMS micro-mirror as a response to torques has insightful analogies with the displacement of a spring as a function of force. The following is a first-principles development of the noise coupling in to the damped harmonic oscillator model as forcing functions on the right-hand-side (RHS) of the dynamic equations. A spring oscillator is employed.

Consider the following problem illustrated by the Figure A1. We have a mounting platform of mass "M" that can move up and down which is attached via a spring with spring constant "K" to a mass "m." "x" measures the displacement of the mounting platform, called vibration, and "y" measures the displacement of the small mass "m" relative to a fixed reference. Both "x" and "y" are functions of time. A force, "F_{vib}," that is a function of time is applied to the mounting platform cause the vibration. The mass "m" is in the shape of a small metallic square plate that serves as one of the plates for a capacitor whose other plate is at a fixed distance "L" from the reference. A control voltage *V(t)* is applied between the fixed capacitor plate and the spring mass plate which induces an electrostatic force. At time equal to zero the mass "m" is at equilibrium meaning that the spring has stretched an amount necessary to compensate for weight of m due to gravity. The initial conditions (for time equal zero) are $x = x(0) = 0$, $y = y(0) = y_0$, and $\dot{y} = \dot{y}(0) = 0$. Molecules experiencing

Brownian motion displace the variable plate by an amount $y_{bm} = y_{bm}(t)$ where a positive value is taken to be in the same direction as that of a positive "y," i.e., in a downward direction further extending the spring.

Problem: Assuming for $t > t_0$ that $V = V(t)$ is given an $x = x(t)$ is measured: is it possible to solve for $y = y(t)$ if $F_{vib} = F_{vib}(t)$ and M are unknown? If so formulate the equations of motion.

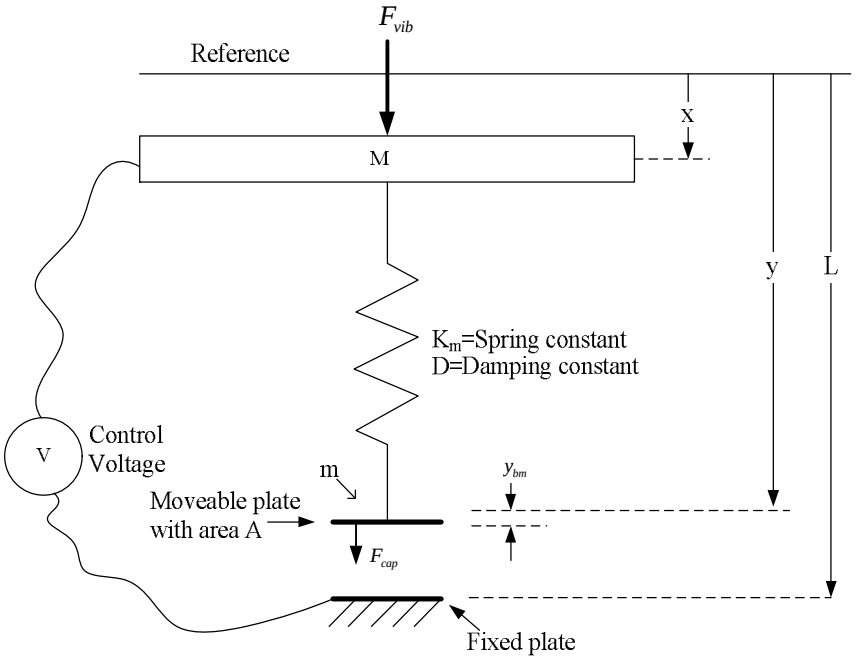


Figure A1. Electrostatically Controlled Spring with Moveable Mounting Platform

Proposed Solution:

The balance of forces at the platform M gives

$$M\ \ddot{x} = F_{vib} + K_m z \tag{A.1}$$

where

$$z(t) = y + y_{bm} - x \tag{A.2}$$

The balance of forces at the moveable plate "m" gives:

$$m\ddot{y} + D\dot{y} = -K_m z + mg + F_{cap} \tag{A.3}$$

where "mg" is the force downward due to gravity and

$$F_{cap} = F_{cap}(y,V) = \frac{\varepsilon A}{2(L-y)^2} V^2 \tag{A.4}$$

is the electro static force downward on the moveable plate of the capacitor due to the voltage, V. If the voltage, $V(t)$, Brownian motion, $y_{bm}(t)$, and vibration motion, $x(t)$, are known then Equation (A.3) is a non-linear differential equation with the forcing terms on the RHS of the equation and it could be solved numerically using SIMULINK, for example.

Substituting z from equation (A.2) into (A.3) gives

$$m\ddot{y} + D\dot{y} + K_m y = K_m x - K_m y_{bm} + mg + F_c(y,V) \tag{A.5}$$

Handling of Transients and Non-Ideal Inputs

Backing up to equation (A.3), consider and ideal environmental situation where there is no vibration or Brownian motion. Imagine that a control voltage $V_0(t)$ is applied for the purpose of creating an eventual fixed displacement once all transient effects have decayed. This could be accomplished if the applied voltage converges to a value of V_∞. If we refer to the intended displacement for the moveable plate to be $y = y_\infty$ then under the ideal environmental conditions

$$V_\infty = \lim_{t\to\infty} V(t) \tag{A.6a}$$

and

$$y_\infty = \lim_{t\to\infty} y(t). \tag{A.6b}$$

In that case y_∞ satisfies

$$K_m y_\infty = mg + \frac{\varepsilon A}{2(L - y_\infty)^2} V_\infty^2 \tag{A.7}$$

A linear approximation of equation (A.5) can be found by assuming that the vibration, $x = x(t)$, and the Brownian motion $y_{bm} = y_{bm}(t)$ are small. We can define $y_0 = y_0(t)$ to be the response when the control voltage $v = v_0(t)$ is applied in the ideal environment. In that case

$$m\ddot{y}_0 + D\dot{y}_0 + K_m y_0 = mg + F_c(y_0, V_0) \tag{A.8a}$$

and that

$$\lim_{t\to\infty}\left(m\ddot{y}_0 + D\dot{y}_0 + K_m y_0\right) = mg + F_c\left(y_\infty, V_\infty\right) \tag{A.8b}$$

we now let "y" represent the incremental change in displacement from y_0 that would result from a non-ideal environment. In this case $y_0(t) + y(t)$ replaces $y(t)$ in equation (A.5). Additionally we can account for possible noise on the control voltage by adding $n = n(t)$ to the voltage variable in the equation. Therefore,

$$\begin{aligned} m\left[\ddot{y}(t) + \ddot{y}_0(t)\right] + D\left[\dot{y}(t) + \dot{y}_0(t)\right] + K_m\left[y(t) + y_0(t)\right] = \\ K_m\left(x - y_{bm}\right) + mg + F_c\left[y(t) + y_0(t), V_0(t) + n(t)\right] \end{aligned} \tag{A.9}$$

Linearization with Taylor Expansion Around (y_∞, V_∞)

Expanding F_c in a Taylor series around the point (y_∞, V_∞) and keeping only the first order terms yields

$$\begin{aligned} F_c\left[y(t) + y_0(t), V(t) + n(t)\right] = F_c\left(y_\infty, V_\infty\right) + K_y \cdot\left[y_0(t) + y(t) - y_\infty\right] \\ + K_V \cdot\left[V_0(t) + n(t) - V_\infty\right] \end{aligned} \tag{A.9}$$

where

$$K_y = \left.\frac{\partial F_c}{\partial y}\right|_{y=y_\infty, V=V_\infty} \tag{A.10}$$

$$K_V = \left.\frac{\partial F_c}{\partial V}\right|_{y=y_\infty, V=V_\infty} \tag{A.11}$$

Substitution of equation (A.9) into (A.8) yields

$$\begin{aligned} m\left[\ddot{y}(t)+\ddot{y}_0(t)\right]+D\left[\dot{y}(t)+\dot{y}_0(t)\right]+K_m\left[y(t)+y_0(t)\right]= \\ K_m\left(x-y_{bm}\right)+mg+F_c\left(y_\infty,V_\infty\right)+K_y\cdot\left[y_0(t)+y(t)-y_\infty\right] \\ +K_V\cdot\left[V_0(t)+n(t)-V_\infty\right] \end{aligned} \qquad \text{(A.12a)}$$

$$\begin{aligned} m\ddot{y}(t)+D\dot{y}(t)+K_m y(t)+m\ddot{y}_0(t)+D\dot{y}_0(t)+K_m y_0(t)= \\ K_m\left(x-y_{bm}\right)+mg+F_c\left(y_\infty,V_\infty\right) \\ +K_y\cdot\left[y_0(t)+y(t)-y_\infty\right]+K_V\cdot\left[V_0(t)+n(t)-V_\infty\right] \end{aligned} \qquad \text{(A.12b)}$$

As time increase we see that

$$m\ddot{y}_0(t)+D\dot{y}_0(t)+K_m y_0(t)\to mg+F_c\left(y_\infty,V_\infty\right)$$

along with

$$y_0(t)\to y_\infty \text{ and } V_0(t)\to V_\infty$$

Therefore, the following equality is true to an arbitrarily good approximation for time sufficiently large so that the ideal solution, $y_0(t)$ can converge sufficiently close to its final value, y_∞.

$$m\ddot{y}(t)+D\dot{y}(t)+K_m y(t)=K_m\left(x-y_{bm}\right)+K_y\cdot y(t)+K_V\cdot n(t) \qquad \text{(A.13a)}$$

or omitting the explicit time notation gives

$$m\ddot{y} + D\dot{y} + K_m y = K_m x - K_m y_{bm} + K_y \cdot y + K_V \cdot n \tag{A.13b}$$

or

$$m\ddot{y} + D\dot{y} + \left(K_m - K_y\right) y = K_m x - K y_{bm} + K_V n \tag{A.13c}$$

$$m\ddot{y} + D\dot{y} + K_E y = K_m x - K y_{bm} + K_V n \tag{A.13d}$$

where K_E is the effective restoring force given by

$$K_E = K_m - K_y \tag{A.13e}$$

Note that as K_y increases the effective restoring constant decreases thereby creating the opportunity for the moveable plate to swing further in response to small stimuli. This is the so called pull in phenomenon, because it becomes possible for the moveable plate to slam into and remain stuck to the bottom plate.

Transfer Function and Input/Out PSD Relationship

If the vibration, Brownian motion, and control voltage noise are independent random processes then the power spectral density of the resulting motion, y, is linearly related to the power spectral densities of the random processes, i.e.,

$$S_{yy}(\omega) = \frac{1}{\left|P(\omega)\right|^2}\left[K_m{}^2 S_{xx}(\omega) + K_m{}^2 S_{y_{bm}y_{bm}}(\omega) + K_V^2 S_{nn}(\omega)\right] \tag{A.14}$$

where the complex polynomial *P* is given by

$$P(\omega) = -m\omega^2 + j\omega D + K_E \tag{A.15}$$

If *P* were known or determined, then measurements of three of the power spectral densities could be used to calculate (i.e., estimate) the other using equation (A.14). For example the Brownian motion power spectral density could be estimated suing

$$\boxed{S_{y_{bm}y_{bm}}(\omega) = \frac{\left|P(\omega)\right|^2 S_{yy}(\omega) - K_m{}^2 S_{xx}(\omega) - K_V^2 S_{nn}(\omega)}{K_m{}^2}} \tag{A.16}$$

Appendix B: Matlab M-files

Table of Contents of Appendix B: Matlab M-Files

1D Torque M-Files and Fitting Curves

M-File 1: compare.m

```
load OutterAxis_TvsR.txt

Vb = 50;
v = 50;
L = 550;
ho = 35;

phi = OutterAxis_TvsR(:,1)*pi/180
torque = zeros(length(phi),1);

for k = 1:length(phi)
torque(k) = OneDTau1(phi(k),v,Vb,ho,L);
end

plot(phi,OutterAxis_TvsR(:,2),'o',phi,torque,'r')
```

M-File 2: OneDTau.m

```
function tau = OneDTau1(phi,v,Vb,ho,L)

% function tau = OneDTau_22Aug06(A, v, Vb,ho,L)
%
% 7 Sept 2006
% Clinton Edwards
% JHU-APL
%
% phi in radians
% v in volts
% Vb in volts
% tau in Newton*meters
% ho in micrometers
% L in micro-meters
% tau in micro-Newton micro-meters
ho = ho*1e-6;   %height of mirror above electrodes (m)
L=L*1e-6;    %length of the mirror facet (m)



e0 = (1/(36*pi))*1e-9;  %  permitivity of space


Lx = 1*L/2;  %0.92 accounting for pad smaller than mirror facet
Lz = 1*L/2;  %0.92 accounting for pad smaller than mirror facet


Const = e0*Lz/2;  %(F)
A = ho;
B = ho+(Lx)*tan(phi);
Bp = ho-(Lx)*tan(phi);

%sets torque to NAN when torque function not valid
%B<=0 or Bp<=0 means mirror facet is swinging past electrode pads
if B<=0|Bp<=0
    tau = nan;
    return
end



tol = 1e-6;  %avoid singularities
if abs(phi)>=tol
    tau1 = -Const*((Vb-v)^2)*(log(B/A)+ho*(1/B-
1/A))*(1/(tan(phi)^2));
    tau2 = Const*((Vb+v)^2)*(log(Bp/A)+ho*((1/Bp)-
(1/A)))*(1/(tan(phi)^2));
```

```
else  %phi is close to zero
    tau1=-Const*((Vb-v)^2)*.5*((Lx)^2)/(ho^2);
    tau2=Const*((Vb+v)^2)*.5*((Lx)^2)/(ho^2);
end

tau = (tau1+tau2);
tau=2*tau;  %Two pads combined for 1d rotation
tau = tau*10^12;  %uN*um

phi=phi;
```

M-File 3: OneDTau_1st.m

```
function OneDTau_1st(v, Vb)

% function OneDTau_1st(phi, v)
%
% 27 Feb 2006
% Clinton Edwards
%
% v in Volts
% Vb in Volts

ho = 32e-6;
eps = (1/(36*pi))*1e-9;
L = 520e-6;
del = 0.001;
phi = [-3:del:3]*pi/180;


phi(find(phi==0))=del/2;  %avoid numerical singularity

Const = eps*L/4;
A = ho;
B = ho+(L/2)*tan(phi);
tau1 = -Const.*((Vb-v).^2).*(1./(tan(phi).^2)).*(log(B/A)+ho*(1./B-
1/A));

Bp = ho-(L/2)*tan(phi);
tau2 =
Const.*((Vb+v).^2).*(1./(tan(phi).^2)).*(log(Bp/A)+ho*((1./Bp)-
(1/A)));

tau = tau1+tau2;


phi_temp = abs(phi);
[X, Y] = find(phi_temp==del/2);

del_tau = tau(Y+2)-tau(Y+1);
del_phi = phi(Y+2)-phi(Y+1);
slope = del_tau/del_phi;
int_val = tau(Y+1);

Lin = slope*phi+int_val;

figure
subplot(2,1,1)
plot(phi*180/pi, tau, phi*180/pi, Lin)
xlabel('angle \phi (degrees)')
ylabel('torque and aproximation (N*m)')
title('Comparison of Torque and First Order Aproximation')
```

```
subplot(2,1,2)
plot(phi*180/pi, tau-Lin)
xlabel('angle \phi (degrees)')
ylabel('Residual (N*m)')
title('Residual of Torque and First Order Aproximation')
```

M-File 4: plot_tau1D.m

```
%plot_tau1D.m

k = 13200;  %uN*um
steps = 25;
phiMax = 4;
phiMin = -4;
VMax = 28;
VMin = -28;
delta_phi = (phiMax-phiMin)/steps;
delta_V = (VMax-VMin)/steps;
[PHI, V] =
meshgrid([phiMin:delta_phi:phiMax]*pi/180,[VMin:delta_V:VMax]);
TAU = zeros(size(PHI));
[x y] = size(PHI);
g = k*PHI;  %2-D
for m=1:y
    for n=1:x
        TAU(n,m) = OneDTau1(PHI(n,m),V(n,m),50,32,520);
    end
end

%3-D Surf of phi, voltage, and torque (uN*um)
figure(1)
surf(PHI*57.3,V,TAU)
xlabel('phi (deg.)')
ylabel('volts (v)')
zlabel('torque uN*um')
axis tight

%plot curves for various voltages
figure(2)
for ii=1:5:x
    plot(PHI(ii,:)*57.3,TAU(ii,:),'--k','linewidth',2)
    hold on
end
plot(PHI(1,:)*57.3,k*PHI(1,:), '--r', 'linewidth',2)
hold off
axis tight
grid on
xlabel('\phi (deg.)')
ylabel('torque (uN*um)')
% legend(num2str(V(1:5:x,1)))

figure(3)
k = 13200;  %uN*um
steps = 100;
VMax = 30;
VMin = -30;
delta_V = (VMax-VMin)/steps;
h=1e-9;
```

```
PHIintn = [];
for v=VMin:delta_V:VMax

    PHIn=0;  %first guess at phi
    f=1;
    counter = 1;
    while abs(f)>1e-3&counter<100
        f = k*PHIn-OneDTau1(PHIn,v,50,32,520);
        fp = k*(PHIn+h)-OneDTau1(PHIn+h,v,50,32,520);
        der = (fp-f)/h;
        PHIn=PHIn-f/der;
        counter = counter+1;
    end
    if(counter==100)
        disp(['overflow at ',num2str(v),' volts'])
    end
    PHIintn = [PHIintn,PHIn];
end
    plot(VMin:delta_V:VMax,PHIintn*57.3,'linewidth',3)

xlabel('V,(volts)')
ylabel('\phi (deg.)')
title('Equilibrium of \phi vs. V')
grid on
```

1D Torque M-files

M-File 5: init_1D_file.m

```
% init_1D_file
% physical parameters for umirror
% param_1D = [Vb,ho,L,I,D,Km];
%
% MEMS Optical 2-axis tip-tilt mirror parameters
% INNER AXIS:
% I = 1.14e-4; kg*um^2
% D = 0.297;  uN*um*s
% Km = 13200;  uN*um
%
%
% OUTTER AXIS:
% I = 1.84e-4; kg*um^2
% D = 0.621;  uN*um*s
% Km = 13200;  uN*um


global param_1D

Vb = 50;  % (v)
ho = 32;  % (um)
L = 520; % (um)
I = 1.14e-4; % (Kg*um^2)
D = 0.297; % (uN*um*s)
Km = 13200; % (uN*um)

param_1D = [Vb,ho,L,I,D,Km];
```

M-File 6: phi_given_v.m

```
function phi = phi_given_v(v)

% 7 Oct 2006
% Clinton Edwards
% JHU-APL
% param = [Vb,ho,L,I,D,K];
% phi in radians
% v in volts
% Vb in volts
% tau in uN*um
% ho in um
% L in um
% tau in uN*um

global param_1D

Vb = param_1D(1);
ho = param_1D(2);
L = param_1D(3);
I = param_1D(4);
D = param_1D(5);
Km = param_1D(6);

    PHIn=0;  %first guess at phi
    h=1e-12; %delta_x   derivative=delta_y/delta_x
    f=1;
    counter = 1;
    while abs(f)>1e-12&counter<200
        f = -Km*PHIn+T(PHIn,v);
        fp = -Km*(PHIn+h)+T(PHIn+h,v);
        der = (fp-f)/h;
        PHIn=PHIn-f/der;
        counter = counter+1;
    end
    if(counter==100)
        disp(['overflow at ',num2str(v),' volts'])
    end

    phi = PHIn;
```

M-File 7: T.m

```
function tau = T(phi,v)
% 7 Oct 2006
% Clinton Edwards
% JHU-APL

% param_1D = [Vb,ho,L,I,D,K];
% phi_o = mean value of phi (radians)
% v = actuation voltage (v)
% Vb = bias voltage (v)
% ho = height of mirror above electrodes (um)
% L = length of mirror facet (um)
% I = mirror facet moment of inertia (Kg*um^2)
% D = damping constant (uN*um*s)
% K = spring restoring constant (uN*um)
% tau = torque (uN*um)
global param_1D

Vb = param_1D(1);
ho = param_1D(2);
L = param_1D(3);
I = param_1D(4);
D = param_1D(5);
Km = param_1D(6);

ho = ho*1e-6;   %height of mirror above electrodes (m)
L=L*1e-6;    %length of the mirror facet (m)

e0 = (1/(36*pi))*1e-9;  %  permitivity of space

Lx = 1*L/2;  %0.92 accounting for pad smaller than mirror facet
Lz = 1*L/2;  %0.92 accounting for pad smaller than mirror facet

Const = e0*Lz/2;  %(F)
A = ho;
B = ho+(Lx)*tan(phi);
Bp = ho-(Lx)*tan(phi);

%sets torque to NAN when torque function not valid
%B<=0 or Bp<=0 means mirror facet is swinging past electrode pads
if B<=0|Bp<=0
    tau = nan;
    return
end

tol = 1e-6;  %avoid singularities
if abs(phi)>=tol
    tau1 = -Const*((Vb-v)^2)*(log(B/A)+ho*(1/B-
1/A))*(1/(tan(phi)^2));
```

```
    tau2 =  Const*((Vb+v)^2)*(log(Bp/A)+ho*(1/Bp-
1/A))*(1/(tan(phi)^2));
else  %phi is close to zero
    tau1=-Const*((Vb-v)^2)*.5*((Lx)^2)/(ho^2);
    tau2=Const*((Vb+v)^2)*.5*((Lx)^2)/(ho^2);
end

tau = (tau1+tau2);
tau=2*tau;  %Two pads combined for 1d rotation
tau = tau*10^12;  %uN*um
```

M-File 8: T_num_sol.m

```
function T_num_sol(v,simout)

% param = [Vb,ho,L,I,D,K];
% phi_o = mean value of phi (radians)
% v = actuation voltage (v)
% Vb = bias voltage (v)
% ho = height of mirror above electrodes (um)
% L = length of mirror facet (um)
% I = mirror facet moment of inertia (Kg*um^2)
% D = damping constant (uN*um*s)
% K = spring restoring constant (uN*um)

global param

% simulation Time
Tsim = .5;
t = 0:0.00001:Tsim;

% magnitude of sinusoidal vibration
phi_vib = 0.001%0.001; %radians

Vb = param_1D(1);
ho = param_1D(2);
L = param_1D(3);
I = param_1D(4);
D = param_1D(5);
Km = param_1D(6);


% determine steady-state phi for given voltage
phi_o = phi_given_v(v);

% determine the partial derivative of Torque (tau) WRT phi at
steady-state
% voltage v and angle phi_o
tau_der_phi = T_phi(phi_o,v);
Ke = tau_der_phi;
K = Km-Ke; %equivalent restoring force
wo = sqrt(K/I);
w=2*pi*10;


To = phi_vib*tau_der_phi; % partial derivative of of Tau WRT phi

% general solution from Boyce and Diprima Sixth Edition
% "Elementary Differnetial Equations" Page 193
% I*phi''+D*phi'+Km*phi = To*sin(w*t)
% c1*exp(r1*t)+c2*exp(r2*t)+(To/Z)*cos(w*t-phase)
```

```
% where Z = sqrt((I^2)*(wo^2-w^2)^2+(D*w)^2);
% and phase = acos(I*(wo^2-w^2)/Z^2);
% and wo = sqrt(Km/I);
% and w=2*pi*10;

% initial conditions for phi
A = 0; %intial conditions for phi(t=0)
B = 0; %intial conditions for phi'(t=0)

% roots to the characteristic equation
r1 = (-D+sqrt(D^2-4*K*I))/(2*I);
r2 = (-D-sqrt(D^2-4*K*I))/(2*I);

% consolidation of a reoccuring constant
Z = sqrt((I^2)*(wo^2-w^2)^2+(D*w)^2);

% constants in front of decaying exponentials in general solution
c1 = A-((-1/(r1+r2))*(B-A*r1+To*r1*I*(wo^2-w^2)/(Z^2)-
To*(w^2)*D/(Z^2)))-To/(Z^2)*(I*(wo^2-w^2));
c2 = (-1/(r1+r2))*(B-A*r1+To*r1/(Z^2)*(I*(wo^2-w^2))-
To*(w^2)*D/(Z^2));
phase = acos(I*(wo^2-w^2)/Z^2);

% solution to 2nd differential equation for arbitrary initial
conditions
phi_plot = phi_o+c1*exp(r1*t)+c2*exp(r2*t)+(To/Z)*cos(w*t-phase);
%phi_plot = phi_o +(To/Z)*cos(w*t-phase);

% plot phi as a function of time
figure(1)
plot(t,real(phi_plot)*180/pi,
simout.time,simout.signals.values,'r','linewidth',3)
xlabel('time (sec.)')
ylabel('phi (deg.)')
size(t)
size(simout.signals.values)

figure(2)
size(real(phi_plot))
semilogy(t,real(phi_plot)*180/pi-simout.signals.values.',
'linewidth',2)
xlabel('time (sec.)')
ylabel('residuals of SIMULINK and Linearized Solution')
```

M-File 9: tau_der_phi.m

```
function tau_der_phi = T_phi(phi,v)

% calculated the partial derivative of 1-D torque with respect to
phi
%
% 7 Oct 2006
% Clinton Edwards
% JHU-APL
%
% phi in radians
% v in volts
% Vb in volts
% tau in uN*um
% ho in um
% L in um
% tau in uN*um

global param_1D

Vb = param_1D(1);
ho = param_1D(2);
L = param_1D(3);
I = param_1D(4);
D = param_1D(5);
K = param_1D(6);

ho = ho*1e-6;   %height of mirror above electrodes (m)
L=L*1e-6;    %length of the mirror facet (m)

e0 = (1/(36*pi))*1e-9;  %  permitivity of space
Lx = 1*L/2;  %0.92 accounting for pad smaller than mirror facet
Lz = 1*L/2;  %0.92 accounting for pad smaller than mirror facet

Const = e0*Lz/2;  %(F)

%tau_der_phi = 2*Const*(Vb-
v)*(log((ho+Lx*tan(phi))/ho)+ho*(1/(ho+Lx*tan(phi))-
1/ho))/tan(phi)^2+2*Const*(Vb+v)*(log((ho-
Lx*tan(phi))/ho)+ho*(1/(ho-Lx*tan(phi))-1/ho))/tan(phi)^2;
tau_der_phi =-Const*(Vb-
v)^2*(1/2*L*(1+tan(phi)^2)/(ho+1/2*L*tan(phi))-
1/2*ho/(ho+1/2*L*tan(phi))^2*L*(1+tan(phi)^2))/tan(phi)^2+2*Const*(V
b-v)^2*(log((ho+1/2*L*tan(phi))/ho)+ho*(1/(ho+1/2*L*tan(phi))-
1/ho))/tan(phi)^3*(1+tan(phi)^2)+Const*(Vb+v)^2*(-
1/2*L*(1+tan(phi)^2)/(ho-1/2*L*tan(phi))+1/2*ho/(ho-
1/2*L*tan(phi))^2*L*(1+tan(phi)^2))/tan(phi)^2-
2*Const*(Vb+v)^2*(log((ho-1/2*L*tan(phi))/ho)+ho*(1/(ho-
1/2*L*tan(phi))-1/ho))/tan(phi)^3*(1+tan(phi)^2);
```

```
tau_der_phi = 2*tau_der_phi*10^12;  %uN*um
```

M-File 10: tau_der_v.m

```
function tau_der_v = T_v(phi,v)

% calculated the partial derivative of 1-D torque with respect to v
% 7 Oct 2006
% Clinton Edwards
% JHU-APL
%
% phi in radians
% v in volts
% Vb in volts
% tau in uN*um
% ho in um
% L in um
% tau in uN*um
global param_1D

Vb = param_1D(1);
ho = param_1D(2);
L = param_1D(3);
I = param_1D(4);
D = param_1D(5);
K = param_1D(6);

ho = ho*1e-6;   %height of mirror above electrodes (m)
L=L*1e-6;    %length of the mirror facet (m)


e0 = (1/(36*pi))*1e-9;  % permitivity of space
Lx = 1*L/2;  %0.92 accounting for pad smaller than mirror facet
Lz = 1*L/2;  %0.92 accounting for pad smaller than mirror facet
Const = e0*Lz/2;  %(F)

tau_der_v = 2*Const*(Vb-
v)*(log((ho+1/2*L*tan(phi))/ho)+ho/(ho+1/2*L*tan(phi))-
1)/tan(phi)^2+2*Const*(Vb+v)*(log((ho-1/2*L*tan(phi))/ho)+ho/(ho-
1/2*L*tan(phi))-1)/tan(phi)^2;

tau_der_v = 2*tau_der_v*10^12;  %uN*um/v
```

2D Torque M-files

M-File 11: G.m

```
function out=G(r1,r2,s1,s2,p,q)
global param_2D
Eo = (1/(36*pi))*1e-9; %permitivity of space
r1=r1*1e-6;
r2=r2*1e-6;
s1=s1*1e-6;
s2=s2*1e-6;
ho=param_2D(2)*1e-6; %(m);
%trap p's anc q's that yield complex logarithms (physical
interpretation:
%the mirror has tilted past the voltage pads)
if (ho-p*r2-q*s2)<=1e-6
    out = NaN;
    return
elseif (ho-p*r1-q*s2)<=1e-6
    out = NaN;
    return
elseif (ho-p*r2-q*s1)<=1e-6
    out = NaN;
    return
 elseif (ho-p*r1-q*s1)<=1e-6
    out = NaN;
    return
end
% param_2D = [Vb,ho,r1,r2,s1,s2,Ix,Dx,Kmx,Iz,Dz,Kmz];
%if p=0 and q=0 numerical solution via L'Hopital's Rule
if abs(p)<10e-6&abs(q)<10e-6
out = (1/(4*ho^2))*Eo*(r1^2-r2^2)*(s2-s1);

%if p=0 numerical solution via L'Hopital's Rule
elseif abs(p)<10e-6
out = (Eo/4)*(r1^2-r2^2)*((s2-s1)/((ho-q*s2)*(ho-q*s1)));

%if q=0 numerical solution via L'Hopital's Rule
elseif abs(q)<10e-6
out = Eo*((s1-s2)/(2*p^2))*(ho/(ho-p*r2)-(ho/(ho-p*r1))+log((ho-
p*r2)/(ho-p*r1)));

else
out = (Eo/(2*p^2*q))*((ho-q*s2)*log((ho-p*r2-q*s2)/(ho-p*r1-
q*s2))...
        -(ho-q*s1)*(log((ho-p*r2-q*s1)/(ho-p*r1-q*s1))));
end %if


% % param_2D = [Vb,ho,r1,r2,s1,s2,Ix,Dx,Kmx,Iz,Dz,Kmz];
% %if p=0 and q=0 numerical solution via L'Hopital's Rule
```

```
% if abs(p)<1e-6&&abs(q)<1e-6
% out = (1/(4*ho^2))*Eo*((r1^2-r2^2)*(s2-s1));
%
% %if p=0 numerical solution via L'Hopital's Rule
% elseif abs(p)<1e-6
% out = (Eo*((ho-q*s2)*(-r2^2/(ho-q*s2)^2+r1^2/(ho-q*s2)^2)-...
%       (ho-q*s1)*(-r2^2/(ho-q*s1)^2+r1^2/(ho-q*s1)^2)))/(4*q);
%
% %if q=0 numerical solution via L'Hopital's Rule
% elseif abs(q)<1e-6
% out = Eo*(-s2*(log(ho-p*r2)-log(ho-p*r1))+(ho)*(-s2/(ho-
p*r2)+s2/(ho-p*r1))...
%       +s1*(log(ho-p*r2)-log(ho-p*r1))-(ho)*(-s1/(ho-p*r2)+s1/(ho-
p*r1)))/(2*p^2);
% else
%
% out = (Eo/(2*p^2*q))*((ho-q*s2)*(log(ho-p*r2-q*s2)-log(ho-p*r1-
q*s2))...
%         -(ho-q*s1)*(log(ho-p*r2-q*s1)-log(ho-p*r1-q*s1)));
% end %if
```

M-File 12: ang_given_volt_SD.m

```
function ANGLE = ang_given_volt_2D_SD(volt)
% function ANGLE = ang_given_volt_2D_NR(volt=[u,v])
%
% physical and dynamic parameters for umirror
% param_2D = [Vb,ho,x1,x2,z1,z2,I2,D2,Km2,I1,D1,Km1];
global param_2D
% u = volt(1);
% v = volt(2);
% phi=ang(1);
% theta=ang(2);

Vb = param_2D(1);
ho = param_2D(2);
Lx = 2*param_2D(4);
Lz = 2*param_2D(6);
I2 = param_2D(7);
D2 = param_2D(8);
Km2 = param_2D(9);
I1 = param_2D(10);
D1 = param_2D(11);
Km1 = param_2D(12);

MAX = 1e3;
%steady-state phi calculation
    ANGn=[.1;.1]*pi/180; %first guess at angles phi and theta
    h=1e-10; % step size
    err=1;
    counter = 1;

    while abs(err)>1e-6&counter<MAX
        c1 = [Km1;0]-(T2D([ANGn(1)+h, ANGn(2)],volt)-T2D([ANGn(1),
ANGn(2)],volt))/h;
        c2 = [0;Km2]-(T2D([ANGn(1), ANGn(2)+h],volt)-T2D([ANGn(1),
ANGn(2)],volt))/h;
        J = [c1,c2];
        if any(any(isnan(J)))
           ANGn=[NaN;NaN];
           break
        end
        Jinv = inv(J);
        T = T2D(ANGn,volt);
        F = ([Km1;Km2].*ANGn)-T;
        ANGn = ANGn-Jinv*F;
        err = max(abs(([Km1;Km2].*ANGn)-T));
        counter = counter+1;
%         if isnan(c1)|isnan(c2)
%             break
%         end
```

```
end

if(counter==MAX)
    disp(['limited convergence for ',num2str(volt),' volts'])
end
 ANGLE = ANGn;
```

M-File 13: Gp.m

```
function PD_Gp=Gp(r1,r2,s1,s2,p,q)
global param_2D
Eo = (1/(36*pi))*1e-9; %permitivity of space
r1=r1*1e-6;
r2=r2*1e-6;
s1=s1*1e-6;
s2=s2*1e-6;
ho=param_2D(2)*1e-6; %(m);
%trap p's anc q's that yield complex logarithms (physical
interpretation:
%the mirror has tilted past the voltage pads)
if (ho-p*r2-q*s2)<=1e-8
    PD_Gp(1,1) = NaN;
    return
elseif (ho-p*r1-q*s2)<=1e-8
    PD_Gp(1,1) = NaN;
    return
elseif (ho-p*r2-q*s1)<=1e-8
    PD_Gp(1,1) = NaN;
    return
 elseif (ho-p*r1-q*s1)<=1e-8
    PD_Gp(1,1) = NaN;
    return
end

PD_Gp = (-Eo/(p^3*q))*((ho-q*s2)*(log((ho-p*r2-q*s2)/(ho-p*r1-
q*s2))).
        -(ho-q*s1)*(log((ho-p*r2-q*s1)/(ho-p*r1-
q*s1))))+(Eo/(2*p^2*q))*...
        ((ho-q*s2)*(-r2/(ho-p*r2-q*s2)+r1/((ho-p*r1-q*s2)))-(ho-
q*s1)*...
        (-r2/(ho-p*r2-q*s1)+r1/(ho-p*r1-q*s1)));

PD_Gp=PD_Gp;
end %(function Gp)
```

M-File 14: Gq.m

```
function PD_Gq=Gq(r1,r2,s1,s2,p,q)
global param_2D
Eo = (1/(36*pi))*1e-9; %permitivity of space
r1=r1*1e-6;
r2=r2*1e-6;
s1=s1*1e-6;
s2=s2*1e-6;
ho=param_2D(2)*1e-6; %(m);

%trap p's anc q's that yield complex logarithms (physical
interpretation:
%the mirror has tilted past the voltage pads)
if (ho-p*r2-q*s2)<=1e-8
    PD_Gq(1,1) = NaN;
    return
elseif (ho-p*r1-q*s2)<=1e-8
    PD_Gq(1,1) = NaN;
    return
elseif (ho-p*r2-q*s1)<=1e-8
    PD_Gq(1,1) = NaN;
    return
 elseif (ho-p*r1-q*s1)<=1e-8
    PD_Gq(1,1) = NaN;
    return
end

PD_Gq = (Eo/(2*p^2))*((-ho/q^2)*(log((ho-p*r2-q*s2)/(ho-p*r1-
q*s2))).../
        -((-ho/q^2)*(log((ho-p*r2-q*s1)/(ho-p*r1-
q*s1))))+(Eo/(2*p^2))*.../
        (((ho-q*s2)/q)*(-s2/(ho-p*r2-q*s2)+s2/((ho-p*r1-q*s2)))-
((ho-q*s1)/q)*.../
        (-s1/(ho-p*r2-q*s1)+s1/(ho-p*r1-q*s1))));

PD_Gq = PD_Gq; %uN*um
end %(function Gq)
```

M-File 15: K1phi.m

```
function PD_K1Phi = K1Phi(volt)
% function PD_K1Phi = K1Phi(volt=[u, v])
%
% Clint Edwards, JHU-APL
% 27 December 2006
%
% K1Phi.m calculated the partial derivative of Tau1 or the Tz with
respect
% to the angle Phi
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% PD_K1Phi (uN*um/rad)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D

%actuating voltages defined
u = volt(1);
v = volt(2);
%bias voltage defined
Vb = param_2D(1);
% voltage amplifier gain
k = 1;

%pad voltages defined
V1 = Vb+k*(v-u);
V2 = Vb+k*(-v-u);
V3 = Vb+k*(-v+u);
V4 = Vb+k*(v+u);

%determine angle for given voltages
ang = ang_given_volt_2D_SD([u v]);

phi = ang(1);
theta = ang(2);

%parametrized angles and differentials defined
A = -tan(phi);
```

```
C = tan(theta)*sec(phi);
% dAdPhi = -sec(phi).^2;
% dCdPhi = tan(theta)*tan(phi)*sec(phi);

%voltage pad lengths defined
x1 = param_2D(3);
x2 = param_2D(4);
z1 = param_2D(5);
z2 = param_2D(6);

%normalized partial derivatives for A and C returned for each
voltage pad
G11A = Gp(x1,x2,z1,z2,A,C)*V1^2;
G11C = Gq(x1,x2,z1,z2,A,C)*V1^2;
G12A = Gp(x1,x2,-z2,-z1,A,C)*V2^2;
G12C = Gq(x1,x2,-z2,-z1,A,C)*V2^2;
G13A = Gp(-x2,-x1,-z2,-z1,A,C)*V3^2;
G13C = Gq(-x2,-x1,-z2,-z1,A,C)*V3^2;
G14A = Gp(-x2,-x1,z1,z2,A,C)*V4^2;
G14C = Gq(-x2,-x1,z1,z2,A,C)*V4^2;

Gt1 = [G11A G12A G13A G14A]*-1*sec(phi)^2;
Gt2 = [G11C G12C G13C G14C]*tan(theta)*tan(phi)*sec(phi);
Gt = [Gt1 Gt2];

%Vector sum of all voltage pads
PD_K1Phi = sum(Gt);
PD_K1Phi = PD_K1Phi*1e12;  %changes units to uN*um/rad
```

M-File 16: K1Theta

```
function PD_K1Theta = K1Theta(volt)
% function PD_K1Theta = K1Theta(volt=[u, v])
%
% Clint Edwards, JHU-APL
% 27 December 2006
%
% K1Theta.m calculated the partial derivative of Tau1 or the Tz with
respect
% to the angle Theta
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% PD_K1Theta (uN*um/rad)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D

%actuating voltages defined
u = volt(1);
v = volt(2);
%bias voltage defined
Vb = param_2D(1);
% voltage amplifier gain
k = 1;

%pad voltages defined
V1 = Vb+k*(v-u);
V2 = Vb+k*(-v-u);
V3 = Vb+k*(-v+u);
V4 = Vb+k*(v+u);

%determine angle for given voltages
ang = ang_given_volt_2D_SD([u v]);
phi = ang(1);
theta = ang(2);

%parametrized angles and differentials defined
A = -tan(phi);
C = tan(theta)*sec(phi);
```

```
%voltage pad lengths defined
x1 = param_2D(3);
x2 = param_2D(4);
z1 = param_2D(5);
z2 = param_2D(6);

%normalized partial derivatives for A and C returned for each
voltage pad
G11C = Gq(x1,x2,z1,z2,A,C)*V1^2;
G12C = Gq(x1,x2,-z2,-z1,A,C)*V2^2;
G13C = Gq(-x2,-x1,-z2,-z1,A,C)*V3^2;
G14C = Gq(-x2,-x1,z1,z2,A,C)*V4^2;

Gt = [G11C G12C G13C G14C]*sec(phi)*sec(theta)^2;
%Vector sum of all voltage pads
PD_K1Theta = sum(Gt);
PD_K1Theta = PD_K1Theta*1e12;  %changes units to uN*um/rad
```

M-File 17: K1Theta_num.m

```
function PD_K1Theta = K1Theta_num(volt)
% function PD_K1Theta = K1Theta_num(volt=[u, v]) NUMERICAL
DERIVATIVE
%
% Clint Edwards, JHU-APL
% 31 December 2006
%
% K1Theta_num.m calculated the partial derivative of Tau1 or the Tz
with respect
% to the angle theta numerically
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% PD_K1Theta (uN*um/rad) (NUMERICAL DERIVATIVE)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D

%actuating voltages defined
u = volt(1);
v = volt(2);
%bias voltage defined
Vb = param_2D(1);
% voltage amplifier gain
k = 1;
h=1e-9;

ang = ang_given_volt_2D_SD([u v]);

phi=ang(1);
theta=ang(2);
t_num = (T2D([phi theta+h],[u v])-T2D([phi theta],[u v]))/h;
PD_K1Theta = t_num(1);
```

M-File 18: K1U.m

```
function PD_K1U = K1U(volt)
% function PD_K1U = K1U(volt=[u, v])
%
% Clint Edwards, JHU-APL
% 27 December 2006
%
% K1U.m calculated the partial derivative of Tau1 or Tz with respect
% to the acutating voltage u
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% PD_K1U (uN*um/volt)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D

%actuating voltages defined
u = volt(1);
v = volt(2);
%bias voltage defined
Vb = param_2D(1);
% voltage amplifier gain
k = 1;

%pad voltages defined
V1 = Vb+k*(v-u);
V2 = Vb+k*(-v-u);
V3 = Vb+k*(-v+u);
V4 = Vb+k*(v+u);

%determine angle for given voltages
ang = ang_given_volt_2D_SD([u v]);
phi = ang(1);
theta = ang(2);

%parametrized angles
A = -tan(phi);
C = tan(theta)*sec(phi);

%voltage pad lengths defined
```

```
x1 = param_2D(3);
x2 = param_2D(4);
z1 = param_2D(5);
z2 = param_2D(6);

%normalized partial derivatives for each voltage pad
K1Un = zeros(4,1);
K1Un(1,1) = G(x1,x2,z1,z2,A,C);
K1Un(2,1) = G(x1,x2,-z2,-z1,A,C);
K1Un(3,1) = G(-x2,-x1,-z2,-z1,A,C);
K1Un(4,1) = G(-x2,-x1,z1,z2,A,C);

%partial derivatives scaled by pad voltage
K1U = zeros(4,1);
K1U(1,1) = K1Un(1,:)*V1*2;
K1U(2,1) = K1Un(2,:)*V2*2;
K1U(3,1) = K1Un(3,:)*V3*2;
K1U(4,1) = K1Un(4,:)*V4*2;

%Vector sum of all voltage pads
PD_K1U = -K1U(1,1)-K1U(2,1)+K1U(3,1)+K1U(4,1);
PD_K1U = PD_K1U*1e12;  %changes units to uN*um/volt
end
```

M-File 19: K1V.m

```
function PD_K1V = K1V(volt)
% function PD_K1V = K1V(volt=[u, v])
%
% Clint Edwards, JHU-APL
% 27 December 2006
%
% K1V.m calculated the partial derivative of Tau2 or Tx with respect
% to the acutating voltage v
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% PD_K1V (uN*um/volt)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D

%actuating voltages defined
u = volt(1);
v = volt(2);
%bias voltage defined
Vb = param_2D(1);
% voltage amplifier gain
k = 1;

%pad voltages defined
V1 = Vb+k*(v-u);
V2 = Vb+k*(-v-u);
V3 = Vb+k*(-v+u);
V4 = Vb+k*(v+u);

%determine angle for given voltages
ang = ang_given_volt_2D_SD([u v]);
phi = ang(1);
theta = ang(2);

%parametrized angles
A = -tan(phi);
C = tan(theta)*sec(phi);

%voltage pad lengths defined
```

```
x1 = param_2D(3);
x2 = param_2D(4);
z1 = param_2D(5);
z2 = param_2D(6);

%normalized partial derivatives for each voltage pad
K1Vn = zeros(4,1);
K1Vn(1,1) = G(x1,x2,z1,z2,A,C);
K1Vn(2,1) = G(x1,x2,-z2,-z1,A,C);
K1Vn(3,1) = G(-x2,-x1,-z2,-z1,A,C);
K1Vn(4,1) = G(-x2,-x1,z1,z2,A,C);

%partial derivatives scaled by pad voltage
K1V = zeros(4,1);
K1V(1,1) = K1Vn(1,:)*V1*2;
K1V(2,1) = K1Vn(2,:)*V2*2;
K1V(3,1) = K1Vn(3,:)*V3*2;
K1V(4,1) = K1Vn(4,:)*V4*2;

%Vector sum of all voltage pads
PD_K1V = K1V(1,1)-K1V(2,1)-K1V(3,1)+K1V(4,1);
PD_K1V = PD_K1V*1e12;  %changes units to uN*um/volt
end
```

M-File 20: K2Phi.m

```
function PD_K2Phi = K2Phi(volt)
% function PD_K2Phi = K2Phi(volt=[u, v])
%
% Clint Edwards, JHU-APL
% 27 December 2006
%
% K2Phi.m calculated the partial derivative of Tau1 or the Tz with
respect
% to the angle Phi
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% PD_K2Phi (uN*um/rad)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D

%actuating voltages defined
u = volt(1);
v = volt(2);
%bias voltage defined
Vb = param_2D(1);
% voltage amplifier gain
k = 1;

%pad voltages defined
V1 = Vb+k*(v-u);
V2 = Vb+k*(-v-u);
V3 = Vb+k*(-v+u);
V4 = Vb+k*(v+u);

%determine angle for given voltages
%if sum(abs([u v]))>6
    ang = ang_given_volt_2D_SD([u v]);
%else
%    ang = ang_given_volt_2D([u v]);
%end
phi = ang(1);
theta = ang(2);
```

```
%parametrized angles and differentials defined
A = -tan(phi);
C = tan(theta)*sec(phi);

%voltage pad lengths defined
x1 = param_2D(3);
x2 = param_2D(4);
z1 = param_2D(5);
z2 = param_2D(6);

%normalized partial derivatives for A and C returned for each
voltage pad
G21A = Gq(z1,z2,x1,x2,C,A)*V1^2;
G21C = Gp(z1,z2,x1,x2,C,A)*V1^2;
G22A = Gq(-z2,-z1,x1,x2,C,A)*V2^2;
G22C = Gp(-z2,-z1,x1,x2,C,A)*V2^2;
G23A = Gq(-z2,-z1,-x2,-x1,C,A)*V3^2;
G23C = Gp(-z2,-z1,-x2,-x1,C,A)*V3^2;
G24A = Gq(z1,z2,-x2,-x1,C,A)*V4^2;
G24C = Gp(z1,z2,-x2,-x1,C,A)*V4^2;

G21 = G(z1,z2,x1,x2,C,A)*V1^2;
G22 = G(-z2,-z1,x1,x2,C,A)*V2^2;
G23 = G(-z2,-z1,-x2,-x1,C,A)*V3^2;
G24 = G(z1,z2,-x2,-x1,C,A)*V4^2;

Gt1 = [G21A G22A  G23A  G24A]*sec(phi);
Gt2 = [G21C G22C G23C G24C]*-tan(theta)*tan(phi);
Gt3 = [G21 G22 G23 G24]*sin(phi);
Gt = [Gt1 Gt2 Gt3];

%Vector sum of all voltage pads
PD_K2Phi = sum(Gt);
PD_K2Phi = PD_K2Phi*1e12;  %changes units to uN*um/rad
```

M-File 21: K2Phi_num.m

```
function PD_K2Phi = K2Phi_num(volt)
% function PD_K2Phi = K2Phi_num(volt=[u, v]) NUMERICAL DERIVATIVE
%
% Clint Edwards, JHU-APL
% 31 December 2006
%
% K2Phi_NUM.m calculated the partial derivative of Tau1 or the Tz
with respect
% to the angle Phi
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% PD_K2Phi_num (uN*um/rad)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D

%actuating voltages defined
u = volt(1);
v = volt(2);

%bias voltage defined
Vb = param_2D(1);

% voltage amplifier gain
k = 1;

% numerical derivative step size
h=1e-9;

%steady-state angle for given voltage using Newton Rapston Method
ang = ang_given_volt_2D_SD([u v]);

phi=ang(1);
theta=ang(2);
t_num = (T2D([phi+h theta],[u v])-T2D([phi theta],[u v]))/h;
PD_K2Phi = t_num(2);
```

M-File 22: K2Theta.m

```
function PD_K2Theta = K2Theta(volt)
% function PD_K2Theta = K2Theta(volt=[u v])
%
% Clint Edwards, JHU-APL
% 27 December 2006
%
% K2Theta.m calculated the partial derivative of Tau2 or the Tz with
respect
% to the angle Theta
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% PD_K2Theta (uN*um/rad)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D

%actuating voltages defined
u = volt(1);
v = volt(2);
%bias voltage defined
Vb = param_2D(1);
% voltage amplifier gain
k = 1;

%pad voltages defined
V1 = Vb+k*(v-u);
V2 = Vb+k*(-v-u);
V3 = Vb+k*(-v+u);
V4 = Vb+k*(v+u);

%determine angle for given voltages
ang = ang_given_volt_2D_SD([u v]);
phi = ang(1);
theta = ang(2);

%parametrized angles and differentials defined
A = -tan(phi);
C = tan(theta)*sec(phi);
```

```
%voltage pad lengths defined
x1 = param_2D(3);
x2 = param_2D(4);
z1 = param_2D(5);
z2 = param_2D(6);

%normalized partial derivatives for A and C returned for each
voltage pad
G21C = Gp(z1,z2,x1,x2,C,A)*V1^2;
G22C = Gp(-z2,-z1,x1,x2,C,A)*V2^2;
G23C = Gp(-z2,-z1,-x2,-x1,C,A)*V3^2;
G24C = Gp(z1,z2,-x2,-x1,C,A)*V4^2;

Gt = [G21C G22C G23C G24C]*-sec(theta)^2;

%Vector sum of all voltage pads
PD_K2Theta = sum(Gt);
PD_K2Theta = PD_K2Theta*1e12;  %changes units to uN*um/rad
```

M-File 23: K2U.m

```
function PD_K2U = K2U(volt)
% function PD_K2U = K2U(volt=[u, v])
%
% Clint Edwards, JHU-APL
% 27 December 2006
%
% K2U.m calculated the partial derivative of Tau2 or Tx with respect
% to the acutating voltage u
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% PD_K2U (uN*um/volt)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D

%actuating voltages defined
u = volt(1);
v = volt(2);
%bias voltage defined
Vb = param_2D(1);
% voltage amplifier gain
k = 1;

%pad voltages defined
V1 = Vb+k*(v-u);
V2 = Vb+k*(-v-u);
V3 = Vb+k*(-v+u);
V4 = Vb+k*(v+u);

%determine angle for given voltages
ang = ang_given_volt_2D_SD([u v]);
phi = ang(1);
theta = ang(2);

%parametrized angles
A = -tan(phi);
C = tan(theta)*sec(phi);

%voltage pad lengths defined
x1 = param_2D(3);
x2 = param_2D(4);
```

```
z1 = param_2D(5);
z2 = param_2D(6);

%normalized partial derivatives for each voltage pad
K2Un = zeros(4,1);
K2Un(1,1) = G(z1,z2,x1,x2,C,A);
K2Un(2,1) = G(-z2,-z1,x1,x2,C,A);
K2Un(3,1) = G(-z2,-z1,-x2,-x1,C,A);
K2Un(4,1) = G(z1,z2,-x2,-x1,C,A);

%partial derivatives scaled by pad voltage
K2U = zeros(4,1);
K2U(1,1) = -2*cos(phi)*V1*K2Un(1,:);
K2U(2,1) = -2*cos(phi)*V2*K2Un(2,:);
K2U(3,1) = -2*cos(phi)*V3*K2Un(3,:);
K2U(4,1) = -2*cos(phi)*V4*K2Un(4,:);

%Vector sum of all voltage pads
PD_K2U = -K2U(1,1)-K2U(2,1)+K2U(3,1)+K2U(4,1);
PD_K2U = PD_K2U*1e12;  %changes units to uN*um/volt
end
```

M-File 24: K2V.m

```
function PD_K2V = K2V(volt)
% function PD_K2V = K2V(volt=[u, v])
%
% Clint Edwards, JHU-APL
% 27 December 2006
%
% K2V.m calculated the partial derivative of Tau2 or Tx with respect
% to the acutating voltage v
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% PD_K2V (uN*um/volt)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D

%actuating voltages defined
u = volt(1);
v = volt(2);
%bias voltage defined
Vb = param_2D(1);
% voltage amplifier gain
k = 1;

%pad voltages defined
V1 = Vb+k*(v-u);
V2 = Vb+k*(-v-u);
V3 = Vb+k*(-v+u);
V4 = Vb+k*(v+u);

%determine angle for given voltages
ang = ang_given_volt_2D_SD([u v]);
phi = ang(1);
theta = ang(2);

%parametrized angles
A = -tan(phi);
C = tan(theta)*sec(phi);

%voltage pad lengths defined
x1 = param_2D(3);
x2 = param_2D(4);
```

```
z1 = param_2D(5);
z2 = param_2D(6);

%normalized partial derivatives for each voltage pad
K2Vn = zeros(4,1);
K2Vn(1,1) = G(z1,z2,x1,x2,C,A);
K2Vn(2,1) = G(-z2,-z1,x1,x2,C,A);
K2Vn(3,1) = G(-z2,-z1,-x2,-x1,C,A);
K2Vn(4,1) = G(z1,z2,-x2,-x1,C,A);

%partial derivatives scaled by pad voltage
K2V = zeros(4,1);
K2V(1,1) = -2*cos(phi)*V1*K2Vn(1,:);
K2V(2,1) = -2*cos(phi)*V2*K2Vn(2,:);
K2V(3,1) = -2*cos(phi)*V3*K2Vn(3,:);
K2V(4,1) = -2*cos(phi)*V4*K2Vn(4,:);

%Vector sum of all voltage pads
PD_K2V = K2V(1,1)-K2V(2,1)-K2V(3,1)+K2V(4,1);
PD_K2V = PD_K2V*1e12;  %changes units to uN*um/volt
end
```

M-File 25: T2D.m

```
function T = T2D(ang,volt)
% function [Tau1 Tau2] = T2D(ang=[phi, theta],volt=[u,v])
%
% 2-D Torque Model
% Clint Edwards, JHU-APL
% 30 November 2006
%
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% Tau1: Tau_z torque about the z-axis (uN*um)
% Tau2: cos(phi)*Tau_x torque about the x-axis (uN*um)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D
u=volt(1);
v=volt(2);
phi=ang(1);
theta=ang(2);
Vb = param_2D(1);
k=1;            %electronics amp gain
A = -tan(phi);
C = tan(theta)*sec(phi);

% EACH PAD HAS AN ABSOLUTE VOLTAGE APPLIED
V1 = Vb+k*(v-u);
V2 = Vb+k*(-u-v);
V3 = Vb+k*(u-v);
V4 = Vb+k*(v+u);
V = [V1 V2 V3 V4];
Vsq = V.^2;

%Tz and Tx are 4x1 vector of torques for each pad
T1 = Tint_1(A,C);
T2 = Tint_2(A,C);

%Sum All Four Pads Vectorially
Tau_1 = Vsq*T1*1e12; %uN*um   (changes units of torque)
Tau_2 = Vsq*T2*1e12; %uN*um  (changes units of torque)
```

```
Tau1 = Tau_1;
Tau2 = cos(phi)*Tau_2;
T = [Tau1;Tau2];

end %(END OF PROGRAM)
```

M-File 26: Tint_1.m

```
function [T1]=Tint_1(A,C)
global param_2D
x1 = param_2D(3);
x2 = param_2D(4);
z1 = param_2D(5);
z2 = param_2D(6);
T1(1,1) = G(x1,x2,z1,z2,A,C);
T1(2,1) = G(x1,x2,-z2,-z1,A,C);
T1(3,1) = G(-x2,-x1,-z2,-z1,A,C);
T1(4,1) = G(-x2,-x1,z1,z2,A,C);
end %(function Tint_z)
```

M-File 27: Tint_2.m

```
function [T2]=Tint_2(A,C)
global param_2D
x1 = param_2D(3);
x2 = param_2D(4);
z1 = param_2D(5);
z2 = param_2D(6);
T2(1,1) = G(z1,z2,x1,x2,C,A);
T2(2,1) = G(-z2,-z1,x1,x2,C,A);
T2(3,1) = G(-z2,-z1,-x2,-x1,C,A);
T2(4,1) = G(z1,z2,-x2,-x1,C,A);
T2 = -1*T2;
end %(function Tint_x)
```

M-File 28: T2DQ1.m

```
function T = T2DQ1(ang,volt)
% function [Tau1 Tau2] = T2DQ1(ang=[phi, theta],volt=[u,v])
% torque for first quadrant voltage pad
% 2-D Torque Model
% Clint Edwards, JHU-APL
% 28 May 2006
%
% init_2D_file.m must be run to initialize mirror physical and
dynamic parameters
%
% INPUTS
% u: actuating voltage around bias causing tipping in phi (volts)
% v: actuating voltage around bias causing tiling in theta (volts)
% phi: tipping of mirror about the z-axis (angle measured from the
x-axis) (radians)
% theta: tilting of mirror about the x''-axis (angle measured from
the z'-axis)
%
% OUTPUTS
% Tau1: Tau_z torque about the z-axis (uN*um)
% Tau2: cos(phi)*Tau_x torque about the x-axis (uN*um)
%
% param_2D = [Vb,ho,x1,x2,z1,z2,Ix,Dx,Kmx,Iz,Dz,Kmz];
global param_2D
u=volt(1);
v=volt(2);
phi=ang(1);
theta=ang(2);
Vb = param_2D(1);
k=1;            %electronics amp gain
A = -tan(phi);
C = tan(theta)*sec(phi);

% EACH PAD HAS AN ABSOLUTE VOLTAGE APPLIED
% V1 = Vb+k*(-v-u);
% V2 = Vb+k*(-u+v);
% V3 = Vb+k*(u-v);
% V4 = Vb+k*(v+u);

V1 = Vb+k*(v-u);
% V2 = Vb+k*(-u-v);
% V3 = Vb+k*(u-v);
% V4 = Vb+k*(v+u);
V = [V1]; % V2 V3 V4];
Vsq = V.^2;

%Tz and Tx are 4x1 vector of torques for each pad
T1 = Tint_1(A,C);
T2 = Tint_2(A,C);
```

```
%Sum All Four Pads Vectorially
Tau_1 = Vsq*T1*1e12; %uN*um   (changes units of torque)
Tau_2 = Vsq*T2*1e12; %uN*um  (changes units of torque)

Tau1 = Tau_1;
Tau2 = cos(phi)*Tau_2;
T = [Tau1;Tau2];

end %(END OF PROGRAM)
```

M-File 29: PartDervVerAC.m

```
%PartDervVerAC.m
%This mfile verifies the theoretical development or the eight
partial
%derivatives that are required for the linearization of the 2D
Torque Model
syms A C ho z1 z2 x1 x2 V Eo

T1=((Eo*V^2)/(2*(A^2)*C))*((ho-C*z2)*(log(ho-A*x2-C*z2)-log(ho-A*x1-
C*z2))...
    -(ho-C*z1)*(log(ho-A*x2-C*z1)-log(ho-A*x1-C*z1)));

T2=-((cos(-atan(A))*Eo*V^2)/(2*A*C^2))*((ho-A*x2)*(log(ho-A*x2-
C*z2)-log(ho-A*x2-C*z1))-...
    (ho-A*x1)*(log(ho-A*x1-C*z2)-log(ho-A*x1-C*z1)));

K1A = diff(T1,'A');
K1C = diff(T1,'C');

K2A = diff(T2,'A');
K2C = diff(T2,'C');
```

M-File 30: PartDervVer.m

```
%PartDervVer.m
%This mfile verifies the theoretical development or the eight
partial
%derivatives that are required for the linearization of the 2D
Torque Model
syms phi theta ho z1 z2 x1 x2 V Eo

T1=((Eo*V^2)/(2*((-tan(phi))^2)*(tan(theta)*sec(phi))))*(((ho-
(tan(theta)...
    *sec(phi))*z2)*(log(ho+tan(phi)*x2-tan(theta)*sec(phi)*z2))-
log(ho+...
    tan(phi)*x1-tan(theta)*sec(phi)*z2))-((ho-
(tan(theta)*sec(phi))*z1)...
    *(log(ho+tan(phi)*x2-tan(theta)*sec(phi)*z1))-
log(ho+tan(phi)*x1-...
    tan(theta)*sec(phi)*z1)));

T2=-((Eo*V^2)/(2*(-
tan(phi))*(tan(theta)*sec(phi))^2))*(((ho+tan(phi)*x2)...
    *(log(ho+tan(phi)*x2-tan(theta)*sec(phi)*z2))-
log(ho+tan(phi)*x2-tan(theta)...
    *sec(phi)*z1))-((ho+tan(phi)*x1)*(log(ho+tan(phi)*x1-
tan(theta)*sec(phi)*z2))...
    -log(ho+tan(phi)*x1-tan(theta)*sec(phi)*z1)));

K1theta = diff(T1,'theta');
K1phi = diff(T1,'phi');

K2theta = diff(T2,'theta');
K2phi = diff(T2,'phi');
```

M-File 31: ang_given_volt_SD.m

```
function ANGLE = ang_given_volt_2D_SD(volt)
% function ANGLE = ang_given_volt_2D_NR(volt=[u,v])
%
% physical and dynamic parameters for umirror
% param_2D = [Vb,ho,x1,x2,z1,z2,I2,D2,Km2,I1,D1,Km1];
global param_2D
% u = volt(1);
% v = volt(2);
% phi=ang(1);
% theta=ang(2);

Vb = param_2D(1);
ho = param_2D(2);
Lx = 2*param_2D(4);
Lz = 2*param_2D(6);
I2 = param_2D(7);
D2 = param_2D(8);
Km2 = param_2D(9);
I1 = param_2D(10);
D1 = param_2D(11);
Km1 = param_2D(12);

MAX = 1e3;
%steady-state phi calculation
    ANGn=[.1;.1]*pi/180; %first guess at angles phi and theta
    h=1e-10; % step size
    err=1;
    counter = 1;

    while abs(err)>1e-6&counter<MAX
        c1 = [Km1;0]-(T2D([ANGn(1)+h, ANGn(2)],volt)-T2D([ANGn(1),
ANGn(2)],volt))/h;
        c2 = [0;Km2]-(T2D([ANGn(1), ANGn(2)+h],volt)-T2D([ANGn(1),
ANGn(2)],volt))/h;
        J = [c1,c2];
        if any(any(isnan(J)))
           ANGn=[NaN;NaN];
           break
        end
        Jinv = inv(J);
        T = T2D(ANGn,volt);
        F = ([Km1;Km2].*ANGn)-T;
        ANGn = ANGn-Jinv*F;
        err = max(abs(([Km1;Km2].*ANGn)-T));
        counter = counter+1;
%         if isnan(c1)|isnan(c2)
%             break
%         end
    end
```

```
if(counter==MAX)
    disp(['limited convergence for ',num2str(volt),' volts'])
end
 ANGLE = ANGn;
```

M-File 32: equil_2D.m

```
% equil_2D.m

u_array = [-20:.5:20];
v_array = [-20:.5:20];
angles =zeros(length(u_array),length(v_array),2);

for ii = 1:length(u_array)
    for jj = 1:length(v_array)
        angles(ii,jj,:) = ang_given_volt_2D_SD([u_array(ii)
v_array(jj)]);
    end
end

figure
surf(v_array,u_array,angles(:,:,1))
save angles_uv
```

M-File 33: Findz.m

```
function Fz = Findz(ANGn)
% function ANGLE = ang_given_volt_2D(volt=[u,v])
%
% physical and dynamic parameters for umirror
% param_2D = [Vb,ho,x1,x2,z1,z2,I2,D2,Km2,I1,D1,Km1];
global param_2D
phi=ANGn(1);
theta = ANGn(2);

u = 10;
v = 0;
volt = [u,v];
Km2 = param_2D(9);
Km1 = param_2D(12);

T = T2D(ANGn,volt);
Fz = max(abs(([Km1;Km2].*ANGn)-T));
```

M-File 34: PD_K1Phi_excel.m

```
%PD_K1Phi_excel.m

u_array = [-18:2:18];
v_array = [-18:2:18];
PD_K1Phi_anal = zeros(length(u_array),length(v_array));
PD_K1Phi_num = zeros(length(u_array),length(v_array));
h=1e-9;

for ii=1:length(u_array)
    for jj=1:length(v_array)
        PD_K1Phi_anal(ii,jj)=K1Phi([u_array(ii),v_array(jj)]);
        ang = ang_given_volt_2D_SD([u_array(ii) v_array(jj)]);

        phi=ang(1);
        theta=ang(2);
        t_num = (T2D([phi+h theta],[u_array(ii) v_array(jj)])-
T2D([phi theta],[u_array(ii) v_array(jj)]))/h;
        PD_K1Phi_num(ii,jj) = t_num(1);
    end
end

xlswrite('PD_K1Phi_anal.xls',PD_K1Phi_anal)
xlswrite('PD_K1Phi_num.xls',PD_K1Phi_num)
```

M-File 35: PD_K1Theta_excel.m

```
%PD_K1Theta_excel.m

u_array = [-18:2:18];
v_array = [-18:2:18];
PD_K1Theta_anal = zeros(length(u_array),length(v_array));
PD_K1Theta_num = zeros(length(u_array),length(v_array));
h=1e-9;

for ii=1:length(u_array)
    for jj=1:length(v_array)
        PD_K1Theta_anal(ii,jj)=K1Theta([u_array(ii),v_array(jj)]);
        ang = ang_given_volt_2D_SD([u_array(ii) v_array(jj)]);

        phi=ang(1);
        theta=ang(2);
        t_num = (T2D([phi theta+h],[u_array(ii) v_array(jj)])-
T2D([phi theta],[u_array(ii) v_array(jj)]))/h;
        PD_K1Theta_num(ii,jj) = t_num(1);
    end
end

xlswrite('PD_K1Theta_anal.xls',PD_K1Theta_anal)
xlswrite('PD_K1Theta_num.xls',PD_K1Theta_num)
```

M-File 36: PD_K1U_excel.m

```
%PD_K1U_excel.m

u_array = [-18:2:18];
v_array = [-18:2:18];
PD_K1U_anal = zeros(length(u_array),length(v_array));
PD_K1U_num = zeros(length(u_array),length(v_array));
h=1e-9;

for ii=1:length(u_array)
    for jj=1:length(v_array)
        PD_K1U_anal(ii,jj)=K1U([u_array(ii),v_array(jj)]);
        if sum(abs([u_array(ii) v_array(jj)]))>6
            ang = ang_given_volt_2D_SD([u_array(ii) v_array(jj)]);
        else
            ang = ang_given_volt_2D([u_array(ii) v_array(jj)]);
        end
        phi=ang(1);
        theta=ang(2);
        t_num = (T2D([phi theta],[u_array(ii)+h v_array(jj)])-
T2D([phi theta],[u_array(ii) v_array(jj)]))/h;
        PD_K1U_num(ii,jj) = t_num(1);
    end
end

xlswrite('PD_K1U_anal.xls',PD_K1U_anal)
xlswrite('PD_K1U_num.xls',PD_K1U_num)
```

M-File 37: PD_K1U_excel.m

```
%PD_K1U_excel.m

u_array = [-18:2:18];
v_array = [-18:2:18];
PD_K1U_plt = zeros(length(u_array),length(v_array));
PD_K1U_num = zeros(length(u_array),length(v_array));
h=1e-9;

for ii=1:length(u_array)
    for jj=1:length(v_array)
        PD_K1U_plt(ii,jj)=K1U([u_array(ii),v_array(jj)]);
        ang = ang_given_volt_2D_SD([u_array(ii) v_array(jj)]);
        phi=ang(1);
        theta=ang(2);
        temp = (T2D([phi theta],[u_array(ii)+h v_array(jj)])-
T2D([phi theta],[u_array(ii) v_array(jj)]))/h;
        PD_K1U_num(ii,jj) = temp(1);
    end
end

xlswrite('PD_K1U_plot.xls',PD_K1U_plt)
xlswrite('PD_K1U_num.xls',PD_K1U_num)
```

M-File 38: PD_K1V_excel.m

```
%PD_K1V_excel.m

u_array = [-18:2:18];
v_array = [-18:2:18];
PD_K1V_anal = zeros(length(u_array),length(v_array));
PD_K1V_num = zeros(length(u_array),length(v_array));
h=1e-9;

for ii=1:length(u_array)
    for jj=1:length(v_array)
        PD_K1V_anal(ii,jj)=K1V([u_array(ii),v_array(jj)]);
        ang = ang_given_volt_2D_SD([u_array(ii) v_array(jj)]);
        phi=ang(1);
        theta=ang(2);
        t_num = (T2D([phi theta],[u_array(ii) v_array(jj)+h])-
T2D([phi theta],[u_array(ii) v_array(jj)]))/h;
        PD_K1V_num(ii,jj) = t_num(1);
    end
end

xlswrite('PD_K1V_anal.xls',PD_K1V_anal)
xlswrite('PD_K1V_num.xls',PD_K1V_num)
```

M-File 39: PD_K2Phi_excel.m

```
%PD_K2Phi_excel.m

u_array = [-18:2:18];
v_array = [-18:2:18];
PD_K2Phi_anal = zeros(length(u_array),length(v_array));
PD_K2Phi_num = zeros(length(u_array),length(v_array));
h=1e-6;

for ii=1:length(u_array)
    for jj=1:length(v_array)
        PD_K2Phi_anal(ii,jj)=K2Phi([u_array(ii),v_array(jj)]);
        ang = ang_given_volt_2D_SD([u_array(ii) v_array(jj)]);
        phi=ang(1);
        theta=ang(2);
        t_num = (T2D([phi+h theta],[u_array(ii) v_array(jj)])-
T2D([phi theta],[u_array(ii) v_array(jj)]))/h;
        PD_K2Phi_num(ii,jj) = t_num(2);
    end
end

xlswrite('PD_K2Phi_anal.xls',PD_K2Phi_anal)
xlswrite('PD_K2Phi_num.xls',PD_K2Phi_num)
```

M-File 40: PD_K2Theta_excel.m

```
%PD_K2Theta_excel.m

u_array = [-18:2:18];
v_array = [-18:2:18];
PD_K2Theta_anal = zeros(length(u_array),length(v_array));
PD_K2Theta_num = zeros(length(u_array),length(v_array));
h=1e-9;

for ii=1:length(u_array)
    for jj=1:length(v_array)
        PD_K2Theta_anal(ii,jj)=K2Theta([u_array(ii),v_array(jj)]);
        ang = ang_given_volt_2D_SD([u_array(ii) v_array(jj)]);

        phi=ang(1);
        theta=ang(2);
        t_num = (T2D([phi theta+h],[u_array(ii) v_array(jj)])-
T2D([phi theta],[u_array(ii) v_array(jj)]))/h;
        PD_K2Theta_num(ii,jj) = t_num(2);
    end
end

xlswrite('PD_K2Theta_anal.xls',PD_K2Theta_anal)
xlswrite('PD_K2Theta_num.xls',PD_K2Theta_num)
```

M-File 41: PD_K2U_excel.m

```
%PD_K2U_excel.m

u_array = [-18:2:18];
v_array = [-18:2:18];
PD_K2U_anal = zeros(length(u_array),length(v_array));
PD_K2U_num = zeros(length(u_array),length(v_array));
h=1e-9;

for ii=1:length(u_array)
    for jj=1:length(v_array)
        PD_K2U_anal(ii,jj)=K2U([u_array(ii),v_array(jj)]);
        ang = ang_given_volt_2D_old([u_array(ii) v_array(jj)]);
        phi=ang(1);
        theta=ang(2);
        t_num = (T2D([phi theta],[u_array(ii)+h v_array(jj)])-
T2D([phi theta],[u_array(ii) v_array(jj)]))/h;
        PD_K2U_num(ii,jj) = t_num(2);
    end
end

xlswrite('PD_K2U_anal.xls',PD_K2U_anal)
xlswrite('PD_K2U_num.xls',PD_K2U_num)
```

M-File 42: PD_K2V_excel.m

```
%PD_K2V_excel.m

u_array = [-18:2:18];
v_array = [-18:2:18];
PD_K2V_anal = zeros(length(u_array),length(v_array));
PD_K2V_num = zeros(length(u_array),length(v_array));
h=1e-9;

for ii=1:length(u_array)
    for jj=1:length(v_array)
        PD_K2V_anal(ii,jj)=K2V([u_array(ii),v_array(jj)]);
        ang = ang_given_volt_2D_SD([u_array(ii) v_array(jj)]);
        phi=ang(1);
        theta=ang(2);
        t_num = (T2D([phi theta],[u_array(ii) v_array(jj)+h])-
T2D([phi theta],[u_array(ii) v_array(jj)]))/h;
        PD_K2V_num(ii,jj) = t_num(2);
    end
end

xlswrite('PD_K2V_anal.xls',PD_K2V_anal)
xlswrite('PD_K2V_num.xls',PD_K2V_num)
```

M-File 43: Cross_PSD.m

```
function OUTPUT = Cross_PSD(SF,N,X,Y)
% function OUTPUT = Cross_PSD(SF,N,temp)
% Clint Edwards, JHU-APL
% 16 June 2007

len = length(X);
T = 1/SF;  % sampling period for time-series (ts) data
win = hanning(N);
X = X-mean(X);
Y = Y-mean(Y);
shift = ceil(N/5);
NN = floor(length(X)/shift);
fft_temp = zeros(NN,N);
index=[0:1:length(X)-1]*T;  %time index
fft_index = [-N/2+1:1:(N/2)]*(1/(N*T));  %freq index

for ii=1:NN-floor(N/shift)
    if (ii-1)*shift<=length(X)
        X_t=win.*X((ii-1)*shift+1:N+(ii-1)*shift);
        Y_t=win.*Y((ii-1)*shift+1:N+(ii-1)*shift);
    else
        X_t=win.*X((ii-1)*shift+1:end);
        Y_t=win.*Y((ii-1)*shift+1:end);
    end
    [Rxy,lags] = xcorr(X,Y,'biased');
    fft_temp(ii,:) = fftshift(T*fft((Rxy-mean(Rxy)),N));
end
nf=1;
Sxy = sum(abs(fft_temp).^2)*(1/NN);
Sxy=filter((1/nf)*ones(nf,1),1,Sxy);

OUTPUT = [fft_index', Sxy'];
```

M-File 44: Lin2D.m

```
% Lin2D.m
% 23 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for the inputs.

init_2D_file
Vb = param_2D(1);
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);

u = volt_noise.signals.values(:,1);  %time-series voltage u from
simulink
v = volt_noise.signals.values(:,2);  %time-series voltage v from
simulink
SimOutPhi = SimOut.signals.values(:,1); %time-series angle phi from
simulink
SimOutTheta = SimOut.signals.values(:,2); % time-series angle theta
from simulink
VibPhi = vib_noise.signals.values(:,1);  %time-series vibration
noise from simulink
VibTheta = vib_noise.signals.values(:,2);  %time-series vibration
noise simulink
BMPhi = bm_noise.signals.values(:,1);  %time-series vibration noise
from simulink
BMTheta = bm_noise.signals.values(:,2);  %time-series vibration
noise simulink

pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
u0 = mean(u(end-pt,end));   % deterministic u (average value)
v0 = mean(v(end-pt,end));   % deterministic v (average value)
volt0 = [u0; v0];

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
```

```
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v


fft_len = 5000;  %Length of FFTs in PSD
OUTPUTu = Standard_PSD(1/ST,fft_len,u);
OUTPUTv = Standard_PSD(1/ST,fft_len,v);
OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
OUTPUTuv = Cross_PSD(1/ST,fft_len,u,v);
OUTPUTvibphi = Standard_PSD(1/ST,fft_len,VibPhi);
OUTPUTvibtheta = Standard_PSD(1/ST,fft_len,VibTheta);
OUTPUTvibphitheta = Cross_PSD(1/ST,fft_len,VibPhi,VibTheta);
OUTPUTvibthetaphi = Cross_PSD(1/ST,fft_len,VibTheta,VibPhi);
OUTPUTphiBM = Standard_PSD(1/ST,fft_len,BMPhi);
OUTPUTthetaBM = Standard_PSD(1/ST,fft_len,BMTheta);
OUTPUTphithetaBM = Cross_PSD(1/ST,fft_len,BMPhi,BMTheta);
OUTPUTthetaphiBM = Cross_PSD(1/ST,fft_len,BMTheta,BMPhi);
freq = OUTPUTu(:,1);

Su = OUTPUTu(:,2);  %Su(jw) PSD
Sv = OUTPUTv(:,2);  %Sv(jw) PSD
Sphi = OUTPUTphi(:,2);  %Sphi(jw) PSD
Stheta = OUTPUTtheta(:,2);  %Stheta(jw) PSD
Suv = OUTPUTuv(:,2);  %Suv(jw) Cross PSD
Svu = Suv; % real valued sequence
Svibphi = OUTPUTvibphi(:,2);
Svibtheta = OUTPUTvibtheta(:,2);
Svibphitheta = OUTPUTvibphitheta(:,2);
Svibthetaphi = OUTPUTvibphitheta(:,2);
SphiBM = OUTPUTphiBM(:,2);
SthetaBM = OUTPUTthetaBM(:,2);
SphithetaBM = OUTPUTphithetaBM(:,2);
SthetaphiBM = OUTPUTphithetaBM(:,2);

w0 = 2*pi*1610;
w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;
P10 = -w0.^2*I1+j*w0*D1+Km1-K1phi0;
P20 = -w0.^2*I2+j*w0*D2+Km2-K2theta0;
Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)
figure
plot(freq,Hjw2)

a1 = (abs(P2).^2)*(Km1^2);
a2 = (P2*K1theta0*Km1*Km2);
a3 = (conj(P2)*K1theta0*Km1*Km2);
a4 = (K1theta0^2)*(Km2^2);
```

```
a5 =
abs((K1theta0^2)*(K2u0^2)+P2*K1theta0*K1u0*K2u0+conj(P2)*K1theta0*K1
u0*K2u0+(abs(P2).^2)*(K1u0^2));
a6 =
abs((K1theta0^2)*K2u0*K2v0+P2*K1theta0*K1u0*K2v0+P2*K1theta0*K1v0*K2
u0+(abs(P2).^2)*K1u0*K1v0);
a7 =
abs((K1theta0^2)*K2u0*K2v0+conj(P2)*K1theta0*K1u0*K2v0+conj(P2)*K1th
eta0*K1v0*K2u0+(abs(P2).^2)*K1u0*K1v0);
a8 =
abs((K1theta0^2)*(K2v0^2)+(P2*K1theta0*K1v0*K2v0)+(conj(P2)*K1theta0
*K1v0*K2v0)+(abs(P2).^2)*(K1v0^2));

a10 = (abs(P20).^2)*(Km1^2)
a20 = abs((P20*K1theta0*Km1*Km2))
a30 = abs((conj(P20)*K1theta0*Km1*Km2))
a40 = (K1theta0^2)*(Km2^2)
a50 =
abs((K1theta0^2)*(K2u0^2)+P20*K1theta0*K1u0*K2u0+conj(P20)*K1theta0*
K1u0*K2u0+(abs(P20).^2)*(K1u0^2))
a60 =
abs((K1theta0^2)*K2u0*K2v0+P20*K1theta0*K1u0*K2v0+P20*K1theta0*K1v0*
K2u0+(abs(P20).^2)*K1u0*K1v0)
a70 =
abs((K1theta0^2)*K2u0*K2v0+conj(P20)*K1theta0*K1u0*K2v0+conj(P20)*K1
theta0*K1v0*K2u0+(abs(P20).^2)*K1u0*K1v0)
a80 =
abs((K1theta0^2)*(K2v0^2)+(P20*K1theta0*K1v0*K2v0)+(conj(P20)*K1thet
a0*K1v0*K2v0)+(abs(P20).^2)*(K1v0^2))

w1_SthetaBM = (K1theta0^2)*(Km2^2);
w1_SphiBM = (abs(P2).^2)*(Km1^2); ;
w1_SphithetaBM = 2*K1theta0*real(P2.*(Km1).*(Km2).*SphithetaBM);
%SphithetaBM
S11BM =
w1_SthetaBM.*Hjw2.*SthetaBM+w1_SphiBM.*Hjw2.*SphiBM+w1_SphithetaBM.*
Hjw2.*SphithetaBM;

w2_Svv =
abs((K2phi0^2)*(K1v0^2)+P1*K2phi0*K2v0*K1v0+conj(P1)*K2phi0*K2v0*K1v
0+(abs(P1).^2)*(K2v0^2));
w2_Suu =
abs((K2phi0^2)*(K1u0^2)+(P1*K2phi0*K2u0*K1u0)+(conj(P1)*K2phi0*K2u0*
K1u0)+(abs(P1).^2)*(K2u0^2));
w2_Svu =
abs(2*(K2phi0^2)*K1v0*K1u0+P1*K2phi0*K2v0*K1u0+P1*K2phi0*K2u0*K1v0+c
onj(P1)*K2phi0*K2v0*K1u0...
   +K2u0*(conj(P1)*K2phi0*K1v0+2*(abs(P1).^2)*K2v0));
w2_Svibphi = (K2phi0^2)*(Km1^2);
w2_Svibtheta = (abs(P1).^2)*(Km2^2);
w2_Svibthetaphi = (P1*K2phi0*Km1*Km2);
w2_Svibthetaphic = (conj(P1)*K2phi0*Km1*Km2);
```

```
w2_SphiBM = (K2phi0^2)*(Km1^2);
w2_SthetaBM = (abs(P1).^2)*(Km2^2); ;
w2_SthetaphiBM = 2*K2phi0*real(P1.*(Km2).*(Km1).*SthetaphiBM);
S22BM =
w2_SphiBM.*Hjw2.*SphiBM+w2_SthetaBM.*Hjw2.*SthetaBM+w2_SthetaphiBM.*
Hjw2;

b1 = (K2phi0^2)*(Km1^2);
b2 = (conj(P1)*K2phi0*Km2*Km1);
b3 = (P1*K2phi0*Km2*Km1);
b4 = (abs(P1).^2)*(Km2^2);
b5 =
abs((K2phi0^2)*(K1u0^2)+(P1*K2phi0*K2u0*K1u0)+(conj(P1)*K2phi0*K2u0*
K1u0)+(abs(P1).^2)*(K2u0^2));
b6 =
abs((K2phi0^2)*K1v0*K1u0+conj(P1)*K2phi0*K2v0*K1u0+conj(P1)*K2phi0*K
2u0*K1v0+(abs(P1).^2)*K2v0*K2u0);
b7 =
abs((K2phi0^2)*K1v0*K1u0+P1*K2phi0*K2v0*K1u0+P1*K2phi0*K2u0*K1v0+(ab
s(P1).^2)*K2v0*K2u0);
b8 =
abs((K2phi0^2)*(K1v0^2)+P1*K2phi0*K2v0*K1v0+conj(P1)*K2phi0*K2v0*K1v
0+(abs(P1).^2)*(K2v0^2));

b10 = (K2phi0^2)*(Km1^2);
b20 = abs((conj(P10)*K2phi0*Km2*Km1));
b30 = abs((P10*K2phi0*Km2*Km1));
b40 = (abs(P10).^2)*(Km2^2);
b50 =
abs((K2phi0^2)*(K1u0^2)+(P10*K2phi0*K2u0*K1u0)+(conj(P10)*K2phi0*K2u
0*K1u0)+(abs(P10).^2)*(K2u0^2));
b60 =
abs((K2phi0^2)*K1v0*K1u0+conj(P10)*K2phi0*K2v0*K1u0+conj(P10)*K2phi0
*K2u0*K1v0+(abs(P10).^2)*K2v0*K2u0);
b70 =
abs((K2phi0^2)*K1v0*K1u0+P10*K2phi0*K2v0*K1u0+P10*K2phi0*K2u0*K1v0+(
abs(P10).^2)*K2v0*K2u0);
b80 =
abs((K2phi0^2)*(K1v0^2)+P10*K2phi0*K2v0*K1v0+conj(P10)*K2phi0*K2v0*K
1v0+(abs(P10).^2)*(K2v0^2));

Sphi_est =
a1.*Hjw2.*Svibphi+a2.*Hjw2.*Svibphitheta+a3.*Hjw2.*Svibthetaphi...

+a4.*Hjw2.*Svibtheta+a5.*Hjw2.*Su+a6.*Hjw2.*Suv+a7.*Hjw2.*Svu+a8.*Hj
w2.*Sv...
    +S11BM;

Sphi_volt_est =
a5.*Hjw2.*Su+a6.*Hjw2.*Suv+a7.*Hjw2.*Svu+a8.*Hjw2.*Sv;
Sphi_bm_est = S11BM;
```

```
Sphi_vib_est =
a1.*Hjw2.*Svibphi+a2.*Hjw2.*Svibphitheta+a3.*Hjw2.*Svibthetaphi...
    +a4.*Hjw2.*Svibtheta;

Stheta_est =
b1.*Hjw2.*Svibphi+b2.*Hjw2.*Svibphitheta+b3.*Hjw2.*Svibthetaphi...

+b4.*Hjw2.*Svibtheta+b5.*Hjw2.*Su+b6.*Hjw2.*Suv+b7.*Hjw2.*Svu+b8.*Hj
w2.*Sv...
    +S22BM;

Stheta_volt_est =
b5.*Hjw2.*Su+b6.*Hjw2.*Suv+b7.*Hjw2.*Svu+b8.*Hjw2.*Sv;
Stheta_bm_est = S22BM;
Stheta_vib_est =
b1.*Hjw2.*Svibphi+b2.*Hjw2.*Svibphitheta+b3.*Hjw2.*Svibthetaphi...
    +b4.*Hjw2.*Svibtheta;

% figure; plot(freq,Sphi,'linewidth',2); hold on;
% plot(freq,Sphi_est,'r','linewidth',2)
% plot(freq,Sphi_volt_est,'k','linewidth',2)
% plot(freq,Sphi_bm_est,'g','linewidth',2)
% plot(freq,Sphi_vib_est,'m','linewidth',2)
% xlabel('Frequency (Hz)')
% ylabel('PSD \phi (W/Hz)')
% title('\phi Power Spectral Density')
% axis([0 5000 0 3e-17])
%
% figure; plot(freq,Stheta,'linewidth',2); hold on;
% plot(freq,Stheta_est,'r','linewidth',2)
% plot(freq,Stheta_volt_est,'k','linewidth',2)
% plot(freq,Stheta_bm_est,'g','linewidth',2)
% plot(freq,Stheta_vib_est,'m','linewidth',2)
% xlabel('Frequency (Hz)')
% ylabel('PSD \theta (W/Hz)')
% title('\theta Power Spectral Density')
% axis([0 5000 0 2e-17])
```

M-File 45: Lin2D_BM.m

```
% Lin2D_BM.m
% 15 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for only the voltage and vibration inputs.

init_2D_file
% param_2D = [Vb,ho,x1,x2,z1,z2,I1,D1,Km1,I2,D2,Km2];
Vb = param_2D(1);
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);

u = volt_noise.signals.values(:,1);  %time-series voltage u from
simulink
v = volt_noise.signals.values(:,2);  %time-series voltage v from
simulink
BMPhi = bm_noise.signals.values(:,1);  %time-series vibration noise
from simulink
BMTheta = bm_noise.signals.values(:,2);  %time-series vibration
noise simulink
SimOutPhi = SimOut.signals.values(:,1); %time-series angle phi from
simulink
SimOutTheta = SimOut.signals.values(:,2); % time-series angle theta
from simulink
pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
u0 = mean(u(end-pt,end));   % deterministic u (average value)
v0 = mean(v(end-pt,end));   % deterministic v (average value)
% BMPhi = BMPhi-mean(BMPhi);  %should already be zero mean
% BMTheta = BMTheta-mean(BMTheta);  %should already be zero mean
volt0 = [u0; v0];

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v
```

```
fft_len = 10000;  %Length of FFTs in PSD
OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
OUTPUTphiBM = Standard_PSD(1/ST,fft_len,BMPhi);
OUTPUTthetaBM = Standard_PSD(1/ST,fft_len,BMTheta);
OUTPUTphithetaBM = Cross_PSD(1/ST,fft_len,BMPhi,BMTheta);
OUTPUTthetaphiBM = Cross_PSD(1/ST,fft_len,BMTheta,BMPhi);

freq = OUTPUTphi(:,1);
Sphi = OUTPUTphi(:,2);  % Sphi(jw) PSD
Stheta = OUTPUTtheta(:,2);  % Stheta(jw) PSD
SphiBM = OUTPUTphiBM(:,2);
SthetaBM = OUTPUTthetaBM(:,2);
SphithetaBM = OUTPUTphithetaBM(:,2);
SthetaphiBM = OUTPUTphithetaBM(:,2);

w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;

Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)
w1_SthetaBM = (K1theta0^2)*(Km2^2);
w1_SphiBM = (abs(P2).^2)*(Km1^2); ;
w1_SphithetaBM = 2*K1theta0*real(P2.*(Km1).*(Km2).*SphithetaBM);
%SphithetaBM

w2_SphiBM = (K2phi0^2)*(Km1^2);
w2_SthetaBM = (abs(P1).^2)*(Km2^2); ;
w2_SthetaphiBM = 2*K2phi0*real(P1.*(Km2).*(Km1).*SthetaphiBM);

Sphi_est =
w1_SthetaBM.*Hjw2.*SthetaBM+w1_SphiBM.*Hjw2.*SphiBM+w1_SphithetaBM.*
Hjw2*1 ;
Stheta_est =
w2_SphiBM.*Hjw2.*SphiBM+w2_SthetaBM.*Hjw2.*SthetaBM+w2_SthetaphiBM.*
Hjw2*1 ;

figure; plot(freq,Sphi,'linewidth',1); hold on;
plot(freq,Sphi_est,'r','linewidth',1)
size(freq)
xlabel('Frequency (Hz)')
ylabel('PSD \phi (W/Hz)')
title('\phi Power Spectral Density')
axis([-4000 4000 0 1e-11])


figure; plot(freq,Stheta,'m','linewidth',2); hold on;
plot(freq,Stheta_est,'g','linewidth',2)
% plot(freq,Stheta_est1,'k','linewidth',1)
```

```
% plot(freq,Stheta_est2,'m','linewidth',1)
% plot(freq,Stheta_est3,'g','linewidth',1)
xlabel('Frequency (Hz)')
ylabel('PSD \theta (W/Hz)')
title('\theta Power Spectral Density')
axis([-4000 4000 0 1e-11])
```

M-File 46: Lin2D_plots.m

```
% Lin2D_plots.m
% 28 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for the inputs.

init_2D_file
Vb = param_2D(1);
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);

u = volt_noise.signals.values(:,1);  %time-series voltage u from
simulink
v = volt_noise.signals.values(:,2);  %time-series voltage v from
simulink
SimOutPhi = SimOut.signals.values(:,1); %time-series angle phi from
simulink
SimOutTheta = SimOut.signals.values(:,2); % time-series angle theta
from simulink
VibPhi = vib_noise.signals.values(:,1);  %time-series vibration
noise from simulink
VibTheta = vib_noise.signals.values(:,2);  %time-series vibration
noise simulink
BMPhi = bm_noise.signals.values(:,1);  %time-series vibration noise
from simulink
BMTheta = bm_noise.signals.values(:,2);  %time-series vibration
noise simulink

pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
u0 = mean(u(end-pt,end));   % deterministic u (average value)
v0 = mean(v(end-pt,end));   % deterministic v (average value)
volt0 = [u0; v0];

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
```

```
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v


fft_len = 10000;  %Length of FFTs in PSD
OUTPUTu = Standard_PSD(1/ST,fft_len,u);
OUTPUTv = Standard_PSD(1/ST,fft_len,v);
OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
OUTPUTuv = Cross_PSD(1/ST,fft_len,u,v);
OUTPUTvibphi = Standard_PSD(1/ST,fft_len,VibPhi);
OUTPUTvibtheta = Standard_PSD(1/ST,fft_len,VibTheta);
OUTPUTvibphitheta = Cross_PSD(1/ST,fft_len,VibPhi,VibTheta);
OUTPUTvibthetaphi = Cross_PSD(1/ST,fft_len,VibTheta,VibPhi);
OUTPUTphiBM = Standard_PSD(1/ST,fft_len,BMPhi);
OUTPUTthetaBM = Standard_PSD(1/ST,fft_len,BMTheta);
OUTPUTphithetaBM = Cross_PSD(1/ST,fft_len,BMPhi,BMTheta);
OUTPUTthetaphiBM = Cross_PSD(1/ST,fft_len,BMTheta,BMPhi);
freq = OUTPUTu(:,1);

Su = OUTPUTu(:,2);  %Su(jw) PSD
Sv = OUTPUTv(:,2);  %Sv(jw) PSD
Sphi = OUTPUTphi(:,2);  %Sphi(jw) PSD
Stheta = OUTPUTtheta(:,2);  %Stheta(jw) PSD
Suv = OUTPUTuv(:,2);  %Suv(jw) Cross PSD
Svu = Suv; % real valued sequence
Svibphi = OUTPUTvibphi(:,2);
Svibtheta = OUTPUTvibtheta(:,2);
Svibphitheta = OUTPUTvibphitheta(:,2);
Svibthetaphi = OUTPUTvibphitheta(:,2);
SphiBM = OUTPUTphiBM(:,2);
SthetaBM = OUTPUTthetaBM(:,2);
SphithetaBM = OUTPUTphithetaBM(:,2);
SthetaphiBM = OUTPUTphithetaBM(:,2);


w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;
Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)

w1_Suu =
abs((K1theta0^2)*(K2u0^2)+P2*K1theta0*K1u0*K2u0+conj(P2)*K1theta0*K1
u0*K2u0+(abs(P2).^2)*(K1u0^2));
w1_Svv =
abs((K1theta0^2)*(K2v0^2)+(P2*K1theta0*K1v0*K2v0)+(conj(P2)*K1theta0
*K1v0*K2v0)+(abs(P2).^2)*(K1v0^2));
w1_Suv =
abs(2*(K1theta0^2)*K2u0*K2v0+P2*K1theta0*K1u0*K2v0+P2*K1theta0*K1v0*
K2u0+conj(P2)*K1theta0*K1u0*K2v0...
   +K1v0*(conj(P2)*K1theta0*K2u0+2*(abs(P2).^2)*K1u0));
```

```
w1_Svibtheta = (K1theta0^2)*(Km2^2);
w1_Svibphi = (abs(P2).^2)*(Km1^2);
w1_Svibphitheta = (P2*K1theta0*Km1*Km2);
w1_Svibphithetac = (conj(P2)*K1theta0*Km1*Km2);
w1_SthetaBM = (K1theta0^2)*(Km2^2);
w1_SphiBM = (abs(P2).^2)*(Km1^2); ;
w1_SphithetaBM = 2*K1theta0*real(P2.*(Km1).*(Km2).*SphithetaBM);
%SphithetaBM

w2_Svv =
abs((K2phi0^2)*(K1v0^2)+P1*K2phi0*K2v0*K1v0+conj(P1)*K2phi0*K2v0*K1v
0+(abs(P1).^2)*(K2v0^2));
w2_Suu =
abs((K2phi0^2)*(K1u0^2)+(P1*K2phi0*K2u0*K1u0)+(conj(P1)*K2phi0*K2u0*
K1u0)+(abs(P1).^2)*(K2u0^2));
w2_Svu =
abs(2*(K2phi0^2)*K1v0*K1u0+P1*K2phi0*K2v0*K1u0+P1*K2phi0*K2u0*K1v0+c
onj(P1)*K2phi0*K2v0*K1u0...
   +K2u0*(conj(P1)*K2phi0*K1v0+2*(abs(P1).^2)*K2v0));
w2_Svibphi = (K2phi0^2)*(Km1^2);
w2_Svibtheta = (abs(P1).^2)*(Km2^2);
w2_Svibthetaphi = (P1*K2phi0*Km1*Km2);
w2_Svibthetaphic = (conj(P1)*K2phi0*Km1*Km2);
w2_SphiBM = (K2phi0^2)*(Km1^2);
w2_SthetaBM = (abs(P1).^2)*(Km2^2); ;
w2_SthetaphiBM = 2*K2phi0*real(P1.*(Km2).*(Km1).*SthetaphiBM);

Sphi_est = w1_Suu.*Hjw2.*Su+w1_Svv.*Hjw2.*Sv+w1_Suv.*Hjw2.*Suv+...
   w1_Svibtheta.*Hjw2.*Svibtheta+w1_Svibphi.*Hjw2.*Svibphi...

+w1_Svibphitheta.*Hjw2.*Svibphitheta+w1_Svibphithetac.*Hjw2.*conj(Sv
ibphitheta)+...

w1_SthetaBM.*Hjw2.*SthetaBM+w1_SphiBM.*Hjw2.*SphiBM+w1_SphithetaBM.*
Hjw2;



Stheta_est = w2_Svv.*Hjw2.*Sv+w2_Suu.*Hjw2.*Su+w2_Svu.*Hjw2.*Svu+...
   w2_Svibphi.*Hjw2.*Svibphi+w2_Svibtheta.*Hjw2.*Svibtheta...

+w2_Svibthetaphi.*Hjw2.*Svibthetaphi+w2_Svibthetaphic.*Hjw2.*conj(Sv
ibthetaphi)+...

w2_SphiBM.*Hjw2.*SphiBM+w2_SthetaBM.*Hjw2.*SthetaBM+w2_SthetaphiBM.*
Hjw2;

% figure; plot(freq,Sphi,'linewidth',2); hold on;
% plot(freq,Sphi_est,'r','linewidth',2)
% xlabel('Frequency (Hz)')
% ylabel('PSD \phi (W/Hz)')
% title('\phi Power Spectral Density')
% %axis([0 3000 0 2e-12])
```

```
% axis([0 5000 0 5e-13])
%
% plot(freq,Stheta,'m','linewidth',2); hold on;
% plot(freq,Stheta_est,'g','linewidth',2)
% xlabel('Frequency (Hz)')
% ylabel('PSD \theta (W/Hz)')
% title('\theta Power Spectral Density')
% %axis([0 3000 0 4e-12])
% %axis([0 5000 0 5e-13])
dat =[Sphi Sphi_est Stheta Stheta_est];
createfigure(freq,dat)
```

M-File 47: Lin2D_SW.m

```
% Lin2D_SW.m
% 23 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for the inputs.

init_2D_file
Vb = param_2D(1);
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);

u = volt_noise.signals.values(:,1);  %time-series voltage u from
simulink
v = volt_noise.signals.values(:,2);  %time-series voltage v from
simulink
SimOutPhi = SimOut.signals.values(:,1); %time-series angle phi from
simulink
SimOutTheta = SimOut.signals.values(:,2); % time-series angle theta
from simulink
VibPhi = vib_noise.signals.values(:,1);  %time-series vibration
noise from simulink
VibTheta = vib_noise.signals.values(:,2);  %time-series vibration
noise simulink
BMPhi = bm_noise.signals.values(:,1);  %time-series vibration noise
from simulink
BMTheta = bm_noise.signals.values(:,2);  %time-series vibration
noise simulink

pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
u0 = mean(u(end-pt,end));   % deterministic u (average value)
v0 = mean(v(end-pt,end));   % deterministic v (average value)
volt0 = [u0; v0];

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v
```

```
fft_len = 10000;  %Length of FFTs in PSD
OUTPUTu = Standard_PSD(1/ST,fft_len,u);
OUTPUTv = Standard_PSD(1/ST,fft_len,v);
OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
OUTPUTuv = Cross_PSD(1/ST,fft_len,u,v);
OUTPUTvibphi = Standard_PSD(1/ST,fft_len,VibPhi);
OUTPUTvibtheta = Standard_PSD(1/ST,fft_len,VibTheta);
OUTPUTvibphitheta = Cross_PSD(1/ST,fft_len,VibPhi,VibTheta);
OUTPUTvibthetaphi = Cross_PSD(1/ST,fft_len,VibTheta,VibPhi);
OUTPUTphiBM = Standard_PSD(1/ST,fft_len,BMPhi);
OUTPUTthetaBM = Standard_PSD(1/ST,fft_len,BMTheta);
OUTPUTphithetaBM = Cross_PSD(1/ST,fft_len,BMPhi,BMTheta);
OUTPUTthetaphiBM = Cross_PSD(1/ST,fft_len,BMTheta,BMPhi);
freq = OUTPUTu(:,1);

Su = OUTPUTu(:,2);  %Su(jw) PSD
Sv = OUTPUTv(:,2);  %Sv(jw) PSD
Sphi = OUTPUTphi(:,2);  %Sphi(jw) PSD
Stheta = OUTPUTtheta(:,2);  %Stheta(jw) PSD
Suv = OUTPUTuv(:,2);  %Suv(jw) Cross PSD
Svu = Suv; % real valued sequence
Svibphi = OUTPUTvibphi(:,2);
Svibtheta = OUTPUTvibtheta(:,2);
Svibphitheta = OUTPUTvibphitheta(:,2);
Svibthetaphi = OUTPUTvibphitheta(:,2);
SphiBM = OUTPUTphiBM(:,2);
SthetaBM = OUTPUTthetaBM(:,2);
SphithetaBM = OUTPUTphithetaBM(:,2);
SthetaphiBM = OUTPUTphithetaBM(:,2);


w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;
Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)

w1_Suu =
abs((K1theta0^2)*(K2u0^2)+P2*K1theta0*K1u0*K2u0+conj(P2)*K1theta0*K1
u0*K2u0+(abs(P2).^2)*(K1u0^2));
w1_Svv =
abs((K1theta0^2)*(K2v0^2)+(P2*K1theta0*K1v0*K2v0)+(conj(P2)*K1theta0
*K1v0*K2v0)+(abs(P2).^2)*(K1v0^2));
w1_Suv =
abs(2*(K1theta0^2)*K2u0*K2v0+P2*K1theta0*K1u0*K2v0+P2*K1theta0*K1v0*
K2u0+conj(P2)*K1theta0*K1u0*K2v0...
   +K1v0*(conj(P2)*K1theta0*K2u0+2*(abs(P2).^2)*K1u0));
w1_Svibtheta = (K1theta0^2)*(Km2^2);
```

```
w1_Svibphi = (abs(P2).^2)*(Km1^2);
w1_Svibphitheta = (P2*K1theta0*Km1*Km2);
w1_Svibphithetac = (conj(P2)*K1theta0*Km1*Km2);
w1_SthetaBM = (K1theta0^2)*(Km2^2);
w1_SphiBM = (abs(P2).^2)*(Km1^2); ;
w1_SphithetaBM = 2*K1theta0*real(P2.*(Km1).*(Km2).*SphithetaBM);
%SphithetaBM

w2_Svv =
abs((K2phi0^2)*(K1v0^2)+P1*K2phi0*K2v0*K1v0+conj(P1)*K2phi0*K2v0*K1v
0+(abs(P1).^2)*(K2v0^2));
w2_Suu =
abs((K2phi0^2)*(K1u0^2)+(P1*K2phi0*K2u0*K1u0)+(conj(P1)*K2phi0*K2u0*
K1u0)+(abs(P1).^2)*(K2u0^2));
w2_Svu =
abs(2*(K2phi0^2)*K1v0*K1u0+P1*K2phi0*K2v0*K1u0+P1*K2phi0*K2u0*K1v0+c
onj(P1)*K2phi0*K2v0*K1u0...
   +K2u0*(conj(P1)*K2phi0*K1v0+2*(abs(P1).^2)*K2v0));
w2_Svibphi = (K2phi0^2)*(Km1^2);
w2_Svibtheta = (abs(P1).^2)*(Km2^2);
w2_Svibthetaphi = (P1*K2phi0*Km1*Km2);
w2_Svibthetaphic = (conj(P1)*K2phi0*Km1*Km2);
w2_SphiBM = (K2phi0^2)*(Km1^2);
w2_SthetaBM = (abs(P1).^2)*(Km2^2); ;
w2_SthetaphiBM = 2*K2phi0*real(P1.*(Km2).*(Km1).*SthetaphiBM);

Sphi_est = w1_Suu.*Hjw2.*Su+w1_Svv.*Hjw2.*Sv+w1_Suv.*Hjw2.*Suv+...
   w1_Svibtheta.*Hjw2.*Svibtheta+w1_Svibphi.*Hjw2.*Svibphi...

+w1_Svibphitheta.*Hjw2.*Svibphitheta+w1_Svibphithetac.*Hjw2.*conj(Sv
ibphitheta)+...

w1_SthetaBM.*Hjw2.*SthetaBM+w1_SphiBM.*Hjw2.*SphiBM+w1_SphithetaBM.*
Hjw2;


Stheta_est = w2_Svv.*Hjw2.*Sv+w2_Suu.*Hjw2.*Su+w2_Svu.*Hjw2.*Svu+...
   w2_Svibphi.*Hjw2.*Svibphi+w2_Svibtheta.*Hjw2.*Svibtheta...

+w2_Svibthetaphi.*Hjw2.*Svibthetaphi+w2_Svibthetaphic.*Hjw2.*conj(Sv
ibthetaphi)+...

w2_SphiBM.*Hjw2.*SphiBM+w2_SthetaBM.*Hjw2.*SthetaBM+w2_SthetaphiBM.*
Hjw2;

figure; plot(freq,Sphi,'linewidth',2); hold on;
plot(freq,Sphi_est,'r','linewidth',2)
xlabel('Frequency (Hz)')
ylabel('PSD \phi (W/Hz)')
title('\phi Power Spectral Density')
%axis([0 3000 0 2e-12])
%axis([0 5000 0 5e-13])
```

```
figure; plot(freq,Stheta,'m','linewidth',2); hold on;
plot(freq,Stheta_est,'g','linewidth',2)
xlabel('Frequency (Hz)')
ylabel('PSD \theta (W/Hz)')
title('\theta Power Spectral Density')
%axis([0 3000 0 4e-12])
%axis([0 5000 0 5e-13])
```

M-File 48: Lin2D_Vib.m

```
% Lin2D_Vib.m
% 16 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for only the vibration inputs.

init_2D_file
% param_2D = [Vb,ho,x1,x2,z1,z2,I1,D1,Km1,I2,D2,Km2];
Vb = param_2D(1);
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);

u = volt_noise.signals.values(:,1);  %time-series voltage u from
simulink
v = volt_noise.signals.values(:,2);  %time-series voltage v from
simulink
VibPhi = vib_noise.signals.values(:,1);  %time-series vibration
noise from simulink
VibTheta = vib_noise.signals.values(:,2);  %time-series vibration
noise simulink
SimOutPhi = SimOut.signals.values(:,1); %time-series angle phi from
simulink
SimOutTheta = SimOut.signals.values(:,2); % time-series angle theta
from simulink

pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
u0 = mean(u(end-pt,end));   % deterministic u (average value)
v0 = mean(v(end-pt,end));   % deterministic v (average value)
volt0 = [u0; v0];
% VibPhi = VibPhi-mean(VibPhi);  %should already be zero mean
% VibTheta = VibTheta-mean(VibTheta);  %should already be zero mean

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v
```

```
fft_len = 10000;  %Length of FFTs in PSD
OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
OUTPUTvibphi = Standard_PSD(1/ST,fft_len,VibPhi);
OUTPUTvibtheta = Standard_PSD(1/ST,fft_len,VibTheta);
OUTPUTvibphitheta = Cross_PSD(1/ST,fft_len,VibPhi,VibTheta);
OUTPUTvibthetaphi = Cross_PSD(1/ST,fft_len,VibTheta,VibPhi);

freq = OUTPUTphi(:,1);
Sphi = OUTPUTphi(:,2);  % Sphi(jw) PSD
Stheta = OUTPUTtheta(:,2);  % Stheta(jw) PSD
Svibphi = OUTPUTvibphi(:,2);
Svibtheta = OUTPUTvibtheta(:,2);
Svibphitheta = OUTPUTvibphitheta(:,2);
Svibthetaphi = OUTPUTvibphitheta(:,2);

w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;

Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)

w1_Svibtheta = (K1theta0^2)*(Km2^2);
w1_Svibphi = (abs(P2).^2)*(Km1^2);
w1_Svibphitheta = (P2*K1theta0*Km1*Km2);
w1_Svibphithetac = (conj(P2)*K1theta0*Km1*Km2);

w2_Svibphi = (K2phi0^2)*(Km1^2);
w2_Svibtheta = (abs(P1).^2)*(Km2^2);
w2_Svibthetaphi = (P1*K2phi0*Km1*Km2);
w2_Svibthetaphic = (conj(P1)*K2phi0*Km1*Km2);

Sphi_est =
w1_Svibtheta.*Hjw2.*Svibtheta+w1_Svibphi.*Hjw2.*Svibphi...

+w1_Svibphitheta.*Hjw2.*Svibphitheta+w1_Svibphithetac.*Hjw2.*conj(Sv
ibphitheta);

Stheta_est =
w2_Svibphi.*Hjw2.*Svibphi+w2_Svibtheta.*Hjw2.*Svibtheta...

+w2_Svibthetaphi.*Hjw2.*Svibthetaphi+w2_Svibthetaphic.*Hjw2.*conj(Sv
ibthetaphi);

Sphi_est1 = w1_Svibtheta.*Hjw2.*Svibtheta;
Sphi_est2 = w1_Svibphi.*Hjw2.*Svibphi;
Sphi_est3 = w1_Svibphitheta.*Hjw2.*Svibphitheta;
Sphi_est4 = w1_Svibphithetac.*Hjw2.*conj(Svibphitheta);
```

```
Stheta_est1 = w2_Svibphi.*Hjw2.*Svibphi;
Stheta_est2 = w2_Svibtheta.*Hjw2.*Svibtheta;
Stheta_est3 = w2_Svibthetaphi.*Hjw2.*Svibthetaphi;
Stheta_est4 = w2_Svibthetaphic.*Hjw2.*conj(Svibthetaphi);


figure; plot(freq,Sphi,'linewidth',2); hold on;
plot(freq,Sphi_est,'r','linewidth',2);
% plot(freq,Sphi_est1,'k');
% plot(freq,Sphi_est2,'m');
% plot(freq,abs(Sphi_est3),'g');
% plot(freq,abs(Sphi_est4),'c');
xlabel('Frequency (Hz)')
ylabel('PSD \phi (W/Hz)')
title('\phi Power Spectral Density')
axis([-1000 1000 0 2e-10]); hold off

figure; plot(freq,Stheta,'m','linewidth',2); hold on;
plot(freq,Stheta_est,'g','linewidth',2)
% plot(freq,Stheta_est1,'k','linewidth',1)
% plot(freq,Stheta_est2,'m','linewidth',1)
% plot(freq,Stheta_es t3,'g','linewidth',1)
xlabel('Frequency (Hz)')
ylabel('PSD \theta (W/Hz)')
title('\theta Power Spectral Density')
axis([-1000 1000 0 2e-10])
```

M-File 49: Lin2D_Volt.m

```
% Lin2D_Volt.m
% 19 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for only the voltage inputs.

init_2D_file
% param_2D = [Vb,ho,x1,x2,z1,z2,I1,D1,Km1,I2,D2,Km2];
Vb = param_2D(1);
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);

u = volt_noise.signals.values(:,1);  %time-series voltage u from
simulink
v = volt_noise.signals.values(:,2);  %time-series voltage v from
simulink
SimOutPhi = SimOut.signals.values(:,1); %time-series angle phi from
simulink
SimOutTheta = SimOut.signals.values(:,2); % time-series angle theta
from simulink
pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
u0 = mean(u(end-pt,end));   % deterministic u (average value)
v0 = mean(v(end-pt,end));   % deterministic v (average value)
volt0 = [u0; v0];

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v

% use psd functions
% function OUTPUT = Standard_PSD(Sampling Frequency,Num points in
FFT,Filename)
% function OUTPUT = Cross_PSD(SF,N,TimeSeriesData1,TimeSeriesData2)

fft_len = 5000;  %Length of FFTs in PSD
```

```
OUTPUTu = Standard_PSD(1/ST,fft_len,u);
OUTPUTv = Standard_PSD(1/ST,fft_len,v);
OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
OUTPUTuv = Cross_PSD(1/ST,fft_len,u,v);
freq = OUTPUTu(:,1);
Su = OUTPUTu(:,2);  %Su(jw) PSD
Sv = OUTPUTv(:,2);  %Sv(jw) PSD
Sphi = OUTPUTphi(:,2);  %Sphi(jw) PSD
Stheta = OUTPUTtheta(:,2);  %Stheta(jw) PSD
Suv = OUTPUTuv(:,2);  %Suv(jw) Cross PSD
Svu = Suv; % real valued sequence

w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;
Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)

w1_Suu =
abs((K1theta0^2)*(K2u0^2)+P2*K1theta0*K1u0*K2u0+conj(P2)*K1theta0*K1
u0*K2u0+(abs(P2).^2)*(K1u0^2));
w1_Svv =
abs((K1theta0^2)*(K2v0^2)+(P2*K1theta0*K1v0*K2v0)+(conj(P2)*K1theta0
*K1v0*K2v0)+(abs(P2).^2)*(K1v0^2));
w1_Suv =
abs(2*(K1theta0^2)*K2u0*K2v0+P2*K1theta0*K1u0*K2v0+P2*K1theta0*K1v0*
K2u0+conj(P2)*K1theta0*K1u0*K2v0...
   +K1v0*(conj(P2)*K1theta0*K2u0+2*(abs(P2).^2)*K1u0));

w2_Svv =
abs((K2phi0^2)*(K1v0^2)+P1*K2phi0*K2v0*K1v0+conj(P1)*K2phi0*K2v0*K1v
0+(abs(P1).^2)*(K2v0^2));
w2_Suu =
abs((K2phi0^2)*(K1u0^2)+(P1*K2phi0*K2u0*K1u0)+(conj(P1)*K2phi0*K2u0*
K1u0)+(abs(P1).^2)*(K2u0^2));
w2_Svu =
abs(2*(K2phi0^2)*K1v0*K1u0+P1*K2phi0*K2v0*K1u0+P1*K2phi0*K2u0*K1v0+c
onj(P1)*K2phi0*K2v0*K1u0...
   +K2u0*(conj(P1)*K2phi0*K1v0+2*(abs(P1).^2)*K2v0));

Sphi_est = w1_Suu.*Hjw2.*Su+w1_Svv.*Hjw2.*Sv+w1_Suv.*Hjw2.*Suv;
Sphi_est1 = w1_Suu.*Hjw2.*Su;  %Su component for aprox of Sphi
Sphi_est2 = w1_Svv.*Hjw2.*Sv;  %Sv component for aprox of Sphi
Sphi_est3 = w1_Suv.*Hjw2.*Suv;  %Suv component for aprox of Sphi

Stheta_est = w2_Svv.*Hjw2.*Sv+w2_Suu.*Hjw2.*Su+w2_Svu.*Hjw2.*Svu;
Stheta_est1 = w2_Svv.*Hjw2.*Sv;  %Su component for aprox of Stheta
Stheta_est2 = w2_Suu.*Hjw2.*Su;  %Sv component for aprox of Stheta
Stheta_est3 = w2_Svu.*Hjw2.*Svu;  %Svu component for aprox of Stheta
```

```
figure; plot(freq,Sphi,'linewidth',2); hold on;
plot(freq,Sphi_est,'r','linewidth',2)
% plot(freq,Sphi_est1,'k','linewidth',1)
% plot(freq,Sphi_est2,'m','linewidth',1)
% plot(freq,Sphi_est3,'g','linewidth',1)
xlabel('Frequency (Hz)')
ylabel('PSD \phi (W/Hz)')
title('\phi Power Spectral Density')
axis([0 5000 0 1e-16])

figure; plot(freq,Stheta,'m','linewidth',2); hold on;
plot(freq,Stheta_est,'g','linewidth',2)
% plot(freq,Stheta_est1,'k','linewidth',1)
% plot(freq,Stheta_est2,'m','linewidth',1)
% plot(freq,Stheta_est3,'g','linewidth',1)
xlabel('Frequency (Hz)')
ylabel('PSD \theta (W/Hz)')
title('\theta Power Spectral Density')
axis([0 5000 0 1e-16])
```

M-File 50: Standard_PSD.m

```
function OUTPUT = Standard_PSD(SF,N,temp)
% function OUTPUT = Standard_PSD(SF,N,temp)
% OUTPUT = [fft_index', fft_temp'];
%
% Clint Edwards, JHU-APL
% 14 June 2007
len = length(temp);
temp = temp-mean(temp);  %mean value removed of 'temp'
T = 1/SF;  % sampling period for time-series (ts) data

% win = hanning(N);
SS = 2;
win = hanning(round(N/SS));
win = padarray(win,floor(N*((1-1/SS)/2)));

while length(win)<N
   win(end+1) = 0;
end

shift = round(N/4);
NN = ceil(length(temp)/shift);
fft_temp = zeros(NN,N);
index=[0:1:length(temp)-1]*T;  %time index
fft_index = [-N/2+1:1:(N/2)]*(1/(N*T));  %freq index

for ii=1:NN-floor(N/shift)
    if (N+(ii-1)*shift)<=length(temp)
        temp_t=win.*temp((ii-1)*shift+1:N+(ii-1)*shift);
    else
        win_size = length(win);
        temp_size = length(temp((ii-1)*shift+1:end));
        while win_size>temp_size
            temp(end+1) = 0;
            temp_size = length(temp((ii-1)*shift+1:end));
        end
        temp_t=win.*temp((ii-1)*shift+1:end);
    end
        fft_temp(ii,:) = fftshift(T*fft(temp_t-mean(temp_t),N));
end
nf=1;
fft_temp = sum(abs(fft_temp).^2)*(1/NN);
fft_temp=filter((1/nf)*ones(nf,1),1,fft_temp);

OUTPUT = [fft_index', fft_temp'];


%Input_Sine_FFT.m
N=2^17;
Tsim = 1e-5;
```

```
nn=[0:N-1];
len = 2e4;

Trunc=input_voltage_noise_1.signals.values(end-len:end);
meanTrunc = mean(Trunc)
B=Trunc-meanTrunc;
BB = (1/(N))*fft(B,N);
f = nn/(N*Tsim);
figure
plot(f(1:100),abs(BB(1:100)))
% figure
% plot(angle(BB))


figure
plot(input_voltage_noise_1.time,
input_voltage_noise_1.signals.values)
```

M-File 51: Lin2D_VibVal.m

```
% Lin2D_VibVal.m
% 18 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for only the vibration inputs.

init_2D_file
% param_2D = [Vb,ho,x1,x2,z1,z2,I1,D1,Km1,I2,D2,Km2];
Vb = param_2D(1);
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);
A = 1e-9;  % Vibration Sinewave Input Amplitude

u = volt_noise.signals.values(:,1);  %time-series voltage u from
simulink
v = volt_noise.signals.values(:,2);  %time-series voltage v from
simulink
VibPhi = vib_noise.signals.values(:,1);  %time-series vibration
noise from simulink
VibTheta = vib_noise.signals.values(:,2);  %time-series vibration
noise simulink
SimOutPhi = SimOut.signals.values(:,1); %time-series angle phi from
simulink
SimOutTheta = SimOut.signals.values(:,2); % time-series angle theta
from simulink

pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
u0 = mean(u(end-pt,end));   % deterministic u (average value)
v0 = mean(v(end-pt,end));   % deterministic v (average value)
volt0 = [u0; v0];
% VibPhi = VibPhi-mean(VibPhi);  %should already be zero mean
% VibTheta = VibTheta-mean(VibTheta);  %should already be zero mean

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0)  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0)  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0)  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0)   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
```

```
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v

fft_len = 20000;  %Length of FFTs in PSD
OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
OUTPUTvibphi = Standard_PSD(1/ST,fft_len,VibPhi);
OUTPUTvibtheta = Standard_PSD(1/ST,fft_len,VibTheta);
OUTPUTvibphitheta = Cross_PSD(1/ST,fft_len,VibPhi,VibTheta);
OUTPUTvibthetaphi = Cross_PSD(1/ST,fft_len,VibTheta,VibPhi);

freq = OUTPUTphi(:,1);
Sphi = OUTPUTphi(:,2);  % Sphi(jw) PSD
Stheta = OUTPUTtheta(:,2);  % Stheta(jw) PSD
Svibphi = OUTPUTvibphi(:,2);
Svibtheta = OUTPUTvibtheta(:,2);
Svibphitheta = OUTPUTvibphitheta(:,2);
Svibthetaphi = OUTPUTvibphitheta(:,2);

w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;

Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)
w0 = 200
ValHjw2 = abs((Km1*A).^2/((-I1*w0^2)+j*w0*D1+Km1-K1phi0));
figure; plot(freq,Hjw2,'--r');

w1_Svibtheta = (K1theta0^2)*(Km2^2);
w1_Svibphi = (abs(P2).^2)*(Km1^2);
w1_Svibphitheta = (P2*K1theta0*Km1*Km2);
w1_Svibphithetac = (conj(P2)*K1theta0*Km1*Km2);

Sphi_est =
w1_Svibtheta.*Hjw2.*Svibtheta+w1_Svibphi.*Hjw2.*Svibphi...

+w1_Svibphitheta.*Hjw2.*Svibphitheta+w1_Svibphithetac.*Hjw2.*conj(Sv
ibphitheta);

Sphi_est1 = w1_Svibtheta.*Hjw2.*Svibtheta;
Sphi_est2 = w1_Svibphi.*Hjw2.*Svibphi;  %problem is likely here with
this calculation
Sphi_est3 = w1_Svibphitheta.*Hjw2.*Svibphitheta;
Sphi_est4 = w1_Svibphithetac.*Hjw2.*conj(Svibphitheta);

figure; plot(freq,Sphi,'linewidth',2); hold on;
plot(freq,(1)*Sphi_est,'r','linewidth',2);
plot(w0,ValHjw2,'ob')% plot(freq,Sphi_est1,'k');
% plot(freq,Sphi_est2,'m');
% plot(freq,abs(Sphi_est3),'g');
```

```
% plot(freq,abs(Sphi_est4),'c');
xlabel('Frequency (Hz)')
ylabel('PSD \phi (W/Hz)')
title('\phi Power Spectral Density')
axis([-1000 1000 0 2e-10])

% figure; plot(freq,Stheta,'m','linewidth',2); hold on;
% plot(freq,Stheta_est,'g','linewidth',2)
% % plot(freq,Stheta_est1,'k','linewidth',1)
% % plot(freq,Stheta_est2,'m','linewidth',1)
% % plot(freq,Stheta_es t3,'g','linewidth',1)
% xlabel('Frequency (Hz)')
% ylabel('PSD \theta (W/Hz)')
% title('\theta Power Spectral Density')
% axis([0 5000 0 1e-16])
```

M-File 52: Lin2D_Volt1.m

```
% Lin2D_Volt1.m
% 15 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for only the voltage inputs.

init_2D_file
% param_2D = [Vb,ho,x1,x2,z1,z2,I1,D1,Km1,I2,D2,Km2];
Vb = param_2D(1);
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);

u = volt_noise.signals.values(:,1);  %time-series voltage u from
simulink
v = volt_noise.signals.values(:,2);  %time-series voltage v from
simulink
SimOutPhi = SimOut.signals.values(:,1); %time-series angle phi from
simulink
SimOutTheta = SimOut.signals.values(:,2); % time-series angle theta
from simulink
pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
u0 = mean(u(end-pt,end));   % deterministic u (average value)
v0 = mean(v(end-pt,end));   % deterministic v (average value)
volt0 = [u0; v0];

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v

% use psd functions
% function OUTPUT = Standard_PSD(Sampling Frequency,Num points in
FFT,Filename)
% function OUTPUT = Cross_PSD(SF,N,TimeSeriesData1,TimeSeriesData2)

fft_len = 15000;  %Length of FFTs in PSD
```

```
OUTPUTu = Standard_PSD(1/ST,fft_len,u);
OUTPUTv = Standard_PSD(1/ST,fft_len,v);
OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
OUTPUTuv = Cross_PSD(1/ST,fft_len,u,v);
freq = OUTPUTu(:,1);
Su = OUTPUTu(:,2);  %Su(jw) PSD
Sv = OUTPUTv(:,2);  %Sv(jw) PSD
Sphi = OUTPUTphi(:,2);  %Sphi(jw) PSD
Stheta = OUTPUTtheta(:,2);  %Stheta(jw) PSD
Suv = OUTPUTuv(:,2);  %Suv(jw) Cross PSD
Svu = Suv; % real valued sequence

w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;

Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)

w1_Suu =
abs((K1theta0^2)*(K2u0^2)+P2*K1theta0*K1u0*K2u0+conj(P2)*K1theta0*K1
u0*K2u0+(abs(P2).^2)*(K1u0^2));
w1_Svv =
abs((K1theta0^2)*(K2v0^2)+(P2*K1theta0*K1v0*K2v0)+(conj(P2)*K1theta0
*K1v0*K2v0)+(abs(P2).^2)*(K1v0^2));
w1_Suv =
abs(2*(K1theta0^2)*K2u0*K2v0+P2*K1theta0*K1u0*K2v0+P2*K1theta0*K1v0*
K2u0+conj(P2)*K1theta0*K1u0*K2v0...
   +K1v0*(conj(P2)*K1theta0*K2u0+2*(abs(P2).^2)*K1u0));

w2_Svv =
abs((K2phi0^2)*(K1v0^2)+P1*K2phi0*K2v0*K1v0+conj(P1)*K2phi0*K2v0*K1v
0+(abs(P1).^2)*(K2v0^2));
w2_Suu =
abs((K2phi0^2)*(K1u0^2)+(P1*K2phi0*K2u0*K1u0)+(conj(P1)*K2phi0*K2u0*
K1u0)+(abs(P1).^2)*(K2u0^2));
w2_Svu =
abs(2*(K2phi0^2)*K1v0*K1u0+P1*K2phi0*K2v0*K1u0+P1*K2phi0*K2u0*K1v0+c
onj(P1)*K2phi0*K2v0*K1u0...
   +K2u0*(conj(P1)*K2phi0*K1v0+2*(abs(P1).^2)*K2v0));

Sphi_est = w1_Suu.*Hjw2.*Su+w1_Svv.*Hjw2.*Sv+w1_Suv.*Hjw2.*Suv;
Sphi_est1 = w1_Suu.*Hjw2.*Su;  %Su component for aprox of Sphi
Sphi_est2 = w1_Svv.*Hjw2.*Sv;  %Sv component for aprox of Sphi
Sphi_est3 = w1_Suv.*Hjw2.*Suv;  %Suv component for aprox of Sphi

Stheta_est = w2_Svv.*Hjw2.*Sv+w2_Suu.*Hjw2.*Su+w2_Svu.*Hjw2.*Svu;
Stheta_est1 = w2_Svv.*Hjw2.*Sv;  %Su component for aprox of Stheta
Stheta_est2 = w2_Suu.*Hjw2.*Su;  %Sv component for aprox of Stheta
Stheta_est3 = w2_Svu.*Hjw2.*Svu;  %Svu component for aprox of Stheta
```

```
figure; plot(freq,Sphi,'linewidth',2); hold on;
plot(freq,Sphi_est,'r','linewidth',2)
% plot(freq,Sphi_est1,'k','linewidth',1)
% plot(freq,Sphi_est2,'m','linewidth',1)
% plot(freq,Sphi_est3,'g','linewidth',1)
xlabel('Frequency (Hz)')
ylabel('PSD \phi (W/Hz)')
title('\phi Power Spectral Density')
axis([0 5000 0 1e-16])

figure; plot(freq,Stheta,'m','linewidth',2); hold on;
plot(freq,Stheta_est,'g','linewidth',2)
% plot(freq,Stheta_est1,'k','linewidth',1)
% plot(freq,Stheta_est2,'m','linewidth',1)
% plot(freq,Stheta_est3,'g','linewidth',1)
xlabel('Frequency (Hz)')
ylabel('PSD \theta (W/Hz)')
title('\theta Power Spectral Density')
axis([0 5000 0 1e-16])
```

```
% Lin2D_VoltBM.m
% 15 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for only the voltage and vibration inputs.

init_2D_file
% param_2D = [Vb,ho,x1,x2,z1,z2,I1,D1,Km1,I2,D2,Km2];
Vb = param_2D(1);
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);

u = volt_noise.signals.values(:,1);  %time-series voltage u from
simulink
v = volt_noise.signals.values(:,2);  %time-series voltage v from
simulink
BMPhi = bm_noise.signals.values(:,1);  %time-series vibration noise
from simulink
BMTheta = bm_noise.signals.values(:,2);  %time-series vibration
noise simulink
SimOutPhi = SimOut.signals.values(:,1); %time-series angle phi from
simulink
SimOutTheta = SimOut.signals.values(:,2); % time-series angle theta
from simulink
pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
u0 = mean(u(end-pt,end));   % deterministic u (average value)
v0 = mean(v(end-pt,end));   % deterministic v (average value)
BMPhi = BMPhi-mean(BMPhi);  %should already be zero mean
BMTheta = BMTheta-mean(BMTheta);  %should already be zero mean
volt0 = [u0; v0];

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v

% use psd functions
```

```
% function OUTPUT = Standard_PSD(Sampling Frequency,Num points in
FFT,Filename)
% function OUTPUT = Cross_PSD(SF,N,TimeSeriesData1,TimeSeriesData2)

fft_len = 15000;  %Length of FFTs in PSD
OUTPUTu = Standard_PSD(1/ST,fft_len,u);
OUTPUTv = Standard_PSD(1/ST,fft_len,v);
OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
OUTPUTphiBM = Standard_PSD(1/ST,fft_len,BMPhi);
OUTPUthetaBM = Standard_PSD(1/ST,fft_len,BMTheta);
OUTPUTphithetaBM = Cross_PSD(1/ST,fft_len,BMPhi,BMTheta);
OUTPUTthetaphiBM = Cross_PSD(1/ST,fft_len,BMTheta,BMPhi);
OUTPUTuv = Cross_PSD(1/ST,fft_len,u,v);
OUTPUTvu = Cross_PSD(1/ST,fft_len,v,u);

freq = OUTPUTu(:,1);
Su = OUTPUTu(:,2);  % Su(jw) PSD
Sv = OUTPUTv(:,2);  % Sv(jw) PSD
Sphi = OUTPUTphi(:,2);  % Sphi(jw) PSD
figure; plot(Sphi)
size(Sphi)
Stheta = OUTPUTtheta(:,2);  % Stheta(jw) PSD
Suv = OUTPUTuv(:,2);  % Suv(jw) Cross PSD
Svu = OUTPUTvu(:,2);  % real valued sequence
SphiBM = OUTPUTphiBM(:,2);
SthetaBM = OUTPUTthetaBM(:,2);
SphithetaBM = OUTPUTphithetaBM(:,2);
SthetaphiBM = OUTPUTphithetaBM(:,2);

w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;

Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)

w1_Suu =
abs((K1theta0^2)*(K2u0^2)+P2*K1theta0*K1u0*K2u0+conj(P2)*K1theta0*K1
u0*K2u0+(abs(P2).^2)*(K1u0^2));
w1_Svv =
abs((K1theta0^2)*(K2v0^2)+(P2*K1theta0*K1v0*K2v0)+(conj(P2)*K1theta0
*K1v0*K2v0)+(abs(P2).^2)*(K1v0^2));
w1_Suv =
abs(2*(K1theta0^2)*K2u0*K2v0+P2*K1theta0*K1u0*K2v0+P2*K1theta0*K1v0*
K2u0+conj(P2)*K1theta0*K1u0*K2v0...
   +K1v0*(conj(P2)*K1theta0*K2u0+2*(abs(P2).^2)*K1u0));
w1_SthetaBM = (K1theta0^2)*(Km2^2);
w1_SphiBM = (abs(P2).^2).*(Km1^2); ;
w1_SphithetaBM = 2*K1theta0*re(P2.*(Km1^2).*(Km2^2).*SphithetaBM);
%SphithetaBM
```

```
w2_Svv =
abs((K2phi0^2)*(K1v0^2)+P1*K2phi0*K2v0*K1v0+conj(P1)*K2phi0*K2v0*K1v
0+(abs(P1).^2)*(K2v0^2));
w2_Suu =
abs((K2phi0^2)*(K1u0^2)+(P1*K2phi0*K2u0*K1u0)+(conj(P1)*K2phi0*K2u0*
K1u0)+(abs(P1).^2)*(K2u0^2));
w2_Svu =
abs(2*(K2phi0^2)*K1v0*K1u0+P1*K2phi0*K2v0*K1u0+P1*K2phi0*K2u0*K1v0+c
onj(P1)*K2phi0*K2v0*K1u0...
   +K2u0*(conj(P1)*K2phi0*K1v0+2*(abs(P1).^2)*K2v0));
% w2_SphiBM =
% w2_SthetaBM =
% w2_SthetaphiBM =


Sphi_est = w1_Suu.*Hjw2.*Su+w1_Svv.*Hjw2.*Sv+w1_Suv.*Hjw2.*Suv+...

w1_SthetaBM.*Hjw2.*SthetaBM+w1_SphiBM.*Hjw2.*SphiBM+w1_SphithetaBM.*
Hjw2*1 ;
% Sphi_est1 = w1_Suu.*Hjw2.*Su;  %Su component for aprox of Sphi
% Sphi_est2 = w1_Svv.*Hjw2.*Sv;  %Sv component for aprox of Sphi
% Sphi_est3 = w1_Suv.*Hjw2.*Suv;  %Suv component for aprox of Sphi

Stheta_est = w2_Svv.*Hjw2.*Sv+w2_Suu.*Hjw2.*Su+w2_Svu.*Hjw2.*Svu;
Stheta_est1 = w2_Svv.*Hjw2.*Sv;  %Su component for aprox of Stheta
Stheta_est2 = w2_Suu.*Hjw2.*Su;  %Sv component for aprox of Stheta
Stheta_est3 = w2_Svu.*Hjw2.*Svu;  %Svu component for aprox of Stheta

figure; plot(freq,Sphi,'linewidth',1); hold on;
% plot(freq,Sphi_est,'r','linewidth',1)
% plot(freq,Sphi_est1,'k','linewidth',1)
% plot(freq,Sphi_est2,'m','linewidth',1)
% plot(freq,Sphi_est3,'g','linewidth',1)
xlabel('Frequency (Hz)')
ylabel('PSD \phi (W/Hz)')
title('\phi Power Spectral Density')
axis([0 5000 0 1e-16])

% figure; plot(freq,Stheta,'m','linewidth',2); hold on;
% plot(freq,Stheta_est,'g','linewidth',2)
% % plot(freq,Stheta_est1,'k','linewidth',1)
% % plot(freq,Stheta_est2,'m','linewidth',1)
% % plot(freq,Stheta_est3,'g','linewidth',1)
% xlabel('Frequency (Hz)')
% ylabel('PSD \theta (W/Hz)')
% title('\theta Power Spectral Density')
% axis([0 5000 0 1e-16])
```

M-File 53: Lin2D_VoltOnly.m

```
% Lin2D_VoltOnly.m
% 12 June 2007
% Clint Edwards, JHU-APL

%Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
%linearized output.

init_2D_file
% param_2D = [Vb,ho,x1,x2,z1,z2,I1,D1,Km1,I2,D2,Km2];
Vb = param_2D(1);
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);

u = volt_noise.signals.values(:,1);  %time-series voltage u from
simulink
v = volt_noise.signals.values(:,2);  %time-series voltage v from
simulink
SimOutPhi = SimOut.signals.values(:,1); %time-series angle phi from
simulink
SimOutTheta = SimOut.signals.values(:,2); % time-series angle theta
from simulink
pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
u0 = mean(u(end-pt,end));   % deterministic u (average value)
v0 = mean(v(end-pt,end));   % deterministic v (average value)
volt0 = [u0; v0];

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v

% use psd functions
% function OUTPUT = Standard_PSD(Sampling Frequency,Num points in
FFT,Filename)
% function OUTPUT = Cross_PSD(SF,N,TimeSeriesData1,TimeSeriesData2)

fft_len = 20000;  %Length of FFTs in PSD
```

```
OUTPUTu = Standard_PSD(1/ST,fft_len,u);
OUTPUTv = Standard_PSD(1/ST,fft_len,v);
OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
OUTPUTuv = Cross_PSD(1/ST,fft_len,u,v);
freq = OUTPUTu(:,1);
Su = OUTPUTu(:,2);  %Su(jw) PSD
Sv = OUTPUTv(:,2);  %Sv(jw) PSD
Sphi = OUTPUTphi(:,2);  %Sphi(jw) PSD
Stheta = OUTPUTtheta(:,2);  %Stheta(jw) PSD
Suv = OUTPUTuv(:,2);  %Suv(jw) Cross PSD
Svu = Suv; % real valued sequence

w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;

Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)

w1_Suu =
abs((K1theta0^2)*(K2u0^2)+P2*K1theta0*K1u0*K2u0+conj(P2)*K1theta0*K1
u0*K2u0+(abs(P2).^2)*(K1u0^2));
w1_Svv =
abs((K1theta0^2)*(K2v0^2)+(P2*K1theta0*K1v0*K2v0)+(conj(P2)*K1theta0
*K1v0*K2v0)+(abs(P2).^2)*(K1v0^2));
w1_Suv =
abs(2*(K1theta0^2)*K2u0*K2v0+P2*K1theta0*K1u0*K2v0+P2*K1theta0*K1v0*
K2u0+conj(P2)*K1theta0*K1u0*K2v0...
   +K1v0*(conj(P2)*K1theta0*K2u0+2*(abs(P2).^2)*K1u0));

w2_Svv =
abs((K2phi0^2)*(K1v0^2)+P1*K2phi0*K2v0*K1v0+conj(P1)*K2phi0*K2v0*K1v
0+(abs(P1).^2)*(K2v0^2));
w2_Suu =
abs((K2phi0^2)*(K1u0^2)+(P1*K2phi0*K2u0*K1u0)+(conj(P1)*K2phi0*K2u0*
K1u0)+(abs(P1).^2)*(K2u0^2));
w2_Svu =
abs(2*(K2phi0^2)*K1v0*K1u0+P1*K2phi0*K2v0*K1u0+P1*K2phi0*K2u0*K1v0+c
onj(P1)*K2phi0*K2v0*K1u0...
   +K2u0*(conj(P1)*K2phi0*K1v0+2*(abs(P1).^2)*K2v0));

Sphi_est = w1_Suu.*Hjw2.*Su+w1_Svv.*Hjw2.*Sv+w1_Suv.*Hjw2.*Suv;
Sphi_est1 = w1_Suu.*Hjw2.*Su;  %Su component for aprox of Sphi
Sphi_est2 = w1_Svv.*Hjw2.*Sv;  %Sv component for aprox of Sphi
Sphi_est3 = w1_Suv.*Hjw2.*Suv;  %Suv component for aprox of Sphi

Stheta_est = w2_Svv.*Hjw2.*Sv+w2_Suu.*Hjw2.*Su+w2_Svu.*Hjw2.*Svu;
Stheta_est1 = w2_Svv.*Hjw2.*Sv;  %Su component for aprox of Stheta
Stheta_est2 = w2_Suu.*Hjw2.*Su;  %Sv component for aprox of Stheta
Stheta_est3 = w2_Svu.*Hjw2.*Svu;  %Svu component for aprox of Stheta
```

```
figure; plot(freq,Sphi,'linewidth',2); hold on;
plot(freq,Sphi_est,'r','linewidth',2)
% plot(freq,Sphi_est1,'k','linewidth',1)
% plot(freq,Sphi_est2,'m','linewidth',1)
% plot(freq,Sphi_est3,'g','linewidth',1)
xlabel('Frequency (Hz)')
ylabel('PSD \phi (W/Hz)')
title('\phi Power Spectral Density')
axis([0 5000 0 1e-16])

figure; plot(freq,Stheta,'m','linewidth',2); hold on;
plot(freq,Stheta_est,'g','linewidth',2)
% plot(freq,Stheta_est1,'k','linewidth',1)
% plot(freq,Stheta_est2,'m','linewidth',1)
% plot(freq,Stheta_est3,'g','linewidth',1)
xlabel('Frequency (Hz)')
ylabel('PSD \theta (W/Hz)')
title('\theta Power Spectral Density')
axis([0 5000 0 1e-16])
```

M-File 54: Lin2D_VoltVib.m

```
% Lin2D_VoltVib.m
% 15 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for only the voltage and vibration inputs.

init_2D_file
% param_2D = [Vb,ho,x1,x2,z1,z2,I1,D1,Km1,I2,D2,Km2];
Vb = param_2D(1);
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);

u = volt_noise.signals.values(:,1);  %time-series voltage u from
simulink
v = volt_noise.signals.values(:,2);  %time-series voltage v from
simulink
VibPhi = vib_noise.signals.values(:,1);  %time-series vibration
noise from simulink
VibTheta = vib_noise.signals.values(:,2);  %time-series vibration
noise simulink
SimOutPhi = SimOut.signals.values(:,1); %time-series angle phi from
simulink
SimOutTheta = SimOut.signals.values(:,2); % time-series angle theta
from simulink
pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
u0 = mean(u(end-pt,end));   % deterministic u (average value)
v0 = mean(v(end-pt,end));   % deterministic v (average value)
VibPhi = VibPhi-mean(VibPhi);  %should already be zero mean
VibTheta = VibTheta-mean(VibTheta);  %should already be zero mean
volt0 = [u0; v0];

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v
```

```
% use psd functions
% function OUTPUT = Standard_PSD(Sampling Frequency,Num points in
FFT,Filename)
% function OUTPUT = Cross_PSD(SF,N,TimeSeriesData1,TimeSeriesData2)

fft_len = 15000;  %Length of FFTs in PSD
OUTPUTu = Standard_PSD(1/ST,fft_len,u);
OUTPUTv = Standard_PSD(1/ST,fft_len,v);
OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
OUTPUTvibphi = Standard_PSD(1/ST,fft_len,VibPhi);
OUTPUTvibtheta = Standard_PSD(1/ST,fft_len,VibTheta);
OUTPUTvibphitheta = Cross_PSD(1/ST,fft_len,VibPhi,VibTheta);
OUTPUTvibthetaphi = Cross_PSD(1/ST,fft_len,VibTheta,VibPhi);
OUTPUTuv = Cross_PSD(1/ST,fft_len,u,v);
OUTPUTvu = Cross_PSD(1/ST,fft_len,v,u);

freq = OUTPUTu(:,1);
Su = OUTPUTu(:,2);  % Su(jw) PSD
Sv = OUTPUTv(:,2);  % Sv(jw) PSD
Sphi = OUTPUTphi(:,2);  % Sphi(jw) PSD
figure; plot(Sphi)
size(Sphi)
Stheta = OUTPUTtheta(:,2);  % Stheta(jw) PSD
Suv = OUTPUTuv(:,2);  % Suv(jw) Cross PSD
Svu = OUTPUTvu(:,2);  % real valued sequence
Svibphi = OUTPUTvibphi(:,2);
Svibtheta = OUTPUTvibtheta(:,2);
Svibphitheta = OUTPUTvibphitheta(:,2);
Svibthetaphi = OUTPUTvibphitheta(:,2);

w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;

Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)

w1_Suu =
abs((K1theta0^2)*(K2u0^2)+P2*K1theta0*K1u0*K2u0+conj(P2)*K1theta0*K1
u0*K2u0+(abs(P2).^2)*(K1u0^2));
w1_Svv =
abs((K1theta0^2)*(K2v0^2)+(P2*K1theta0*K1v0*K2v0)+(conj(P2)*K1theta0
*K1v0*K2v0)+(abs(P2).^2)*(K1v0^2));
w1_Suv =
abs(2*(K1theta0^2)*K2u0*K2v0+P2*K1theta0*K1u0*K2v0+P2*K1theta0*K1v0*
K2u0+conj(P2)*K1theta0*K1u0*K2v0...
   +K1v0*(conj(P2)*K1theta0*K2u0+2*(abs(P2).^2)*K1u0));
w1_Svibtheta = (K2theta0^2)*(Km2^2);
w1_Svibphi = (abs(P1).^2)*(Km2^2);
w1_Svibphitheta = (P2*K1theta0*Km1*Km2+conj(P2)*K1theta0*Km1*Km2);
```

```
w2_Svv =
abs((K2phi0^2)*(K1v0^2)+P1*K2phi0*K2v0*K1v0+conj(P1)*K2phi0*K2v0*K1v
0+(abs(P1).^2)*(K2v0^2));
w2_Suu =
abs((K2phi0^2)*(K1u0^2)+(P1*K2phi0*K2u0*K1u0)+(conj(P1)*K2phi0*K2u0*
K1u0)+(abs(P1).^2)*(K2u0^2));
w2_Svu =
abs(2*(K2phi0^2)*K1v0*K1u0+P1*K2phi0*K2v0*K1u0+P1*K2phi0*K2u0*K1v0+c
onj(P1)*K2phi0*K2v0*K1u0...
   +K2u0*(conj(P1)*K2phi0*K1v0+2*(abs(P1).^2)*K2v0));
% w2_Svibphi =
% w2_Svibtheta =
% w2_Svibthetaphi =


Sphi_est = w1_Suu.*Hjw2.*Su+w1_Svv.*Hjw2.*Sv+w1_Suv.*Hjw2.*Suv+...

w1_Svibtheta.*Hjw2.*Svibtheta+w1_Svibphi.*Hjw2.*Svibphi+w1_Svibphith
eta.*Hjw2.*Svibphitheta ;
Sphi_est1 = w1_Suu.*Hjw2.*Su;  %Su component for aprox of Sphi
Sphi_est2 = w1_Svv.*Hjw2.*Sv;  %Sv component for aprox of Sphi
Sphi_est3 = w1_Suv.*Hjw2.*Suv;  %Suv component for aprox of Sphi

Stheta_est = w2_Svv.*Hjw2.*Sv+w2_Suu.*Hjw2.*Su+w2_Svu.*Hjw2.*Svu;
Stheta_est1 = w2_Svv.*Hjw2.*Sv;  %Su component for aprox of Stheta
Stheta_est2 = w2_Suu.*Hjw2.*Su;  %Sv component for aprox of Stheta
Stheta_est3 = w2_Svu.*Hjw2.*Svu;  %Svu component for aprox of Stheta

figure; plot(freq,Sphi,'linewidth',1); hold on;
% plot(freq,Sphi_est,'r','linewidth',1)
% plot(freq,Sphi_est1,'k','linewidth',1)
% plot(freq,Sphi_est2,'m','linewidth',1)
% plot(freq,Sphi_est3,'g','linewidth',1)
xlabel('Frequency (Hz)')
ylabel('PSD \phi (W/Hz)')
title('\phi Power Spectral Density')
axis([0 5000 0 1e-16])

% figure; plot(freq,Stheta,'m','linewidth',2); hold on;
% plot(freq,Stheta_est,'g','linewidth',2)
% % plot(freq,Stheta_est1,'k','linewidth',1)
% % plot(freq,Stheta_est2,'m','linewidth',1)
% % plot(freq,Stheta_est3,'g','linewidth',1)
% xlabel('Frequency (Hz)')
% ylabel('PSD \theta (W/Hz)')
% title('\theta Power Spectral Density')
% axis([0 5000 0 1e-16])
```

M-File 55: Lin_Resp_2D_PhiBM.m

```
function [mag phase]=Lin_Resp_2D_PhiBM(freq, volt)
% Lin_Resp_2D_PhiBM.m
%
% Clint Edwards
% 4 January 2007

init_2D_file
I2 = param_2D(7);
D2 = param_2D(8);
Km2 = param_2D(9);
I1 = param_2D(10);
D1 = param_2D(11);
Km1 = param_2D(12);

K1Phi0=K1Phi(volt);
K2Phi0=K2Phi_num(volt);
K1Theta0=K1Theta_num(volt);
K2Theta0=K2Theta(volt);

w=freq*2*pi;
P1=-w^2*I1+j*w*D1+Km1-K1Phi0;
P2=-w^2*I2+j*w*D2+Km2-K2Theta0;

Hjw=1/(P1*P2-K1Theta0*K2Phi0);
Hjw_mir_phi = Hjw*(K1Theta0*K2Phi0+P2*K1Phi0);
Hjw_mir_theta = Hjw*(K2Phi0*K1Phi0+P1*K2Phi0);

mag = [abs(Hjw_mir_phi); abs(Hjw_mir_theta)];
phase = [angle(Hjw_mir_phi); angle(Hjw_mir_theta)];
```

M-File 56: Lin_Resp_2D_PhiVib.m

```
function [mag phase]=Lin_Resp_2D_PhiVib(freq, volt)
% Lin_Resp_2D_PhiVib.m
%
% Clint Edwards
% 4 January 2007

init_2D_file
I2 = param_2D(7);
D2 = param_2D(8);
Km2 = param_2D(9);
I1 = param_2D(10);
D1 = param_2D(11);
Km1 = param_2D(12);

K1Phi0=K1Phi(volt);
K2Phi0=K2Phi_num(volt);
K1Theta0=K1Theta_num(volt);
K2Theta0=K2Theta(volt);

w=freq*2*pi;
P1=-w^2*I1+j*w*D1+Km1-K1Phi0;
P2=-w^2*I2+j*w*D2+Km2-K2Theta0;

Hjw=1/(P1*P2-K1Theta0*K2Phi0);
Hjw_mir_phi = Hjw*(K1Theta0*K2Phi0+P2*K1Phi0);
Hjw_mir_theta = Hjw*(K2Phi0*K1Phi0+P1*K2Phi0);

mag = [abs(Hjw_mir_phi); abs(Hjw_mir_theta)];
phase = [angle(Hjw_mir_phi); angle(Hjw_mir_theta)];
```

M-File 57: Lin_Resp_2D_ThetaBM.m

```
function [mag phase]=Lin_Resp_2D_ThetaBM(freq, volt)
% Lin_Resp_2D_ThetaBM.m
%
% Clint Edwards
% 4 January 2007

init_2D_file
I2 = param_2D(7);
D2 = param_2D(8);
Km2 = param_2D(9);
I1 = param_2D(10);
D1 = param_2D(11);
Km1 = param_2D(12);

K1Phi0=K1Phi(volt);
K2Phi0=K2Phi_num(volt);
K1Theta0=K1Theta_num(volt);
K2Theta0=K2Theta(volt);

w=freq*2*pi;
P1=-w^2*I1+j*w*D1+Km1-K1Phi0;
P2=-w^2*I2+j*w*D2+Km2-K2Theta0;

Hjw=1/(P1*P2-K1Theta0*K2Phi0);
Hjw_mir_phi = Hjw*(K1Theta0*K2Theta0+P2*K1Theta0);
Hjw_mir_theta = Hjw*(K2Phi0*K1Theta0+P1*K2Theta0);

mag = [abs(Hjw_mir_phi); abs(Hjw_mir_theta)];
phase = [angle(Hjw_mir_phi); angle(Hjw_mir_theta)];
```

M-File 58: Lin_Resp_2D_ThetaVib.m

```
function [mag phase]=Lin_Resp_2D_ThetaVib(freq, volt)
% Lin_Resp_2D_ThetaVib.m
%
% Clint Edwards
% 4 January 2007

init_2D_file
I2 = param_2D(7);
D2 = param_2D(8);
Km2 = param_2D(9);
I1 = param_2D(10);
D1 = param_2D(11);
Km1 = param_2D(12);

K1Phi0=K1Phi(volt);
K2Phi0=K2Phi_num(volt);
K1Theta0=K1Theta_num(volt);
K2Theta0=K2Theta(volt);

w=freq*2*pi;
P1=-w^2*I1+j*w*D1+Km1-K1Phi0;
P2=-w^2*I2+j*w*D2+Km2-K2Theta0;

Hjw=1/(P1*P2-K1Theta0*K2Phi0);
Hjw_mir_phi = Hjw*(K1Theta0*K2Theta0+P2*K1Theta0);
Hjw_mir_theta = Hjw*(K2Phi0*K1Theta0+P1*K2Theta0);

mag = [abs(Hjw_mir_phi); abs(Hjw_mir_theta)];
phase = [angle(Hjw_mir_phi); angle(Hjw_mir_theta)];
```

M-File 59: Lin_Resp_2D_U.m

```
function [mag phase]=Lin_Resp_2D_U(freq, volt)
% Lin_Resp_2D_U.m
%
% Clint Edwards
% 31 December 2006

init_2D_file
I2 = param_2D(7);
D2 = param_2D(8);
Km2 = param_2D(9);
I1 = param_2D(10);
D1 = param_2D(11);
Km1 = param_2D(12);

K1Phi0=K1Phi(volt);
K2Phi0=K2Phi_num(volt);
K1Theta0=K1Theta_num(volt);
K2Theta0=K2Theta(volt);
K2U0=K2U(volt);
K1U0=K1U(volt);

%cross coupling terms
ProdCrossTerms = K1Theta0*K2Phi0;

w=freq*2*pi;
P1=-w^2*I1+j*w*D1+Km1-K1Phi0;
P2=-w^2*I2+j*w*D2+Km2-K2Theta0;

% non cross coupling terms
P1P2 = abs(P1*P2);
P2K1U0=abs(P2*K1U0);

Hjw=1/(P1*P2-K1Theta0*K2Phi0);
Hjw_mir_phi = Hjw*(K1Theta0*K2U0+P2*K1U0);
Hjw_mir_theta = Hjw*(K2Phi0*K1U0+P1*K2U0);

mag = [abs(Hjw_mir_phi); abs(Hjw_mir_theta)];
phase = [angle(Hjw_mir_phi); angle(Hjw_mir_theta)];
```

M-File 60: Lin_Resp_2D_V.m

```
function [mag phase]=Lin_Resp_2D_V(freq, volt)
% Lin_Resp_2D_V.m
%
% Clint Edwards
% 3 January 2007

init_2D_file
I2 = param_2D(7);
D2 = param_2D(8);
Km2 = param_2D(9);
I1 = param_2D(10);
D1 = param_2D(11);
Km1 = param_2D(12);

K1Phi0=K1Phi(volt);
K2Phi0=K2Phi_num(volt);
K1Theta0=K1Theta_num(volt);
K2Theta0=K2Theta(volt);
K2V0=K2V(volt);
K1V0=K1V(volt);

w=freq*2*pi;
P1=-w^2*I1+j*w*D1+Km1-K1Phi0;
P2=-w^2*I2+j*w*D2+Km2-K2Theta0;

Hjw=1/(P1*P2-K1Theta0*K2Phi0);
Hjw_mir_phi = Hjw*(K1Theta0*K2V0+P2*K1V0);
Hjw_mir_theta = Hjw*(K2Phi0*K1V0+P1*K2V0);

mag = [abs(Hjw_mir_phi); abs(Hjw_mir_theta)];
phase = [angle(Hjw_mir_phi); angle(Hjw_mir_theta)];
```

M-File 61: plot_sinewave_input.m

```
%plot_sinewave_input.m
init_2D_file
global param_2D
w=2*pi*50;
Amp = 1e-5
t = input_voltage_noise_1.time;
Out=Sinewave_Input_volt([1 1])
LinSine = Out(1,1)*Amp*sin(w*t+Out(1,2));
plot(t,LinSine)

Sim_output_phi = OUTPUT.signals.values(:,1);
size(Sim_output_phi)
mean_Sim_output_phi = mean(Sim_output_phi(end-35e3:end))
hold on
plot(t,Sim_output_phi-mean_Sim_output_phi, 'r')
```

M-File 62: Sinewave_Input_volt.m

```
function Out=Sinewave_Input_volt(volt)
%Sinewave_Input_volt
init_2D_file
global param_2D

Vb = param_2D(1);
ho = param_2D(2);
Lx = 2*param_2D(4);
Lz = 2*param_2D(6);
I2 = param_2D(7);
D2 = param_2D(8);
Km2 = param_2D(9);
I1 = param_2D(10);
D1 = param_2D(11);
Km1 = param_2D(12);

u=volt(1);
v=volt(2);
u=1;  %volt
v=1;  %volt
w=2*pi*50;  %hertz

format long
Angle = ang_given_volt_2D_SD(volt)

K1phi0 = K1Phi(volt);
K1theta0 = K1Theta_num(volt);  %numerical aproximation required
K2phi0 = K2Phi_num(volt);  %numerical aproximation required
K2theta0 = K2Theta(volt);
K1u0 = K1U(volt);
K1v0 = K1V(volt);
K2v0 = K2V(volt);
K2u0 = K2U(volt);

P1 = (w.^2)*I1+j*w*D1+(Km1-K1phi0);
P2 = (w.^2)*I2+j*w*D2+(Km2-K2theta0);
Hjw = 1/(P1.*P2-K1theta0*K2phi0);

TranFuncSine_Phi_U = Hjw*(K1theta0*K2u0+P2*K1u0);
TranFuncSine_Phi_V = Hjw*(K1theta0*K2v0+P2*K1v0);

mag_trans_Phi_U = abs(TranFuncSine_Phi_U);
phase_trans_Phi_U = angle(TranFuncSine_Phi_U);
Out(1,1) = mag_trans_Phi_U;
Out(1,2) = phase_trans_Phi_U;
A = ['|H(jw)|=|Phi/U|= ',num2str(mag_trans_Phi_U)];
B = ['<)H(jw)=<)Phi/U= ',num2str(phase_trans_Phi_U), 'radians'];
disp(' ')
disp(A)
```

```
disp(B)
disp(' ')

mag_trans_Phi_V = abs(TranFuncSine_Phi_V);
phase_trans_Phi_V = angle(TranFuncSine_Phi_V);
Out(2,1) = mag_trans_Phi_V;
Out(2,2) = phase_trans_Phi_V;
C = ['|H(jw)|=|Phi/V|= ',num2str(mag_trans_Phi_V)];
D = ['<)H(jw)=<)(Phi/V)= ',num2str(phase_trans_Phi_V), 'radians'];
disp(C)
disp(D)
disp(' ')


%%%%%%%%%%%%%%%%%%%%%%%%
TranFuncSine_Theta_U = Hjw*(K2phi0*K1u0+P1*K2u0);
TranFuncSine_Theta_V = Hjw*(K2phi0*K1v0+P1*K2v0);

mag_trans_Theta_U = abs(TranFuncSine_Theta_U);
phase_trans_Theta_U = angle(TranFuncSine_Theta_U);
Out(3,1) = mag_trans_Theta_U;
Out(3,2) = phase_trans_Theta_U;
E = ['|H(jw)|=|Theta/U|= ',num2str(mag_trans_Theta_U)];
F = ['<)H(jw)=<)Theta/U= ',num2str(phase_trans_Theta_U), 'radians'];
disp(' ')
disp(E)
disp(F)
disp(' ')

mag_trans_Theta_V = abs(TranFuncSine_Theta_V);
phase_trans_Theta_V = angle(TranFuncSine_Theta_V);
Out(4,1) = mag_trans_Theta_V;
Out(4,2) = phase_trans_Theta_V;
G = ['|H(jw)|=|Theta/V|= ',num2str(mag_trans_Theta_V)];
H = ['<)H(jw)=<)(Theta/V)= ',num2str(phase_trans_Theta_V),
'radians'];
disp(G)
disp(H)
disp(' ')
```

M-File 63: QC_AngConv.m

```
function [phi, theta]=QC_AngConv(File, OAL)
% function [phi, theta]=QC_AngConv(V_Xerr, V_Yerr, OAL)
% INPUTS
% V_Xerr:  error voltage from quad cell indicating movement in the
horizontal
% direction
% V_Yerr:  error voltage from quad cell indicating movement in the
vertical
% direction
% OAL:  Optical Arm Length from MEMS mirror to Quad cell
%
% ANGLE = atan(QC_Displacment/OAL)
%
% OUTPUTS
% phi (movment in the outter axis)
% theta (movement in the inner axis)
%
% Clinton Edwards, JHU-APL
% 20 April 2007


load QuadCellCal.mat
V_Xerr=File(:,1);
V_Yerr=File(:,2);

Norm_X_Volt=data(:,1);
Delta_X=data(:,2);
Norm_Y_Volt=data(:,3);
Delta_Y=data(:,4);

Interp_Delta_X = interp1(Norm_X_Volt,Delta_X,V_Xerr);
Interp_Delta_Y = interp1(Norm_Y_Volt,Delta_Y,V_Yerr);

phi = atan(Interp_Delta_X/OAL);
theta = atan(Interp_Delta_Y/OAL);
```

M-File 64: exp_volt_settings.m

```
function [Vhv1,Vhv2,Vdc,Vpp,Vb] = exp_volt_settings(V1, V2, Vamp)
% function exp_volt_settings(V1, V2, Vamp, Vb)
% example function call: [Vhv1,Vhv2,Vdc,Vpp,Vb] =
exp_volt_settings(45, 55, 5)
%
% This function takes user inputs of average voltages on each pad
for the
% mems device (1D) model and outputs the voltage settings for lab
equipment
%
% INPUTS:
%   V1 - Average voltage on pad 1
%   V2 - Average voltage on pad 2
%   Vamp - Voltage amplitude modulating each mirror
%
% OUTPUT:
%   Vhv1 - Voltage for high voltage piezo amplifier setting #1
%   Vhv2 - Voltage for high voltage piezo amplifier setting #2
%   Vpp - Voltage setting for HP function generator which is set for
a 50 ohm load and outputs twice of what is required
%   Vdc - Voltage setting for DC power supply
%   Vb - Bias Voltage which is the average of
%
% Lab Instrumentation
%   Power Supply:  Agilent E3631A
%   Func. Gen:  Agilent 33250A
%   High Volt Amp:  Thorlabs MDT693A
%
%   Clinton Edwards, JHU-APL
%   24 April 2007

Vb = (V1+V2)/2;

Vdc = -Vamp/10;

Vpp = Vamp/10;

Vhv1 = V1-Vamp;

Vhv2 = V2-Vamp;
```

M-File 65: TransAnalysis2.m

```
% TransAnalysis2.m
% comparison of transient analysis of experimental measurements and
% simulink model
%
% MATLAB
%
%
% Clint Edwards, JHU-APL
% 5 May 2007
load Trans35_65_1Vpp.txt
global param_2D


% param_2D = [Vb,ho,x1,x2,z1,z2,I1,D1,Km1,I2,D2,Km2];
I1 = param_2D(7);
I2 = param_2D(10);
Km1 = param_2D(9);
Km2 = param_2D(12);

N = 25000;  %number of points in fft
sam_freq = 25000;
T = 1/sam_freq;  % sampling period for time-series (ts) data

A = Trans35_65_1Vpp(:,2);
B = Trans35_65_1Vpp(:,1);
index=[0:1:length(A)-1]*T;  %time index
[phi, theta]=QC_AngConv(Trans35_65_1Vpp, .53);
delta_time = 435;%account for the offset in simulation vs.
measurement time

SCALE = 50;

sim_time = Sim_out.time;
sim_phi = Sim_out.signals.values(:,1);
sim_theta = Sim_out.signals.values(:,2);
simSCALE = 0.0075;

figure; plot(index(1:end-delta_time),SCALE*(phi(delta_time:end-1)-
mean(phi)),sim_time,simSCALE*SCALE*(sim_phi-
mean(sim_phi)),'linewidth',2)
% hold on  %cross coupling plot
% plot(index(1:end-delta_time),SCALE*(theta(delta_time:end-1)-
mean(theta)),sim_time,simSCALE*SCALE*(sim_theta-
mean(sim_theta)),'linewidth',2)
xlabel('time (s)')
ylabel('\phi (deg.)')
%axis([0.1 0.44 -.18 .18])
```

```
nat_freq1 = sqrt(Km1/I1)
gamma1 = D1/(2*sqrt(Km1*I1))

% init_2D_file.m that fits this data the best...
% global param_2D
%
% Vb = 50;  % (v)
% ho = 35;  % (um)
% x1 = 1;   % (um)
% x2 = 260; % (um)
% z1 = 1;   % (um)
% z2 = 260; % (um)
%
% % INNER AXIS DYNAMICS PARAMETERS:
% I2 = 1.14e-4; %kg*um^2
% D2 = 0.5;  %uN*um*s
% Km2 = 35000;  %uN*um
%
% % OUTTER AXIS DYNAMICS PARAMETERS:
% I1 = 1.84e-4; %kg*um^2
% D1 = .87;  %uN*um*s
% Km1 = 15500;  %uN*um
%
% param_2D = [Vb,ho,x1,x2,z1,z2,I1,D1,Km1,I2,D2,Km2];
%
```

M-File 66: SS_Fitting.m

```
function SS_Fitting(L,Km)
% Fitting steady-state measurements of MEMS Optical Mirror to 1D
Torque
% Model.
%
% Clint Edwards, JHU-APL

global param_1D

load SteadyStateMeasurements1.txt  %load steady-state (SS)
measurements
SSdata = SteadyStateMeasurements1;  %rename to simpler name
volt = SSdata(:,1);       % voltage data
SSphi = SSdata(:,2);
SStheta = SSdata(:,3);

%Fitting for Phi
Vb = 50;  % (v)
ho = 30;  % (um)
%L = 520*1; % (um)
I = 1.84e-4; % (Kg*um^2)
D = 0.497; % (uN*um*s)
%Km = 22200; % (uN*um)
% param_1D = [Vb,ho,L,I,D,Km];
 volt_fit=[-0:.5:40];
% phi_fit = zeros(length(volt_fit),1);
% for k=1:length(volt_fit)
% phi_fit(k,1) = phi_given_v(volt_fit(k))*180/pi;
% end
% figure; plot(volt,SSphi,'or',volt_fit,phi_fit,'b')
% xlabel('actuating voltage "u"')
% ylabel('\phi (deg.)')
% title('Fitting of Steady-State \phi vs. u')

% Fitting for Theta
Vb = 50;  % (v)
%L = 560; % (um)
I = 1.14e-4; % (Kg*um^2)
D = 0.297; % (uN*um*s)
% Km = 45200; % (uN*um)
param_1D = [Vb,ho,L,I,D,Km];
theta_fit = zeros(length(volt_fit),1);
for m=1:length(volt_fit)
theta_fit(m,1) = phi_given_v(volt_fit(m))*180/pi;
end
figure; plot(volt,SStheta,'or',volt_fit,theta_fit,'b')
xlabel('actuating voltage "v"')
ylabel('\theta (deg.)')
% title('Fitting of Steady-State \theta vs. v')
```

```
% %polynomial fitting
% p = polyfit(volt,SSphi,3)
% x2 = -49:1:49;
% y2 = polyval(p,x2);
% figure; plot(volt,SSphi,'*',x2,y2)
% grid on
%
%
% p = polyfit(volt,SStheta,3)
% x2 = -49:1:49;
% y2 = polyval(p,x2);
% figure; plot(volt,SStheta,'o',x2,y2)
% grid on
```

M-File 67: SS_Fitting_Abr_phi.m

```
function SS_Fitting_Abr_phi(L,Km)
% Fitting steady-state measurements of MEMS Optical Mirror to 1D
Torque
% Model.
%
% Clint Edwards, JHU-APL

global param_1D

load Abr_SteadyStateMeasurementsCPI.txt  %load steady-state (SS)
measurements
SSdata = Abr_SteadyStateMeasurementsCPI;  % rename CPI Close To
Pull-in
% load Abr_SteadyStateMeasurementsCTZ.txt  %load steady-state (SS)
measurements
% SSdata = Abr_SteadyStateMeasurementsCTZ;  %rename to simpler name
CTZ "Close To Zero"
volt = SSdata(:,1);       % voltage data
SSphi = SSdata(:,2);
%SStheta = SSdata(:,3);

% Fitting for Phi
Vb = 50;  % (v)
ho = 30;  % (um)
I = 1.84e-4; % (Kg*um^2)
D = 0.497; % (uN*um*s)
param_1D = [Vb,ho,L,I,D,Km];
volt_fit=[0:.25:50];
phi_fit = zeros(length(volt_fit),1);
for k=1:length(volt_fit)
phi_fit(k,1) = phi_given_v(volt_fit(k))*180/pi;
end
figure; plot(volt,SSphi,'or',volt_fit,phi_fit,'b')
xlabel('actuating voltage "u"')
ylabel('\phi (deg.)')

% % Fitting for Theta
% Vb = 50;  % (v)
% I = 1.14e-4; % (Kg*um^2)
% D = 0.297; % (uN*um*s)
% param_1D = [Vb,ho,L,I,D,Km];
% theta_fit = zeros(length(volt_fit),1);
% for m=1:length(volt_fit)
% theta_fit(m,1) = phi_given_v(volt_fit(m))*180/pi;
% end
% % figure;
% plot(volt,SStheta,'or',volt_fit,theta_fit,'b')
% xlabel('actuating voltage "v"')
% ylabel('\theta (deg.)')
```

```
% hold on
%
```

M-File 68: SS_Fitting_Abr_theta.m

```
function SS_Fitting_Abr_theta(L,Km)
% Fitting steady-state measurements of MEMS Optical Mirror to 1D
Torque
% Model.
%
% Clint Edwards, JHU-APL

global param_1D

% load Abr_SteadyStateMeasurementsCPI.txt  %load steady-state (SS)
measurements
% SSdata = Abr_SteadyStateMeasurementsCPI;  % rename CPI Close To
Pull-in
load Abr_SteadyStateMeasurementsCTZ.txt  %load steady-state (SS)
measurements
SSdata = Abr_SteadyStateMeasurementsCTZ;  %rename to simpler name
CTZ "Close To Zero"
volt = SSdata(:,1);       % voltage data
SSphi = SSdata(:,2);
SStheta = SSdata(:,3);

% Fitting for Phi
Vb = 50;  % (v)
ho = 30;  % (um)
volt_fit=[0:.25:50];
I = 1.14e-4; % (Kg*um^2)
D = 0.297; % (uN*um*s)
param_1D = [Vb,ho,L,I,D,Km];
theta_fit = zeros(length(volt_fit),1);

for m=1:length(volt_fit)
theta_fit(m,1) = phi_given_v(volt_fit(m))*180/pi;
end
% figure;
plot(volt,SStheta,'or',volt_fit,theta_fit,'b')
xlabel('actuating voltage "v"')
ylabel('\theta (deg.)')
hold on
```

M-File 69: MEMS_OPTICAL_Fit.m

```
% MEMS_OPTICAL_Fit.m
% Fitting MEMS Optical Mirror to 1D Torque (Measurements made by
MEMS OPTICAL INC.)
% Model.
% 1 May 2007
% Clint Edwards, JHU-APL

global param_1D

load MEMSOPTICAL_theta.mat   %load steady-state (SS) measurements
load MEMSOPTICAL_phi.mat

SSdata_phi = MEMSOPTICAL_phi;  %rename to simpler name
SSphi = SSdata_phi(:,1);
volt_phi = SSdata_phi(:,2);       % voltage data

SSdata_theta = MEMSOPTICAL_theta;
SStheta = SSdata_theta(:,1);
volt_theta = SSdata_theta(:,2);

%plot(volt_phi, SSphi, 'or', volt_theta, SStheta, 'ob')  %Looks like
this
%is plotting okay

%Fitting for Phi
Vb = 40;  % (v)
ho = 35;  % (um)
L = 525; % (um)
I = 1.84e-4; % (Kg*um^2)
D = 0.497; % (uN*um*s)
Kmphi = 12555; % (uN*um)

%Fitting for Phi
volt_fit=[0:.5:33];
phi_fit = zeros(length(volt_fit),1);
param_1D = [Vb,ho,L,I,D,Kmphi];

for k=1:length(volt_fit)
   phi_fit(k,1) = phi_given_v(volt_fit(k))*180/pi;
end
figure; plot(volt_phi,SSphi,'or',volt_fit,phi_fit,'b')
xlabel('actuating voltage "u"')
ylabel('\phi (deg.)')
title('Fitting of MEMS Optical Inc. Measurements Steady-State \phi
vs. u')

error = zeros(length(SSphi),1);
phi_fit_error = zeros(length(SSphi),1);
```

```
%Fitting for Theta
volt_fit=[0:1:43];
theta_fit = zeros(length(volt_fit),1);
Kmtheta = 20000;
L=558;
param_1D = [Vb,ho,L,I,D,Kmtheta];
for k=1:length(volt_fit)
   theta_fit(k,1) = phi_given_v(volt_fit(k))*180/pi;
end
figure; plot(volt_theta,SStheta,'om',volt_fit,theta_fit,'b')
xlabel('actuating voltage "v"')
ylabel('\theta (deg.)')
title('Fitting of MEMS Optical Inc. Measurements Steady-State \theta
vs. v')

error = zeros(length(SStheta),1);
theta_fit_error = zeros(length(SStheta),1);
```

M-File 70: MEMS_OPTICAL_Fit_opt.m

```
function [output]=MEMS_OPTICAL_Fit_opt(x)
% Fitting MEMS Optical Mirror to 1D Torque (Measurements made by
MEMS OPTICAL INC.)
% Model.
% 1 May 2007
% Clint Edwards, JHU-APL

global param_1D

load MEMSOPTICAL_theta.mat   %load steady-state (SS) measurements
load MEMSOPTICAL_phi.mat

L=x(1);
Kmtheta=x(2);
% SSdata_phi = MEMSOPTICAL_phi;  %rename to simpler name
% SSphi = SSdata_phi(:,1);
% volt_phi = SSdata_phi(:,2);       % voltage data

SSdata_theta = MEMSOPTICAL_theta;
SStheta = SSdata_theta(:,1);
volt_theta = SSdata_theta(:,2);

%plot(volt_phi, SSphi, 'or', volt_theta, SStheta, 'ob')  %Looks like
this
%is plotting okay

% %Fitting for Phi
Vb = 40;  % (v)
ho = 35;  % (um)
%L = 520; % (um)
I = 1.84e-4; % (Kg*um^2)
D = 0.497; % (uN*um*s)
%Kmphi = 12152; % (uN*um)

%Fitting for Phi
%volt_fit=[0:.5:33];
% phi_fit = zeros(length(volt_phi),1);
% param_1D = [Vb,ho,L,I,D,Kmphi];
%
% for k=1:length(volt_phi)
%    phi_fit(k,1) = phi_given_v(volt_phi(k))*180/pi;
% end
% output = sum(abs(phi_fit-SSphi));


%Fitting for Theta
%volt_theta=[0:1:43];
theta_fit = zeros(length(volt_theta),1);
```

```
% Kmtheta = 15292;
param_1D = [Vb,ho,L,I,D,Kmtheta];
for k=1:length(volt_theta)
   theta_fit(k,1) = phi_given_v(volt_theta(k))*180/pi;
end

output = sum(abs(theta_fit-SStheta));
```

M-File 71: Disp_AngAccel.m

```
% Disp_AngAccel.m

load AngAccel_X_Floating.txt
load AngAccel_Y_Floating.txt

G = 10;  %Accelerometer Gain 100mV/g
g = 9.81;
ST = 1/(25000);

Ax = AngAccel_X_Floating*G*g;
Ay = AngAccel_Y_Floating*G*g;
Ax = Ax-mean(Ax);
Ay = Ay-mean(Ay);

DisplacementX = (1/2)*Ax*ST^2;
DisplacementY = (1/2)*Ay*ST^2;

angX = atan(DisplacementX/.025);
angY = atan(DisplacementY/.025);

mu_angX = mean(angX)
std_angX = std(angX)
mu_angY = mean(angY)
std_angY = std(angY)

figure;
plot(angX)
OUTPUT = Standard_PSD(1/ST,5e4,angX);
figure;
plot(OUTPUT(:,1),OUTPUT(:,2))

figure;
plot(angX)
OUTPUT = Standard_PSD(1/ST,5e4,angY);
figure;
plot(OUTPUT(:,1),OUTPUT(:,2))
```

M-File 72: plot_PSD_AngAccel.m

```
% plot_PSD_AngAccel.m

load AngAccel_X_Floating.txt
X = AngAccel_X_Floating;
Xn = X*(10);
plot(Xn)
OUTPUT = Standard_PSD(1/25000,1e4,Xn);
OUTPUT = Standard_PSD(25000,1e4,Xn);
OUTPUT = Standard_PSD(1/25000,1e4,Xn);
plot(OUTPUT(:,1),OUTPUT(:,2))
OUTPUT = Standard_PSD(25000,1e4,Xn);
plot(OUTPUT(:,1),OUTPUT(:,2))
```

M-File 73: ts_quadcell_swept.m

```
% ts_quadcell_swept.m
% Input text file from Quad Cell and then take the fft and plot it
with
% MATLAB
%
% Plot of both the direct coupling noise and cross coupling LR
acutation.
%
% Clint Edwards, JHU-APL
% 15 April 2007
load temp.txt

N = 1e4;  %number of points in fft
sam_freq = 1e4;
T = 1/sam_freq;  % sampling period for time-series (ts) data

A = temp(:,2);
B = temp(:,1);

V1 = temp(:,3);
V2 = temp(:,4);
V3 = temp(:,5);
V4 = temp(:,6);
index=[0:1:length(A)-1]*T;

fftA = fftshift(T*fft(A-mean(A),N));
fft_index = [-N/2+1:1:(N)/2]*(1/(N*T));
plot(fft_index,abs(fftA(1:N)),'b')
hold on

fftB = fftshift(T*fft(B-mean(B),N));
fft_index = [-N/2+1:1:(N)/2]*(1/(N*T));
plot(fft_index,abs(fftB(1:N)),'r')
xlabel('Frequency (Hz)')
ylabel('\phi (red) PSD and \theta (blue) PSD (Watts/sqrt(Hz))')
title('2-Axis Tip-Tilt MEMS Mirror Freq. Resp.')

hold off
```

M-File 74: CrossSpectrums.m

```
% CrossSpectrums.m
% Compares the square of FFT of time-series data and the FFT of the
bias correlation
% of time-series data.  They are equivalent functions.

load SS45_55data.txt
data = SS45_55data;
[A, B]=QC_AngConv(data, .53);

N = 1e4;  %number of points in fft
sam_freq = 1e4;
T = 1/sam_freq;  % sampling period for time-series (ts) data


A = data(:,1)-mean(data(:,1));   %phi movement (voltage)
B = data(:,2)-mean(data(:,2));   %theta movement (voltage)
V1 = data(:,3)-mean(data(:,3));
V2 = data(:,4)-mean(data(:,4));
V3 = data(:,5)-mean(data(:,5));
V4 = data(:,6)-mean(data(:,6));
index=[0:1:length(A)-1]*T;
fft_index = [-N/2+1:1:(N)/2]*(1/(N*T));

% [BV1corr, lagsB] = xcorr(B(1:1e4,1),V1(1:1e4,1),'biased');
% figure; plot(lagsB,BV1corr)
% [BV2corr, lagsB] = xcorr(B(1:1e4,1),V2(1:1e4,1),'biased');
% figure; plot(lagsB,BV2corr)
% [BV3corr, lagsB] = xcorr(B(1:1e4,1),V3(1:1e4,1),'biased');
% figure; plot(lagsB,BV3corr)
% [BV4corr, lagsB] = xcorr(B(1:1e4,1),V4(1:1e4,1),'biased');
% figure; plot(lagsB,BV4corr)
%
% [AV1corr, lagsA] = xcorr(A(1:1e4,1),V1(1:1e4,1),'biased');
% figure; plot(lagsA,AV1corr)
% [AV2corr, lagsA] = xcorr(A(1:1e4,1),V2(1:1e4,1),'biased');
% figure; plot(lagsA,AV2corr)
% [AV3corr, lagsA] = xcorr(A(1:1e4,1),V3(1:1e4,1),'biased');
% figure; plot(lagsA,AV3corr)
% [AV4corr, lagsA] = xcorr(A(1:1e4,1),V4(1:1e4,1),'biased');
% figure; plot(lagsA,AV4corr)

[ABcorr, lagsAB] = xcorr(A(1:1e4,1),B(1:1e4,1),'biased');
figure; plot(lagsAB,ABcorr)
```

M-File 75: PSDphi_HuPSDu.m

```
% PSDphi_HuPSDu.m
% Input text file from Quad Cell and then take the fft and plot it
with
% MATLAB
%
% Plot of both the direct coupling noise and cross coupling LR
acutation.
%
% Clint Edwards, JHU-APL
% 15 April 2007
load Battery14V_50VBias_6June2007_1.txt
temp = Battery14V_50VBias_6June2007_1;
N = 1e4;  %number of points in fft
sam_freq = 1e4;
T = 1/sam_freq;  % sampling period for time-series (ts) data

% assign column in data text file to
V1 = temp(:,3);
V2 = temp(:,4);
V3 = temp(:,5);
V4 = temp(:,6);
u = V1-mean(V1);
Var_u = mean(u.^2)
Var_u = 8e-11
figure
plot(u)

% convert from voltages to angles for phi and theta
[phi, theta]=QC_AngConv(temp, .53);

win = hanning(N);
shift = 500;
NN = floor(length(phi)/shift);
PSD_phi = zeros(NN,N);
PSD_theta = zeros(NN,N);
PSD_u = zeros(NN,N);
index=[0:1:length(phi)-1]*T;  %time index
fft_index = [-N/2+1:1:(N)/2]*(1/(N*T));  %freq index
nf = 1;  %smoothing filtering 1= no filtering

for ii=1:NN-ceil(N/shift)
   phi_t=win.*phi((ii-1)*shift+1:N+(ii-1)*shift);
   theta_t=win.*theta((ii-1)*shift+1:N+(ii-1)*shift);
   ut = win.*u((ii-1)*shift+1:N+(ii-1)*shift);
   PSD_phi(ii,:) = abs(fftshift(T*fft(phi_t-mean(phi_t),N))).^2;
   PSD_theta(ii,:) = abs(fftshift(T*fft(theta_t-
mean(theta_t),N))).^2;
   PSD_u(ii,:) = abs(fftshift(T*fft(ut,N))).^2;
end
```

```
% average multiple PSD
PSD_phi = sum(PSD_phi)*(1/NN);
PSD_u = sum(PSD_u)*(1/NN);
figure; plot(fft_index,PSD_u,'k'); hold on;
xlabel('freq (Hz)')
ylabel('PSD of u Voltage')
% axis([1000 2500 0 4e-8])

PSD_phi=filter((1/nf)*ones(nf,1),1,PSD_phi);
PSD_u=filter((1/nf)*ones(nf,1),1,PSD_u);
figure; plot(fft_index,PSD_phi,'r'); %hold on;
% axis([1000 2500 0 4e-15])

%linearized approximation of PSD_phi
u = .5*mean(V1-V3)
v = .5*mean(V1-V2)
Tran_Hu = Hu([u, v],fft_index);
figure; plot(fft_index,abs(Tran_Hu).^2)

EST_PSD_phi=((abs(Tran_Hu)).^2).*10^(-10.27);
figure;
plot(fft_index,EST_PSD_phi)
% axis([-2500 2500 0 4e-15])
%title('Estimation of S\phi Given Su(jw)')
%plot_index = [5000+1200:5000+2400];
 figure;
plot(fft_index,2*abs(PSD_phi),'b',fft_index,2*abs(EST_PSD_phi),'r')
% xlabel('Frequency (Hz)')
% ylabel('\phi PSD (Watts/sqrt(Hz))')
% axis([0 5000 0 1e-10])
```

M-File 76: D_lookup.m

```
function OUTPUT = D_lookup(Pressure)
% function OUTPUT = D_lookup(Pressure)
% Pressure should be units of torr
%
% Clinton L. Edwards, JHU-APL
% 3 July 2007
load DampingCoeffData.txt

Pdat = DampingCoeffData(:,1);
D1dat = DampingCoeffData(:,2);
D2dat = DampingCoeffData(:,3);

% YI = INTERP1(X,Y,XI) interpolates to find YI, the values of the
%      underlying function Y at the points in the array XI. X must
be a
%      vector of length N.
D1 = interp1(Pdat,D1dat, Pressure);
D2 = interp1(Pdat,D2dat, Pressure);

OUTPUT = [D1;D2];


I2 = 1.14e-4; %kg*um^2
Km2 = 22461;  %uN*um

% OUTTER AXIS DYNAMICS PARAMETERS:
I1 = 1.84e-4; %kg*um^2
Km1 = 22461;  %uN*um
w1 = 1680;
w2=2100;

Q1 = I1*w1./D1
Q2 = I2*w2./D2
```

M-File 77: PressureInvariancePlot.m

```
% PressureInvariancePlot.m
% Clinton Edwards, JHU-APL

load PressureInvariance.txt
data = PressureInvariance;

noise = .5e-14*randn(length(data(:,1)),2);
data(:,2) = 1.79e-13+noise(:,2);
data(:,3) = 2.16e-13+noise(:,1);

% Plot Pressure Invariance
loglog(data(:,1),data(:,2),'^-r','linewidth',2)
hold on;
loglog(data(:,1),data(:,3),'o-b','linewidth',2)
xlabel('Pressure (Torr)')
ylabel('Variance (Radians^2)')
title('Pressure Invariance of Jitter')
legend('Var(\phi)','Var(\theta)')
axis([6.3e-6 7.6e2 4e-14 1e-12])
```

M-File 78: Var_ang.m

```
% VAR_ang.m
% Averaged FFT of MEMS OPTICAL Inc. Micro-mirror facet jitter on 8
May 2007
% Lorenzian should develope over the resonant frequency as subsequen
fft's
% are averaged.
% frequency domain is integrated over to determine the variance of
the
% jitter in the angles phi and theta
%
% 2 July 2007
% Clint Edwards, JHU-APL

load D760.txt
temp = D760;
sw1 = temp(:,1);
sw2 = temp(:,2);
temp(:,1) = sw2;
temp(:,2) = sw1;
len = length(temp(:,1));
V1 = D760(:,3);
V2 = D760(:,4);
V3 = D760(:,5);
V4 = D760(:,6);
std_V1 = std(V1)
std_V2 = std(V2)
std_V3 = std(V3)
std_V4 = std(V4)
std_phi = std(temp(:,1))
std_theta = std(temp(:,2))

[phi, theta]=QC_AngConv(temp, .75);  % I think I recorded the data
files with the phi and theta reversed
% std_V = std(V1)
% std_phi = std(phi)

N = 1e4;  %number of points in fft
sam_freq = 1e4;
T = 1/sam_freq;  % sampling period for time-series (ts) data
A = phi;
B = theta;
OUTPUTphi = Standard_PSD(sam_freq,N,phi);
OUTPUTtheta = Standard_PSD(sam_freq,N,theta);
freq = OUTPUTphi(:,1);
Sphi = OUTPUTphi(:,2);
Stheta = OUTPUTtheta(:,2);
figure; plot(freq,Sphi,freq,Stheta)
xlabel('Frequency (Hz)')
ylabel('PSD (Watts/Hz)')
title('MEMS Mirror Freq. Resp. to Air')
```

```
axis([1000 3000 0 8e-16])

delta = abs(freq(1,1)-freq(2,1));
p1 = find(freq==5e2);
p2 = find(freq==3e3);
% Var_phi = 4*pi*delta*(sum(Sphi(1:10)))
% Var_theta = 4*pi*delta*(sum(Stheta(1:10)))
Var_phi = 2*2*pi*delta*(sum(Sphi(p1:p2)))
Var_theta = 2*2*pi*delta*(sum(Stheta(p1:p2)))
```

M-File 79: Lin2D_Est.m

```
% Lin2D_Est.m
% 28 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for the inputs for voltage noise and vibration
noise.
% The different betweent the output and linearized model is
attributed to
% brownian motion of air or the mechanical thermal noise caused by
% molecular bombardment.m
% sim('Torque2D_SIMULINK.mdl')
% MeasFileOUTPUT
load temp.txt  %file from experimental measurements
data = temp;
Pressure = 760;
init_2D_file
Ds = D_lookup(Pressure);
D1 = Ds(1);
D2 = Ds(2);
ST = 1/(5e4);
I1 = param_2D(7);
% D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
% D2 = param_2D(11);
Km2 = param_2D(12);

phi = data(:,1);
theta = data(:,2);
V1 = data(:,3);
V2 = data(:,4);
V3 = data(:,5);
V4 = data(:,6);
VibPhi = data(:,7);
VibTheta = data(:,7);

pt = round(.9*length(V1));   % avoid transient values when
aproximating deterministic u,v,phi,theta
V1n = mean(V1);
V2n = mean(V2);
V3n = mean(V3);
V4n = mean(V4);

Vb = (1/4)*(V1n+V2n+V3n+V4n);
u = (V3+V4-2*Vb)/2;
v = (V2+V3 -2*Vb)/(-2);
u0 = (V3n+V4n-2*Vb)/2;
v0 = (V2n+V3n -2*Vb)/(-2);
```

```
volt0 = [u0; v0];

%Calculate Constants for Linearized Model (K1theta, K2theta, K1phi,
K2phi, K1u, K2u, K1v, K2v,)
K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v

fft_len = 1e4;  %Length of FFTs in PSD
OUTPUTu = Standard_PSD(1/ST,fft_len,u);
OUTPUTv = Standard_PSD(1/ST,fft_len,v);
OUTPUTphi = Standard_PSD(1/ST,fft_len,phi);
OUTPUTtheta = Standard_PSD(1/ST,fft_len,theta);
OUTPUTuv = Cross_PSD(1/ST,fft_len,u,v);
OUTPUTvibphi = Standard_PSD(1/ST,fft_len,VibPhi);
OUTPUTvibtheta = Standard_PSD(1/ST,fft_len,VibTheta);
OUTPUTvibphitheta = Cross_PSD(1/ST,fft_len,VibPhi,VibTheta);
OUTPUTvibthetaphi = Cross_PSD(1/ST,fft_len,VibTheta,VibPhi);

freq = OUTPUTu(:,1);

Su = OUTPUTu(:,2);  %Su(jw) PSD
Sv = OUTPUTv(:,2);  %Sv(jw) PSD
Sphi = OUTPUTphi(:,2);  %Sphi(jw) PSD
Stheta = OUTPUTtheta(:,2);  %Stheta(jw) PSD
Suv = OUTPUTuv(:,2);  %Suv(jw) Cross PSD
Svu = Suv; % real valued sequence
Svibphi = OUTPUTvibphi(:,2);
Svibtheta = OUTPUTvibtheta(:,2);
Svibphitheta = OUTPUTvibphitheta(:,2);
Svibthetaphi = OUTPUTvibphitheta(:,2);
SphiBM = .3*(1/(2*pi))*ST*4.38e-14;
SthetaBM = .2*(1/(2*pi))*ST*5.95e-14;
SphithetaBM = 0;
SthetaphiBM = 0;

w = freq*2*pi;
P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
P2=-w.^2*I2+j*w*D2+Km2-K2theta0;
Hjw = 1./(P1.*P2-K1theta0*K2phi0);
Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and Sw2(jw)
to Sphi(jw) and Stheta(jw)

w1_Suu =
abs((K1theta0^2)*(K2u0^2)+P2*K1theta0*K1u0*K2u0+conj(P2)*K1theta0*K1
u0*K2u0+(abs(P2).^2)*(K1u0^2));
```

```
w1_Svv =
abs((K1theta0^2)*(K2v0^2)+(P2*K1theta0*K1v0*K2v0)+(conj(P2)*K1theta0
*K1v0*K2v0)+(abs(P2).^2)*(K1v0^2));
w1_Suv =
abs(2*(K1theta0^2)*K2u0*K2v0+P2*K1theta0*K1u0*K2v0+P2*K1theta0*K1v0*
K2u0+conj(P2)*K1theta0*K1u0*K2v0...
   +K1v0*(conj(P2)*K1theta0*K2u0+2*(abs(P2).^2)*K1u0));
w1_Svibtheta = (K1theta0^2)*(Km2^2);
w1_Svibphi = (abs(P2).^2)*(Km1^2);
w1_Svibphitheta = (P2*K1theta0*Km1*Km2);
w1_Svibphithetac = (conj(P2)*K1theta0*Km1*Km2);
w1_SthetaBM = (K1theta0^2)*(Km2^2);
w1_SphiBM = (abs(P2).^2)*(Km1^2); ;
w1_SphithetaBM = 2*K1theta0*real(P2.*(Km1).*(Km2).*SphithetaBM);
%SphithetaBM

w2_Svv =
abs((K2phi0^2)*(K1v0^2)+P1*K2phi0*K2v0*K1v0+conj(P1)*K2phi0*K2v0*K1v
0+(abs(P1).^2)*(K2v0^2));
w2_Suu =
abs((K2phi0^2)*(K1u0^2)+(P1*K2phi0*K2u0*K1u0)+(conj(P1)*K2phi0*K2u0*
K1u0)+(abs(P1).^2)*(K2u0^2));
w2_Svu =
abs(2*(K2phi0^2)*K1v0*K1u0+P1*K2phi0*K2v0*K1u0+P1*K2phi0*K2u0*K1v0+c
onj(P1)*K2phi0*K2v0*K1u0...
   +K2u0*(conj(P1)*K2phi0*K1v0+2*(abs(P1).^2)*K2v0));
w2_Svibphi = (K2phi0^2)*(Km1^2);
w2_Svibtheta = (abs(P1).^2)*(Km2^2);
w2_Svibthetaphi = (P1*K2phi0*Km1*Km2);
w2_Svibthetaphic = (conj(P1)*K2phi0*Km1*Km2);
w2_SphiBM = (K2phi0^2)*(Km1^2);
w2_SthetaBM = (abs(P1).^2)*(Km2^2); ;
w2_SthetaphiBM = 2*K2phi0*real(P1.*(Km2).*(Km1).*SthetaphiBM);

Sphi_est = w1_Suu.*Hjw2.*Su+w1_Svv.*Hjw2.*Sv+w1_Suv.*Hjw2.*Suv+...

w1_SthetaBM.*Hjw2.*SthetaBM+w1_SphiBM.*Hjw2.*SphiBM+w1_SphithetaBM.*
Hjw2+...
   w1_Svibtheta.*Hjw2.*Svibtheta+w1_Svibphi.*Hjw2.*Svibphi...

+w1_Svibphitheta.*Hjw2.*Svibphitheta+w1_Svibphithetac.*Hjw2.*conj(Sv
ibphitheta);


Sphi_volt_est = w1_Suu.*Hjw2.*Su+w1_Svv.*Hjw2.*Sv+w1_Suv.*Hjw2.*Suv;
Sphi_bm_est =
w1_SthetaBM.*Hjw2.*SthetaBM+w1_SphiBM.*Hjw2.*SphiBM+w1_SphithetaBM.*
Hjw2;
Sphi_res = Sphi-Sphi_volt_est;
Sphi_vib_est =
w1_Svibtheta.*Hjw2.*Svibtheta+w1_Svibphi.*Hjw2.*Svibphi...
```

```
+w1_Svibphitheta.*Hjw2.*Svibphitheta+w1_Svibphithetac.*Hjw2.*conj(Sv
ibphitheta);

Stheta_est = w2_Svv.*Hjw2.*Sv+w2_Suu.*Hjw2.*Su+w2_Svu.*Hjw2.*Svu+...

w2_SphiBM.*Hjw2.*SphiBM+w2_SthetaBM.*Hjw2.*SthetaBM+w2_SthetaphiBM.*
Hjw2+...
   w2_Svibphi.*Hjw2.*Svibphi+w2_Svibtheta.*Hjw2.*Svibtheta+...

w2_Svibthetaphi.*Hjw2.*Svibthetaphi+w2_Svibthetaphic.*Hjw2.*conj(Svi
bthetaphi);

Stheta_volt_est =
w2_Svv.*Hjw2.*Sv+w2_Suu.*Hjw2.*Su+w2_Svu.*Hjw2.*Svu;
Stheta_bm_est =
w2_SphiBM.*Hjw2.*SphiBM+w2_SthetaBM.*Hjw2.*SthetaBM+w2_SthetaphiBM.*
Hjw2;
Stheta_res = Stheta-Stheta_volt_est;
Stheta_vib_est =
w2_Svibthetaphi.*Hjw2.*Svibthetaphi+w2_Svibthetaphic.*Hjw2.*conj(Svi
bthetaphi);

figure; plot(freq,Sphi,'b','linewidth',2); hold on;
plot(freq,Sphi_est,'r','linewidth',2)
plot(freq,Sphi_volt_est,'k','linewidth',2)
plot(freq,Sphi_vib_est,'m','linewidth',2)
plot(freq,Sphi_bm_est,'--g','linewidth',2)
plot(freq,Sphi_res,'c','linewidth',2)
xlabel('Frequency (Hz)')
ylabel('PSD \phi (W/Hz)')
title('Pressure 10 Torr')
axis([1200 2200 0 6e-15])
legend('S\phi','Model Est','Volt Comp','Vib Comp','MT Est')

figure; plot(freq,Stheta,'b','linewidth',2); hold on;
plot(freq,Stheta_est,'r','linewidth',2)
plot(freq,Stheta_volt_est,'k','linewidth',2)
plot(freq,Stheta_vib_est,'m','linewidth',2)
plot(freq,Stheta_bm_est,'--g','linewidth',2)
plot(freq,Stheta_res,'c','linewidth',2)
xlabel('Frequency (Hz)')
ylabel('PSD \theta (W/Hz)')
title('Pressure 10 Torr')
axis([1500 2500 0 6e-15])
legend('S\theta','Model Est','Volt Comp','Vib Comp','MT Est')
```

M-File 80: Lin2D_voltvib.m

```
% Lin2D_voltvib.m
% 28 June 2007
% Clint Edwards, JHU-APL
%
% Taking the 2D analytic model and outputs from SIMULINK and
caluclating the
% linearized output for the inputs.

load data.txt

Vb = 50;
I1 = param_2D(7);
D1 = param_2D(8);
Km1 = param_2D(9);
I2 = param_2D(10);
D2 = param_2D(11);
Km2 = param_2D(12);

phi = data(:,1);
theta = data(:,2);
V1 = data(:,3);
V2 = data(:,4);
V3 = data(:,5);
V4 = data(:,6);

pt = round(.9*length(u));   % avoid transient values when
aproximating deterministic u,v,phi,theta
V1n = mean(V1);
V2n = mean(V2);
V3n = mean(V3);
V4n = mean(V4);

% V1 = Vb+k*(v-u);
% V2 = Vb+k*(-v-u);
% V3 = Vb+k*(-v+u);
% V4 = Vb+k*(v+u);
u = V3n+V4n-2*Vb;
v = V2+V3 -2*Vb;

% u0 = mean(u(end-pt,end));   % deterministic u (average value)
% v0 = mean(v(end-pt,end));   % deterministic v (average value)
% volt0 = [u0; v0];
%
% %Calculate Constants for Linearized Model (K1theta, K2theta,
K1phi, K2phi, K1u, K2u, K1v, K2v,)
% K1phi0 = K1Phi(volt0);  %PD of K1 WRT phi
% K2phi0 = K2Phi_num(volt0);  %PD of K2 WRT phi
% K1theta0 = K1Theta_num(volt0);  %PD of K1 WRT theta
% K2theta0 = K2Theta(volt0);   %PD of K2 WRT theta
```

```
% K1u0 = K1U(volt0);   %PD of K1 WRT acutating voltage u
% K2u0 = K2U(volt0);   %PD of K2 WRT acutating voltage u
% K1v0 = K1V(volt0);   %PD of K1 WRT acutating voltage v
% K2v0 = K2V(volt0);   %PD of K2 WRT acutating voltage v
%
%
% fft_len = 5000;  %Length of FFTs in PSD
% OUTPUTu = Standard_PSD(1/ST,fft_len,u);
% OUTPUTv = Standard_PSD(1/ST,fft_len,v);
% OUTPUTphi = Standard_PSD(1/ST,fft_len,SimOutPhi);
% OUTPUTtheta = Standard_PSD(1/ST,fft_len,SimOutTheta);
% OUTPUTuv = Cross_PSD(1/ST,fft_len,u,v);
% OUTPUTvibphi = Standard_PSD(1/ST,fft_len,VibPhi);
% OUTPUTvibtheta = Standard_PSD(1/ST,fft_len,VibTheta);
% OUTPUTvibphitheta = Cross_PSD(1/ST,fft_len,VibPhi,VibTheta);
% OUTPUTvibthetaphi = Cross_PSD(1/ST,fft_len,VibTheta,VibPhi);
% OUTPUTphiBM = Standard_PSD(1/ST,fft_len,BMPhi);
% OUTPUTthetaBM = Standard_PSD(1/ST,fft_len,BMTheta);
% OUTPUTphithetaBM = Cross_PSD(1/ST,fft_len,BMPhi,BMTheta);
% OUTPUTthetaphiBM = Cross_PSD(1/ST,fft_len,BMTheta,BMPhi);
% freq = OUTPUTu(:,1);
%
% Su = OUTPUTu(:,2);  %Su(jw) PSD
% Sv = OUTPUTv(:,2);  %Sv(jw) PSD
% Sphi = OUTPUTphi(:,2);  %Sphi(jw) PSD
% Stheta = OUTPUTtheta(:,2);  %Stheta(jw) PSD
% Suv = OUTPUTuv(:,2);  %Suv(jw) Cross PSD
% Svu = Suv; % real valued sequence
% Svibphi = OUTPUTvibphi(:,2);
% Svibtheta = OUTPUTvibtheta(:,2);
% Svibphitheta = OUTPUTvibphitheta(:,2);
% Svibthetaphi = OUTPUTvibphitheta(:,2);
% SphiBM = OUTPUTphiBM(:,2);
% SthetaBM = OUTPUTthetaBM(:,2);
% SphithetaBM = OUTPUTphithetaBM(:,2);
% SthetaphiBM = OUTPUTphithetaBM(:,2);
%
%
% w = freq*2*pi;
% P1=-w.^2*I1+j*w*D1+Km1-K1phi0;
% P2=-w.^2*I2+j*w*D2+Km2-K2theta0;
% Hjw = 1./(P1.*P2-K1theta0*K2phi0);
% Hjw2 = abs(Hjw).^2;  %Transfer function relating Sw1(jw) and
Sw2(jw) to Sphi(jw) and Stheta(jw)
%
% w1_Suu =
abs((K1theta0^2)*(K2u0^2)+P2*K1theta0*K1u0*K2u0+conj(P2)*K1theta0*K1
u0*K2u0+(abs(P2).^2)*(K1u0^2));
% w1_Svv =
abs((K1theta0^2)*(K2v0^2)+(P2*K1theta0*K1v0*K2v0)+(conj(P2)*K1theta0
*K1v0*K2v0)+(abs(P2).^2)*(K1v0^2));
```

```
% w1_Suv =
abs(2*(K1theta0^2)*K2u0*K2v0+P2*K1theta0*K1u0*K2v0+P2*K1theta0*K1v0*
K2u0+conj(P2)*K1theta0*K1u0*K2v0...
%    +K1v0*(conj(P2)*K1theta0*K2u0+2*(abs(P2).^2)*K1u0));
% w1_Svibtheta = (K1theta0^2)*(Km2^2);
% w1_Svibphi = (abs(P2).^2)*(Km1^2);
% w1_Svibphitheta = (P2*K1theta0*Km1*Km2);
% w1_Svibphithetac = (conj(P2)*K1theta0*Km1*Km2);
% w1_SthetaBM = (K1theta0^2)*(Km2^2);
% w1_SphiBM = (abs(P2).^2)*(Km1^2); ;
% w1_SphithetaBM = 2*K1theta0*real(P2.*(Km1).*(Km2).*SphithetaBM);
%SphithetaBM
%
% w2_Svv =
abs((K2phi0^2)*(K1v0^2)+P1*K2phi0*K2v0*K1v0+conj(P1)*K2phi0*K2v0*K1v
0+(abs(P1).^2)*(K2v0^2));
% w2_Suu =
abs((K2phi0^2)*(K1u0^2)+(P1*K2phi0*K2u0*K1u0)+(conj(P1)*K2phi0*K2u0*
K1u0)+(abs(P1).^2)*(K2u0^2));
% w2_Svu =
abs(2*(K2phi0^2)*K1v0*K1u0+P1*K2phi0*K2v0*K1u0+P1*K2phi0*K2u0*K1v0+c
onj(P1)*K2phi0*K2v0*K1u0...
%    +K2u0*(conj(P1)*K2phi0*K1v0+2*(abs(P1).^2)*K2v0));
% w2_Svibphi = (K2phi0^2)*(Km1^2);
% w2_Svibtheta = (abs(P1).^2)*(Km2^2);
% w2_Svibthetaphi = (P1*K2phi0*Km1*Km2);
% w2_Svibthetaphic = (conj(P1)*K2phi0*Km1*Km2);
% w2_SphiBM = (K2phi0^2)*(Km1^2);
% w2_SthetaBM = (abs(P1).^2)*(Km2^2); ;
% w2_SthetaphiBM = 2*K2phi0*real(P1.*(Km2).*(Km1).*SthetaphiBM);
%
% Sphi_est = w1_Suu.*Hjw2.*Su+w1_Svv.*Hjw2.*Sv+w1_Suv.*Hjw2.*Suv+...
%    w1_Svibtheta.*Hjw2.*Svibtheta+w1_Svibphi.*Hjw2.*Svibphi...
%
+w1_Svibphitheta.*Hjw2.*Svibphitheta+w1_Svibphithetac.*Hjw2.*conj(Sv
ibphitheta)+...
%
w1_SthetaBM.*Hjw2.*SthetaBM+w1_SphiBM.*Hjw2.*SphiBM+w1_SphithetaBM.*
Hjw2;
%
% Sphi_volt_est =
w1_Suu.*Hjw2.*Su+w1_Svv.*Hjw2.*Sv+w1_Suv.*Hjw2.*Suv;
% Sphi_bm_est =
w1_SthetaBM.*Hjw2.*SthetaBM+w1_SphiBM.*Hjw2.*SphiBM+w1_SphithetaBM.*
Hjw2;
% Sphi_vib_est =
w1_Svibtheta.*Hjw2.*Svibtheta+w1_Svibphi.*Hjw2.*Svibphi...
%
+w1_Svibphitheta.*Hjw2.*Svibphitheta+w1_Svibphithetac.*Hjw2.*conj(Sv
ibphitheta);
%
% Stheta_est =
w2_Svv.*Hjw2.*Sv+w2_Suu.*Hjw2.*Su+w2_Svu.*Hjw2.*Svu+...
```

```
%    w2_Svibphi.*Hjw2.*Svibphi+w2_Svibtheta.*Hjw2.*Svibtheta...
%
+w2_Svibthetaphi.*Hjw2.*Svibthetaphi+w2_Svibthetaphic.*Hjw2.*conj(Sv
ibthetaphi)+...
%
w2_SphiBM.*Hjw2.*SphiBM+w2_SthetaBM.*Hjw2.*SthetaBM+w2_SthetaphiBM.*
Hjw2;
%
% Stheta_volt_est =
w2_Svv.*Hjw2.*Sv+w2_Suu.*Hjw2.*Su+w2_Svu.*Hjw2.*Svu;
% Stheta_bm_est =
w2_SphiBM.*Hjw2.*SphiBM+w2_SthetaBM.*Hjw2.*SthetaBM+w2_SthetaphiBM.*
Hjw2;
% Stheta_vib_est =
w2_Svibthetaphi.*Hjw2.*Svibthetaphi+w2_Svibthetaphic.*Hjw2.*conj(Svi
bthetaphi)+...
%
w2_SphiBM.*Hjw2.*SphiBM+w2_SthetaBM.*Hjw2.*SthetaBM+w2_SthetaphiBM.*
Hjw2;
%
% figure; plot(freq,Sphi,'linewidth',2); hold on;
% plot(freq,Sphi_est,'r','linewidth',2)
% plot(freq,Sphi_volt_est,'k','linewidth',2)
% plot(freq,Sphi_bm_est,'g','linewidth',2)
% plot(freq,Sphi_vib_est,'m','linewidth',2)
% xlabel('Frequency (Hz)')
% ylabel('PSD \phi (W/Hz)')
% title('\phi Power Spectral Density')
% axis([0 5000 0 3e-17])
%
% figure; plot(freq,Stheta,'m','linewidth',2); hold on;
% plot(freq,Stheta_est,'g','linewidth',2)
% plot(freq,Stheta_volt_est,'k','linewidth',2)
% plot(freq,Stheta_bm_est,'b','linewidth',2)
% plot(freq,Stheta_vib_est,'m','linewidth',2)
% xlabel('Frequency (Hz)')
% ylabel('PSD \theta (W/Hz)')
% title('\theta Power Spectral Density')
% axis([0 5000 0 2e-17])
```

M-File 81: ComparisonOfBackgroundNoiseAndThermalNoise.m

```
% ComparisonOfBackgroundNoiseAndThermalNoise
% NI 9239 DAQ System
% 29 June 2007
load BackgroundSystemNoise.txt
bck = BackgroundSystemNoise;

load ThermalNoiseMeasurement.txt
TN = ThermalNoiseMeasurement;

OUTPUT = Standard_PSD(1e4,1e4,bck);
TN1_PSD = Standard_PSD(1e4,1e4,TN(:,1));
TN2_PSD = Standard_PSD(1e4,1e4,TN(:,2));
TN3_PSD = Standard_PSD(1e4,1e4,TN(:,3));
TN4_PSD = Standard_PSD(1e4,1e4,TN(:,4));

figure; plot(TN1_PSD(:,1),TN1_PSD(:,2))
ylabel('PSD Channel V1 (Watts/Hz)')
xlabel('frequency (Hz)')
figure; plot(TN2_PSD(:,1),TN2_PSD(:,2))
ylabel('PSD Channel V2 (Watts/Hz)')
xlabel('frequency (Hz)')
figure; plot(TN3_PSD(:,1),TN3_PSD(:,2))
ylabel('PSD Channel V3 (Watts/Hz)')
xlabel('frequency (Hz)')
figure; plot(TN4_PSD(:,1),TN4_PSD(:,2))
ylabel('PSD Channel V4 (Watts/Hz)')
xlabel('frequency (Hz)')
figure; plot(OUTPUT(:,1),OUTPUT(:,2))
ylabel('PSD (Watts/Hz)')
xlabel('frequency (Hz)')
title('All Control Voltages Grounded')
```

M-File 82: VoltageScale.m

```
%VoltageScale
%Scales the voltage inputs from voltage divider ckt
% G = [10.5878   10.6080   10.6310   10.5795]
%load 'datafile'
load Lin4p3Em2.txt
datafile = Lin4p3Em2;
G =  [10.5878 10.6080 10.6310 10.5795];
V1 = datafile(:,3)*G(1);
V2 = datafile(:,4)*G(2);
V3 = datafile(:,5)*G(3);
V4 = datafile(:,6)*G(4);

DataOUT = [datafile(:,1),datafile(:,2),V1,V2,V3,V4];
save('Lin4p3Em2S.txt', 'DataOUT', '-ascii', '-double', '-tabs')
```

M-File 83: BM_FFTs.m

```
% BM_FFTs.m
%
% Averaged FFT of MEMS OPTICAL Inc. Micro-mirror facet jitter on 8
May 2007
% Lorenzian should develope over the resonant frequency as subsequen
fft's
% are averaged.
% Clint Edwards, JHU-APL
clear all
close all
clc

load temp.txt
temp = temp;
len = length(temp(:,1));
[phi, theta]=QC_AngConv(temp, .53);
% figure; plot(phi);

N = 1e4;  %number of points in fft
sam_freq = 1e4;
T = 1/sam_freq;  % sampling period for time-series (ts) data
A = phi;
B = theta;
win = hanning(N);
shift = 2000;
NN = floor(length(A)/shift);
fftA = zeros(NN,N);
fftB = zeros(NN,N);
index=[0:1:length(A)-1]*T;  %time index
fft_index = [-N/2+1:1:(N)/2]*(1/(N*T));  %freq index
nf = 1;  %filtering
for ii=1:NN-ceil(N/shift)
   At=win.*A((ii-1)*shift+1:N+(ii-1)*shift);
   Bt=win.*B((ii-1)*shift+1:N+(ii-1)*shift);
   fftA(ii,:) = fftshift(T*fft(At-mean(At),N));
   fftB(ii,:) = fftshift(T*fft(Bt-mean(Bt),N));
end

ave_fftA = sum(abs(fftA).^2)*(1/NN);
ave_fftA=filter((1/nf)*ones(nf,1),1,ave_fftA);
ave_fftB = sum(abs(fftB).^2)*(1/NN);
plot_index = [5000+10:5000+4000];
TotalPower = sum(2*abs(ave_fftA(plot_index)));
figure; plot(fft_index(plot_index),2*abs(ave_fftA(plot_index)),'b')
hold on; plot(fft_index(plot_index),2*abs(ave_fftB(plot_index)),'r')
% plot(fft_index(N/2:N),2*abs(ave_fftA(N/2:N)),'b')
% hold on; plot(fft_index(N/2:N),2*abs(ave_fftB(N/2:N)),'r')
% plot(fft_index,abs(ave_fftB(1:N)),'r')
xlabel('Frequency (Hz)')
ylabel('\phi (blue) PSD (Watts/sqrt(Hz))')
```

```
ylabel('\theta (red) PSD and \phi (blue) PSD (Watts/sqrt(Hz))')
title('MEMS Mirror Freq. Resp. to Air')
```

M-File 84: BM_FFTsFitted.m

```
% BM_FFTsFitted.m
%
% Averaged FFT of MEMS OPTICAL Inc. Micro-mirror facet jitter on 8
May 2007
% Lorenzian should develope over the resonant frequency as subsequen
fft's
% are averaged.
% Clint Edwards, JHU-APL
clear all
close all
clc
load NEW_BM_18June2007.txt
temp = NEW_BM_18June2007;
len = length(temp(:,1));
[phi, theta]=QC_AngConv(temp, .53);

N = 1e4;  %number of points in fft
sam_freq = 1e4;
T = 1/sam_freq;  % sampling period for time-series (ts) data
A = phi;
B = theta;
win = hanning(N);
shift = 2000;
NN = floor(length(A)/shift);
fftA = zeros(NN,N);
fftB = zeros(NN,N);
index=[0:1:length(A)-1]*T;  %time index
fft_index = [-N/2+1:1:(N)/2]*(1/(N*T));  %freq index
nf = 1;  %filtering
for ii=1:NN-ceil(N/shift)
   At=win.*A((ii-1)*shift+1:N+(ii-1)*shift);
   Bt=win.*B((ii-1)*shift+1:N+(ii-1)*shift);
   fftA(ii,:) = fftshift(T*fft(At-mean(At),N));
   fftB(ii,:) = fftshift(T*fft(Bt-mean(Bt),N));
end

ave_fftA = sum(abs(fftA).^2)*(1/NN);
ave_fftA=filter((1/nf)*ones(nf,1),1,ave_fftA);
ave_fftB = sum(abs(fftB).^2)*(1/NN);
ave_fftB=filter((1/nf)*ones(nf,1),1,ave_fftB);

TotalPower = sum(2*abs(ave_fftA));
figure; plot(fft_index,2*abs(ave_fftA),'b','linewidth',2); hold on
plot(fft_index,2*abs(ave_fftB),'r');hold on;
xlabel('Frequency (Hz)')
ylabel('\phi (blue) PSD (Watts/Hz)')
ylabel('\theta (red) PSD and \phi (blue) PSD (Watts/sqrt(Hz))')
title('MEMS Mirror Freq. Resp. to Air')
```

M-File 85: Fitting.m

```
% fitting for the phi axis
% Lorenzian Curve Fitting
gamma1 = 160;
offset1 = 1.1e-16;
x0 = 1700;
x = fft_index;
Lorentz = (1.9e-13)*(1/pi)*(gamma1./((x-x0).^2+gamma1.^2))+offset1;
plot(fft_index,Lorentz,'--k','linewidth',2); hold on
axis([100 4000 0 1.5e-15])
%Gaussian Curve Fitting
mu = 1700;
sigma = 150;
var = sigma^2;
Gaussian = (1.7e-13)*(1/(sqrt(2*pi*var)))*exp(-(x-
mu).^2/(2*var))+offset1;
plot(fft_index,Gaussian,'--r','linewidth',2); hold on;

% fitting for the theta axis
% Lorenzian Curve Fitting
gamma2 = 115;
offset2 = .3e-16;
x0 = 2050;
x = fft_index;
Lorentz = (3.1e-13)*(1/pi)*(gamma2./((x-x0).^2+gamma2.^2))+offset2;
plot(fft_index,Lorentz,'--y','linewidth',2); hold on
% Gaussian Curve Fitting
mu = 2050;
sigma = 110;
var = sigma^2;
Gaussian = (2.8e-13)*(1/(sqrt(2*pi*var)))*exp(-(x-
mu).^2/(2*var))+offset2;
plot(fft_index,Gaussian,'--c','linewidth',2); hold on;
```

M-File 86: QC_AngConv1.m

```
function [phi, theta]=QC_AngConv1(File, OAL)
% function [phi, theta]=QC_AngConv1(V_Xerr, V_Yerr, OAL)
% INPUTS
% V_Xerr:  error voltage from quad cell indicating movement in the
horizontal
% direction
% V_Yerr:  error voltage from quad cell indicating movement in the
vertical
% direction
% OAL:  Optical Arm Length from MEMS mirror to Quad cell
%
% ANGLE = atan(QC_Displacment/OAL)
%
% OUTPUTS
% phi (movment in the outter axis)
% theta (movement in the inner axis)
%
% Clinton Edwards, JHU-APL
% 20 April 2007


load QuadCellCal.mat
V_Xerr=File(:,1);
V_Yerr=File(:,2);

Norm_X_Volt=data(:,1);
Delta_X=data(:,2);
Norm_Y_Volt=data(:,3);
Delta_Y=data(:,4);

Interp_Delta_X = interp1(Norm_X_Volt,Delta_X,V_Xerr);
Interp_Delta_Y = interp1(Norm_Y_Volt,Delta_Y,V_Yerr);

phi = atan(Interp_Delta_X/OAL);
theta = atan(Interp_Delta_Y/OAL);
```

M-File 87: ts_quadcell_swept.m

```
% ts_quadcell_swept.m
% Input text file from Quad Cell and then take the fft and plot it
with
% MATLAB
%
% Plot of both the direct coupling noise and cross coupling LR
acutation.
%
% Clint Edwards, JHU-APL
% 15 April 2007
load temp.txt

N = 35000;  %number of points in fft
sam_freq = 35000;
T = 1/sam_freq;  % sampling period for time-series (ts) data

A = temp(:,2);
B = temp(:,1);

% V1 = temp(:,3);
% V2 = temp(:,4);
% V3 = temp(:,5);
% V4 = temp(:,6);
index=[0:1:length(A)-1]*T;

fftA = fftshift(T*fft(A-mean(A),N));
fft_index = [-N/2+1:1:(N)/2]*(1/(N*T));
plot(fft_index,abs(fftA(1:N)),'b')
hold on

fftB = fftshift(T*fft(B-mean(B),N));
fft_index = [-N/2+1:1:(N)/2]*(1/(N*T));
plot(fft_index,abs(fftB(1:N)),'r')
xlabel('Frequency (Hz)')
ylabel('\phi (red) PSD and \theta (blue) PSD (Watts/sqrt(Hz))')
title('2-Axis Tip-Tilt MEMS Mirror Freq. Resp.')

hold off
```

M-File 88: ChanCheck.m

```
% ChanCheck.m
% Input text file from Quad Cell and then take the fft and plot it
with
% MATLAB
% Verifying that all the channels work the same.
% Plot of both the direct coupling noise and cross coupling LR
acutation.
%
% Clint Edwards, JHU-APL
% 15 April 2007
load
EMI6_AllInputsGroundedQCHighVoltWithMirrorWithLaser10June2007_2.txt
temp =
EMI6_AllInputsGroundedQCHighVoltWithMirrorWithLaser10June2007_2;
N = 1e4;  %number of points in fft
sam_freq = 1e4;
T = 1/sam_freq;  % sampling period for time-series (ts) data

% assign column in data text file to
C0 = temp(:,1)-mean(temp(:,1));
C1 = temp(:,2)-mean(temp(:,2));
C3 = temp(:,3)-mean(temp(:,3));
C4 = temp(:,4)-mean(temp(:,4));
C5 = temp(:,5)-mean(temp(:,5));
C6 = temp(:,6)-mean(temp(:,6));

win = hanning(N);

shift = 5000;
NN = floor(length(C0)/shift);
C0t = zeros(NN,N);
C1t = zeros(NN,N);
C3t = zeros(NN,N);
C4t = zeros(NN,N);
C5t = zeros(NN,N);
C6t = zeros(NN,N);

index=[0:1:length(C0)-1]*T;  %time index
fft_index = [-N/2+1:1:(N)/2]*(1/(N*T));  %freq index

for ii=1:NN-ceil(N/shift)
    C0t = win.*C0((ii-1)*shift+1:N+(ii-1)*shift);
    C1t = win.*C1((ii-1)*shift+1:N+(ii-1)*shift);
    C3t = win.*C3((ii-1)*shift+1:N+(ii-1)*shift);
    C4t = win.*C4((ii-1)*shift+1:N+(ii-1)*shift);
    C5t = win.*C5((ii-1)*shift+1:N+(ii-1)*shift);
    C6t = win.*C6((ii-1)*shift+1:N+(ii-1)*shift);
    PSD_C0(ii,:) = abs(fftshift(T*fft(C0t,N))).^2;
    PSD_C1(ii,:) = abs(fftshift(T*fft(C1t,N))).^2;
```

```
    PSD_C3(ii,:) = abs(fftshift(T*fft(C3t,N))).^2;
    PSD_C4(ii,:) = abs(fftshift(T*fft(C4t,N))).^2;
    PSD_C5(ii,:) = abs(fftshift(T*fft(C5t,N))).^2;
    PSD_C6(ii,:) = abs(fftshift(T*fft(C6t,N))).^2;
end


% Channel 0
PSD_C0 = sum(PSD_C0)*(1/NN);
figure; plot(fft_index,PSD_C0,'k'); % hold on;
xlabel('freq (Hz)')
ylabel('PSD of Chan0/Chan2')
title('Channel 0')

% Channel 1
PSD_C1 = sum(PSD_C1)*(1/NN);
figure; plot(fft_index,PSD_C1,'k'); % hold on;
xlabel('freq (Hz)')
ylabel('PSD of Chan1/Chan2')
title('Channel 1')

% Channel 3
PSD_C3 = sum(PSD_C3)*(1/NN);
figure; plot(fft_index,PSD_C3,'k'); % hold on;
xlabel('freq (Hz)')
ylabel('PSD of Chan 3')
title('Channel 3')

% Channel 4
PSD_C4 = sum(PSD_C4)*(1/NN);
figure; plot(fft_index,PSD_C4,'k'); % hold on;
xlabel('freq (Hz)')
ylabel('PSD of Chan 4')
title('Channel 4')

% Channel 5
PSD_C5 = sum(PSD_C5)*(1/NN);
figure; plot(fft_index,PSD_C5,'k'); % hold on;
xlabel('freq (Hz)')
ylabel('PSD of Chan 5')
title('Channel 5')

% Channel 6
PSD_C6 = sum(PSD_C6)*(1/NN);
figure; plot(fft_index,PSD_C6,'k'); % hold on;
xlabel('freq (Hz)')
ylabel('PSD of Chan 6')
title('Channel 6')
```

M-File 89: JitterAnalysis.m

```
JitterAnalysis.m
% Jitter Justification Analysis
%
% Clint Edwards, JHU-APL
% 28 April 2007
%
% This file analyzes the rms jitter from the MEMS mirror and a large
mass
% rigid mirror.  This is establish the presents of jitter and the
% importance of it consideration in MEMS systems design.  The MEMS
device
% is positioned in three difference operating angle voltages with a
3Vpp
% squarewave oscillating the facet.  Jitter will be more pronounced
closer
% to pull-in.
%
% text files from labview are:
% MEMSMIRROR_BaseLine_QC_45_55_3.txt  (with MEMS mirror)
% MEMSMIRROR_BaseLine_QC_40_60_3.txt  (with MEMS mirror)
% MEMSMIRROR_BaseLine_QC_30_70_3.txt  (with MEMS mirror)
% MIRROR_Baseline_QC_0.txt  (with rigid mirror)
% MIRROR_Baseline_QC_1.txt  (with rigid mirror)
% MIRROR_Baseline_QC_2.txt  (with rigid mirror)

% IN-SITU MIRROR JITTER FOR FLOATING AND NON-FLOATING OPTICAL TABLE
load MEMSMIRROR_BaseLine_QC_45_55_3.txt
load MEMSMIRROR_BaseLine_QC_40_60_3.txt
load MEMSMIRROR_BaseLine_QC_30_70_3.txt
load MEMSMIRROR_BaseLine_QC_45_55_3f.txt
load MEMSMIRROR_BaseLine_QC_40_60_3f.txt
load MEMSMIRROR_BaseLine_QC_30_70_3f.txt

N = 20000;  %number of points in fft
sam_freq = 20000;
T = 1/sam_freq;  % sampling period for time-series (ts) data

A = MEMSMIRROR_BaseLine_QC_45_55_3(:,1);
index=[0:1:length(A)-1]*T;

[phi, theta]=QC_AngConv(MEMSMIRROR_BaseLine_QC_45_55_3, .53);
phi1 = phi(700:1500)*180/pi;
index1 = index(700:1500);
figure; plot(index1,phi1, 'k')
std_45_55_3 = std(phi1)

[phi, theta]=QC_AngConv(MEMSMIRROR_BaseLine_QC_40_60_3, .53);
phi2 = phi(700:1500)*180/pi;
index2 = index(700:1500);
```

```
figure; plot(index2,phi2, 'k')
std_40_60_3 = std(phi2)

[phi, theta]=QC_AngConv(MEMSMIRROR_BaseLine_QC_30_70_3, .53);
phi3 = phi(700:1300)*180/pi;
index3 = index(700:1300);
figure; plot(index3,phi3, 'k')
std_30_70_3 = std(phi3)

[phi, theta]=QC_AngConv(MEMSMIRROR_BaseLine_QC_45_55_3f, .53);
phi7 = phi(1100:1800)*180/pi;
index7 = index(1100:1800);
figure; plot(index7,phi7, 'k')
std_45_55_3f = std(phi7)

[phi, theta]=QC_AngConv(MEMSMIRROR_BaseLine_QC_40_60_3f, .53);
phi8 = phi(1200:1900)*180/pi;
index8 = index(1200:1900);
figure; plot(index8,phi8, 'k')
std_40_60_3f = std(phi8)

[phi, theta]=QC_AngConv(MEMSMIRROR_BaseLine_QC_30_70_3f, .53);
phi9 = phi(600:1300)*180/pi;
index9 = index(600:1300);
figure; plot(index9,phi9, 'k')
std_30_70_3f = std(phi9)


% FIXED MASSIVE RIGID MIRROR
load MIRROR_Baseline_QC_0.txt
load MIRROR_Baseline_QC_1.txt
load MIRROR_Baseline_QC_2.txt

N = length(MIRROR_Baseline_QC_0(:,1));  %number of points in fft
sam_freq = N;
T = 1/sam_freq;  % sampling period for time-series (ts) data

[phi4, theta]=QC_AngConv(MIRROR_Baseline_QC_0, .53);
phi4 = phi4*180/pi;
figure; plot(phi4(end-1000:end), 'm')
MIRROR1_std = std(phi4(end-1000:end))

[phi5, theta]=QC_AngConv(MIRROR_Baseline_QC_1, .53);
phi5 = phi5*180/pi;
figure; plot(phi5(end-1000:end), 'm')
MIRROR2_std = std(phi5(end-1000:end))

[phi6, theta]=QC_AngConv(MIRROR_Baseline_QC_2, .53);
phi6 = phi6*180/pi;
figure; plot(phi6(end-1000:end), 'm')
MIRROR2_std = std(phi6(end-1000:end))
```

```
% JitterAnalysis.m Results  (Incorporated into a table)
% std_45_55_3 = 1.4755e-004 (deg)
% std_45_55_3f = 1.1748e-004 (deg)
% std_40_60_3 = 1.5838e-004 (deg)
% std_40_60_3f = 1.3240e-004 (deg)
% std_30_70_3 = 5.3423e-004 (deg)
% std_30_70_3f = 4.9404e-004 (deg)
% MIRROR1_std = 2.8660e-005 (deg)
% MIRROR2_std = 3.0569e-005 (deg)
% MIRROR2_std = 3.0569e-005 (deg)
```

M-File 90: StandardPSD1.m

```
function OUTPUT = Standard_PSD1(SF,N,temp)
% function OUTPUT = Standard_PSD1(SF,N,temp)
% OUTPUT = [fft_index', fft_temp'];
%
% Clint Edwards, JHU-APL
% 14 June 2007
len = length(temp);
temp = temp-mean(temp);  %mean value removed of 'temp'
T = 1/SF;  % sampling period for time-series (ts) data

% win = hanning(N);
SS = 2;
win = hanning(round(N/SS));
win = padarray(win,floor(N*((1-1/SS)/2)));

while length(win)<N
   win(end+1) = 0;
end

shift = round(N/4);
NN = ceil(length(temp)/shift);
fft_temp = zeros(NN,N);
index=[0:1:length(temp)-1]*T;  %time index
fft_index = [-N/2+1:1:(N/2)]*(1/(N*T));  %freq index

for ii=1:NN-floor(N/shift)
    if (N+(ii-1)*shift)<=length(temp)
        temp_t=win.*temp((ii-1)*shift+1:N+(ii-1)*shift);
    else
        win_size = length(win);
        temp_size = length(temp((ii-1)*shift+1:end));
        while win_size>temp_size
            temp(end+1) = 0;
            temp_size = length(temp((ii-1)*shift+1:end));
        end
        temp_t=win.*temp((ii-1)*shift+1:end);
    end
        fft_temp(ii,:) = fftshift(T*fft(temp_t-mean(temp_t),N));
end
nf=1;
fft_temp = sum(abs(fft_temp).^2)*(1/NN);
 fft_temp=filter((1/nf)*ones(nf,1),1,fft_temp);

OUTPUT = [fft_index', fft_temp'];
```

Bibliography

1. Hao, B. Winfield, M. Whitely, J. Brooks, J. A. Hammer "A Design Methodology for a Bulk-Micromachined Two-Dimensional Electrostatic Torsion Micro-mirror," *IEEE Journal of Microelectromechanical Systems*, Vol 12, No. 5, 692-702, (2003)
2. Moog, Inc. Chatsworth Operations, "A fine pointing mechanism for intersatellite laser communication," 21339 Nordhoff St., Chatsworth, CA 91311.
3. A. Allison, D. Abbott, "A MEMS Brownian Ratchet," *Elsevier, Journal of Microelectronics,* Vol. 33, 235-243, (2002)
4. A. Allison, D. Abbott "A MEMS implementation of a Brownian Ratchet," *Proceedings of the SPIE, Smart Electronics and MEMS II*, Vol. 4236, 319-329, (2001)
5. Y. Nemirovsky, O. Bochobza-Degani, "A Methodology and Model for the Pull-In Parameters of Electrostatic Actuators", *IEEE Journal of Microelectromechanical Systems*, Vol. 10, No. 4, 601-616, (2001)
6. K. Petersen, "A New Age for MEMS," *IEEE International Conference on Solid-State Sensors, Actuators, and Microsystems*, Seoul, Korea, June 5-9, (2005)

7. G. E. Uhlenbeck, S. Goudsmit, “A problem in Brownian Motion”, *Physical Review*, Vol. 34, 145-151, (1929)

8. X. M. Zhang, F. S. Chau, C. Quan, Y. L. Lam, A. Q. Liu, “A study of the static characteristics of a torsional micro-mirror,” *Elsevier, Journal of Sensors and Actuators*, Vol. 90, 73-81, (2001)

9. K. Takahashi, H. N. Kwon, K. Saruta, M. Mita, H. Fujita, H. Toshiyoshi, “A Two-Dimensional f-theta micro optical lens scanner with electrostatic comb-drive XY-stage,” *IEICE Electronics Express*, Vol. 2, No. 21, 542-547, (2005)

10. C. Marxer, P. Griss, N. F. de Rooij, “A Variable Optical Attenuator Based on Silicon Micromechanics,” *IEEE Photonics Technology Letters*, Vol. 11, No. 2, 233-235, (1999)

11. Z. Xiao, X. T. Wu, W. Peng, K. R. Farmer, “An Angle-Based Design Approach for Rectangular Electrostatic Torsion Actuators,” *IEEE Journal of Microelectromechanical Systems*, Vol. 10, No. 4, 561-598, (2001)

12. H. Schenk, P. Durr, D. Kunze, H. Lakner, H. Kuck, “An Electrostatically Excited 2D Micro-Scanning Mirror with an In-Plane Configuration of the Driving Electrodes,” *IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*, 473-478, (2000)

13. M. Tavakoli, R. Sarpeshkar, “An Offset-Canceling Low-Noise Lock-In Architecture for Capacitive Sensing,” *IEEE Journal of Solid-State Circuits, Vol. 38, No. 2, 244-253*, (2003)

14. H. Camon, F. Larnaudie, R. Rivoirard, B. Jammes, "Analytical Simulation of a 1D Single Crystal Silicon Electrostatic Micro-mirror," *Proceedings of the Second International Conference on Modeling and Simulation of Microsystems*, (1999)

15. N. N. Sharma, R. K. Mittal, "Brownian Motion Model of Nanoparticle Considering Nonrigidity of Matter-A Systems Modeling Approach," *IEEE Transactions on Nanotechnology*, Vol. 4. No. 2, 180-186, (2005)

16. H. Rokhsari, M. Hossein-Zadeh, A. Hajimiri, K. Vahala, "Brownian noise in radiation-pressure-driven micromechanical oscillators," Applied Physics Letter, Vol. 89, No. 261109, 1-3, (2006)

17. Z. Kadar, W. Kindt, A. bossche, J. Mollinger, "Calculation of the Quality Factor of Torsional Resonators in the Low-Pressure Region," *The 8th International Conference on Solid-State Sensors and Actuators, and Eurosensors IX*, Stockhom, Sweden, June 25-29, 29-32, (1995)

18. K. B. Brown, W. Allegretto, F. E. Vermeulen, R. P. W. Lawson, A. M. Robinson, "Cantilever-in-Cantilever Micromachined Pressure Sensors Fabricated in CMOS Technology", *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, Edmonton, Alberta, Canada, May 9-12, 1686-1691, (1999)

19. J. Stockley, S. Serati, "Cascaded One-Dimensional Liquid Crystal OPAs for 2-D Beam Steering,", *Proceedings of the IEEE, Aerospace Conference*, Vol. 4, No. 3, 1817-1822, (2003)

20. J. F. Vignola, X. Liu, S. F. Morse, B. H. Houston, J. A. Bucaro, M. H. Marcus, S. M. Photiadis, L. Sekaric, "Characterization of Silicon Micro-Oscillators by Scanning Laser Vibrometry," *Review of Scientific Instruments*, Vol. 73, No. 10, 3584-3588, (2002)

21. K. Liao, Y. Wang, C. Yeh, R. Chen, "Closed-Loop Adaptive Control For Electrostatically Driven Torsional Micro-mirrors", Journal of Microlithography, Microfabrication, and Microsystems, Vol 4. No. 4, 041503-1-9, (2005)

22. R. B. Darling, C. Hivick, J. Xu, "Compact Analytical Models for Squeeze Film Damping," IEEE International Conference of Solid-State Sensors and Actuators, 1113-1116, (1997)

23. H. DeWeerd, "Compact, Lower Power Precision Beam Steering Mirror," SPIE Beam Deflection and Scanning Technologies, Vol. 1454, 207-304, (1991)

24. K. B. Fielhauer, B. G. Boone, J. R. Bruzzi, B. E. Kluga, J. R. Connely, M.M. Bierbaum, J. J. Gorman, N. Dagalakis, "Comparison of Macro-Tip Tilt and Meso-Scale Position Beam Steering Transducers for Free Space Optical Communications Using a Quardrant Photodiode Sensor," *Proceedings of the SPIE Free-Space Laser Communication and Active Laser Illumination III*, Vol. 5160, (2003)

25. B. Borovic, C. Hong, A. Q. Liu, L. Xie, F. L. Lewis, “Control of MEMS Optical Switch,” IEEE Conference on Decision and Control, 3039-3044, (2004)

26. A. Minikes, I. Bucher, and G. Avivi, “Damping of a Micro-Resonator Torsion Mirror in Rarefied Gas Ambient,” Journal of Micromechanics and Microengineering, 1762-1769, (2005)

27. V. A. Skormin, M. A. Tascillo, T. E. Busch, “Demonstration of a jitter rejection technique for free-space laser communication,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 33, No. 2, 568-576, (1997)

28. F. R. Blom, S. Bouwstra, M. Elwenspoek, J. H. J. Fluitman, “Dependence of Quality Factor of Micromachined Silicon Beam Resonators on Pressure and Geometry,” *Journal of Vacuum Science Technology*, Vol. 10, No. 1, (1992)

29. P. B. Chu, et al. “Design and Nonlinear Servo Control of MEMS Mirrors and Their Performance in a Large Port-Count Optical Switch,” *Journal of Microelectromechanical Systems*, Vol. 14, No. 2, 261-273, (2005)

30. O. Degani, Y. Nemirovsky, “Design Considerations of Rectangular Electrostatic Torsion Actuators Based on New Analytical Pull-in Expressions,” Journal of Microelectromechanical Systems, Vol. 11, No. 1, 20-26, (2002)

31. J. Tsai, M. C. Wu, “Design, Fabrication, and Characterization of a High Fill-Factor, Large Scan-Angle, Two-Axis Scanner Array Driven by a Leverage Mechanism,” Journal of Microelectromechanical Systems, Vol. 15, No. 5, 1209-1213, (2006)

32. F. Pan, J. Kubby, E. Peeters, J. K. Chen, O. Vitomirov, “Design, Modeling and Verification of MEMS Silicon Torsion Mirror,” Proceedings of the SPIE, Vol. 3226, 114-124, (1997)

33. J. J. Sniegowski, S. M. Rodgers, B. G. Boone, J. R. Bruzzi, C. W. Drabenstadt, B. E. Kluga, E. W. Rogala, R. Osiander, K. J. Rebello, M. A. Darrin, “Development, Test and Evaluation of MEMS Micro-Mirrors for Free-Space Optical Communications” *Proceedings of the SPIE,* Vol. 5550, 299-312, (2004)

34. T. Juneau, K. Unterkofler, T. Seliverstov, S. Zhang, M. Judy, “Dual-axis Optical Mirror Positioning Using A Nonlinear Closed-Loop Controller,” *IEEE Conference of Solid State, Sensors, Actuators, and Microsystems*, Boston, Vol. 1, 560-563, (2003)

35. R. A. Conant, J. T. Nee, K. Y. Lau, R. S. Muller, “Dynamic Deformation of Scanning Mirrors”, IEEE/LEOS International Conference on Optical MEMS Vol. 1, 49-50, (2000)

36. J. Chen, S. Kang, “Dynamic Macromodeling of MEMS Mirror Devices,” *IEEE Electron Devices Meeting Technical Digest*, 41.5.1-41.5.4, (2001)

37. G. Leveque, P. Girard, S. Belaidi, G. C. Solal, “Effects of Air Damping in Noncontact Resonant Force Microscopy,” Review of Scientific Instrumentation, Vol. 68, No. 11, 4137-4144, (1997)

38. J. Mertens, E. Finot, T. Thundat, A. Fabre, M. Nadal, V. Eyraud, E. Bourillot, “Effects of Temperature and Pressure on Microcantilever Resonance Response,” *Elesevier*, *Journal of Ultramicroscopy*, Vol. 97, 119-126, (2003)

39. C. Pu, S. Park, P. B. Chu, S. Lee, M. Tsai, D. Peale, N. H. Bonadeo, I. Brener, “Electrostatic Actuation of Three-Dimensional MEMS Mirrors Using Sidewalls Electrodes,” IEEE Journal of Selected Topics in Quantum Electronics, Vol. 10, No. 3, 472-477, (2004).

40. M. Kiang, O. Solgaard, K. Y. Lau, R. S. Muller, “Electrostatic Combdrive-Actuated Micro-mirrors for Laser-Beam Scanning and Positioning,” IEEE Journal of Micromechanical Systems, Vol. 7, No. 1, 27-37, (1998)

41. K. B. Lee, Y. Cho, “Electrostatic Control of Mechanical Quality Factors for Surface-Micromachined Lateral Resonators,” Journal of Micromechanical and Microengineering, Vol. 6, 426-430, (1996)

42. H. Toshiyoshi, H. Fujita, “Electrostatic Micro Torsion Mirrors For An Optical Switch Matrix,” Journal of Microelectromechanical Systems, Vol. 5, No. 4, 231-237, (1996)

43. W. Lang, H. Pavlicek, T. Marx, H. Scheithauer, B. Schmidt, "Electrostatically Actuated Micro-mirror Devices in Silicon Technology," *Elsevier Journal of Sensors and Actuators*, Vol. 74, 216-218, (1999)

44. M Fischer, M. Giousouf, J. Schaepperle, D. Eichner, M. Weinmann, W. von Munch, F. Assumus, "Electrostatically Deflectable Polysilicon Micro-mirrors Dynamic Behavior and Comparison with the Results," Elsevier, Journal of Sensors and Actuators A, Vol. 67, 89-95, (1998)

45. U. Hofmann, S. Muehlmann, M. Witt, K. Dorschel, R. Schutz, B. Wagner, "Electrostatically Driven Micro-mirrors for a Miniaturized Confocal Laser Scanning Microscope," *Proceedings of the SPIE, Conference on Miniaturized Systems with Micro-Optics and MEMS*, Vol. 3878, 29-38, (1999)

46. T. Xie, H. Xie, G. K. Fedder and Y. Pan, "Endoscopic Optical Coherence Tomography with a New MEMS Mirror," *IEEE Electronic Letters*, Vol. 39, No. 21, (2003)

47. M. Bao, H. Yang, H. Yin, Y. Sun, "Energy Transfer Model for Squeeze-Film Air Damping in Low Vacuum," Institute of Physics, Journal of Micromechanical and Microengineering, Vol. 12, 341-346, (2002)

48. T. Veijola, et al. "Equivalent circuit model of the squeezed gas film in a silicon accelerometer," Elsevier, Journal of Sensors and Actuators A, Vol. 48, 239-248, (1995)

49. O. B. Ozdoganlar, B. D. Hansche, T. G. Carne, “Experimental Model Analysis for Microsystems,” Proceedings of International Modal Analysis conference (IMAC), (2003)

50. R. W. Cochran, R. H. Vassar, “Fast Steering Mirrors in Optical Control Systems,” *Proceedings of the SPIE, Advances in Optical Structure Systems*, Vol. 1303, 245-251 (1990)

51. L. German, J. Braccio, “Fine-Steering Mirror Technology Supports 10 Nanoradian Systems,” *SPIE, Journal of Optical Engineering,* Vol. 29, No. 11, 1351-1359, (1990)

52. S. Lee, L. Huang, C. Kim, M. C. Wu, “Free-Space Fiber-Optic Switches Based on MEMS Vertical Torsion Mirrors,” IEEE Journal of Lightwave Technology, Vol. 17, No. 1, 7-13, (1999)

53. Y. Uenishi, J. Yamaguchi, T. Yamanoto, N. Takeuchi, A. Shimizu, E. Higurashi, R. Sawada, “Free-Space Optical Cross Connect Switch based on a 3D MEMS Mirror Array,” *IEEE, Lasers and Electro-Optics Society,* Vol. 1, 59-60, 2002

54. L. J. Hornbeck, “Digital Light Processing and MEMS: An Overview,” *IEEE, Lasers and Electro-Optics Society, Advanced Applications of Lasers in Matericals Processing…MEMS and Their Applications*, 7-8, (1996)

55. C. Chen, C. S. Gardner, “Impact of Random Pointing and Tracking Errors on the Design of Coherent and Incoherent Optical Intersatellite

Communication Links," IEEE Transactions On Communciations, Vol. 37, No. 3, 252-260, (1989)

56. M. R. Pinto, "Integrated Communications Microsystems," *Proceedings of the IEEE Solid-State and Integrated-Circuit Technology Conference*, Vol. 1, 17-18, (2001)

57. H. Toshiyoshi, W. Piyawattanametha, C. Chan, M. C. Wu, "Linearization of Electrostatically Actuated Surface Micromachined 2-D Optical Scanner," *IEEE Journal of Micoelectromechanical Systems*, Vol. 10, No. 2, 205-214, (2001)

58. J. Bernstein, R. Miller, W. Kelley, P. Ward, "Low-Noise MEMS Vibration Sensor for Geophysical Applications," *IEEE Journal of Micoelectromechanical Systems*, Vol. 8, No. 4, 433-438, (1999)

59. C. M. Van, "Macroscopic and Microscopic Methods for Noise in Devices," *IEEE Transactions on Electron Devices*, Vol. 41, No. 11, 1902-1915,(1994)

60. H. Baili, J. Julliard, "Management of Uncertainty within Estimation in Dynamical Context Application to MEMS," Proceedings of the 6th Nordic Signal Processing Symposium, 41-44, Espoo, Finland, June9-11, 2004

61. A. Mehta, S. Cherian, D. Hedden, T. Thundat, "Manipulation and Controlled Amplification of Brownian Motion of Microcantilever Sensors," Applied Physics Letters, Vol. 78, No. 1, 1637-1640, (2001)

62. L. A. Rocha, E. Cretu, R. F. Wolffenbuttel, "Measuring and Interpreting the Mechanical-Thermal Moise spectrum in a MEMS," *Institute of Physics, Journal of Micromechanics and Microengineering*, Vol. 15, 30-38, (2005)

63. Y. Reznichenko, M. Judy, S. Zhang, "Measuring the Optical and Electromechanical Properties of MEMS Mirrors," *IEEE, 12th International Conference on Solid State Sensors, Actuators and Microsystems*, 1466-1469, Boston, June 8-12, 2003

64. R. P. Leland, "Mechanical Thermal Noise in MEMS Gyroscopes," *IEEE Sensors Journal*, Vol. 5, No. 3, 493-500, (2005)

65. T. B. Gabrielson, "Mechanical-Thermal Noise in Micromachined Acoustic and Vibration Sensors," *IEEE Transactions on Electron Devices*, Vol. 40, No. 5, (1993)

66. Z. Djuric, "Mechanisms of Noise Sources in Microelectromechanical Systems," *Elsevier, Journal of Microelectronics Reliability*, Vol. 40, 919-932, (2000)

67. A. Neukermans, "MEMS Devices for All Optical Networks," *Proceedings of the SPIE, MOEMS and Miniaturized Systems II*, Vol. 4561, 1-10, (2001)

68. M. Whitely, J. Hammer, Z. Hao, B. Wingfield, L. Nelson, "A Single Two-Axis Micromachined Tilt Mirror and Linear Array," Proceedings of the SPIE, Vol. 4985, 83-94, (2003)

69. Z. Hao, R. Clark, J. Hammer, M. Whitley, B. Wingfield, "Modeling Air-Damping Effect in a Bulk Micromachined 2D Tilt Mirror," Elsevier, Journal of Sensors and Actuators A, Vol. 102, 42-48, (2002)

70. D. L. Dickensheets, G. S. Kino, "Microfabricated Biaxial Electrostatic Torsional Scanning Mirror," Proceedings of the SPIE, Vol. 3009, 141-150, (1997)

71. J. T. W. Yeow, et al. "Micromachined 2-D scanner for 3-D optical Coherence Tomography," *Elsevier, Journal of Sensors and Actuators A*, Vol. 117, 331-340, (2005)

72. P. Huber, U. Gerlach, "Micromachined Scanning Mirrors for Laser Beam Deflection," *Proceeings of the SPIE Optical Space Communications II*, Vol. 1522, 135-141, (1991)

73. D. J. Johnson, C. L. L. Ngo, J. Yang, Q. Li, "Micro-mirror for Laser Beam Scanning for Robotics and Other Applications," *2nd International Conference on Autonomous Robots, and Agents*, December 13-15, Palmerston North, New Zealand, 408-412, (2004)

74. T. Gessner, S. Kurth, C. Kaufmann, J. Markert, A. Ehrlich, W. Dotzel, "Micro-mirrors and Micro-mirror Arrays for Scanning Applications," *Proceedings of the SPIE: MOEMS and Miniaturized Systems*, Vol. 4178, 338-349, (2000)

75. J. Wang, J. M. Kahn, K. Y. Lau, “Minimization of Acquisition Time in Short-Range Free-Space Optical Communication,” *Journal of Applied Optics*, Vol. 41, No. 36, (2002)

76. X. M. Zhang, F. S. Chau, C. Quan, A. Q. Liu, “Modeling of the Optical Torsion Micro-mirror,” *Proceedings of the SPIE*, Vol. 3899, 109-116, (1999)

77. O. Bochobza-Degani, Y. Nemirovsky, “Modeling the Pull-In Parameters of Electrostatic Actuators with a Novel Lumped Two Degrees of Freedom Pull-In Model,” *Elsevier, Journal of Sensors and Actuators A*, Vol. 97-98, 569-579, (2002)

78. H.-L. Chau, K. D. Wise, “Noise Due to Brownian Motion in Ultrasensitive Solid-State Pressure Sensors,” IEEE Transactions On Electron Devices, Vol. 24, No. 4, 859-865, (1987)

79. J. R. Vig, Y. Kim, “Noise in Microelectromechanical System Resonators,” IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, Vol. 46, No. 6, (1999)

80. I. Brener, et al. “Nonlinear Servo Control of MEMS Mirrors and Their Performance in a Large Port-Count Optical Switch,” *Optical Fiber Communication Conference*, Vol. 1, 385-386, (2003)

81. Y. Zhang, Y.-P. Zhao, “Numerical and Analytical Study on the Pull-In Instability of Micro-Structure Under Electrostatic Loading,” *Elsevier, Journal of Sensors and Actuators*, Vol. 127, 366-380, (2006)

82. S. Hutcherson, W. Ye, "On the Squeeze-Film Damping of Micro-Resonators in the Free-Molecule Regime," *Journal of Micromechanical and Microengineering*, Vol. 14, 1726-1733, (2004)

83. G. E. Uhlenbeck, L. S. Ornstein, "On the Theory of the Brownian Motion," *Physical Review*, Vol. 36, 823-841, (1930)

84. M. C. Wu, "Optical MEMS for Lightwave Communication," *IEEE Journal of Lightwave Technology*, Vol. 24, No. 12, 4433-4454, (2006)

85. O. Degani, D. J. Seter, E. Socher, S. Kaldor, Y. Nemirovsky, "Optimal Design and Noise Consideration of Micromachined Vibrating Rate Gyroscope with Modulated Integrative Differential Optical Sensing," IEEE Journal of Microelectromechanical Sytems, Vol. 7, No. 3, 329-339, (1998)

86. S. R. Bhalotra, J. D. Mansell, H. L. Kung, D. A. B. Miller, "Parallel-plate MEMS Mirror Design for Large On-resonance Displacement," *IEEE/LEOS International Conference on Optical MEMS*, 93-94, (200)

87. S. Serati, J. Stockley, "Phased Array of Phased Arrays for Free Space Optical Communications," Proceedings of the IEEE, Aerospace Conference, Vol. 4, 1809-1816, (2003)

88. W. Zhang, K. L. Turner, "Pressure-Dependent Damping Characteristics of Micro Silicon Beam Resonator for Different Resonant Modes," *IEEE Journal of Sensors*, 1-4, (2005)

89. R. M. Bevensee, "Probabilistic Potential Theory Applied to Electrical Engineering Problems," *Proceedings of the IEEE*, Vol. 61, No. 4, (1973)

90. O. Degani, E. Socher, A. Lopsin, T. Leitner, D. J. Setter, S. Kaldor, Y. Newmirovsky, "Pull-in Study of an Electrostatic Torsion Micro Actuator," Journal of Microelectromechanical Systems, Vol. 7, No. 4, 373-379, (1998)

91. R. K. Gupta, S. D. Senturia, "Pull-In Time Dynamics as a Measure of Absolute Pressure," 10th IEEE International Workshop on MEMS, Nogoya, Japan, January 26-30, (1997)

92. K. Y. Yasumura, T. D. Stowe, E. M. Chow, T. Pfafman, T. W. Kenny, B. c. Stipe, D. Rugar, "Quality Factors in Micron- and Submicron-Thick Cantilevers," *IEEE Journal of Microelectromechanical Systems*, Vol. 9, No. 1, 117-125, (2000)

93. M. Suhonen, J. Graeffe, T. Sillanpaa, H. Sipola, M. Eiden, "Scanning Micromechanical Mirror for Fine-Pointing Units of Intersatellite Optical Links," Institute of Physics, Journal of Smart Materials, and Structures, Vol. 10, 1204-1210, (2001)

94. J. Kahn, "Secure Free-Space Optical Communication Between Moving Platforms," Lasers, and Electro-Optics Society, Vol. 2, 455-456, (2002)

95. W. Dotzel, T. Gessner, R. Hahn, C. Kaufmann, K. Kehr, S. Kurth, J. Mehner, "Silicon Mirrors and Micro-mirror Arrays for Spatial Laser Beam Modulation," *Proceedings of the IEEE, 1997 International Conference on Solid-State Sensors and Actuators*, Chicago, June 16-19, 1997

96. K. E. Peterson, "Silicon Torsional Scanning Mirror," *IBM Journal of Research and Development*, Vol. 24, No. 5, 631-637, (1980)

97. "Squeeze Film Damping Effect on the Dynamic Response of a MEMS Torsion Mirror," *Technical Proceedings of the 1998 International Conference on Modeling and Simulation of Microsystems*, Chapter 11, 474-479, (1998)

98. J. I. Seeger, S. B. Crary, "Stabilization of Electrostatically Actuated Mechanical Devices," *International Conference on Transducers, Solid-State Sensors and Actuators*, Vol. 2, 1133-1136, (1997)

99. G. G. Ortiz, S. Lee, J. W. Alexander, "Sub-Microradian Pointing for Deep Space Optical Telecommunications Network," *19th AIAA International Communications Satellite Systems Conference*, Toulouse, France, (2001)

100. G.-D. J. Su, H. Toshiyoshi, M. C. Wu, "Surface-Micromachined 2-D Optical Scanners with High-Performance Single-Crystalline Silicon Micro-mirrors," IEEE Photonics Technology Letters, Vol. 13, No. 6, (2001)

101. H.-Y. Lin, W. Fang, "Torsional Mirror with an Electrostatically Driven Lever-Mechanism," IEEE/LEOS International Conference on Optical MEMS, 113-114, (2000)

102. L. Zhou, M. Last, V. Milanovic, J. M. Kahn, K. S. J. Pister, "Two Axis Scanning Mirror for Free Space Optical Communication Between UAVs," IEEE/LEOS International Conference on Optical MEMS, (2003)

103. W. Piyawattanametha, L. Fan, S. Hsu, M. Fujino, M. C. Wu, P. R. Herz, A. D. Aguirre, Y. Chen, J. G. Fujimoto, "Two-Axis MEMS Scanning Catheter

for Ultrahigh Resolution Three-Dimensional and En Face Imaging," Optial Society of America Optical Express, Vol. 15, 2445-2453, (2007)

104. W. Piyawattanametha, W. L. Fan, S. Hsu, M. Fujino, M. C. Wu, P. R. Herz, A. D. Aquirre, Y. Chen, J. G. Fujimoto, "Two-Dimensional Endoscopic MEMS Scanner for High Resolution Optical Coherence Tomography," CLEO Conference on Lasers and Electro-Optics, Vol. 1, (2004)
105. T. A. Johnson, J. Antol, T. M. Hotz, V. Cuda, "Using Fast Steering Mirror Control to Reduce Instrument Pointing Errors Caused by Spacecraft Jitter," NASA Contractor Report 201583, 1-40, (1996)
106. MOEMS, Micro-Opto-Electro-Mechanical Systems, Manouchehr E. Motamedi, Editor, SPIE Press, Bellingham, Washington, USA, (2005)
107. S. D. Senturia, "Microsystem Design" Kluwer Academic Publishers, Boston, (2001)
108. C. L. Edwards, B. G. Boone, W. S. Levine and C. C. Davis, "Modeling of Electrostatically actuated two-axis (Tip-Tilt) MEMS Torsion Micro-Mirrors for Laser Beam Steering", *Proceedings of the SPIE*, Vol. 6555, No. 10, April, 2007
109. C. L. Edwards, B. G. Boone, W. S. Levine and C. C. Davis, "An Analytic Torque Model for Two-axis MEMS Mirrors", Accepted for Publication at *SPIE Journal of Micro/Nanolithography, MEMS, and MOEMS*, August, 2007.

110. C. L. Edwards, B. G. Boone, W. S. Levine and C. C. Davis, “First Principle Jitter Characterization of Two-Axis MEMS Mirrors”, Submitted for Publication at SPIE *Journal of Micro/Nanolithography, MEMS, and MOEMS*, July, 2007.

111. Daniel Hahn, C.L. Edwards, D. D. Duncan, “Link Availability Model for Optical Communications Through Clouds,” Proceedings of the SPIE, 4821-4837, (2002)

112. C. L. Edwards, C. C. Davis, “Free-Space Optical Communications through a forest Canopy.” Journal of Applied Optics, Vol. 45, No. 1, 191-200, (2006)

113. M. Suhonen, J. Graeffe, T. Sillanpaa, H. Sipola, M. Eiden, “Scanning Micromechanical Mirror for Fine-Pointing Units of Intersatellite Optical Links”, *Institute of Physics, Smart Materials and Structures*, 10, 1204-1210 (2001)

114. G. G. Ortiz, S. Lee, J. W. Alexander, “Sub-microradian Pointing for Deep Space Optical Telecommunications Network”, *19th AIAA International Communications Satellite Systems Conference*, Toulouse, France (2001)

115. J. Sniegowski, S. Rodgers, B.G. Boone, et al., “Development, Test and Evaluation of MEMS Micro-Mirrors for Free-space Optical Communications,” *Proceedings of SPIE, Vol. 1550, Free-Space Laser Communications and Laser Imaging IV, Symposium on Optical Science and Technology, SPIE 49th Annual Meeting* (2004)