

Embedding Desired Dynamics into Physical and Control Parameters of a Mechanical System

A dissertation

submitted by

Frank Saunders

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Mechanical Engineering

T U F T S U N I V E R S I T Y

February 2013

Advisor: Dr. Jason Rife

Committee Member: Dr. Robert White

Cross-Department Committee Member: Dr. Barry Trimmer

External Committee Member: Dr. John Rieffel

Abstract

To date, no method has been identified that utilizes morphological and control parameters of an under-actuated system in order to create periodic oscillations of states by altering a system's mode shapes. Phased periodic oscillations of system states are relevant in robot and animal locomotion. For example, the gait of a snake robot utilizes phased oscillations of segments in order to generate forward movement. As is evident in many natural systems, the interactions between direct actuation, sensory feedback, and morphological characteristics are all vital to create periodic oscillations that generate locomotion. This thesis introduces novel methodologies of creating periodic oscillations in under-actuated systems by altering mode shapes through design of physical and control parameters. Currently, it is not well understood how to design a system's modes to create desired periodic movements given a prescribed forcing function. It is also not well understood how to embed desired modes into system dynamics. The benefits of such methods are important to a large range of potential dynamic systems spanning from soft robots to rigid structures. In order to achieve this goal the conversion of designer-specified trajectories into mode shapes, and then the creation of those mode shapes in hardware, will be addressed.

Acknowledgements

Throughout my career at Tufts University I have had the privilege of working with a group of extremely gifted individuals. Thank you to all of the Automated Systems and Robotics and Trimmer Lab members who have contributed to my research. I would like to specifically thank all of my committee members for their advice and guidance. I would like to especially thank my lead advisor, Professor Jason Rife, for his contributions to my growth both as a researcher and as a person. Professor Rife has always been insightful and inspired to tackle tough problems. He has been extremely friendly and truly cared about my life on and off of campus. For this I am appreciative.

I would not be able to achieve a Ph.D. in Mechanical Engineering without my family. I am truly appreciative of the support I have received from my brother Jason, mother Carolyn, and father Frank. This support did not start at the beginning of my college career, but at the beginning of my life. My family always provided me with unconditional love and support.

Finally, I would like to thank my wife Danielle. Without Danielle I would have never pursued a Ph.D. in Mechanical Engineering. Danielle has had to deal with a full time graduate student husband for the last five years. She never once objected to my decision to pursue a Ph.D., and was completely supportive of every move. She has provided me with friendship, love, and a rock to lean on. I only hope that I can make her as proud of me as I am of her.

Table of Contents

Chapter I: Introduction	2
I. Motivation	2
II. Problem Statement	5
III. Research Areas of Interest	6
1. Biological Inspiration	6
2. Soft Robots	7
3. Mode Shapes	9
IV. Approach	11
V. Reader's Guide	13
Chapter II: Biological Inspiration	16
I. Introduction	17
II. Methods	19
1. Extensible-Link Model	19
2. Data Sets	23
a. Motion Capture	23
b. Ground Reaction Force Data	24
3. Computation of Internal and External Forces	25
a. Inverse-Kinematics	25
b. Inverse-Dynamics	27
4. Verification Procedure	30
a. Comparison of Simulated to Measured GRF	30
b. Quasi-static Testing	31
III. Results	31
1. Ground Reaction Force Verification	31
2. Quasi-Static Assumption Verification	33
3. Internal Forces	34
IV. Discussion	35
V. Conclusion	39
Chapter III: Modal Eigenstate Determination for Reoccurring Dynamics	40
I. Introduction	40
II. MEDFRD Problem	43
1. Qualitative Statement of MEDFRD Problem	43
2. Mathematical Model of MEDFRD Problem	44
III. Conclusion	50
Chapter IV: Specifying Eigenstates Based on Fourier Decomposition of a Desired Periodic Trajectory	52
I. Introduction	52
II. Fourier-Eigenstate Specification	55
III. Case Study	65
1. Model System	65
2. Application to Model System	66
IV. Conclusion	72

Chapter V: Embedding Desired Eigenstates into Active and Passive Dynamics of a Feedback System Subject to Equality Constraints.....	73
I. Introduction.....	73
II. Dual-Domain Eigenstate Factorization.....	75
1. Mathematical Description of Dual-Domain Eigenstate Factorization.....	77
2. Dual-Domain Eigenstate Factorization for LTI Systems.....	79
III. Proposed Algorithm.....	81
1. Compile all Equations.....	82
2. Eliminate Cases of Direct Assignment.....	87
3. Check System Rank and Augment the System as Necessary.....	88
4. Numerical Analysis.....	88
IV. Case Study.....	90
1. Specified System Response.....	90
2. Simulation.....	93
3. Eigenstate Matching.....	94
4. System Feasibility.....	95
V. Discussion.....	96
VI. Conclusion.....	98
Chapter VI: Mitigating Unwanted Dynamics While Embedding Eigenstates into a Feedback System in the Presence of Inequality Constraints.....	99
I. Introduction.....	99
II. Dual-Domain Eigenstate Factorization Optimization Variant.....	103
III. Simulation Based Performance Assessment.....	106
1. Optimization Solvers.....	107
2. Model System.....	107
a. Three-Link Pendulum.....	107
b. Baseline Eigenstate Specification.....	111
c. Baseline Constraints.....	112
3. Monte Carlo Trials.....	113
IV. Results.....	113
1. Representative Solution.....	113
2. Results of Monte Carlo Comparison Study.....	115
V. Discussion.....	118
VI. Conclusion.....	120
Chapter VII: Conclusion.....	122
I. Overview.....	122
II. Contributions.....	123
1. Bio-Inspiration.....	124
2. Eigenstate Specification.....	127
3. Eigenstate Matching.....	130
III. Future Work.....	133
IV. Summary.....	135
Appendix A: Manduca sexta Model Parameters.....	136
Appendix B: Matrix Norm Investigation.....	137
I. Analysis.....	137
1. Specified Eigenstates.....	137
2. Scenarios to be Evaluated.....	138

3. Alternative Matrix Norms.....	139
II. Results.....	139
III. Discussion.....	142
IV. Conclusion.....	143
Appendix C: Modal Transfer Function Derivation.....	144
Appendix D: State-Space Representation of Cavity System.....	145
Appendix E: State-Space of Mass-Spring-Damper System.....	147
Appendix F: Determining the Number of Specifiable Eigenstates for Linear Dynamic Systems.....	148
I. Introduction.....	148
II. Eigenstate Specification.....	151
1. Degree of Shapeability.....	151
2. Equality Constraints.....	152
3. Inequality Constraints.....	155
III. Conclusion.....	158
Lemma 1 Proof.....	159
Appendix G: Shapeability Example.....	161
I. MEDFRD Problem Overview.....	161
1. Exact Solution.....	161
2. Approximate Solution.....	163
II. Model System.....	163
III. Shapeability Investigation.....	165
1. Specified Eigenstate.....	165
2. Experimental Procedure.....	166
IV. Shapeability Experiment Results.....	167
V. Conclusion.....	170
References.....	171

List of Figures

Figure 1: Relationship to Alternative Research Areas.....	5
Figure 2: <i>Manduca sexta</i> Caterpillar.....	7
Figure 3: Pneumatically Actuated Deformable Robot.....	8
Figure 4: Research Area Investigated.....	9
Figure 5: Concepts Covered.....	12
Figure 6: Thesis Overview.....	15
Figure 7: Caterpillar Model Setup.....	21
Figure 8: Free Body Diagram of Representative Segment.....	22
Figure 9: <i>Manduca</i> Point Tracking Locations.....	24
Figure 10: Lifted Proleg Compression.....	29
Figure 11: Vertical Comparison.....	32
Figure 12: Horizontal Comparison.....	33
Figure 13: Simulated Internal Forces.....	35
Figure 14: Thoracic Normal and Axial Force Contribution.....	36
Figure 15: Caterpillar Locomotion Traveling Wave.....	41
Figure 16: Fourier-Eigenstate Specification Steps.....	57
Figure 17: Cavity Model System.....	66
Figure 18: State Trajectories.....	68
Figure 19: Sine Wave Representations of Each State Trajectory.....	69
Figure 20: Sinusoidal Forcing Function Response.....	70
Figure 21: System Response Compared to Desired Response.....	71
Figure 22: Region of Interest.....	77
Figure 23: Pseudo-code for Dual-Domain Eigenstate Factorization.....	86
Figure 24: Mass-Spring-Damper Configuration.....	90
Figure 25: Relative Displacements of the Masses with Respect to Time.....	91
Figure 26. Desired Responses.....	92
Figure 27. Method Response Comparison.....	94
Figure 28. Research Topics Covered by Dual-Domain Eigenstate Factorization.....	102
Figure 29. Three-Link Pendulum.....	108
Figure 30: Representative Solution to the MEDFRD Problem.....	115
Figure 31: Box Plots of J_{comp} Values Achieved.....	116
Figure 32: Frequency Matching 3 Actuators.....	140
Figure 33: Frequency Matching 2 Actuators.....	141
Figure 34: Amplitude/Phasing of Links 3 Actuators.....	141
Figure 35: Amplitude/Phasing of Links 2 Actuators.....	142
Figure 36: Pneumatically Actuated Deformable Robot.....	149
Figure 37: The Effect of Shapeability on the Solution of the MEDFRD Problem.....	155
Figure 38: The Effect of Tightening Inequality Constraints of $\tilde{\mathbf{A}}$	157

List of Tables

Table I: Physical and Control Parameter Values Matching Direct Solution $\tilde{\mathbf{A}}$	95
Table II. Tight Inequality Constraints on the $\tilde{\mathbf{A}}$ Matrix.....	113
Table III. Maximum Unspecified Eigenvalue Limit.....	118
Table IV: Manduca Model Parameters.....	136
Table V: Varying Constraints and Shapeability.....	168

Embedding Desired Dynamics into Physical and Control Parameters of a Mechanical System

Chapter 1: Introduction

I. Motivation

The purpose of this thesis document is to introduce novel methodologies to create phased periodic oscillations in under-actuated dynamic systems. Phased periodic oscillations of system states are relevant in robot and animal locomotion. For example, the gait of a snake robot utilizes phased oscillations of segments in order to generate locomotion [1]. Humans also utilize phased periodic oscillations of leg segments in order to walk and run [2, 3]. These periodic oscillations are accomplished through the use of passive musculoskeletal dynamics and active neural feedback. As is evident in many natural systems, the interactions between directly activated components, sensory feedback, and morphological characteristics are vital in creating periodic oscillations that generate locomotion.

Understanding morphological and control parameter interactions is critical when trying to create periodic oscillations in robotic systems with substantially more degrees of freedom than actuators. One group of robotic platforms inherently under-actuated, and highly impacted by physical parameters, is soft robots (robots entirely composed of elastically deformable materials) [4]. The ability of a soft robot to navigate through spaces smaller than its original cross-sectional area is a distinguishing characteristic from a rigid robot, and may allow it to perform missions such as penetration into rubble piles at disaster sites [5]. Periodic oscillations generated by soft robots can potentially be used to traverse the confined spaces of a rubble pile in order to reach a trapped survivor and deliver lifesaving supplies. However, current methods do not exist to design

periodic oscillations for highly under-actuated soft robotic systems that utilize both physical and control parameters.

To date, little investigation has gone into concurrent tuning of physical and control parameters to create complex dynamics for under-actuated systems. This is true because most traditional robotic systems are fully actuated; in which each degree of freedom has an associated actuator. By actuating each degree of freedom the physical characteristics of a system become less critical, so long as the actuators can still cause the system to achieve the desired motions. However, once actuators are lost the physical parameters of a system become increasingly more important. That is, the natural resonant modes affected by the physical parameters cannot be fully compensated and mitigated by control authority. Once actuators are lost, new methodologies are needed in which physical and control parameters are both utilized for the generation of periodic oscillations of system states.

One way to create complex periodic oscillating motions with minimal actuation may be to exploit resonance modes. In linear systems, resonance modes are described by Eigenstates, each comprising an Eigenvalue-Eigenvector pair. Eigenstate-based movement patterns are dynamic, inherently rejecting certain impulsive disturbances, and hence are appropriate for robotic platforms [6]. It may be possible to specify Eigenstates by leveraging both control and physical parameters in order to access a larger design space, even for under-actuated systems.

In this document new methods for Eigenstructure assignment are motivated by under-actuated system control, in which a relatively small number of actuators must be used to control a higher

number of degrees of freedom. Typically, controllers for under-actuated systems rely upon strong inertial coupling between actuated and un-actuated system states [7]. In general, under-actuated systems are controlled by performing partial feedback linearization and then placing Eigenvalues of the system [8]. Although under-actuated systems have been extensively studied [8-10], existing methods do not allow for full Eigenstate specification (including the assignment of Eigenvectors).

The emerging discipline of *co-design* offers the potential to address the limitations of existing passive design, Eigen-structure placement, and under-actuated control design methods when specifying full Eigenstates for under-actuated systems. Co-design involves the simultaneous design of a systems physical parameters as well as control parameters in order to achieve desired system goals [11, 12]. Co-design techniques have recently been developed to select control and physical architectures that reduce energy inputs or create minimal changes to physical structures while accomplishing previously impossible tasks for a given system [13, 14]. However, co-design has not yet been explored in detail for modal control or under-actuated systems.

This thesis seeks to introduce a new generalization of co-design tuned for creating repeated motion patterns for under-actuated systems by designing system Eigenstates. Combining together tools for tuning passive dynamic parameters and feedback control parameters to achieve desired Eigenstates is a compelling area for new research, especially if systems can be under-actuated. As can be seen in Figure 1, this thesis ties together related research from a number of other disciplines in order to generate periodic oscillations of under-actuated system states. The blue squares in Figure 1 represent areas that have been thoroughly investigated and the white

areas are areas currently not investigated given each specified field. This thesis attempts to investigate the tuning of physical parameters, control parameters, mode shapes, and under-actuation (as is indicated by the thesis block having all blue sections).

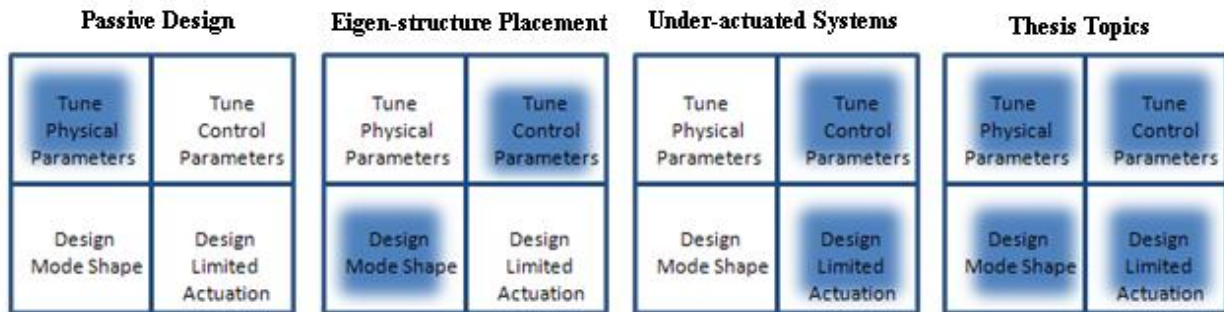


Figure 1: Relationship to Alternative Research Areas

II. Problem Statement

If it is desired to develop periodic oscillating motions for systems with a very large number of degrees of freedom, and a much smaller number of actuators, new tools are needed. This thesis introduces novel methodologies to concurrently tune physical and control parameters to design mode shapes of under-actuated systems with the goal of creating periodic oscillations. In order to achieve this goal the conversion of designer specified trajectories into mode shapes, and then the creation of those mode shapes in hardware, must be addressed.

III. Research Areas of Interest

In order to explore Eigenstate specification for under-actuated systems, I choose to span several different research areas. These include, but are not limited to, soft robots, biological systems, and mode shapes. This section will offer an overview of some pertinent background pertaining to periodic oscillation specifications for under-actuated systems.

1. Biological Inspiration

New methodologies for controlling under-actuated systems have been inspired by an investigation of the *Manduca sexta* caterpillar, which can be seen in Figure 2 (figure from [15]). *Manduca sexta* is able to traverse a variety of terrains in a number of orientations (vertical, horizontal, or inverted). *Manduca sexta* is also capable of many other types of motion, such as reaching, probing, rolling and strike reflex movements [16]. Such a wide variety of relatively complex movements are implemented by *Manduca sexta* using a surprisingly simple neuromuscular system in which each muscle is typically actuated by only one motor neuron [17, 18]. That this simple, distributed control system can generate such a wide variety of motions, suggests that some control functions are embedded in the animal's material composition and body mechanics [19]. An understanding of how complex movements are created by offloading control authority to physical aspects of a system must be investigated if under-actuated soft systems are to have some semblance of the success of soft animals like *Manduca sexta*. This thesis will investigate the *Manduca sexta* to determine if any fundamental principles of the generation of its locomotion can be applied to robotic platforms.



Figure 2: *Manduca sexta* Caterpillar [courtesy of Dr. Barry Trimmer]

2. Soft Robots

The design and control of soft robotic platforms is a compelling new research area for incorporating physical and control parameter design to achieve periodic oscillations for under-actuated systems. A soft robot is a platform comprised primarily of highly deformable materials, an example of which can be seen in Figure 3 [4]. Soft robot locomotion strategies include the use of gel-like actuators comprised of electro-active polymers that enable simultaneous deformation of the entire robot volume, the use of toroidal flexure devices that operate in the manner of a radially symmetric treadmill, and the use of peristalsis in the manner of leeches and worms [20-22].



Figure 3: Pneumatically Actuated Deformable Robot

To date, soft robot technology has focused primarily on stationary manipulator arms due a lack of efficient actuation and control methods [23]. These manipulators are inspired by elephant trunks and octopus arms, and are often analyzed with continuum models [24-27]. Consequently, the ability to control a locomoting soft robot is also not well understood. The primary reason for this lack of knowledge is the fact that soft robots are generally comprised of non-linear materials and have significantly more degrees of freedom in terms of achievable motions in comparison to the number of actuators. The high mismatch between actuation number and degrees of freedom results in a highly under-actuated system. Novel methods for generating locomotion patterns for systems spanning all actuation ranges must be pursued in order to create realizable soft robots.

Due to the complexity of fully soft systems, this thesis will focus on flexible (or *hyper-redundant*) structures, comprised of a number of rigid elements directly connected to one another in order to allow a wide range of movements (specifically phased periodic displacements of system states). Highly flexible systems, such as snake robots, have significantly more degrees of freedom than rigid systems with limited numbers of joints (referred to as *discrete* systems), but are still

mathematically tractable in comparison to *continuum* (systems with deformations made up by continuous curves) or *soft robots* (continuum robots made entirely of soft materials) [4]. A depiction of the relationships between system types can be seen in Figure 4 (where Figure 4 is adapted from [4]). The investigation of flexible structures given in this document may be able to inform the design of soft structures for certain applications (e.g. if the soft robots motions are constrained), and is considered a suitable first pass at trying to control under-actuated systems similar to soft robots.

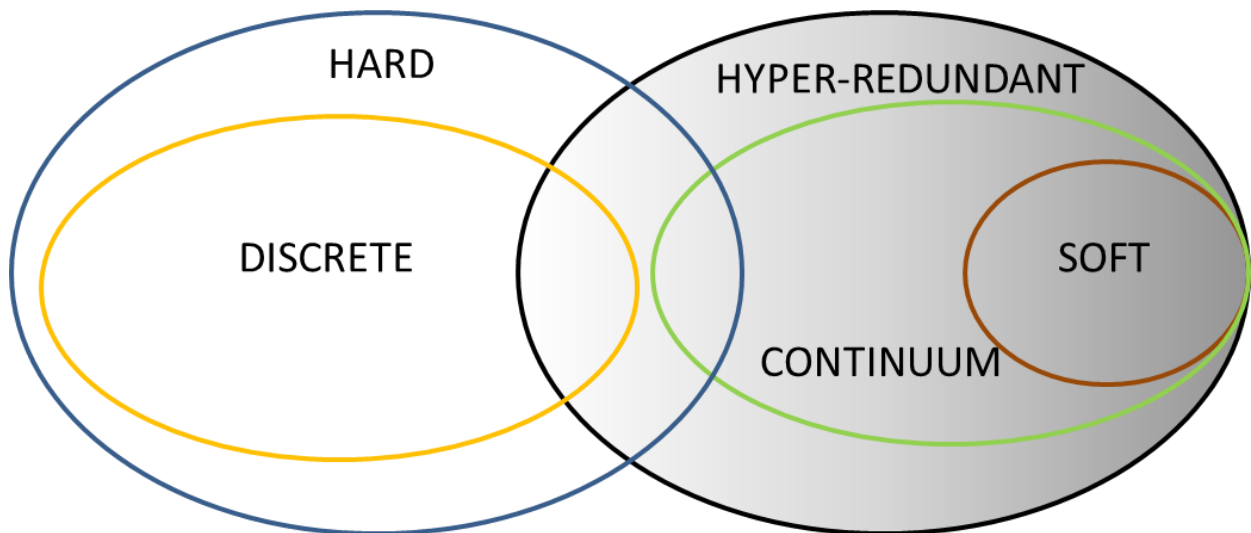


Figure 4: Research Area Investigated

3. Mode Shapes

One possible way to create periodic movements is to alter the dynamic response of a system. The dynamic response is governed by the mode shapes of a system which are directly affected by both morphological and control parameters [28]. If mode shapes can be altered in an intelligent way, it may be feasible to create periodic movements with limited control authority. For example, a variety of waves can be created on a string by oscillating its end with a single

actuator at different frequencies. However, for certain vibrating structures and undulating mechanisms, it may be desirable to specify more than one mode shape to create complex periodic movements. Currently, it is also not well understood how to design multiple mode shapes to illicit desired system responses given a periodic forcing function.

The two primary approaches for specifying mode shapes for systems currently include; modifying passive dynamics (through changes to mechanical design, for instance) or introducing appropriate feedback control.

Modifying passive dynamics is a relatively simple approach, and the range of possible mode shapes is limited only by the physical constraints on the system. Tuning passive physical parameters of a system, such as element masses or stiffnesses, can allow alteration of mode shape, resonant frequency, and decay rate [29]. However, all physical parameters are subject to inequality constraints that are either imposed by fundamental physics or by design requirements. For example, lengths are constrained by the physical volume in which the system must fit and mass must be positive. Additionally, kinematic constraints can further limit possible Eigenstates. Consider a pendulum, for example; the Eigenvector must obey the kinematic constraint that angular velocity be equal to the derivative of angle. These constraints severely limit the mode shapes possible for a system.

As an alternative to tuning passive system parameters, feedback control methods have been defined to modify the Eigenstates, and consequently mode shapes, of a system. It has long been known that Eigenvalues can be placed for controllable linear state-space systems. This ability has

been leveraged to modify system responses using methods including but not limited to LQR and direct pole placement [30]. More recently control over both Eigenvectors and Eigenvalues has been considered. This area of controls is sometimes referred to as modal control or Eigenstructure assignment [31, 32]. The author is not aware of any existing methods to tune both the passive dynamical and active feedback properties of a system simultaneously for the purposes of specifying a desired set of mode shapes. Combining together tools for tuning passive dynamic parameters and feedback control parameters to achieve desired mode shapes is a compelling area for new research. The benefits of such a new approach would be significant, especially for systems in which variations to either passive parameters or to feedback parameters alone cannot achieve desired mode shapes (e.g. a soft robot). Such applications are quite common, due for example to physical system constraints or under-actuation.

IV. Approach

The areas of research discussed throughout this thesis focus on four major topics: designing physical parameters, designing control parameters, designing mode shapes, and designing for under-actuation (as is demonstrated in Figure 5). Although a number of disciplines have touched upon some of these core concepts, no methodology has operated at the intersection of all of these research areas. This document closes the gap and introduces novel methodologies in order to create periodic phased oscillations of system states in a designed manner for potentially under-actuated and constrained systems.

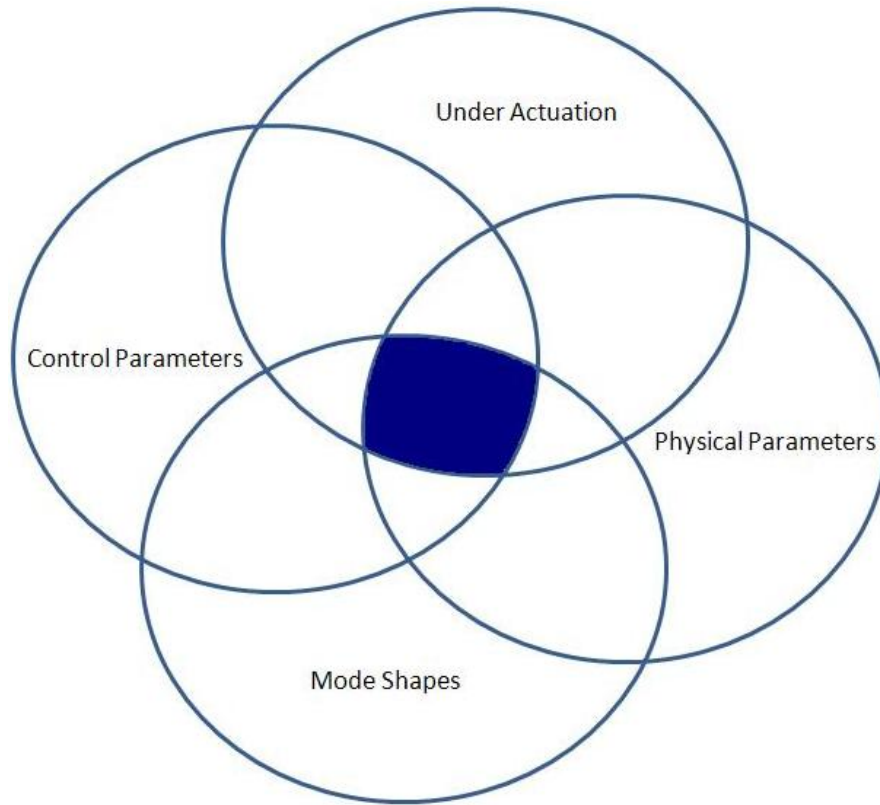


Figure 5: Concepts Covered

Throughout this thesis the creation of periodic oscillations of under-actuated and/or constrained systems will be investigated from biological implementation, to Eigenstate specification, and then to Eigenstate matching.

The main contributions of this thesis are the following.

1. Obtained model of internal forces within a crawling *Manduca sexta* caterpillar by applying inverse-kinematic model to experimentally acquired motion capture data.

- *Developed extensible-link model of caterpillar locomotion*
- *Generated first model of internal forces within a crawling caterpillar, forces which cannot otherwise be measured directly but that are relevant to*

- *Engineering inspiration for crawling robots*
- *Biological understanding of the “Environmental Skeleton” model of caterpillar locomotion*

2. *Invented a new approach for specifying the modal dynamics of a linear system to match periodic state trajectories as closely as possible.*

- *Proposed concept of embedding desired trajectories into closed-loop system modes*
- *Proposed Fourier-Eigenstate specification, a new method that sets Eigenvalues to multiples of a base frequency as a means to map a Fourier Series description of a trajectory into an Eigenstate description*
- *Adapted Eigenstate specification to apply not only to undamped systems but also to weakly damped systems*

3. *Identified and solved a Dual-Domain Eigenstate problem as a means of converting dynamic system specifications into a realizable set of system parameters.*

- *Identified that it is necessary to solve a Dual-Domain Eigenstate problem (a problem in which some values of state-update matrix are known and in which some of the Eigenvectors and Eigenvalues of that matrix are known, and in which the remaining matrix values and Eigenstates must be computed) in order to map desired modes into state-space description of system*
- *Proposed solving a set of non-linear algebraic equations in order to address the Dual-Domain problem involving equality constraints*
- *Adapted solutions to optimization problem format in order to include inequality constraints*

V. Reader’s Guide

This thesis introduces methods for designing and implementing mode shapes on under-actuated hardware to create periodic oscillations. The remainder of this document comprises five primary chapters. Each chapter introduces a different hyper-redundant model system to demonstrate

proposed Eigenstate specification methodologies. Each chapter also includes an extensive overview of previous research in the areas discussed. The remaining chapters include:

- an introduction to the caterpillar *Manduca sexta*, and modeling of its internal force distribution using an extensible-link model (Chapter II)
- an explicitly described problem statement of trying to match system Eigenstates to desired Eigenstates, both qualitatively and quantitatively, referred to as Modal/Eigenstate Determination for Reoccurring Dynamics or MEDFRD (Chapter III)
 - The problem of specifying mode shapes becomes increasingly complex for systems with few actuators and constrained physical parameters. In this thesis, the problem of Eigenstate specification is defined, which is referred to as Modal/Eigenstate Definition for Recurring Dynamics (MEDFRD). MEDFRD considers both a system's passive dynamics and its feedback control in order to generate desired system dynamics. For linear systems MEDFRD uses system Eigenstates in order to generate desired dynamics.
- a strategic methodology for converting designer generated state trajectories into the natural dynamics of a Linear Time-Invariant system, referred to as Fourier-Eigenstate specification (Chapter IV)
- a methodology for matching specified Eigenstates to system Eigenstates through solving a set of non-linear algebraic equations subject to equality constraints (Chapter V)
- a methodology of matching specified Eigenstates to system Eigenstates through solving a non-convex optimization problem subject to inequality constraints (VI)

Additional material that supplements these main results is presented in the Appendices.

The material covered in order to achieve the goal of Eigenstate specification and the corresponding chapters can be seen in Figure 6.

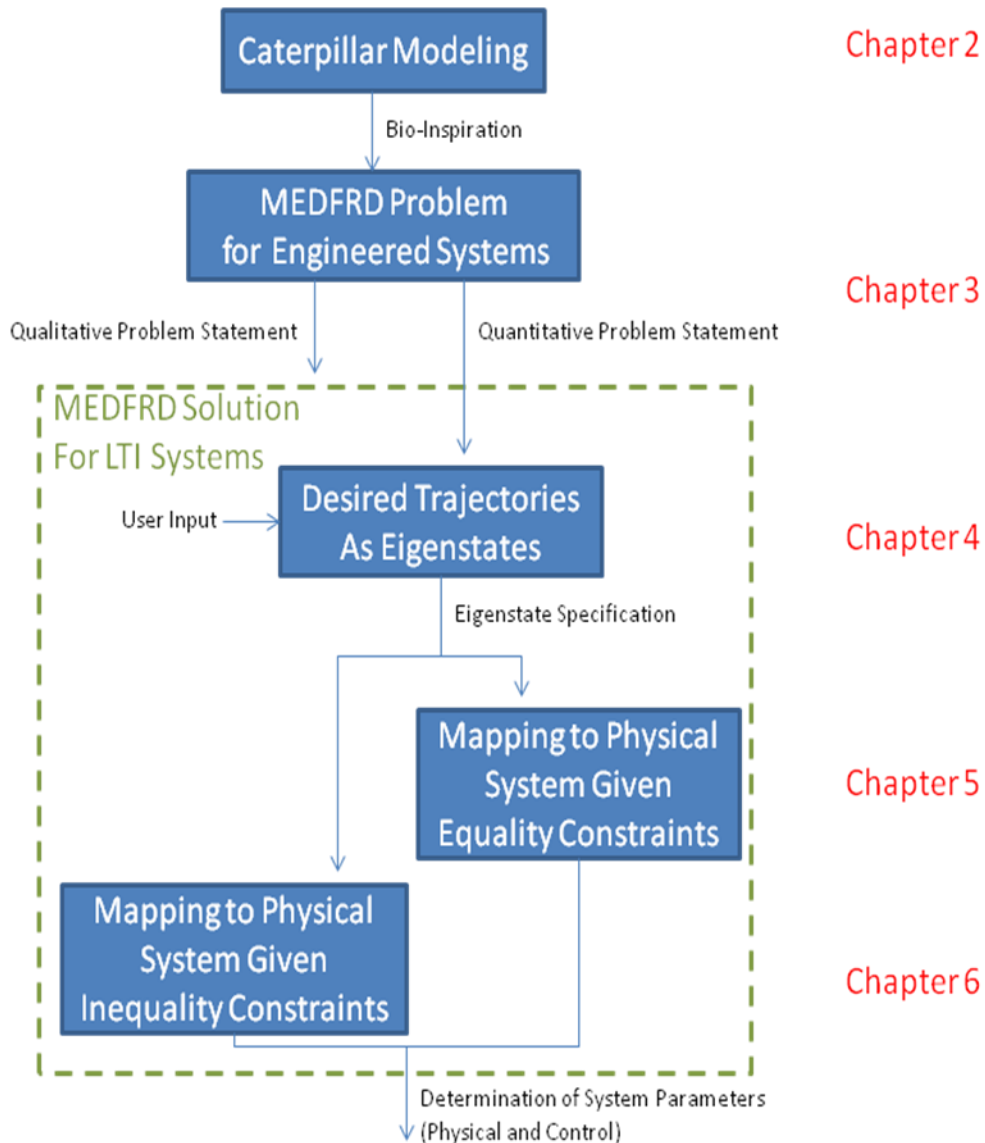


Figure 6: Thesis Overview

Chapter II: Biological Inspiration

Executive Summary- Most bio-inspired robots have been based on animals with jointed, stiff skeletons. There is now an increasing interest in mimicking the robust performance of animals in natural environments by incorporating compliant materials into robot locomotion. However, the mechanics of moving, highly conformable structures are particularly difficult to analyze. This chapter proposes a planar, extensible-link model for the soft-bodied tobacco hornworm caterpillar, *Manduca sexta* to provide insight for biologists and engineers studying locomotion of highly deformable animals and caterpillar-like robots containing more degrees of freedom than actuators. Using inverse-dynamics to process experimentally acquired point-tracking data, ground reaction forces and internal forces were determined for a crawling caterpillar. Computed ground reaction forces were compared to experimental data to validate the model. The results show that a system of linked extendable joints can describe the general form and magnitude of the contact forces produced by a crawling caterpillar. Furthermore, the model can be used to compute internal forces that cannot be measured experimentally. It is predicted that between different body segments in stance phase the body is mostly kept in tension and that compression only occurs during the swing phase when the prolegs release their grip. This finding supports a recently proposed mechanism for locomotion by soft animals in which the substrate transfers compressive forces from one part of the body to another (the environmental skeleton) thereby minimizing the need for hydrostatic stiffening. The model also clearly demonstrates the complexity of control needed to create caterpillar-like movement with traditional control methods.

*The material in this chapter was published as a reviewed article in the Journal of Biomimetics & Bioinspiration

I Introduction

Caterpillars are a compelling model for bio-inspired soft robot design. They can climb in complex structures in any orientation [32, 33] and they can even burrow several body lengths into soil. Caterpillars do not have a rigid skeleton and are therefore able to conform to the substrate and maneuver through small openings [34]. Unlike terrestrial worms or mollusks they have limbs (prolegs) widely spaced along the body that can grip at many locations simultaneously to provide failsafe attachment in different terrains.

This chapter focuses on the tobacco hornworm caterpillar *Manduca sexta* because it provides a tractable system for analyzing both the neural and mechanical features of soft-bodied locomotion. Soft-bodied locomotion is interesting because of the new locomotion modes possible and the inherent problem of controlling a highly under-actuated system. The musculature of the body has been described in detail [25-29] and it is possible to record the firing patterns of motor neurons driving these muscles [40, 41]. The responses of each muscle to these commands can be characterized under conditions that replicate normal strain cycling during locomotion [18, 42-43] and the mechanical properties of the body wall have been measured and modeled [44]. Importantly, the kinematics [45] and dynamics of crawling have now been recorded in sufficient detail to formulate a new theory of soft locomotion (the Environmental Skeleton) that can be used without hydrostatic control [15]. An emerging picture is that some control functions are embedded in the animal's material composition and body mechanics, a principle sometimes called *embodiment* [46].

If a reduced-order physical model for *Manduca* can be generated, it may enable us to infer how combined passive mechanical forces and active neuromuscular forces contribute to caterpillar motion. Creating a reduced-order model for *Manduca* is challenging because of the animal's highly deformable and compliant body; caterpillars stretch as well as bend. This means conventional robotics models, consisting of revolute joints connecting a series of rigid links, are too constraining to simulate *Manduca* motion, even though such models have been demonstrated to be useful in simulating both undulating vertebrates, like snakes, and undulating vertebrate-inspired robots [47, 48]. By contrast, previous simulations of invertebrate motion have generally employed continuum mechanics methods, such as finite difference or finite element analysis [23-26, 49]. These continuum mechanics methods introduce many more degrees of freedom than necessary for our application, making the inference of unknown control forces difficult. Perhaps because of the perceived necessity of performing a continuum mechanics simulation to describe invertebrate motion accurately, no prior attempts have been made to model the crawling mechanics of *Manduca sexta*.

In this chapter a planar, extensible-link model for *Manduca sexta* is described. This approach, which simulates *Manduca sexta* as a series of deformable links connected by revolute joints, strikes a balance between the extremes of rigid-body and continuum mechanics simulations. The extensible-link approach has sufficient degrees of freedom to capture the essential characteristics of caterpillar motion (stretching and bending), and yet it is intentionally made as simple as possible to promote abstraction. It is anticipated that such abstraction will lead to motion templates that inform the design of soft crawling robots, much as bio-inspired templates have previously been used to model the target dynamics for walking robots [50-52].

To demonstrate the utility of the extensible-link model, it will be applied to obtain internal (control) and external (reaction) forces for a crawling caterpillar. In this approach, internal and external forces are computed from motion capture data through a process known as inverse-dynamics [53]. The model provides a new analysis capability for studying caterpillar gait, in that no experimental methods yet exist to measure internal forces *in vivo* for a crawling caterpillar. Fortunately, it is possible to measure external forces. To validate the results, simulated external ground reaction forces will be compared to measured values obtained using a multi-sensor force plate customized for the study of caterpillar crawling [15]. Fundamental principles of caterpillar-like motion will then be discussed, along with the role of body mechanics as well as neural control when creating caterpillar-like locomotion.

II. Methods

1. Extensible-Link Model

Creating a caterpillar model required making several simplifying assumptions. *Manduca* has sixteen segments, each with over seventy muscles. It has ten abdominal prolegs (each featuring a passive gripping system) and six smaller thoracic legs [17]. One salient assumption is that only the abdomen of *Manduca* is modeled, from the A3 segment to the terminal proleg (the section in Figure 7 between the two blue lines, labeled 0 and 12). This decomposition simplifies the problem and is justified by prior observations that the abdominal prolegs are the primary means of locomotion [17] (After removing its thoracic legs, the animal still crawls). As will be discussed in section II.3.b, the choice not to model the caterpillar's thorax introduces a new boundary condition at the front of the A3 segment (indicated in Figure 7 by vertical line 12).

To further simplify the representation of the caterpillar, a 2D extensible-link model is used, comprised of twelve deformable links connected by revolute joints. During straight-line crawling there are negligible lateral movements and the prolegs always operate in symmetrical bilateral pairs so a planar representation is sufficient for our purposes. All links were deformable in the sense that their axial lengths were allowed to change. The center of mass of each link was assumed always to lie at the midpoint between successive revolute joints. By contrast, the alternative approach of adding prismatic joints to a rigid-link model allows for axial length changes but does not maintain the center of mass at the midpoint between successive revolute joints [54]. Additionally, each abdominal segment is modeled with two links. The links were numbered 1 through 12 (as identified in Figure 7 by the label on the right boundary of each link). Revolute joints allow the model to bend; axial deformability allows the model to stretch.

Reference lengths (\overline{L}_n), masses (M_n) and inertias (I_n) for each link n were measured for a representative fifth instar specimen. Parameter values for the representative specimen are summarized in Appendix A.

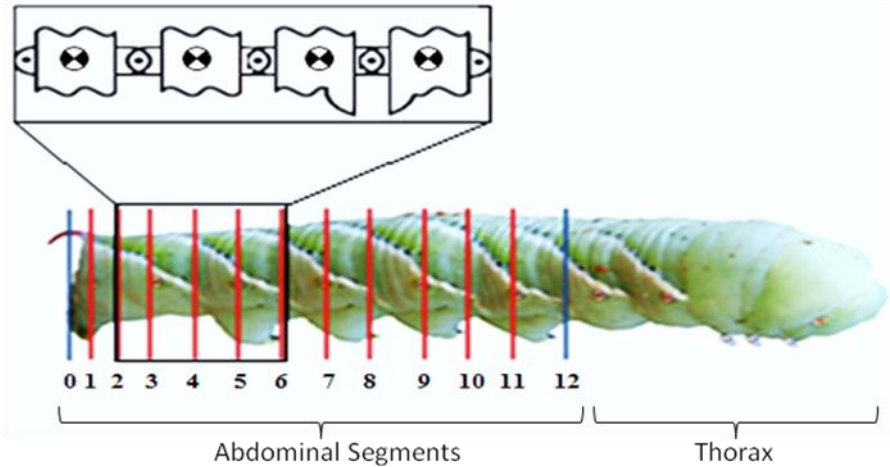


Figure 7: Caterpillar Model Setup

Forces and torques on each link can be computed by solving the 2D Newton-Euler equations, given that the accelerations for each link are known. This process is known as inverse-dynamics (in contrast with forward dynamics, in which the accelerations are computed from known forces). A free-body diagram for one representative link is illustrated in Figure 8. Internal force vectors (\mathbf{F}_n) and torques (T_n) are computed at each joint. The gravitational term, proportional to the acceleration \mathbf{g} , acts at the center of mass of each link. Individual prolegs are each split by a cutting plane. Accordingly, prolegs are each associated with a particular joint. The total proleg ground reaction forces ($2\mathbf{R}_n$) is distributed equally between each of the adjacent links. The exception is the terminal proleg, for which the total ground reaction force (\mathbf{R}_0) is applied only to the first link. Ground reaction forces were set to zero for prolegs not in contact with the ground.

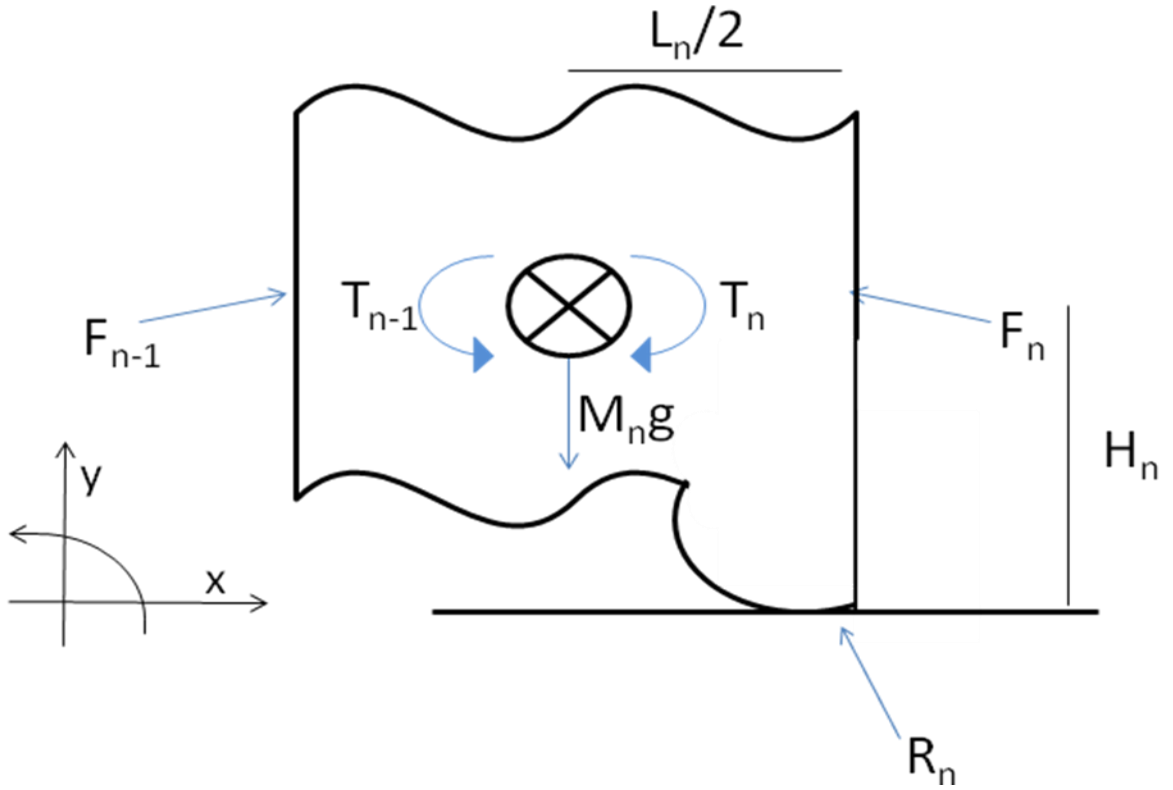


Figure 8: Free Body Diagram of Representative Segment

The 2D Newton-Euler equations for each link are given below. These equations are written in terms of the position vector \mathbf{x}_n describing the center-of-mass location and the angle θ_n of each link relative to the inertial reference frame.

$$M_n \ddot{\mathbf{x}}_n = \mathbf{F}_{n-1} - \mathbf{F}_n + \mathbf{R}_{n-1} + \mathbf{R}_n - M_n \mathbf{g} \quad (1)$$

$$I_n \ddot{\theta}_n = T_{n-1} - T_n + \frac{\mathbf{u}_{\perp,n}(-F_{n-1} + F_n - R_{n-1} + R_n)L_n}{2} + \mathbf{u}_{\parallel,n}(R_{n-1} - R_n)H_n \quad (2)$$

The vector $\mathbf{u}_{\perp,n}$ is the unit vector perpendicular to each link.

$$\mathbf{u}_{\perp,n} = \begin{bmatrix} \sin(\theta_n) \\ -\cos(\theta_n) \end{bmatrix} \quad (3)$$

The unit vector in the axial direction for each link is

$$\mathbf{u}_{\parallel,n} = \begin{bmatrix} \cos(\theta_n) \\ \sin(\theta_n) \end{bmatrix} \quad (4)$$

No link is attached to more than one proleg. The reaction force terms (\mathbf{R}_{n-1} and \mathbf{R}_n) are zero for joints without an attached proleg. For the case shown in Figure 8, for instance, a proleg is present only at joint n , so \mathbf{R}_{n-1} is zero.

2. Data Sets

a. Motion Capture

Motion capture of a live, crawling *Manduca* specimen was performed by tracking fluorescent markers [17] with DLTdv3 software [55]. Additional marker points were defined using visible features on the surface of the caterpillar, such as its spiracles. In total thirteen markers were tracked. The locations of these markers, when the caterpillar was at rest, are illustrated in Figure 9. Marker points were tracked through one complete gait period, consisting of 354 video frames sampled at 100 Hz.

All proleg contact and detachment times were determined using a standard kinematics template for caterpillar locomotion [15]. This template was constructed by averaging proleg lift-off and

placement times (percentages through gait period) for a large number of animals. The determined percentage values were then implemented on the gait period used for point tracking.

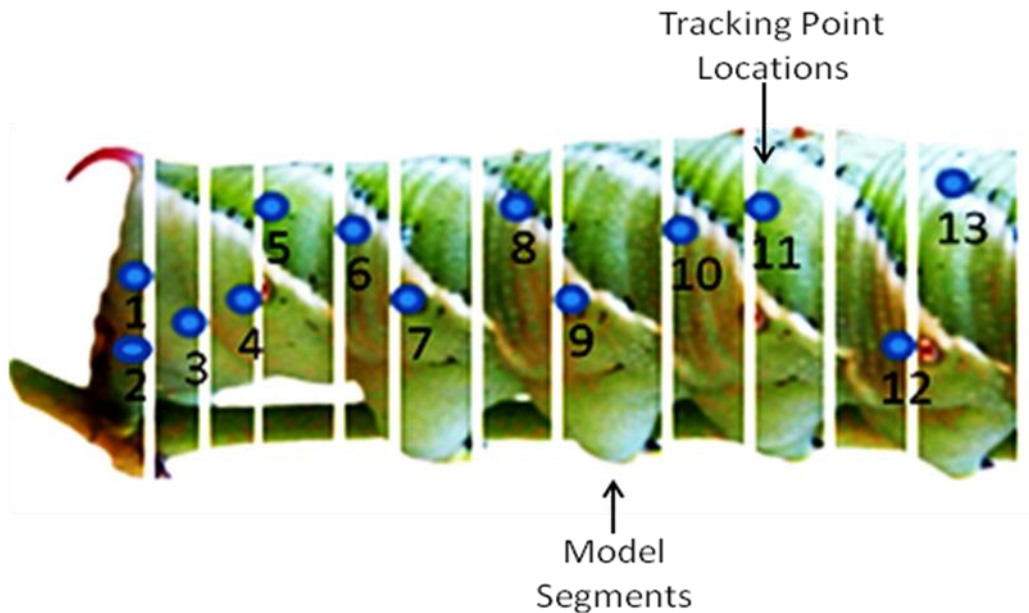


Figure 9: *Manduca* Point Tracking Locations

b. Ground Reaction Force Data

For the purposes of defining the boundary conditions (at joint 12) and validating the ground reaction forces computed for each of the prolegs, data collected from a custom force-plate apparatus was used. Simultaneous video and force-plate data were not available for a single specimen. As such, a database of ground reaction forces collected for eight trials involving several fifth instar caterpillars was used. Each force-plate trial provided reaction force data in two directions (horizontal and vertical) for all caterpillar prolegs in contact with the substrate [15]. For this analysis, the data to obtain a single trace of mean vertical and horizontal ground reaction forces were phase-aligned and averaged for each proleg.

3. Computation of Internal and External Forces

a. Inverse-kinematics

As a first step in computing forces and torques from the Newton-Euler equations, (1) and (2), a minimization procedure was applied to obtain link configuration. This process of solving for link configuration variables from tracking point data is called *inverse-kinematics* [56]. Altogether, four variables are needed to define the configuration of a single extensible link: center-of-mass horizontal and vertical location (\mathbf{x}_n), link length (L_n), and orientation (θ_n). For a series of extensible links connected by revolute joints, only two of these parameters are needed to describe each link after the first (e.g., link length L_n and orientation θ_n). Thus, the twelve link caterpillar model is fully described by twenty-six configuration variables.

Each tracking point provides two independent equations that can be used to solve for these configuration variables; hence thirteen tracking points are needed, as a minimum, to solve the configuration variables. An equation that relates the tracking-point position vector \mathbf{t}_j to the set of link configuration variables is:

$$\mathbf{t}_j = \mathbf{p}_{n(j)} - \mathbf{u}_{\parallel,n(j)} w_{n(j)} + \mathbf{u}_{\perp,n(j)} h_j \quad (5)$$

Because the number of tracking points is not the same as the number of links, the index j is used to refer to each tracking point, and the index n is used to refer to each joint and to the associated link. The notation $n(j)$ refers to the link index to which a particular tracking point j is attached. The parameters w_j and h_j describe the axial and transverse distances of the tracking point j from the link endpoint $\mathbf{p}_{n(j)}$, which is a function of the configuration variables for all links through n .

$$\mathbf{p}_n = \mathbf{x}_1 + \frac{L_1 \mathbf{u}_{\parallel,1}}{2} + \sum_{m=2}^n L_m \mathbf{u}_{\parallel,m} \quad (6)$$

Because the link is extensible, the axial position of the tracking point is allowed to change in time; however, it is assumed that the fractional displacement of the tracking point along the length (Δ_j) is constant.

$$\Delta_j = \frac{w_j}{L_n} \quad (7)$$

By contrast, the transverse distance h_j , is assumed constant. In *Manduca*, axial extension does not necessarily result in transverse compression, because of compressible air channels (trachea) that exist throughout the arthropod's body.

To obtain the configuration variables for the extensible-link model, equation (4) must be simultaneously solved for all tracking points. This nonlinear set of equations is solved as an optimization problem. Framing the problem as an optimization problem allows for the accommodation of more tracking points and incorporation of constraint equations. As part of the solution, for instance, constraints are included that the link end points were required to remain above the ground. The optimization problem which obtains a least-squares solution to the tracking-point equations subject to this constraint is written as follows.

$$\begin{aligned} \min J_1(\mathbf{z}) \\ \text{s. t. } \hat{\mathbf{y}} \mathbf{p}_n > 0 \end{aligned} \quad (8)$$

The cost function, J_j , is the square of the residual error associated with each tracking point.

$$J_1(\mathbf{z}) = \left\| \begin{array}{c} \mathbf{t}_1 - (\mathbf{p}_1 - \mathbf{u}_{\parallel,1} L_1 \Delta_1 + \mathbf{u}_{\perp,1} h_1 \\ \vdots \\ \mathbf{t}_j - (\mathbf{p}_{n(j)} - \mathbf{u}_{\parallel,n(j)} L_{n(j)} \Delta_j + \mathbf{u}_{\perp,n(j)} h_j) \\ \vdots \end{array} \right\|_2 \quad (9)$$

The optimization obtains the following configuration-variable vector \mathbf{z} .

$$\mathbf{z} = [\mathbf{x}_1^T L_1 \theta_1 \dots \mathbf{x}_n^T L_n \theta_n \dots]^T \quad (10)$$

b. Inverse-Dynamics

To compute internal and external forces an inverse-dynamics problem for the vector \mathbf{u} must be solved.

$$\mathbf{u} = [R_0^T F_1^T T_1 R_1^T \dots F_{11}^T T_{11} R_{11}^T F_{12}^T T_{12}]^T \quad (11)$$

Note that there are no joint forces or torques at the free end (joint 0) and no proleg reaction forces at the abdomen/thorax boundary (joint 12).

It is first useful to rewrite the force and torque balances for each link, (1) and (2), as a system of linear equations in terms in terms of \mathbf{u} . Assuming quasi-static conditions, the inertial terms are

set to zero. The remaining terms include the gravity vector \mathbf{g} and a configuration dependent matrix \mathbf{A} , which multiplies the force vector \mathbf{u} .

$$\mathbf{A}(\theta_n, L_n)\mathbf{u} = \mathbf{g} \quad (12)$$

In principle, the unknown forces \mathbf{u} could be computed by inverting the matrix \mathbf{A} , as long as there are a sufficient number of dynamic equations to ensure that \mathbf{A} is nonsingular. In this case, there are only 36 scalar equations, obtained by evaluating equations (1) - (3) for each link. However, there are 60 variables in the vector \mathbf{u} .

Because there are fewer equations than unknowns, additional constraints are needed to solve equation (12). The following assumptions were applied to constrain the inverse-dynamics analysis.

- The caterpillar may push but not pull on the ground. Thus, when a proleg is in contact with the ground, the vertical reaction force is positive $0 \leq \mathbf{R}_n$.
- Reaction forces are zero for joints with no proleg or for which the proleg is not in contact with the ground ($\mathbf{R}_{n-1} = \mathbf{R}_n = 0$). Contact is determined by visual inspection of video data and the application of a *Manduca* standard kinematics template.
- The torque at the boundary between the abdomen and thorax, T_{12} , is zero $T_{12} = 0$.
- The force at the boundary between the abdomen and thorax, \mathbf{F}_{12} , is computed directly from the ground-reaction force data $\mathbf{G}_{n,meas}$, by computing a force balance over the entire caterpillar body (given a quasi-static assumption).

$$\sum_{n=0}^{11} (\mathbf{G}_{n,meas} - M_n \mathbf{g}) = \mathbf{F}_{12} \quad (13)$$

In order to further restrict the space of allowable solutions for (12), an additional physical constraint related to the soft structure of the caterpillar was imposed. In the absence of a rigid internal skeleton, it is assumed that segments of the caterpillar raised off the ground are always in compression. This notion is illustrated by the concept diagram in Figure 10. Mathematically, the axial force component is positive at each joint for a link without a proleg in contact with the ground ($F_{\perp, n} \mathbf{u}_{\perp}(\parallel, n) > 0$).

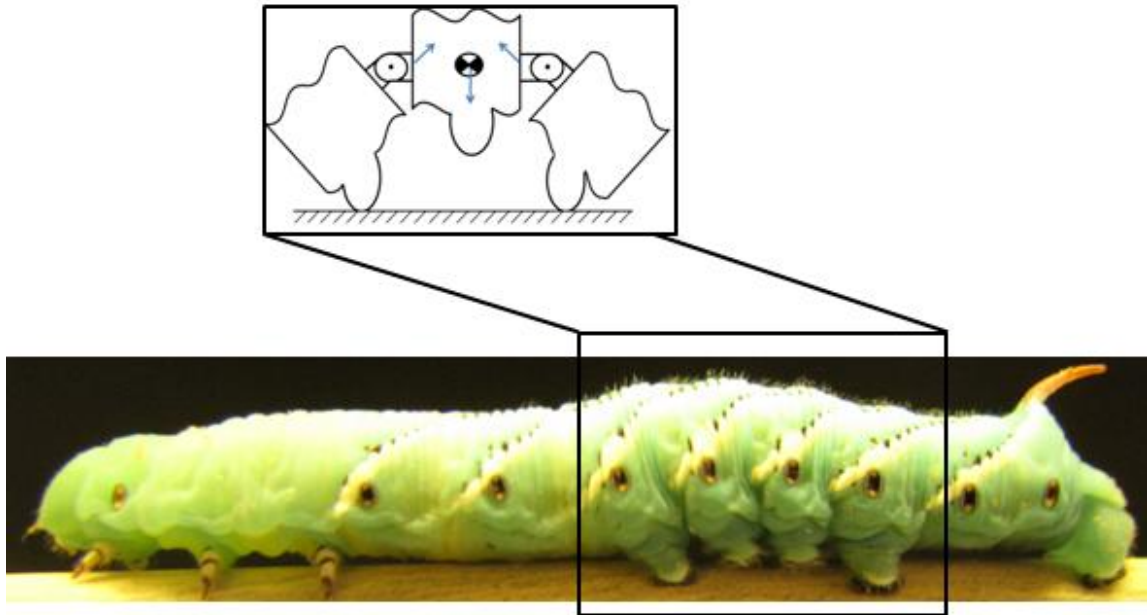


Figure 10: Lifted Proleg Compression

In practice, this set of equations is typically slightly over-constrained (53 or more equalities, and as many as 12 inequalities used to solve for 60 unknowns). As such, we use an optimization method to solve for the unknown forces.

$$\begin{aligned} \min J_2(\mathbf{u}) \\ \text{s. t. } \mathbf{G}_1 = 0, \mathbf{G}_2 \leq 0 \end{aligned} \quad (14)$$

Here \mathbf{G}_1 is the set of equality constraints discussed in this section and \mathbf{G}_2 is the set of inequality constraints. The cost function, J_2 , is the square of the residual errors for the torque and force balances of equation (12).

$$J_2(\mathbf{u}) = \|\mathbf{A}(\theta_n, L_n)\mathbf{u} - \mathbf{g}\|_2 \quad (15)$$

4. Verification Procedure

a. Comparison of Simulated to Measured GRF

The ground reaction forces collected for horizontal crawling (described in section II.2.b.) were used to verify the model performance. These data were compared directly to the outputs of the inverse-dynamics code. It should be noted that sum of reaction forces was applied as a boundary condition to the simulation, through equation (14), but that the individual reaction force values were not otherwise used in obtaining the inverse-dynamics solution. Simulated and experimentally measured values for ground reaction forces as a function of time are plotted for each set of prolegs in both the vertical (Figure 11) and horizontal directions (Figure 12). A correlation coefficient ρ for each pair of simulated and experimentally measured reaction forces was computed to quantify the comparison. The equation to compute the correlation coefficient for each simulated and measured time trace, $s(t)$ and $m(t)$ respectively, can be seen below.

$$\rho = \frac{\int s(t)m(t)dt}{\sqrt{\int s^2(t)dt} \sqrt{\int m^2(t)dt}} \quad (16)$$

The correlation coefficient has a value of unity if the simulated and measured traces are identical; it differs from unity if the phase or shape of the two traces differ.

b. Quasi-static Testing

It has been hypothesized that the *Manduca sexta* is quasi-static during horizontal crawling [17], meaning that inertia is negligibly small compared to the internal and external forces applied to each segment of the caterpillar's body. This hypothesis can be tested by re-computing the inverse dynamics, incorporating the values for the inertial terms on the left side of equations (1) and (2), rather than assuming these terms are equal to zero. To assess the degree to which the dynamic and quasi-static solutions differ, we can compare the correlation coefficients for the two cases.

III. Results

1. Ground Reaction Force Verification

Using inverse-dynamics simulation, ground reaction forces were computed for each of the caterpillar's five prolegs. Each proleg is associated with a particular joint (1, 5, 7, 9, and 11, Figure 7) in the extensible-link model. The ground reaction forces in both vertical and horizontal directions computed using the inverse-dynamics simulation were an excellent fit to those measured from crawling *Manduca* (Figure 11 and 12). The average value of the correlation coefficient over all of the vertical (normal) ground reaction force data sets was 0.914. The average value of the correlation coefficient over all of the horizontal ground reaction forces was

0.755. The total vertical (normal) reaction forces are equal the animal's weight (quasi-static) and the profiles of the vertical ground reaction forces result from varying proleg contact combinations.

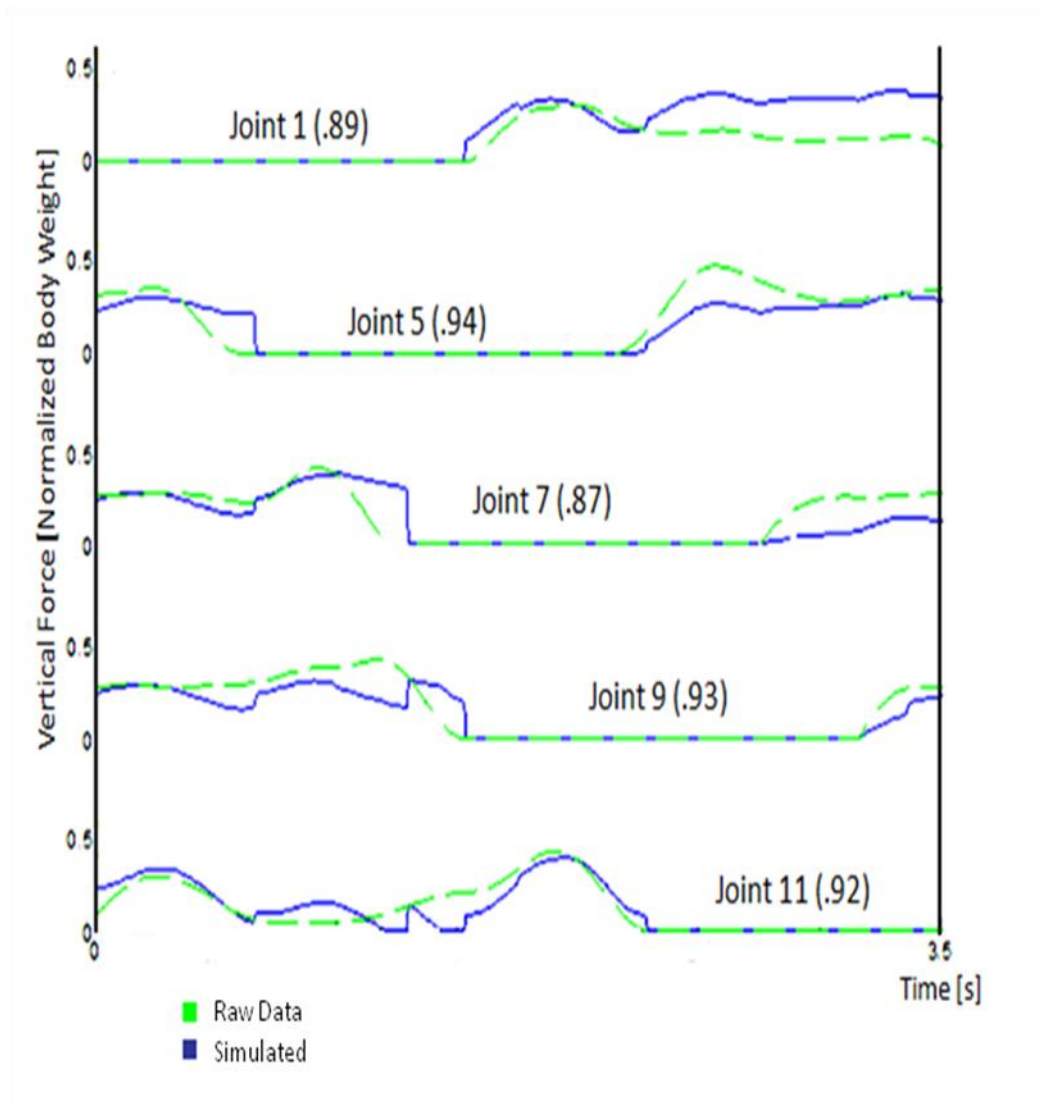


Figure 11: Vertical Comparison; Comparison of vertical ground reaction forces computed by inverse dynamics (solid line) to those measured experimentally on crawling caterpillars (dashed line). Force values are normalized by the caterpillar's body weight, which was 0.0416 N. A correlation coefficient was computed (next to the joint number) for each pair of time traces (simulation and experimental data) using equation (16).

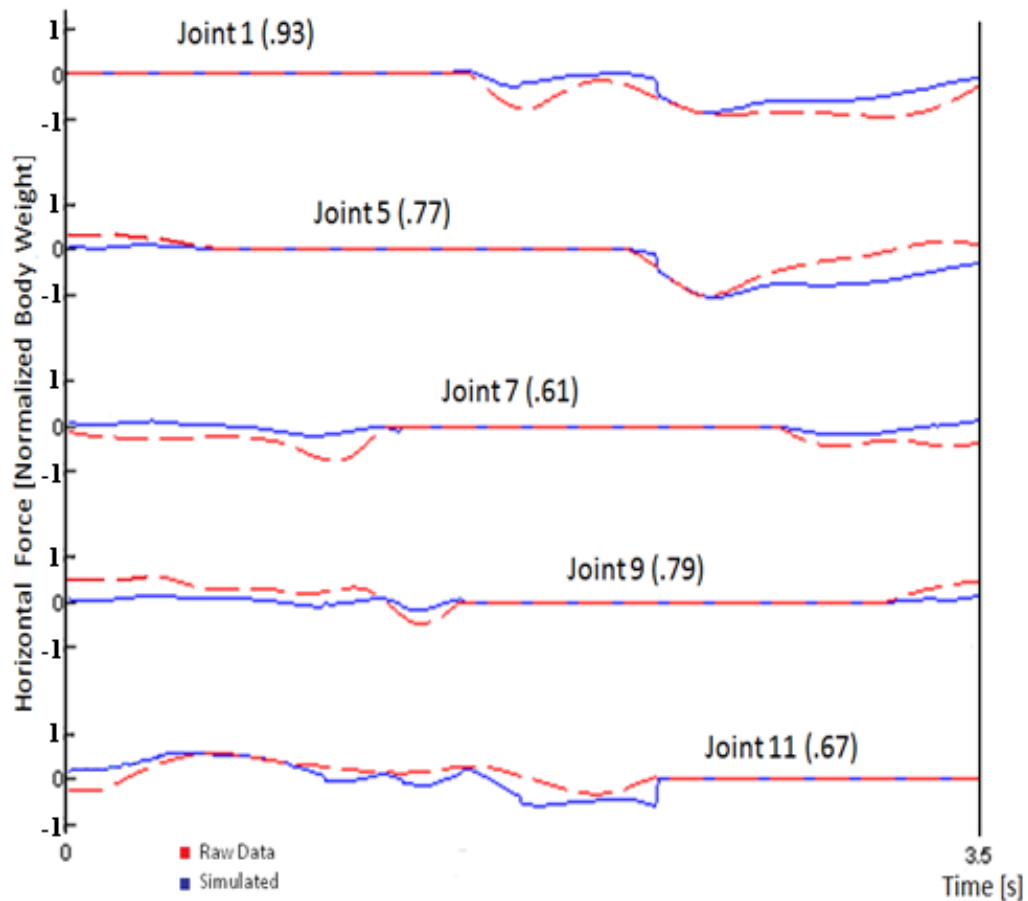


Figure 12: Horizontal Comparison; Comparison of horizontal (propulsive) ground reaction forces computed by inverse dynamics (solid line) to those measured on crawling caterpillars (dashed line). Force values are normalized by the caterpillar’s body weight, which was 0.0416 N. A correlation coefficient was computed (next to the joint number) for each pair of time traces (simulation and experimental data) using equation (16).

2. Quasi-Static Assumption Verification

The difference in ground reaction force values was negligible, whether or not inertial terms were considered in the inverse-dynamics simulation. The normal ground reaction force correlation coefficient changed by 0.02 percent from one case to the other; the axial ground reaction force

correlation coefficient changed by 1.6 percent. The maximum horizontal inertial force was 0.25 percent the animal's body weight. However, the maximum horizontal force produced by the prolegs was -132.80 percent the animal's body weight (the negative sign indicates the direction of the applied force). The maximum vertical inertial force was 0.26 percent the body weight of the animal. While the maximum normal force produced by a proleg was 39.63 percent of the animal's body weight. Taking a ratio of the horizontal and vertical inertial forces to proleg forces further verifies the quasi-static assumption.

3. Internal Forces

Internal forces were also computed by inverse-dynamics. These values comprise both active (muscle) and passive (structural) forces, which together produce caterpillar crawling. Computed forces are highly coordinated along the length of the caterpillar. To illustrate this coordination, axial internal forces for all links are plotted in Figure 13. These values are plotted for five time steps, which correspond to the lift off of each proleg from front to back (starting with the terminal proleg, at joint 1 and proceeding forward toward the anterior proleg at joint 11).

The axial internal force values can be described as a compressive wave traveling from the posterior toward the anterior of the caterpillar. Because of the constraints imposed (see Section II.3.b), links which are not in contact with the ground are in compression. Links with prolegs in contact with the ground, by contrast, are not constrained; however, these links are generally in tension, presumably to balance the compressive forces observed in other segments.

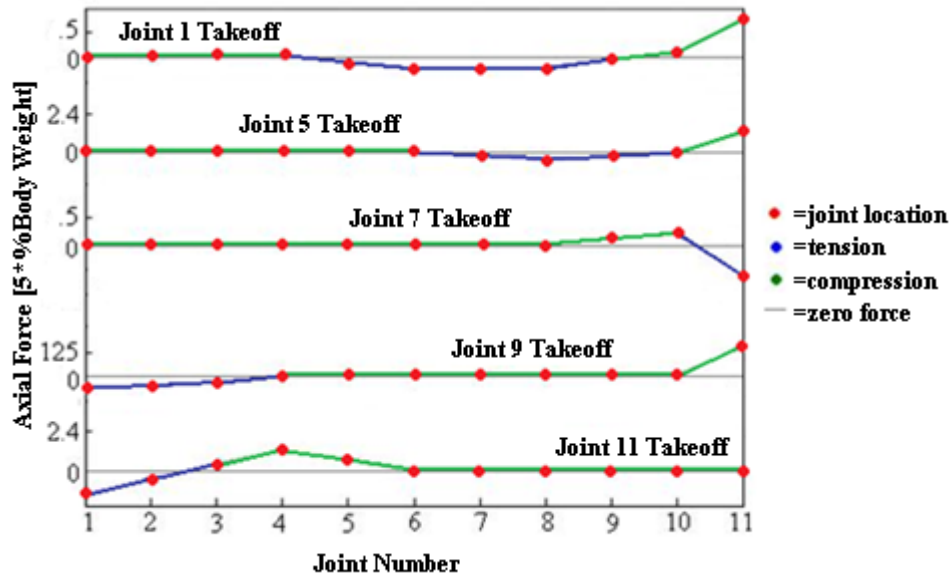


Figure 13: Simulated Internal Forces

IV. Discussion

Ground reaction force data collected from live specimens of *Manduca* has been used to verify the performance of a planar, extensible link model of caterpillar crawling. Given the model's relatively low complexity, it was able to match both normal and horizontal ground reaction forces for all prolegs with surprising accuracy (91% correlation in the vertical direction 76% correlation in the horizontal).

Remaining discrepancies between the predicted and actual ground reaction force data might be attributed to a number of modeling assumptions. These include; the number of extensible links selected, the assumption of tracking points fixed to links with constant h_j and Δ_j , the assumption of an even split of proleg reaction forces across joints, the assumption of compression in lifted prolegs, and the assumption of quasi-static motion. Despite its shortcomings, the model provides

a clear benefit, in that it provides a mechanism for estimating internal forces within a crawling caterpillar, forces which cannot currently be measured by experimental means.

Boundary conditions had a significant role in matching the raw ground reaction force data. Specifically, boundary conditions were the assumed vertical and horizontal forces contributed by the thorax (acting on joint 12), which were computed from the ground reaction force data using equation (13). This boundary condition was not a trivial function of time, as can be seen in Figure 14. In fact, the boundary condition function features several peaks which apparently correspond to liftoff of individual prolegs. This suggests a relationship between internal forces and proleg liftoff that was not explicitly included in the model. In the future, if a full-body (rather than just abdomen) model were constructed there would be no need to apply a boundary condition at joint 12; however, it is possible that the relationships between internal forces and proleg liftoff would need to be incorporated into the model as a new constraint.

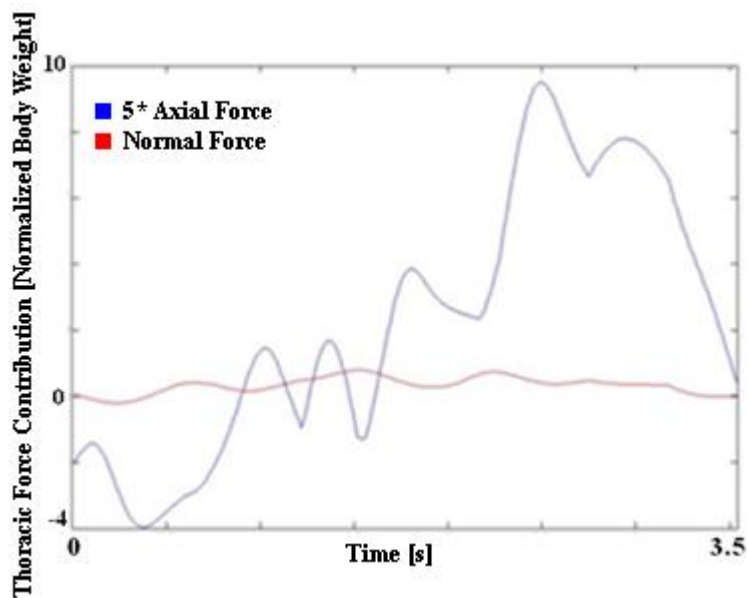


Figure 14: Thoracic Normal and Axial Force Contribution

This analysis strongly supports the validity of applying a quasi-static assumption to solve the Newton-Euler equations for a crawling caterpillar; the addition of inertial terms had a negligible effect on the correlation coefficient between simulated and measured ground reaction forces. The quasi-static assumption provides a computational simplification in equations, 1 and 2, and in applying the boundary condition at joint 12, as described by equation (13). A quasi-static model assumption is also useful for analyzing the mechanisms underlying caterpillar neuromuscular control. Experiments on the proprioceptive stretch receptor organs of *Manduca* show that changes in body segment length or transient force are not reliably reported to the central nervous system on a time scale necessary to control locomotion [57]. If caterpillar crawling is essentially quasi-static, then feedback based on inertial cues is unlikely to be a useful control parameter.

A significant outcome of the analysis is a prediction of how tensile forces are distributed throughout the caterpillar's body. During most of the gait cycle the net ground reaction force data indicate that at least some of the caterpillar's body segments must be in tension (after the first second, Figure 13). It might be expected that these tension forces are distributed evenly throughout the body; however, this is not the case. Rather, the simulation suggests that tension is concentrated in a highly localized region, adjacent to the compression region. The proximity of the compressive and tension terms is immediately evident in all of the frames of Figure 13 except the fourth. In the fourth frame, the links that are in compression and those that are in tension can be considered adjacent in a periodic sense (in which case link 11 is "adjacent" to link 1). This periodic interpretation may be critical transition in setting up each new wave pattern as the previous wave passes into the thorax at joint 11.

The model predicts that the distribution of internal tension is potentially an important mechanical parameter. This is supported by experimental findings from the neural patterns of muscle activation in *Manduca* [18] and from ground reaction forces during crawling [15]. Both sets of data imply that *Manduca* keeps its body largely in tension (during stance) and that compressive forces are passed from one part of the body to another through the substrate (the Environmental Skeleton strategy). Locomotion is achieved by broad waves of muscle contraction that produce this tension and by the precise timing of grip release by the prolegs [15]. Unlike classical hydrostatic locomotion in which the body is stiffened by pressurization, this strategy allows caterpillars to remain comparatively soft and to conform to their substrate. It also uncouples body pressurization from the motor control program (important in an animal that has internal air sacs) and potentially saves energy that would be expended to maintain a stiff body. When necessary, caterpillars can switch to hydrostatic stiffening to cantilever or burrow. The environmental skeleton strategy may not be confined to insect larvae; it is possible that octopus and other soft species with controllable grip can also exploit tension-based locomotion. The modeling results described here provide new insights into the possible distribution and shift of internal forces during such soft-bodied locomotion.

A second significant outcome of the analysis is that it establishes a basis for designing control forces for a caterpillar-inspired crawling robot. In principle, computed internal forces could be used directly (or scaled appropriately) to construct a control law. However, the model does not explicitly decompose passive forces, applied by structure, from active forces, generated by actuators. Recreating computed internal forces in a robot would thus require a careful decomposition between design of structure and design of control [58]. In order to create a

traveling wave motion capable of creating robotic locomotion a thorough understanding of a systems physical as well as control parameters must be completed.

V. Conclusion

A planar, extensible-link model for the *Manduca* caterpillar was introduced. The model was validated by comparing computed ground reaction forces to those measured by experiment. The model enables the estimation of internal forces within a crawling caterpillar, forces which cannot yet be measured experimentally. Internal force profiles (as a function of time) suggest that compressive and tensile forces are highly localized, rather than being distributed evenly along the caterpillar body. This evidence supports a new locomotion strategy for soft bodied locomotion in which compressive forces are primarily carried by the substrate allowing the body to remain soft and conformable. Both the caterpillar's physical characteristics as well as neural control create its ability to generate retrograde traveling compressive waves. Future research is necessary to understand the tuning of control and physical parameters in robotic platforms in order to create desired modes of locomotion.

Chapter III: Modal Eigenstate Determination for Reoccurring Dynamics

Executive Summary- For certain vibrating structures and undulating mechanisms, it is desirable to specify one or more mode shapes, but impossible to do so without consideration of both the system's passive dynamics and its feedback control. The problem of specifying mode shapes becomes increasingly complex for systems with few actuators and constrained physical parameters. In this chapter, the problem of Eigenstate specification is defined, which is referred to as Modal/Eigenstate Definition for Recurring Dynamics (MEDFRD). Multiple solution approaches to solve the MEDFRD problem for linear systems are then offered in the subsequent chapters.

I. Introduction

Many animals utilize phased periodic oscillations of body segments in order to create locomotion. One model animal that locomotes in this manner is the *Manduca sexta* caterpillar. *Manduca sexta* body segments displace in a periodic posterior-to-anterior traveling wave, as can be observed in Figure 15. The generation of this traveling wave involves both neural control and evolved biomechanical characteristics of the animal (e.g. gripping system and flexibility of body wall) [17]. The creation of periodic movements for engineered systems can be informed by the neuromechanics of the caterpillar; which utilizing both physical and feedback control parameters.

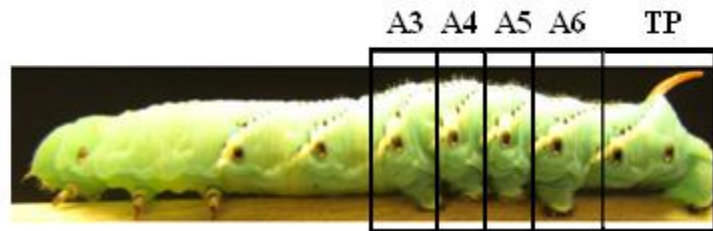
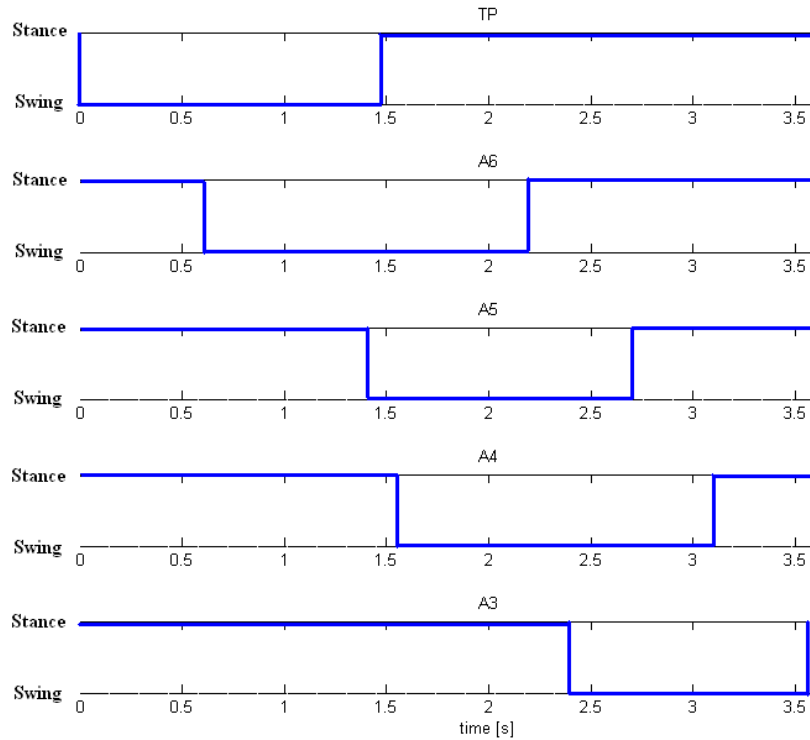


Figure 15: Caterpillar Locomotion Traveling Wave; Each pro-leg of the animal (TP, A6, A5, A4, and A3) is either attached to the substrate (stance) or lifted off (swing). The duration of liftoff is relatively consistent and phased, indicating a traveling wave movement.

The *Manduca sexta* caterpillar is able to generate a periodic phased oscillation of system states despite relatively simple neural control and severe under-actuation. This is only possible due to biomechanical aspects of the caterpillar that aid in the generation of its periodic displacement of segments. For many systems, especially those that are under-actuated, it is essential to generate

prescribed system dynamics through the manipulation of physical and control parameters. A thorough understanding of how physical and control parameters effect periodic movements must be completed before oscillations representative of those created by *Manduca sexta* can be designed into system dynamics.

One way to describe the oscillations of a system's states is to identify its modal dynamics. In a linear system, these modes are more specifically referred to as Eigenstates, which consist of Eigenvalue and Eigenvector pairs. Eigenstates determine the frequency of oscillations, the relative phasing of states, the relative amplitude of states, and the general shapes of state displacements, given a forcing function. Nonlinear systems may exhibit similar modal characteristics, although the values of modal properties (frequency, phasing etc.) may be a function of amplitude in nonlinear systems, whereas they are independent of amplitude in linear systems [59]. For linear systems it may be possible to embed the desired response of a system to a periodic forcing function into system modes/Eigenstates, and elicit behaviors similar to those observed in Figure 15.

Currently, it is also not well understood how to design modal behaviors to elicit desired system responses, similar to those of the caterpillar, given a periodic forcing input. However, new methods for embedding desired modal responses into dynamic feedback systems would be useful, particular for applications involving under-actuated mechanisms or robots. This chapter will discuss the problem of embedding desired periodic oscillations into the dynamics of a system through physical and control parameter alterations. This problem will be referred to as Modal/Eigenstate Determination for Reoccurring Dynamics (MEDFRD).

II. MEDFRD Problem

The first step in trying to specify modal dynamics for systems is to define the MEDFRD problem in a qualitative sense. The following section will apply this problem description to LTI systems, where the modal dynamics are well defined as Eigenstates, in order to obtain a quantitative mathematical problem description, to which solutions will be posed in subsequent chapters.

1. Qualitative Statement of MEDFRD Problem

The term *MEDFRD problem* refers to the challenging design problem of assigning patterns of oscillations to multi-degree of freedom physical systems. A solution to the MEDFRD problem should accomplish the following objectives.

1. Enable design engineers to specify reoccurring motion patterns for a system. It will be assumed all patterns are based on the same fundamental period and that the same modal amplitudes reoccur at any given phase within this period.
2. Be general enough to encompass un-actuated, under-actuated, fully-actuated, and over-actuated systems. The term *un-actuated* will refer to systems having no actuators; *under-actuated*, to systems having fewer independent actuators than modes; *fully-actuated*, to systems having exactly as many independent actuators as modes; and *over-actuated*, to systems having as many independent actuators as modes plus additional redundant actuators.

3. Allow for co-design through tuning of physical parameters (e.g. dimensions, mass, inertia) and of software defined parameters (control gains).
4. Enforce physical constraints of the system (e.g. positive mass, maximum size constraints).
5. Provide an approximate solution if an exact solution is not possible. The approximate solution should be “close” to the desired pattern of reoccurring motion.
6. Ensure that undesired dynamic modes, if they persist, are small. The designer may not desire (or be able) to specify all dynamic modes. To ensure that designer-specified modes dominate reoccurring dynamics, undesired modes must be mitigated.

In general, system dynamics may be described by linear or nonlinear models, ordinary or partial differential equations, and continuous or discrete time. As a first step in realizing solutions to the more general MEDFRD problem, this thesis restricts itself to analysis of continuous, linear time-invariant (LTI) systems.

2. Mathematical Model of MEDFRD Problem

This section will translate the qualitative statements of the previous section into quantitative representations, based on the assumption of a continuous LTI model for system dynamics. It is further assumed that desired mode shapes are specified to oscillate sinusoidally in time.

1. Specify Reoccurring Dynamics

Assume the system dynamics are LTI, such that passive dynamic evolution of the state vector \mathbf{z} is governed by a matrix of ordinary differential equations, where \mathbf{A} is a time-invariant matrix.

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} \quad (17)$$

By definition, reoccurring dynamics are periodic, but they need not be sinusoidal. However, if we restrict ourselves to superposition of sinusoidal oscillations, then reoccurring system dynamics can be specified as Eigenstates. This formulation is sufficiently general that mode shapes can be specified by defining the relative phase and amplitude of each system state. Consider the following case in which the desired reoccurring dynamics for the state vector $\mathbf{z}_{\text{desired}}$ are specified by a designer in terms of a single Eigenmode.

$$\mathbf{z}_{\text{desired}} = \mathbf{x}_1 \exp(\sigma_1 t) \quad (18)$$

The designer specifies a desired Eigenvector of the system \mathbf{x}_1 (relative phase and amplitude of each state) and a desired Eigenvalue σ_1 (decay rate and frequency of oscillations). More generally, it may be desirable to specify more than one Eigenstate, such that

$$\mathbf{z}_{\text{desired}} = \sum_{i=1}^N \mathbf{x}_i \exp(\sigma_i t). \quad (19)$$

Here N is the number of specified Eigenstates, which must be equal to or smaller than the number of states in the system. Note: to specify oscillatory dynamics (via a complex σ_i), it is

actually necessary to specify complex-conjugate Eigenstates, as well, so N must be at least two. In general, a system might have a large number of degrees of freedom (e.g. a snake robot), but it may only be possible (or desirable) to specify a reduced number of Eigenstates, as few as one complex-conjugate Eigenstate pair in some cases (e.g. one sinusoidal gait pattern with a particular fixed period).

2. *Allow for varying levels of actuation*

We have considered only passive dynamics, as in equation (17), but to account for software (feedback control) parameters, we must add in actuators. The modified system is

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u}, \quad (20)$$

where \mathbf{B} is a constant linear matrix and \mathbf{u} is a vector of actuators. Any number of actuators is permitted. In the limit of no actuators the system is equivalent to a purely passive design. When the number of actuators of \mathbf{u} is the same as the elements of \mathbf{z} , the conditions for full modal control (e.g., for specification of all system modes) are satisfied. In this sense, we unify two existing design problems (passive modal design and conventional feedback-based modal design), while also bridging the space between these problems by allowing for under-actuated or over-actuated systems.

3. Allow for tuning of physical and control parameters

A state-space model structure explicitly tabulates physical parameters in the \mathbf{A} and \mathbf{B} matrices. Software parameters are introduced through a control law. With the intention of preserving a continuous LTI model, we will assume full-state, linear feedback of the form:

$$\mathbf{u} = -\mathbf{Kz}. \quad (21)$$

Here \mathbf{K} is a constant gain matrix. The Eigenstates of the closed loop system are thus the Eigenstates of the matrix $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{BK}. \quad (22)$$

The dynamic system can be rewritten in the standard manner as:

$$\dot{\mathbf{z}} = \tilde{\mathbf{A}}\mathbf{z}. \quad (23)$$

By specifying Eigenstates of $\tilde{\mathbf{A}}$, we in effect specify quadratic equations for the elements of \mathbf{A} , \mathbf{B} , and \mathbf{K} . This type of problem, in which physical and control parameters are tunable is often referred to as co-design [14]. However, this variant of co-design is unique in that it collapses physical and control parameters into elements of $\tilde{\mathbf{A}}$, leaving their unique identification to the designer. For this thesis we suppress differences between tuning one parameter and another. As a result, we seek solutions that solve for an $\tilde{\mathbf{A}}$ matrix that achieves the desired Eigenstates as

closely as possible and that leaves the designer the freedom to set elements of \mathbf{A} , \mathbf{B} , and \mathbf{K} to achieve this $\tilde{\mathbf{A}}$.

4. Physical Constraints

Constraints may exist on elements of \mathbf{A} , \mathbf{B} , and \mathbf{K} . These can be mapped to elements of $\tilde{\mathbf{A}}$. Examples may be constraints on mass/inertia being positive (inequality constraints) or that a particular element of $\tilde{\mathbf{A}}$ be zero (equality constraint). We will assume that all constraints on $\tilde{\mathbf{A}}$ elements can be modeled in one way or another, either by equalities or inequalities on linear combinations of elements of $\tilde{\mathbf{A}}$. We will concatenate inequality constraints in a vector \mathbf{f} and equality constraints in a vector \mathbf{g} as follows.

$$\begin{aligned}\mathbf{f}(\tilde{\mathbf{A}}) &< 0 \\ \mathbf{g}(\tilde{\mathbf{A}}) &= 0\end{aligned}\tag{24}$$

A specialized subset of constraints are kinematic constraints that relate one state to another. For instance a velocity state may be simply the derivative of a position state. Such relationships cannot be modified by adjusting the elements of $\tilde{\mathbf{A}}$. Because they cannot be tuned, we instead embed these relationship directly into the desired Eigenvector \mathbf{x} . This requires a slight modification of equation (18).

When some Eigenstates may be kinematically coupled, it may be desirable to decompose the set of Eigenstates into those that can be freely specified and those that are kinematically constrained.

To specify consistent Eigenvectors it is necessary to define a vector \mathbf{x}_s (where the s subscript indicates specified elements of the Eigenvector); the remaining elements are clustered in a vector \mathbf{x}_{kd} (where kd indicates the kinematically defined components). Which elements are placed in \mathbf{x}_s and \mathbf{x}_{kd} is arbitrary, as long as all kinematic constraints are included in relating the two vectors as in the following equation.

$$\mathbf{x}_{kd} = (\mathbf{A}_{22} - \mathbf{I}\lambda)^{-1} \mathbf{A}_{21} \mathbf{x}_s \quad (25)$$

Here the $\tilde{\mathbf{A}}$ matrix has been decomposed into four sub-blocks.

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \quad (26)$$

The final target Eigenvectors (\mathbf{x}) have the form:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_s \\ \mathbf{x}_{kd} \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ (\mathbf{A}_{22} - \mathbf{I}\lambda)^{-1} \mathbf{A}_{21} \end{bmatrix} \mathbf{x}_s. \quad (27)$$

5. Approximate Solution Acceptable

Given under-actuation and design constraints, it may be possible that desired Eigenstates cannot be obtained exactly. In this case we want a solution that is “close” to the desired motion pattern. In order to define a notion of close, we note that the exact solution would result in

$$\tilde{\mathbf{A}}\tilde{\mathbf{X}} = \tilde{\mathbf{X}}\tilde{\Lambda}. \quad (28)$$

Here $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{\Lambda}}$ represent matrices of specified Eigenvectors and Eigenvalues. These matrices may be square, if Eigenstates are fully specified, or rectangular, if only a subset of Eigenstates is specified. By relaxing the equality in equation (28), we obtain a cost or penalty function J_1 , which should be minimized.

$$J_1 = \|\tilde{\mathbf{A}}\tilde{\mathbf{X}} - \tilde{\mathbf{X}}\tilde{\mathbf{\Lambda}}\|_F \quad (29)$$

Here the subscript ‘‘F’’ means the Frobenius norm. The Frobenius matrix norm is defined as the root-sum-of-squares of the elements of a matrix. Although many other means of evaluating ‘‘close’’ are possible, in this thesis we will focus on the Frobenius norm (because it is an effective measure of ‘‘close’’ as discussed in Appendix B).

6. Unspecified dynamics should be negligible

A designer may choose to specify the behavior of all system modes or only a subset of those modes. When some Eigenstates are unspecified by the designer, it is important that these Eigenstates not have slow or unstable dynamics, which might swamp specified Eigenstates. One way to achieve this is to ensure that Eigenvalues of unspecified dynamics are as small as possible. In this sense we might constrain the worst case Eigenvalue λ_k to be less than ϕ .

II. Conclusion

In this chapter the problem of recreating desired mode shapes in a system, by varying control and physical parameters has been both qualitatively and quantitatively introduced. In the subsequent

chapters alternative methods for solving the MEDFRD problem will be thoroughly discussed. Sub-problem 1 (that of enabling designers to effectively specify modal dynamics) will be addressed in the next chapter, Chapter IV. Sub-problems 2-4 (that of assigning system Eigenstates to match the designer's specification) will be addressed subsequently in Chapter V. Note, initially only equality constraints will be considered. A solution to the complete MEDFRD problem for LTI systems, including inequality constraints (an additional aspect of sub-problem 4), approximation (sub-problem 5), and mitigation of unspecified dynamics (sub-problem 6) will be presented in Chapter VI.

Chapter IV: Specifying Eigenstates Based on Fourier Decomposition of a Desired Periodic Trajectory

Executive Summary- Many animals use phased oscillations of body segments in order to produce locomotion. In order for a dynamic system to emulate the motions of these animals it is necessary to create coordinated periodic body movements. By tuning physical and control parameters (an idea sometimes referred to as co-design) it may be possible to specify the modal response of a system to generate desired periodic movements. The key innovation in this chapter is the introduction of a method that converts designer specified periodic motions, such as traveling wave patterns, into a set of system Eigenstates (Eigenvalues and Eigenvectors). In this thesis we will refer to the proposed Eigenstate design methodology as Fourier-Eigenstate Specification.

I. Introduction

A useful approach for generating periodic patterns of motion in an under-actuated dynamic system is to embed those patterns directly into the system's modal dynamics. In order to do this, new tools are needed to map desired trajectories into the dynamic system parameters that govern modal response. This chapter introduces a new approach for mapping designer-specified trajectories into dynamic system parameters for Linear Time Invariant (LTI) systems.

By designing modal responses of a system it may even be possible to eliminate the need to track desired reference trajectories for each state, while still generating periodic movements. Instead of creating controllers that drive each state to a desired reference value, it may be possible to design body plans and controllers that drive state trajectories over an extended period of time without the need for reference values. Essentially, the natural dynamics of the system and the control

inputs can utilize both the open and closed-loop responses of systems to generate periodic displacements.

The idea of embedding desired trajectories into modal system dynamics is, in part, inspired by biological systems, whose body shape and materials have evolved to generate complex periodic motions (e.g. walking) when excited by relatively simple neural signals. This idea is utilized by the *Manduca sexta* and many other animal species, as was described in Chapter II. Essentially, biomechanical characteristics offload the need for direct control of all actions. For example, it has been shown that the compliant nature of cockroach legs offer substantial disturbance rejection without active neural control [60]. The concept of building desired system responses into mechanical properties of a system has been investigated, and is commonly referred to as embodiment [4]. Embodiment, however, generally focuses on stability and energy recovery, rather than creating periodic displacements [4].

The embodiment concept offers potentially significant benefits for engineering applications, in the sense that exploiting physical structures and materials to embed desired dynamic behaviors can reduce controller complexity, reduce actuator power requirements, and reduce the number of actuators needed to generate desired motions. Reducing the number of actuators needed is a particularly significant benefit, as a reduced actuator count reduces mechanism cost while typically enhancing its reliability. However, designing a dynamic system that exhibits desired periodic motions is challenging, even when feedback control can be implemented. Some of the challenges include inherent limits on physical parameter values, actuator saturation, and control of under-actuated states. However, the primary challenge is a lack of understanding in

manipulating system dynamics through physical and control parameters to match desired periodic patterns.

Given the challenges of embedding desired dynamics into physical systems, particularly in the case of under-actuated systems, new tools are needed to assist designers in realizing physical systems that exhibit desired dynamic behaviors. Accordingly, this chapter focuses on defining design procedures that cleanly map desired specifications (e.g. periodic motion trajectories) into system parameters. In order to address this goal, it is first helpful to understand what impacts the natural dynamics of a system. Physical and control parameters determine the Eigenstates, and consequently natural dynamics, of a system. Eigenstates govern the modal and state responses of the system to a prescribed input, and are made up of Eigenvalue and Eigenvector pairs. Thus, the system response can be tuned by altering the Eigenstates [6].

For linear time-invariant systems, Eigenstate parameters provide a compact description of modal dynamics; hence, the process of embedding a desired trajectory into a system's natural dynamics may be viewed as a problem of converting that trajectory into a set of approximately equivalent Eigenstates. This chapter introduces such a method, referred to as *Fourier-Eigenstate Specification*, which creates state trajectories by embedding the desired system responses into the Eigenstates of the system. By doing so, phased periodic movements may be generated without the need for reference trajectory based control of each state, and complex state relationships can exhibit themselves based on simple periodic forcing functions applied to a single state.

This chapter will first give a complete description of Fourier-Eigenstate Specification method. Next, a case study of the method involving a hydraulic cavity system will be presented along with a brief discussion of potential applications and limitations of the proposed methodology. A conclusion will follow.

II. Fourier-Eigenstate Specification

This section introduces a methodology to specify a dynamic system based on a periodic designer-specified trajectory for each system state. The concept of the method is to match Fourier modes of the designer-specified trajectory to the natural resonance modes of the dynamic system. Then the desired trajectory can, in concept, be re-created simply by exciting the natural resonance modes with a sinusoidal actuator input. This surprisingly simple concept has not previously been presented in the research literature.

In this method, called Fourier-Eigenstate Specification, it is assumed that the dynamic system can be modeled as a linear time-invariant. If we represent a linear time-invariant system of differential state equations in a state-space representation, then the change of the vector states \mathbf{z} with respect to time can be described with

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u} \quad (30)$$

where \mathbf{A} and \mathbf{B} are constant matrices. It is assumed that both physical dynamics and state-feedback dynamics are both modeled by the \mathbf{A} matrix. An additional actuator input \mathbf{u} is still

required to excite the natural modes. The elements of this input vector are assumed to be sinusoidal functions of time. The actuator input \mathbf{u} is a scalar value described as:

$$u_i = \sum_i a_i \sin(\omega_i t + \varphi_i). \quad (31)$$

Here a represents the amplitude, ω represents the frequency and φ represents the phasing of i sinusoids. Where i is an index corresponding to column elements of the vector \mathbf{u} , each having a unique sine wave function. If a system is sufficiently controllable, then it may suffice to input the excitation signal using only a single actuator input.

It is assumed that the dynamic system of interest can be modeled by (30) and (31) and that physical and feedback parameters can be varied such that the designer is free to choose the values of the elements of the \mathbf{A} matrix. Given a dynamic system of this form and a set of specified periodic trajectories, the Fourier-Eigenstate method transforms determines the form of the \mathbf{A} matrix for which the natural dynamics represent a close approximation of the specified state trajectories. The algorithm for implementing this transformation is described in the remainder of this section. The primary steps include specifying desired trajectories, Fourier series decomposition, Eigenstate specification, and Bode analysis. A representation of these steps can be seen in Figure 16.

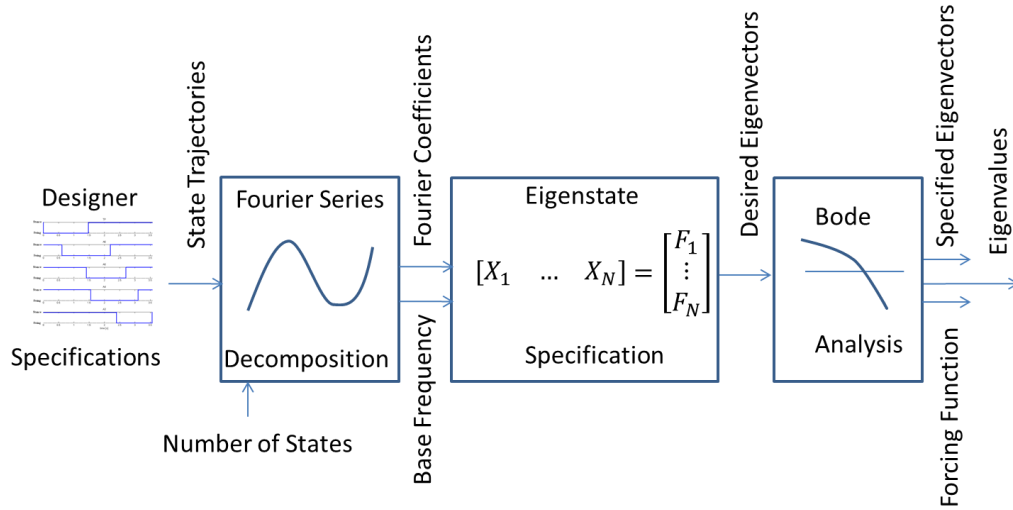


Figure 16: Fourier-Eigenstate Specification Steps

A more detailed description of capturing desired state trajectories into the natural dynamics of a system follows.

1. *Designer Specifications:* Desired state trajectories for a fixed period length must be determined. The desired trajectories for the states can be specified as follows,

$$\bar{z}_j = f_j(t) \quad \{0 < t \leq P\} \quad (32)$$

where there are state trajectories indicated from each state j for a time period P . Here \bar{z} stands for the nominal trajectory of each state and not the instantaneous value of the states. A vector of state trajectories can then be created in which each row represents the trajectory for a given state.

$$\bar{z} = f(t) \quad (33)$$

2. *Fourier Series Expansion:* The next step of Fourier-Eigenstate Specification is to compute a Fourier Series expansion for each state trajectory. A separate Fourier series expansion must be created for each state trajectory j . In order to attain a square Eigenvector matrix (capable of inversion) it is necessary that Fourier series expansion should be performed to $N/2+1$ terms. Given N is the total number of states, which is provided by the designer as an input to the Fourier decomposition step. If $N/2+1$ terms are used for Fourier series decomposition and the DC component of the Fourier series expansion is disregarded (described in detail in this section), $N/2$ specifications will be left. Because of a need for complex conjugate Eigenstates, these $N/2$ Fourier terms will fully define the system dynamics.

For any desired state trajectory $\bar{z}_j(t)$ described by the function $f_j(t)$ the Fourier Series Expansion for that state to $N/2+1$ terms can be determined by using:

$$\bar{f}_j(t) \approx \frac{1}{2} a_{0,j} + \sum_{n=1}^{\frac{N}{2}} (a_{n,j} \cos\left(\frac{n\pi t}{L}\right) + b_{n,j} \sin\left(\frac{n\pi t}{L}\right)) \quad (34)$$

$$a_{0,j} = \frac{1}{L} \int_{-L}^L \bar{f}_j(t) dt$$

$$a_{n,j} = \frac{1}{L} \int_{-L}^L \bar{f}_j(t) \cos\left(\frac{n\pi t}{L}\right) dt$$

$$b_{n,j} = \frac{1}{L} \int_{-L}^L \bar{f}_j(t) \sin\left(\frac{n\pi t}{L}\right) dt$$

$$L = \frac{P}{2}$$

The Fourier series expansion for each state, which is comprised of sines and cosines, must then be converted into a representation of purely sine waves for each frequency. In order to do this

each sine and cosine pairing with the same frequency is converted into a single sine wave with shifted phasing and amplitude according to:

$$a_n \cos\left(\frac{n\pi t}{L}\right) + b_n \sin\left(\frac{n\pi t}{L}\right) = \sqrt{a_n^2 + b_n^2} \sin\left(\frac{n\pi t}{L} + \tan^{-1}\left(\frac{-b_n}{a_n}\right) + \frac{\pi}{2}\right) \quad (35)$$

After this step is complete each desired state trajectory $\bar{z}_j(t)$ can be described by:

$$\bar{f}_j(t) \approx \frac{1}{2} a_{0,j} + \sum_{n=1}^{\frac{N}{2}} \sqrt{a_{n,j}^2 + b_{n,j}^2} \sin\left(\frac{n\pi t}{L} + \tan^{-1}\left(\frac{-b_{n,j}}{a_{n,j}}\right) + \frac{\pi}{2}\right) \quad (36)$$

$$a_{0,j} = \frac{1}{L} \int_{-L}^L \bar{f}_j(t) dt$$

$$a_{n,j} = \frac{1}{L} \int_{-L}^L \bar{f}_j(t) \cos\left(\frac{n\pi t}{L}\right) dt$$

$$b_{n,j} = \frac{1}{L} \int_{-L}^L \bar{f}_j(t) \sin\left(\frac{n\pi t}{L}\right) dt$$

$$L = \frac{P}{2}$$

It is assumed that the physical design for the systems of interest is such that the true DC value of all states must be zero. However, this constraint is not necessarily enforced when the designer specifies state trajectories (to simplify the user input process). Because the specifications do not necessarily obey the 0 DC value constraint, this constraint must be enforced and equation (36) is re-written with $a_{0,j} = 0$ for all states.

After compiling the Fourier decompositions for each state into sine waves and removing the DC components, each state trajectory can be represented by a summation of Fourier coefficients

($F_{n,j}$) and base frequencies $\left(\frac{n\pi}{L}\right)$. $F_{n,j}$ is a complex number that represents the phasing and amplitude values depicted in equation (36). Specifically, the magnitude of the complex number is the amplitude of the sine wave $\left(\sqrt{a_{n,j}^2 + b_{n,j}^2}\right)$. The angle of the complex number is the phase angle $\left(\tan^{-1}\left(\frac{-b_{n,j}}{a_{n,j}}\right) + \frac{\pi}{2}\right)$.

$$\bar{f}_j(t) \approx \sum_{n=1}^{\frac{N}{2}} F_{n,j} e^{\left(\frac{n\pi}{L}jt\right)} \quad (37)$$

The Fourier coefficients for each state can be compiled in a vector \mathbf{F}_j , as follows.

$$\mathbf{F}_j = [F_{1,j} \quad \dots \quad F_{N/2,j}] \quad (38)$$

A matrix of Fourier coefficients can then be generated by considering all states of the system.

The form of the Fourier coefficient matrix is as follows.

$$\mathcal{F} = \begin{bmatrix} \mathbf{F}_1 \\ \vdots \\ \mathbf{F}_N \end{bmatrix} \quad (39)$$

Given the Fourier coefficient matrix an approximation of the desired trajectories can be generated as follows.

$$\bar{\mathbf{z}} = \mathcal{F}\boldsymbol{\omega}(\mathbf{t}) \quad (40)$$

Where

$$\boldsymbol{\omega}(\mathbf{t}) = \begin{bmatrix} e^{\left(\frac{\pi j}{L}\mathbf{t}\right)} \\ \vdots \\ e^{\left(\frac{N/2\pi j}{L}\mathbf{t}\right)} \end{bmatrix}. \quad (41)$$

3. Eigenstate Specification: The output of the Fourier decomposition procedure was a matrix of Fourier coefficients, \mathcal{F} , and a base frequency ω_{base} (equal to π/L). The task of the Eigenstate Specification block is to convert \mathcal{F} into a proposed Eigenvector matrix \mathbf{X} . Eventually, the full \mathbf{A} matrix will be generated from the Eigenvector matrix \mathbf{X} and an Eigenvalue matrix Λ (which will be determined in the next section).

$$\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1} \quad (42)$$

The key result for Eigenstate specification is that the Eigenvector matrix \mathbf{X} can be directly derived from the Fourier coefficient matrix \mathcal{F} . The rationale for this mapping is described below.

First, it is assumed that the sinusoidal perturbation input can be defined in a way that allows arbitrary control over the phase and amplitude of the oscillations. (The details of defining the input sinusoid are managed in the following section.) Given that the amplitude and phase of the input frequencies are fully tunable, then the input frequencies can be matched to the harmonics of ω_{base} , such that each mode (e.g. Eigenstate pair) oscillates at a particular harmonic. Given this

assumption the dynamic system can be matched to the finite-term Fourier expansion of the desired user trajectory by setting

$$\mathbf{z} = \tilde{\mathbf{X}}[\boldsymbol{\omega}(t)\overline{\boldsymbol{\omega}(t)}]^T = \mathcal{F}\boldsymbol{\omega}(t) \quad (43)$$

Here it is assumed that every Eigenstate is a complex conjugate and hence, those Eigenstates are repeated, such that the number of Eigenstates is twice as large as the number of Fourier modes. Since every Eigenstate has a complex conjugate we can write the following for simplicity.

$$\tilde{\mathbf{X}} = [\mathbf{X} \quad \bar{\mathbf{X}}] \quad (44)$$

This notation splits one set of Eigenvalues ($\boldsymbol{\omega}(t)$) into the script matrix \mathbf{X} and their complex conjugates ($\overline{\boldsymbol{\omega}(t)}$) into the companion matrix (where the $\bar{\mathbf{X}}$ indicates the complex conjugate). Because each of the complex conjugate modes contributes half of the response, equation (43) and (44) can be re-written as:

$$\mathbf{z} = 2\mathbf{X}\boldsymbol{\omega}(t) = \mathcal{F}\boldsymbol{\omega}(t) \quad (45)$$

It is clear from this equation, that the desired trajectory can be embedded in the natural dynamics by selecting the Eigenvectors such that

$$\mathbf{X} = \frac{\mathcal{F}}{2}. \quad (46)$$

4. *Bode Analysis*: In order to control the system it is necessary to insert a sine wave forcing function. This section will demonstrate the effects of a forcing function on system Eigenvectors, determine the Eigenvalues of the system, and determine the appropriate forcing function to elicit the desired dynamic behaviors.

Given that the periodic steady-state response can be approximated as

$$\mathbf{z} = \mathbf{B}\boldsymbol{\omega}(t), \quad (47)$$

if you excite with a superposed set of sinusoids of the form of equation (31)

$$u_i(t) = \sum_i a_i \sin(\omega_i t + \varphi_i),$$

then the systems modal responses can be found to be

$$\mathbf{z} = \sum_i \sum_n \tilde{X}_n \mathbf{G}_n(\omega_i) / 2 a_i e^{j(\omega_i t + \varphi_i)}. \quad (48)$$

Here the vector \mathbf{G} is the vector of transfer functions from the input “ i ” to Eigenmode n . The value of each element of \mathbf{G} is directly related to the corresponding Eigenvalues which are the base frequencies of the Fourier decomposition (and is the same for complex conjugates). A derivation of \mathbf{G} can be found in Appendix C. If it is assumed that the excitation frequencies are

harmonics of the base frequency for the designer-specified trajectories, then equation (48) can be re-written in terms of only a single entry from each Eigenstate pair:

$$\mathbf{z} = \mathbf{X}\mathcal{G}\mathbf{U}\boldsymbol{\omega}(t) \quad (49)$$

where \mathcal{G} is the matrix of \mathbf{G} vectors

$$\mathcal{G} = [\mathbf{G}(\omega_1) \mathbf{G}(\omega_2) \dots \mathbf{G}(\omega_{N/2})] \quad (50)$$

and where \mathbf{U} is a diagonal matrix of complex actuator coefficients that embed the amplitude and phase of each frequency in the actuator input u_i .

$$\mathbf{U} = \text{diag}([a_1 \exp(j\varphi_1) \ a_2 \exp(j\varphi_2) \ \dots \ a_{N/2} \exp(j\varphi_{N/2})]) \quad (51)$$

For a general system that it would be more appropriate to have set

$$\mathcal{F} = \mathbf{X}\mathcal{G}\mathbf{U}. \quad (52)$$

However, the assertion that $\mathcal{F} = 2\mathbf{X}$ is still of interest if the input amplitudes and phases of the exciting signal can be defined such that

$$\mathbf{G} \mathbf{U} = 2\mathbf{I} \quad (53)$$

Resonance can be exploited in order to reduce the size of the input signal. For this reason, the Eigenvalues are intentionally selected such that they are lightly damped (damping ratio ζ is near zero) and have natural frequencies tuned to the harmonics of the specified trajectories. That is

$$\lambda_n = -\zeta n\omega_{base} \pm jn\omega_{base}. \quad (54)$$

A damping value of zero is not desired due to the inherent risk of instability, but a very low damping value leverages resonance and create a matrix which is largely decoupled. For this case, when the excitation frequencies are also harmonics of the specified trajectories, the matrix becomes approximately diagonal. In other words, the diagonal terms $G_n(n\omega_{base})$ are all much larger than the off-diagonal elements with $G_n(m\omega_{base})$ for $m \neq n$. If is diagonal and \mathbf{U} is also diagonal, then each excitation frequency has an amplitude/phase relationship that is easily found. In short, for small ζ ,

$$\mathbf{u}_n = 2\mathbf{G}_n(n\omega_{base})^{-1}. \quad (55)$$

III. Case Study

1. Model System

In this section a series-cavity hydraulic actuator will be investigated. Changes in cavity volumes are reminiscent of the lateral displacements of caterpillar pro-leg segments during crawling (see Chapter II). For this example we will assume that the deformable elastic cavities are connected to

one another in series, and a gear pump is also connected to each cavity (as can be seen in Figure 17).

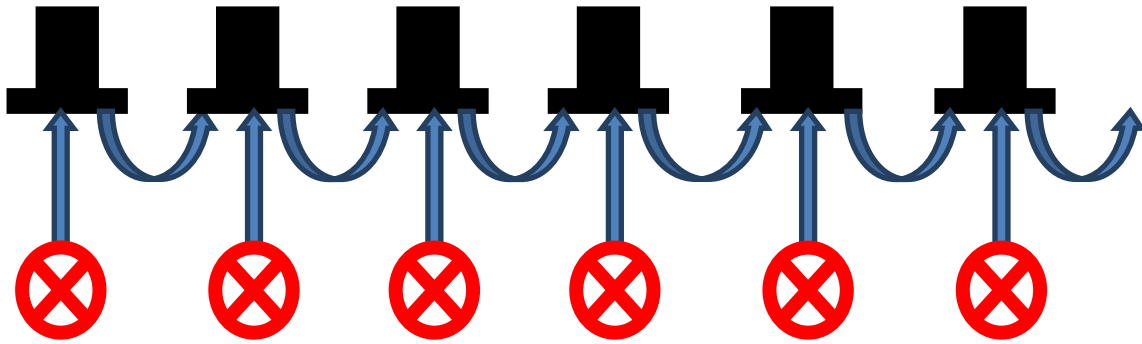


Figure 17: Cavity Model System; Pumping water through a number of dynamically coupled cavities to produce phased deformation of elastic regions. The gear pumps are indicated in red, and the blue arrows indicate tubes in which water may flow.

The proposed hydraulic actuator is assumed to comprise a set of deformable elastomeric cavities, a set of feedback controlled gear pumps, and pressure sensors in each cavity allowing for closed-loop control. Full actuation was selected for this model system so that it is immediately evident that each element of the \mathbf{A} matrix of the system is linearly independent. However, since morphological parameters can be altered it may not be necessary to include a pump on each cavity. The state-space representation of this system can be found in Appendix D. The state-space model has dimension N equal to 6, which constrains the Fourier model of trajectories to 3 modes (each with a complex conjugate).

2. Application to Model System

In order to demonstrate the Fourier-Eigenstate specification method a traveling wave of state trajectories for the cavity system will be specified and embedded into the system's modal dynamics. Each step in the Fourier-Eigenstate specification algorithm (depicted by Figure 16) will be described and applied to the model system in order to assess how well the method functions.

Desired state trajectories for the six cavity system were specified and can be seen in Figure 18 as blue square waves for each state. For this application on/off square wave cavity volume displacements were selected, without concern for the DC component of the trajectories. Without considering the DC components, the designer can simply specify an on time as positive and an off time as zero. The DC value for each state will later be removed by the Fourier analysis step in the design process (as an added constraint). However, any desired reference trajectory form is acceptable in the state trajectory specification step, as can be seen in equation (32). For this example, the total period for each state oscillation desired was equal to 3π sec, and the duration of each displacement was equal to π sec.

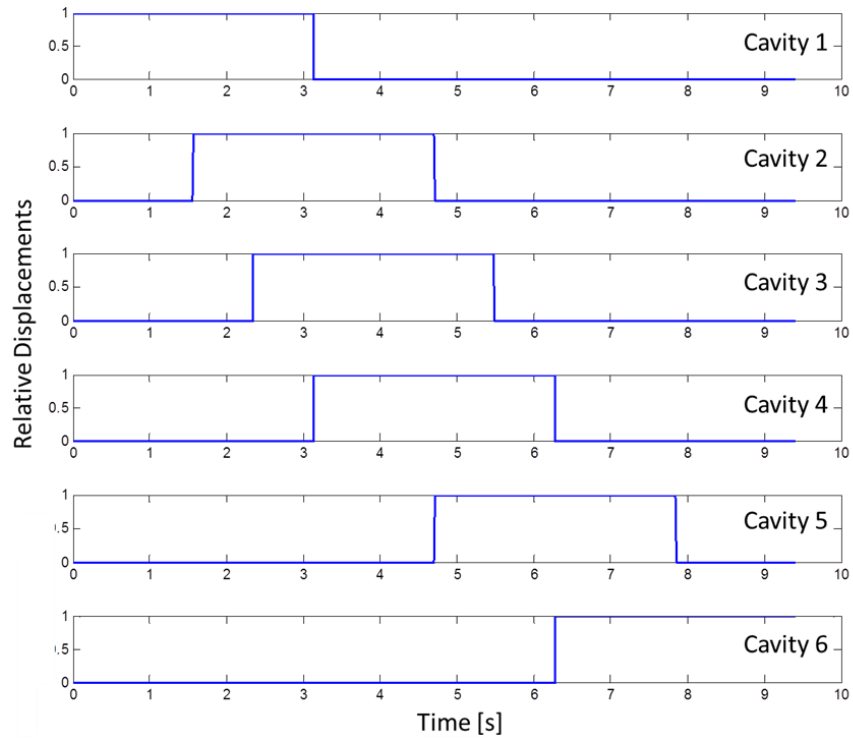


Figure 18: State Trajectories; Blue lines are prescribed relative volume changes of the cavities [normalized by $3.5 \times 10^{-4} \text{ m}^3$] with respect to time in seconds. Note that the cavity displacements correspond to the creation of a traveling wave similar to that of Figure 15.

Fourier Series expansion was performed on the system to give the sine wave representations of each state trajectory seen in Figure 19. The number of modes used to fit each desired trajectory was 3, or half of the total number of states. It was specified that a 30% duty cycle (30% “on” and 70% “off”) would be used for the originally desired trajectories. This ensured that higher modes would be necessary, since only a standing wave with 50% duty cycle can be created with a single mode. Equation (40) was used to generate representative sine waves from the Fourier decomposition of the desired trajectories.

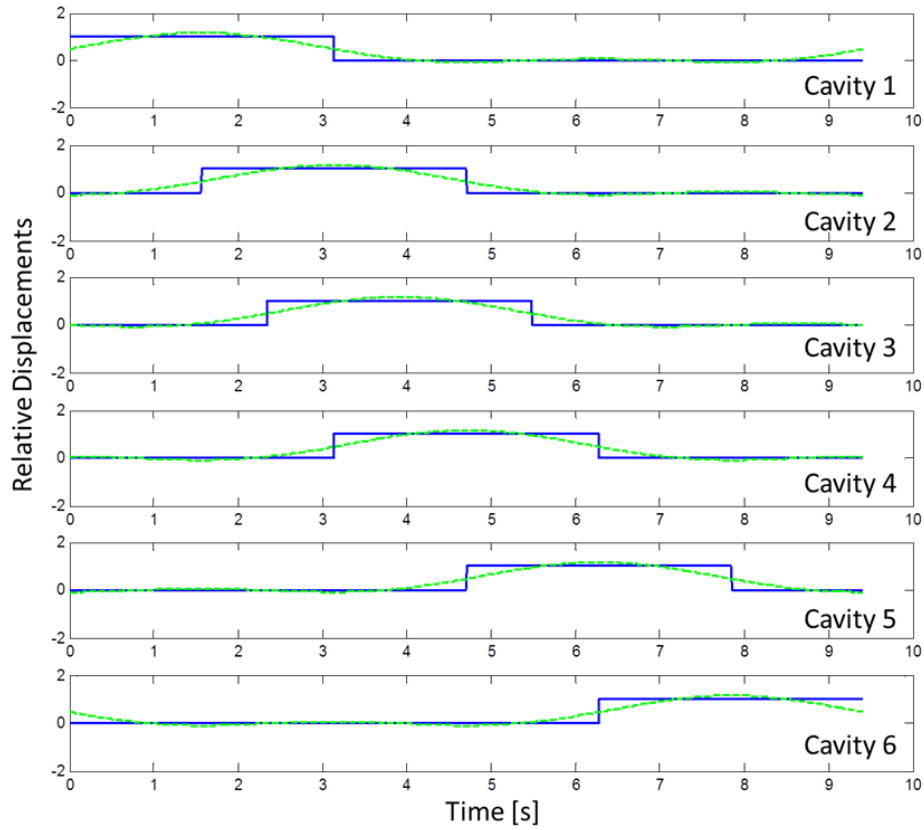


Figure 19: Sine Wave Representations of Each State Trajectory; The green line is the match based on Fourier series expansion of the desired trajectories shown in blue. The y-axis is relative volume change of the cavities when normalized by $3.5 \times 10^{-4} m^3$.

The sine wave representations were used to specify all desired Eigenvectors using equation (46) and all desired Eigenvalues using equation (54). These Eigenvalues and Eigenvectors, along with an assumed damping coefficient of 0.001 (a low damping value used to ensure diagonalization of the matrix \mathbf{A}), can be mapped into a desired \mathbf{A} matrix for the system using equation (42). After applying the Bode analysis to the desired system, and determining the appropriate periodic forcing function using equation (55), the system responded as can be seen in Figure 20.

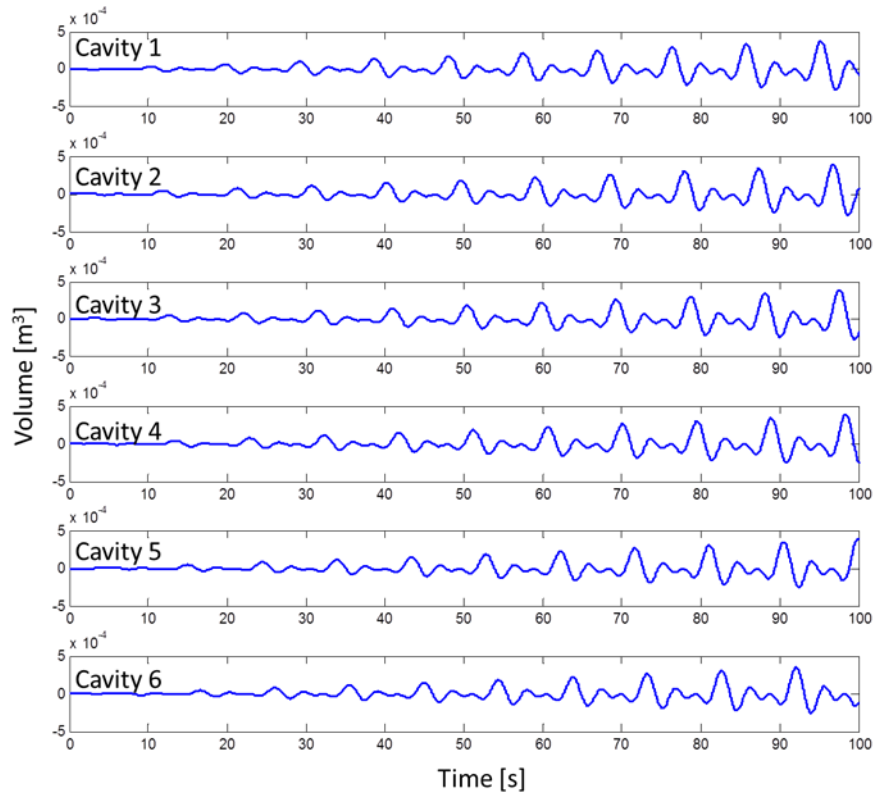


Figure 20: Sinusoidal Forcing Function Response

Based on the system response displayed in Figure 20 it is clear that the system did not approximately reach steady-state operation until 75 seconds. The steady state time is dependent upon the forcing function applied which is a function of the damping values. A natural consequence of using Fourier-Eigenstate specification is that a steady state response is only achieved after a finite amount of time. The amplitude reached by the system states is a function of the amplitude of the input forcing function. As damping values for the prescribed Eigenvalues increase it become extremely important to apply equation (48), so that the desired responses can be achieved given that the modes will respond with different decay rates for high damping systems.

A comparison of the desired response and the Fourier representation to the system response can be seen in Figure 21. The system response data was taken from one full period from 75 to 84.42 seconds on Figure 20 and is indicated in red. The desired trajectories are indicated in blue and the sine wave representations are green. It is clear that the system response is indicative of the desired and Fourier based responses in terms of amplitudes of oscillations, relative phasing of oscillations, and frequency of oscillations, as well as the general shape of the displacements.

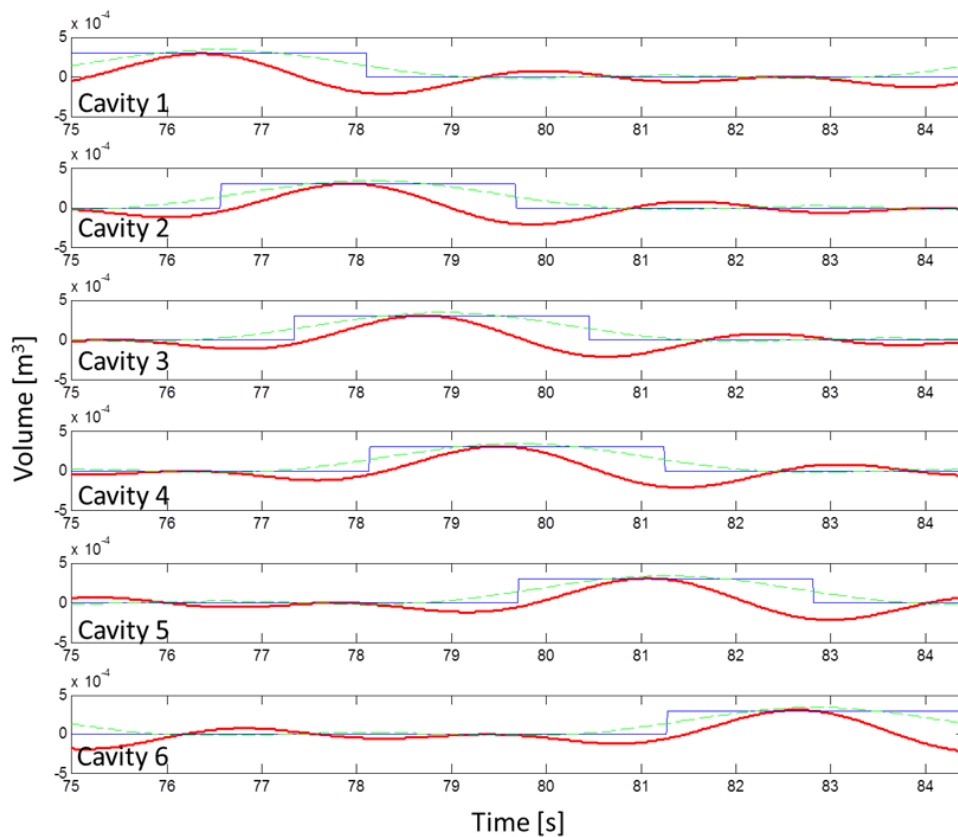


Figure 21: System Response Compared to Desired Response

A loss of fidelity is associated with each step in the transformation from the designer's desired trajectories to the system responses. When the desired trajectory is converted into a

representation of sine waves (green in Figure 21) a noticeable loss in fidelity can occur. This is because only a limited number of sine waves can be captured into a system's Eigenstates and consequently only a limited number of sine waves can be used to represent the desired trajectories. The consequence of using fewer sine waves is a rounding off of sharp trajectory changes (blue lines in Figure 21) and a lack of ability to match highly complex trajectories. Fidelity is also lost as the sine wave representations are converted into Eigenstates. The primary loss in fidelity is the need to include damping into the Eigenvalues of the system; a characteristic not captured in the Fourier series decomposition of the state trajectories. The effects of including damping can be seen in the red lines of Figure 21, and forces the system and desired trajectories to alter slightly over time.

IV. Conclusion

A method for tuning the response of a system to match specified state trajectories has been introduced, and is referred to as Fourier-Eigenstate Specification. A model system has been used to demonstrate the capabilities of Fourier-Eigenstate Specification. Eigenstates of the model system were determined to result in the desired state trajectories a priori, and were validated in simulation with the addition of a periodic sinusoidal input. Future work addresses the ability of a system to match the desired Eigenstates given limited actuation and parameter constraints.

Chapter V: Embedding Desired Eigenstates into Active and Passive Dynamics of a Feedback System Subject to Equality Constraints

Executive Summary- As was previously discussed in Chapter IV, Eigenstates (Eigenvalues and Eigenvectors) of a feedback control system can be specified to create repeated patterns of motion. Once, however, the Eigenstates needed to produce a desired system response to a given input are specified, new methods are needed to tune physical and feedback parameters to match the specification. No method currently exists to select feedback and physical parameters to instantiate desired Eigenstates for an under-actuated LTI system. This chapter introduces a novel methodology that allows Eigenstate specification in previously unexplored ways; by altering physical and control parameters concurrently. The method requires the solution of a particular class of Eigenvalue problem, referred to as Dual-Domain Eigenstate Factorization, in which both certain matrix parameters and also certain elements of that matrix's Eigenstate elements are constrained.

I. Introduction

Previous chapters have described how a desired periodic trajectory might be specified as a set of system dynamics (in terms of Eigenvectors and Eigenvalues) that can be excited by a simple sinusoidal input. The challenge is how to assign those desired dynamics for high degree of freedom systems that are under-actuated (e.g. systems for which not every degree of freedom is controlled by a separate actuator).

For some under-actuated systems it's easier to simulate the performance of a system given guessed parameter values and iterate until a desired behavior occurs. However, the interactions

between performance and parameters often leads to a large non-convex search space with a substantial number of locally optimal solutions [61]. Consequently, if system parameters that result in a desired behavior can be determined directly, iteration of the inverse problem is not necessary.

This chapter proposes an approach for designing dynamic systems that tunes feedback parameters and physical parameters to instantiate a desired set of dynamics. As a first attempt to solve the MEDFRD problem, we limit our focus to the simplest set of dynamic systems, that of linear time-invariant (LTI) systems.

No method currently exists to select feedback and physical parameters to instantiate desired Eigenstates for an under-actuated LTI system. Although Eigenstructure assignment methods have used controllers to specify whole system Eigenstates (assuming full actuation) [62], to my knowledge, no methods allow for Eigenstate specification while tuning control and physical system parameters of a system concurrently. Alternatively, although research has considered the design of physical and control parameters of a system simultaneously, an idea referred to as *co-design*, this idea has not been leveraged for Eigenstate specification [11]. By allowing physical and control parameters to alter, issues such as under-actuation may be resolved, so that a system's Eigenstates can still be specified even when every degree of freedom is not controlled [63].

One way to get as close to a desired trajectory as possible, even when under-actuation occurs, is to design desired dynamics into the physical system such that need for feedback control is reduced.

This idea is sometimes referred to as *embodiment* [5]. For example, by building compliant materials into a walking or crawling robot's legs disturbance rejection can in some cases be accomplished without the need for active control [60]. In a similar vein, if resonant modes can be designed into a physical system, then it may be possible to create periodic oscillations, even if the system is under-actuated. As will be shown in the next section, the process of assigning resonant modes to a linear time invariant (LTI) system gives rise to a generalization of the MEDFRD problem, which I label as the *Dual-Domain Eigenstate Factorization Problem*.

The remainder of this chapter will focus on specifying Eigenstates of systems while including constraints on system parameters; whether control or physical. The solution approach offered in this chapter will allow for both physical and control parameters to be concurrently tuned. In the next section a brief overview of the problem statement will be given; a more complete description can be found in [64]. Following the problem definition fundamental mathematical arguments to solving the proposed problem will be introduced. A specific algorithm that attempts to achieve the mathematical representation will then be discussed. A case study will then be implemented to investigate the proposed solution methodology. The methodology as well as problem relaxations will then be discussed. Finally, a conclusion will summarize the findings of this chapter.

II. Dual-Domain Eigenstate Factorization

Achieving a desired set of dynamics is nontrivial, even for LTI models of physical systems, because constraints exist in two domains. Specifically, constraints limit the choice of both the dynamic modal parameters (the Eigenstates of the LTI system) and the time-domain coefficients

of the LTI model. For instance, in typical LTI models, certain time-domain coefficients may be constrained equal to zero or one. To understand why the distinction between constraints on modal parameters and time-domain coefficients is relevant, consider the structure of a conventional Eigenstate factorization of the form:

$$\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}. \quad (56)$$

The time-domain coefficients may be thought of as elements of the \mathbf{A} matrix, and the modal parameters as elements of the Eigenvalue matrix $\mathbf{\Lambda}$ or the Eigenvector matrix \mathbf{X} . In conventional Eigenstate factorization, the matrices \mathbf{X} and $\mathbf{\Lambda}$ are computed from a given \mathbf{A} matrix. In other words, all time-domain coefficients are constrained and all Eigenstate parameters are free. A complementary problem is the inverse, in which the matrix \mathbf{A} is computed from \mathbf{X} and $\mathbf{\Lambda}$ via matrix multiplication. In this complementary problem, the time-domain coefficients are free and the Eigenstate parameters are fixed.

The problem of designing modes into a physical system differs from both the conventional Eigenstate factorization problem and its inverse (matrix multiplication), in that constraints exist on both sides of equation (56). If a subset of Eigenstates is assigned to the system, then some values of \mathbf{X} and $\mathbf{\Lambda}$ are constrained while others are free. Similarly, if some coefficients of the plant model cannot be altered by the design process or by the introduction of feedback control, then these elements of the \mathbf{A} matrix are constrained (whereas other elements of the \mathbf{A} matrix may

be free). Thus, a new generalization of the conventional Eigenstate factorization problem arises. I call this generalization the *Dual-Domain Eigenstate Factorization Problem*. The defining characteristic of this problem is that it bridges the conventional and inverse Eigenstate factorization problems, as illustrated in Figure 22.

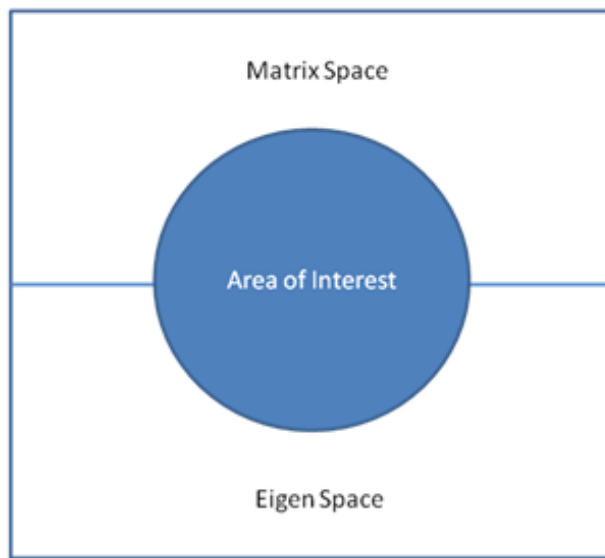


Figure 22: Region of Interest

1. Mathematical Description of Dual-Domain Eigenstate Factorization

For some systems, elements of \mathbf{A} (from equation 56) may be constrained by equality constraints such as kinematic relationships or fixed parameter values. These relationships can be captured by equality constraints of the form:

$$\mathbf{g}(\mathbf{A}) = \mathbf{0} \quad (57)$$

In this chapter, it is assumed that equality constraints are linear. The solution to equation (56) is not unique unless constraints are imposed upon the system Eigenvectors. In order to determine a unique solution to equation (56) constraints on Eigenvectors must be imposed with the form of:

$$\|\mathbf{X}\| = \boldsymbol{\gamma}, \quad (58)$$

where $\boldsymbol{\gamma}$ is a vector of magnitudes corresponding to each column of \mathbf{X} . In general, the magnitude of an Eigenvector is arbitrary [65]. As such, computational packages typically report Eigenvectors with unit length. In this example, $\boldsymbol{\gamma}$ would be a vector of ones.

Using the problem description given in equations (56) through (58) it is possible to go from the Eigen space to the Matrix space and vice versa in the conventional manner. Given N^2 Eigenvector elements (where N equals the number of states) and N Eigenvalue elements, a unique \mathbf{A} matrix and $\boldsymbol{\gamma}$ vector can be determined (in the absence of equality constraints). Also if N^2 \mathbf{A} element values and N $\boldsymbol{\gamma}$ values are specified then the Eigenvectors and Eigenvalues can be fully determined. The total number of Eigenstate equations in the Dual-Domain Eigenstate Factorization problem (without considering Equality constraints) is consequently N^2+N . Of these equations, N^2 result from (56), since an equation can be written for each element of \mathbf{A} . Another N equations result from (58), corresponding to each Eigenvector.

In order to solve Dual-Domain Eigenstate Factorization a mixing of specifications from both the Matrix space and Eigen space must be possible. That is, elements of the \mathbf{A} matrix and $\boldsymbol{\gamma}$ vector along with Eigenvector and Eigenvalue elements must be specifiable. By doing this constraints such as those observed in equation (57) can be preserved and any unspecified \mathbf{A} , \mathbf{X} , $\boldsymbol{\gamma}$, and $\boldsymbol{\Lambda}$ values can be solved for directly.

2. Dual-Domain Eigenstate Factorization for LTI System with Linear Feedback Control

The state space representation of a system, described by equation (56), can be constructed by both morphological and feedback parameters of a system. Morphological parameters are all the physical parameters of the system including material properties and system geometries. This section will demonstrate the impact of both morphological and control parameters on the state-space representation of an LTI system.

Let us first assume that the dynamics of a LTI system can be re-written as

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u}, \quad (59)$$

where \mathbf{A} and \mathbf{B} are time-invariant matrices of physical parameters, and \mathbf{u} is an actuation vector.

If it is then assumed that the systems of interest have full-state, linear feedback of the form:

$$\mathbf{u} = -\mathbf{K}\mathbf{z}, \quad (60)$$

where \mathbf{K} is a constant gain matrix. Then, the Eigenstates of the closed loop system are the Eigenstates of the matrix $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{BK}. \quad (61)$$

Consequently, $\tilde{\mathbf{A}}$ is comprised of feedback parameters in the matrix \mathbf{K} and physical parameters in \mathbf{A} and \mathbf{B} . It is only by utilizing both feedback and control parameters that the full search space of possible solutions can be reached. For example, the introduction of feedback parameters may reduce the number of equality constraints on the elements of $\tilde{\mathbf{A}}$, allowing for more Eigenstate elements to be directly specified.

For this chapter it is assumed that the number of states, and thus Eigenstates, is assumed to be even. Consequently, each mode has a complex conjugate partner, and no Eigenvalues must be made purely real to compensate for an odd number of modes. Odd numbers of states (like inequality constraints) will be addressed in Chapter VI.

For this chapter control gains and physical parameters are not specifically determined. Rather linear combinations of feedback and control gains are solved for as elements of $\tilde{\mathbf{A}}$. It is up to the designer to determine the final parameter selection by decomposing the elements of $\tilde{\mathbf{A}}$ appropriately. By solving for $\tilde{\mathbf{A}}$ parameters rather than physical and control parameters the search space effectively collapses. This may be an advantage over previously proposed co-design

strategies [11], by simplifying the Eigenstate equations and making a solution more mathematically tractable.

When one or more modes is specified for an LTI system with feedback control, the Dual-Domain Eigenvalue Factorization problem takes the following form

$$\begin{aligned}\tilde{\mathbf{A}} &= \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1} \\ \mathbf{g}(\tilde{\mathbf{A}}) &= \mathbf{0} \\ \|\mathbf{X}\| &= \gamma.\end{aligned}\tag{62}$$

III. Proposed Algorithm

This section presents a description of an algorithm designed to solve the Dual-Domain Eigenvalue Factorization problem using co-design of the physical plant and feedback control law for an LTI system. The guiding principle for this algorithm is to match the number of equations to the number of unknowns. Once a complete set of equations is identified, these nonlinear algebraic equations can be solved using an iterative method (e.g., Newton-Raphson). There are four steps to this algorithm. First, all equations of interest must be compiled. Second, all cases of direct assignment must be eliminated. Third, the rank of the system must be checked and augmented as necessary. Finally, numerical methods can be used to determine a unique solution. Each one of these steps is described in detail in this section.

1. Compile all Equations

Dual-Domain Eigenstate Factorization will be phrased as a nonlinear algebraic equation of the following form:

$$\mathbf{h}(\mathbf{y}) = \mathbf{0}. \quad (63)$$

It is first important to identify all of the parameters involved in the solution of equation (63). In order to identify the parameters in the \mathbf{y} vector of equation (63), let us first partitioned the matrix $\tilde{\mathbf{A}}$ on a row by row basis as

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_N \end{bmatrix}. \quad (64)$$

Given that each row \mathbf{a}_i is a \mathcal{R}^N vector, there are N^2 parameter values based on the systems state-space definition.

If we similarly partition the columns of the Eigenvector matrix \mathbf{X} as

$$\mathbf{X} = [\mathbf{x}_1 \quad \dots \quad \mathbf{x}_N], \quad (65)$$

and separate the complex conjugate Eigenvector pairs as

$$\mathbf{X} = [\mathbf{x}_{real,1} + j\mathbf{x}_{imag,1} \quad \dots \quad \mathbf{x}_{real,N/2} + j\mathbf{x}_{imag,N/2} \mid \mathbf{x}_{real,1} - j\mathbf{x}_{imag,1} \quad \dots \quad \mathbf{x}_{real,N/2} - j\mathbf{x}_{imag,N/2}], \quad (66)$$

where $\mathbf{x}_{real,i}$ and $\mathbf{x}_{imag,i}$ are \mathcal{R}^N vectors, then there is an additional N^2 parameters based on the Eigenvector specifications.

An additional N parameters are related to the Eigenvector magnitudes shown in equation (58).

Finally, if there exists N Eigenvalues for a system which are fully described by

$$\Lambda = \text{diag} \left(\begin{bmatrix} \lambda_{real} + j \lambda_{imag} \\ \lambda_{real} - j \lambda_{imag} \end{bmatrix} \right), \quad (67)$$

where λ_{real} and λ_{imag} are $\mathcal{R}^{N/2}$ vectors, then N more parameters must be considered.

In total there are $2N^2 + 2N$ total parameters in the \mathbf{y} vector with \mathbf{y} having the form of

$$\mathbf{y} = [\lambda_{real} \quad \lambda_{imag} \quad \mathbf{x}_{real,1} \quad \dots \quad \mathbf{x}_{real,N/2} \quad \mathbf{x}_{imag,1} \quad \dots \quad \mathbf{x}_{imag,N/2} \quad \gamma \quad \mathbf{a}_1^T \quad \dots \quad \mathbf{a}_N^T]^T. \quad (68)$$

After identifying all of the variables of interest, equation (63) can be separated into distinct sets of equations. Namely, those pertaining to the Dual-domain problem ($\mathbf{h}_{DD}(\mathbf{y})$), those pertaining to $\tilde{\mathbf{A}}$ ($\mathbf{h}_{\tilde{\mathbf{A}}}(\tilde{\mathbf{A}})$), and those pertaining to Eigenvectors ($\mathbf{h}_{\mathbf{X}}(\mathbf{X}), \mathbf{h}_{\gamma}(\mathbf{X}, \gamma)$) and Eigenvalues ($\mathbf{h}_{\Lambda}(\Lambda)$).

$$\mathbf{h}(\mathbf{y}) = [\mathbf{h}_{DD}(\mathbf{y}) \quad \mathbf{h}_{\tilde{\mathbf{A}}}(\tilde{\mathbf{A}}) \quad \mathbf{h}_{\Lambda}(\Lambda) \quad \mathbf{h}_{\mathbf{X}}(\mathbf{X}) \quad \mathbf{h}_{\gamma}(\mathbf{X}, \gamma)]^T \quad (69)$$

The Dual-Domain problem statements are of the form

$$\mathbf{h}_{DD}(\mathbf{y}) = [\mathbf{h}_{dd,1} \quad \dots \quad \mathbf{h}_{dd,N}]^T = \mathbf{0}, \quad (70)$$

where

$$\mathbf{h}_{dd,n} = (\tilde{\mathbf{A}} - \lambda_n \mathbf{I}) \mathbf{X}_n. \quad (71)$$

The matrix $\tilde{\mathbf{A}}$ equations are of the form

$$\mathbf{h}_{\tilde{\mathbf{A}}}(\tilde{\mathbf{A}}) = \mathbf{0}, \quad (72)$$

and represent equality constraints present in the $\tilde{\mathbf{A}}$ matrix. In some cases, these equations are simply definitions setting the particular time-domain coefficient to a constant. Directly defined variables are flagged in a binary vector \mathbf{b}_{direct} of length $2N^2+2N$, whose elements correspond to elements of \mathbf{y} . In other words,

$$\mathbf{b}_{direct,i} = \begin{cases} 1 & \text{if } \mathbf{y}_i \text{ is directly defined} \\ 0 & \text{otherwise} \end{cases}. \quad (73)$$

The form of the nonlinear algebraic Eigenvalue equations is that of

$$\mathbf{h}_{\Lambda}(\Lambda) = \mathbf{0}. \quad (74)$$

For this algorithm, it is desired that the system is stable. In order to ensure stability all Eigenvalues are fully defined to specified values designated by $\overline{\lambda_{real}}$ and $\overline{\lambda_{imag}}$.

$$\begin{aligned}\lambda_{real} &= \overline{\lambda_{real}} \\ \lambda_{imag} &= \overline{\lambda_{imag}}\end{aligned}\tag{75}$$

According, the values of \mathbf{b}_{direct} corresponding to λ_{real} and λ_{imag} are also set to 1.

The Eigenvector equations must also be considered. In general these equations are nonlinear algebraic of the form

$$\mathbf{h}_X(\mathbf{X}) = \mathbf{0}.\tag{76}$$

For this application, it is assumed that a full set of desired Eigenvectors \mathbf{X} has been specified (as is demonstrated in chapter IV). However, it is also assumed that these specified Eigenvectors are rank ordered. For example, in Fourier-Eigenstate specification, each higher harmonic is progressively less important to the overall mode shape. Assuming the Eigenvectors in \mathbf{X} are listed in this rank order (with the highest priority in column 1 and the lowest priority in column $N/2$) and that there is no priority difference among the elements of the Eigenvector, as many terms as possible are specified.

If a new variable is defined to represent the Eigenvector specifications as

$$x_{cmp,n} = [x_{real,1}(n) \quad x_{imag,1}(n) \quad \dots \quad x_{real,N/2}(n) \quad x_{imag,N/2}(n)]^T \quad (77)$$

and assume that L constraints have been imposed by Eigenvalue and time-domain specifications, then an algorithm for Eigenvector specification can be implemented as follows.

```

for all i
for all n
     $x_{cmp,n}(i) = \overline{x_{cmp,n}(i)}$ 
    m=m+1;
    if  $m \geq N^2 - L$  then break
end for
end for

```

Figure 23: Pseudo-code for Dual-Domain Eigenstate Factorization

Note: $\overline{x_{cmp,n}(i)}$ corresponds to the Eigenvector elements specified in the Fourier-Eigenstate specification method of chapter IV.

For each value assigned in this pseudo-code, the corresponding elements of \mathbf{b}_{direct} are set as true. In effect, this pseudo-code steps through each element in each Eigenvector and then moves onto the next Eigenvector, until all possible assignments are made.

The last set of equations to consider pertains to the Eigenvector magnitudes. These equations have the form of

$$\mathbf{h}_\gamma(\mathbf{X}, \gamma) = \mathbf{0}. \quad (78)$$

In this work it is assumed that the two-norm is used to determine the magnitude of each Eigenvector. Thus,

$$h_{\gamma,n} = x_{cmp,n}^T x_{cmp,n} - \gamma_n^2 = 0 \quad (79)$$

2. Eliminate Cases of Direct Assignment

Now that the full system of nonlinear algebraic equations ($\mathbf{h}(\mathbf{y})$) has been constructed, the next step is to remove all cases of direct assignment. After removing all directly assigned variable (indicated by \mathbf{b}_{direct} values equal to 1), the set of remaining solution variables will be referred to as $\tilde{\mathbf{y}}$.

$$\tilde{\mathbf{y}} = \{\mathbf{y}_i | \mathbf{b}_{direct,i} = 0\} \quad (80)$$

Similarly, the set of nonlinear equations is reduced in size, eliminating all cases of direct assignment.

$$\tilde{\mathbf{h}}(\mathbf{y}) = \{\mathbf{h}_i(\mathbf{y}) | \mathbf{b}_{direct,i} = 0\} \quad (81)$$

All directly assigned parameters can be treated as equation parameters, such that the new set of nonlinear algebraic equations is now

$$\tilde{\mathbf{h}}(\tilde{\mathbf{y}}) = \mathbf{0}. \quad (82)$$

3. Check System Rank and Augment the System as Necessary

The previous steps guarantee that the number of equations and the number of unknowns are matched. However, it is possible that some of these nonlinear equations become linearly dependent (at least in a localized region). Consequently, rank of the locally linearized system should be checked to confirm that a sufficient number of equations are present to solve for all unknowns. This check can be performed by obtaining the Jacobian matrix \mathbf{H} .

$$\mathbf{H}_{i,j} = \frac{d\tilde{h}_i(\tilde{\mathbf{y}})}{d\tilde{y}_j} \quad (83)$$

If the rank of this matrix is full, then a solution is possible. If the rank is not full, then the system can be augmented with an additional equation. Simply, specify another Eigenvector element to a desired value. Then, recheck the rank and repeat the process until the rank of the Jacobian is full. During this process, the algorithm may assign extra Eigenvector elements until local linear independence is achieved.

4. Numerical Analysis

A number of numerical methods can be used to solve the proposed algorithms set of non-linear algebraic equations seen in equation (82). The non-linear algebraic set of equations can be solved

by algorithms including but not limited to Richardson's method and Newton's method. Additionally, many conventional engineering mathematics solvers have explicit packages for solving nonlinear equations (for instance `fsolve` in Matlab), making implementation of the proposed algorithm quite easy.

The Jacobian for the set of equations, seen in equation (83), can be computed directly through linearization or indirectly through parameter perturbation. By checking the Jacobian matrix the convergence upon a solution can be checked by comparing the maximum Eigenvalue magnitudes and ensuring they are less than one. If convergence criteria is broken at any iteration, then alternative numerical methods can be used, or different iteration criteria can be implemented. For this chapter we will use Richardson's method with iterative updating until convergence on a solution has occurred. We will assume convergence is given when the magnitudes of the tunable parameters alter by less than $1e-6$ between successive iterations. This approach has the form of

$$\tilde{h}(\tilde{y}) = \tilde{h}(\tilde{y}^*) + H(\tilde{y} - \tilde{y}^*) + HOT = 0 \quad (84)$$

$$\tilde{y} = \tilde{y}^* - H^{-1}\tilde{h}(\tilde{y}^*) \quad (85)$$

where \tilde{y}^* is the solution at the previous iteration, and \tilde{y} is the solution at the new iteration. The process is repeated until convergence occurs.

IV. Case Study

A spring-mass-damper system will be used as a model to demonstrate the proposed algorithm to solving Dual-Domain Eigenstate Factorization. The model will include three masses attached by tunable linear springs and dampers. Forces will only be applied to the first and third mass, allowing the second mass to have no control input. It is assumed that the masses slide on a frictionless surface. The model system can be seen in Figure 24. The full state-space representation of the model system can be found in Appendix E.

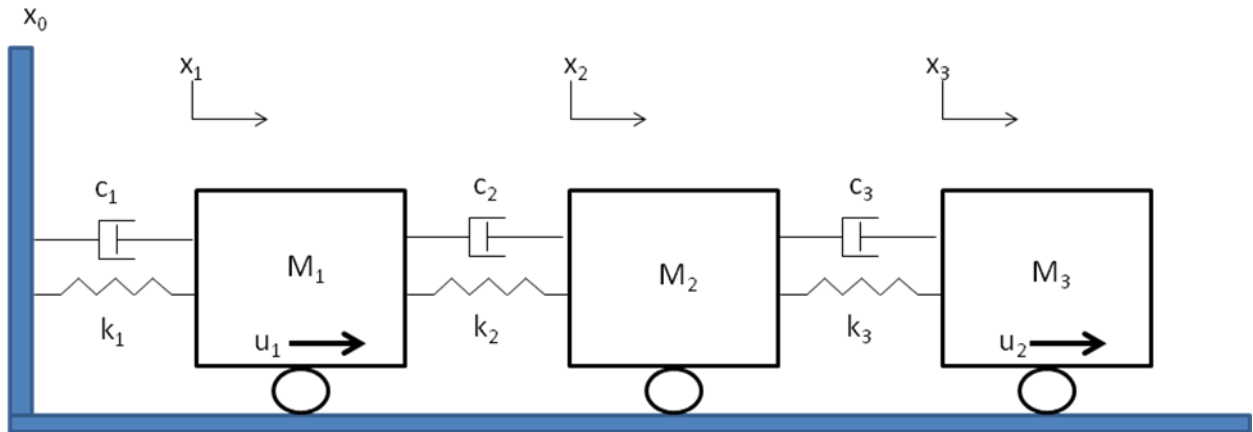


Figure 24: Mass-Spring-Damper Configuration; with tunable passive dynamical parameters (mass M_i , linear damper c_i , and linear spring k_i) and feedback-controlled forces acting on the center of mass of mass one and three (u_i). For this model it is assumed that $x_3 > x_2 > x_1$ and $\dot{x}_3 > \dot{x}_2 > \dot{x}_1$

1. Specified System Response

A set of desired Eigenstates for the system were selected in order to determine if the proposed Dual-Domain Eigenstate Factorization algorithm could converge on $\tilde{\mathbf{A}}$ matrices that have Eigenstates representative of those desired. The Eigenvectors were selected in way that ensures that the kinematic relationships between position and velocity were preserved, which is detailed

further in [19]. The Eigenvalues and Eigenvectors selected were representative of a traveling wave of relative state displacements, representative of those observed in Figure 25. The damping coefficient for the selected Eigenvalues was chosen to be 1×10^{-4} . The method in which the Eigenstates were selected in order to achieve the desired traveling wave pattern of states can be found was discussed in Chapter IV.

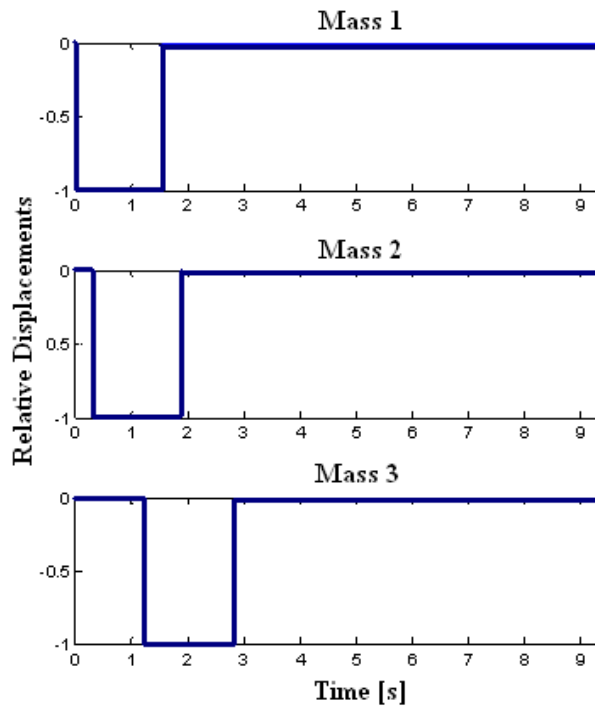


Figure 25: Relative Displacements of the Masses with Respect to Time [normalized by .1 m]; given the specified Eigenstates.

As a basis for comparison the proposed solution to Dual-Domain Eigenstate Factorization was compared to the system response with the given Eigenstates. These Eigenvalues and Eigenvectors, along with the assumed damping, were mapped into a desired $\tilde{\mathbf{A}}$ matrix for the system using equation (42). After applying Bode analysis to the desired system, and determining the appropriate periodic forcing function using equation (55), the system responded as can be

seen in Figure 26. Given that the desired system Eigenstates and forcing function were determined based on Fourier-Eigenstate Specification a phased oscillation of system states should be observed in Figure 26.

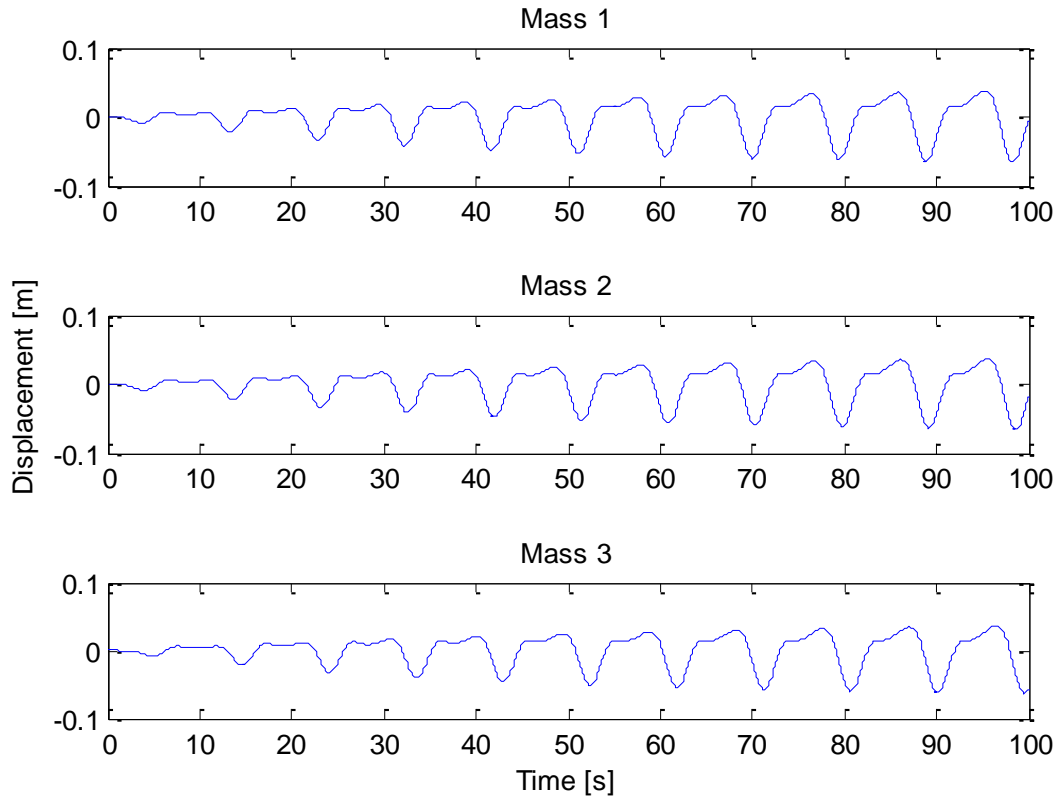


Figure 26: Desired Response

The desired response is that of a traveling wave, where the period of deactivation is longer than the period of activation and the displacement of links occurs in a phased manner from the first mass to the last. The phasing between mass 1 and 2 is equal to 0.1π , and between mass 1 and 3 is equal to 0.4π . The total period of each oscillation is equal to 3π .

2. Simulation

In order to evaluate if the proposed solution methodology can achieve the desired Eigenstates a simple simulation has been performed. A primary goal of the simulation was to assess how much the approximation of the original gait pattern deteriorates at each step of the algorithm. For the simulation the under-actuated three-mass system had the proposed Dual-Domain Eigenstate Factorization method applied to match the system Eigenstates to the desired Eigenstates. The solution output by the methodology was then converted into a set of physical and control parameters needed to create the solution. The accuracy of the proposed methodology solution in achieving the desired Eigenstates and the feasibility of the proposed method's solution were investigated.

In order to apply the proposed methodology all equality constraints on the systems $\tilde{\mathbf{A}}$ matrix were identified. It was then specified that the Eigenvalues of the system match those desired. Given all of the Eigenvalue and equality constraints there were 16 Eigenvector elements which could be directly specified until the number of linearly independent equations equaled the number of unknowns (determined by the mathematical reasoning laid out in Section II). Since the desired Eigenvectors were selected in a manner in which the kinematic relationships between position and velocity were preserved only 18 Eigenvector elements were needed for full specifications. However, since only 16 Eigenstate elements were available from $\mathbf{x}_{real,1:N/2}$ and $\mathbf{x}_{imag,1:N/2}$ it was decided not to specify the two highest frequency Eigenvector components associated with the third mode. This selection assumes that the highest frequency oscillations contribute to the system response the least amount, as is evident in Fourier series decomposition.

3. Eigenstate Matching

After applying the proposed Dual-Domain Eigenstate Factorization method to the model system it was clear that the desired Eigenstates could be representatively achieved. This is verified by referring to Figure 27, in which the proposed methodology's responses to the prescribed forcing function are plotted against the desired responses. In Figure 27, the desired systems response to the forcing function is displayed in blue. The response achieved by the Dual-Domain Eigenstate factorization method is shown in red. Each masses displacement (meters) with respect to time (seconds) indicates the presence of a traveling wave matching.

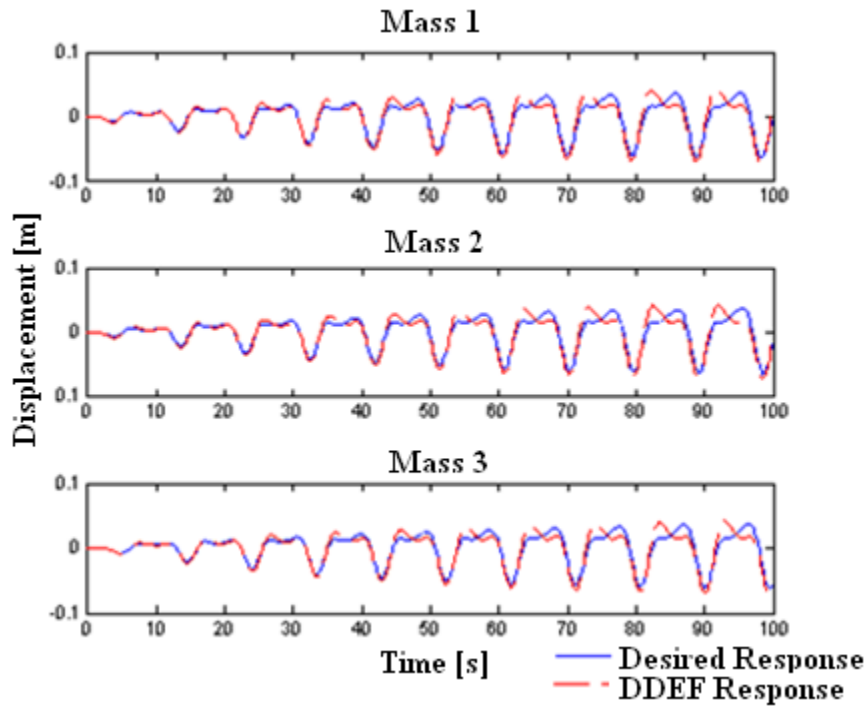


Figure 27: Method Responses

Figure 27 indicates that the proposed methodology is capable of converging on a system configuration in which its Eigenstates are representative of the desired Eigenstates, and phased

periodic oscillations of system states can be achieved. The system response does not match the desired response exactly, due to all Eigenvectors not being completely specified, but it is clear that the frequency, phasing, amplitudes, and general shape of the desired response are recreated by the proposed Dual-Domain Eigenstate Factorization algorithm.

4. System Feasibility

In order to determine the feasibility of the proposed method's $\tilde{\mathbf{A}}$ matrix, it may be necessary to also consider inequality constraints. The $\tilde{\mathbf{A}}$ matrix was compared to the state-space representation of the system found in Appendix E to determine if it could be created with all masses, spring constants, and damping constant values being positive. This system can be achieved with the values seen in Table I, and thus is considered feasible to create. Although there may be many parameter combinations that can create the Dual-Domain Eigenstate Factorization method's solution, only one valid solution according to parameter constraints is needed to verify that it is in fact feasible for this test case.

Table I: Physical and Control Parameter Values Matching Direct Solution $\tilde{\mathbf{A}}$

	1	2	3
Mass	0.2529	0.1488	0.4486
Spring Constant	0.2619	0.1506	0.0871
Damping Constant	0.6679	0.9442	0.315

Control Gains	1	2	3	4	5	6
	-2.62	-0.24	0.08	7.10	-8.83	0.32
Control Gains	7	8	9	10	11	12
	4.58	14.08	3.27	-48.48	41.54	7.23

V. Discussion

The proposed solution methodology to Dual-Domain Eigenstate Factorization has many advantages to its use. When implementing the proposed methodology it is possible to specify specific Eigenvector elements over others. That is, it may be desired to match all low frequency Eigenvector elements and less important to match high frequency Eigenvector elements. The proposed solution methodology is also mathematically tractable.

The proposed methodology also has some clear limitations. Depending on the numerical method used to solve the set of non-linear equations it may be difficult to converge on a solution to the set of non-linear state equations. Additionally, inequality constraints are not considered in the proposed methodology. As was demonstrated by the example given in this chapter, it is sometimes possible to use the proposed methodology to match desired Eigenstates while generating physically realizable systems. However, if the values output by the proposed Dual-Domain Eigenstate Factorization method are not feasible (e.g. mass must be positive) then a variant to the proposed methodology may be needed as an adequate substitute.

A slight modification can be made to the proposed solution methodology which converts it into an optimization problem can be made, that allows for inequality constraints to be present. Once the proposed solution approach is written as an optimization problem then inequality constraints can easily be added with the form of:

$$\mathbf{f}(\tilde{\mathbf{A}}) < 0 \tag{86}$$

A cost function for the desired Eigenvectors must then be specified. The cost function can have the form of the Frobenius norm between the desired Eigenvectors and the specified Eigenvectors where:

$$J = \|\tilde{\mathbf{X}} - \mathbf{X}\|_F \quad (87)$$

The entire form of the Optimized solution, including inequality constraints, is:

$$\begin{aligned} J &= \|\tilde{\mathbf{X}} - \mathbf{X}\|_F \quad (88) \\ \text{s.t.} \quad & \mathbf{f}(\tilde{\mathbf{A}}) < 0 \\ & [\mathbf{h}_{DD}(\boldsymbol{\gamma}) \quad \mathbf{h}_{\tilde{\Lambda}}(\tilde{\mathbf{A}}) \quad \mathbf{h}_{\Lambda}(\Lambda) \quad \mathbf{h}_{\boldsymbol{\gamma}}(\mathbf{X}, \boldsymbol{\gamma})] = \mathbf{0} \end{aligned}$$

The optimization problem given in (88) is a non-convex quadratically constrained quadratic problem. There is no closed form solution to this optimization problem and a global optimum cannot be guaranteed. There are techniques to relax the problem into a convex form [60]. However, these techniques still do not guarantee a global optimal can be found.

A complete description of solving the Dual-Domain Eigenstate Specification problem with an optimization method will be discussed in the next chapter in full detail. Details of how the optimization variant may allow for a loosening of assumptions made during Dual-Domain Eigenstate Factorization (e.g. set all Eigenvalues, number of states is even) will also be explained in the next chapter.

VI. Conclusion

A method for matching system Eigenstates to desired Eigenstates was introduced, with the goal of tuning physical and control parameters of a system. The method was unique in that it allows for the tuning of both physical and control parameters, as well as Eigenstate parameters. This problem was referred to as Dual-Domain Eigenstate Factorization. A proposed solution approach was offered and demonstrated with a model system. The proposed approach matches the number of linearly independent equations to the number of unknowns to result in a unique system configuration that matches as many specified Eigenstate elements as possible.

Chapter VI: Mitigating Unwanted Dynamics While Embedding Eigenstates into a Feedback System in the Presence of Inequality Constraints

Executive Summary- An alternative form to Dual-Domain Eigenstate Factorization can be created in which not all Eigenvalues are set. Rather, Eigenvalues pertaining to modes which are desired to be mitigated, can be made large and negative. This method of Eigenstate assignment is a significant variation on Dual-Domain Eigenstate Factorization that addresses several limitations of the method described in the previous chapter: allowing for inequality constraints to be present, allowing the number of States/Eigenstates to be odd, and allowing for not all Eigenvalues to be specified. The solution to this variant of Dual-Domain Eigenstate Factorization is a non-convex quadratically constrained quadratic optimization problem. This chapter will introduce the optimization variant and then use a model system (a three-link pendulum) to investigate alternative means of applying solution algorithms, as well as alternative optimization methods for applying the variant of Dual-Domain Eigenstate Factorization to specify Eigenstates for mechanical systems.

I. Introduction

New methods for designing undulating robots and mechanisms are needed that enable dynamic modes to be defined precisely using a minimum of actuation. One way to create complex motion with minimal actuation may be to exploit Eigenstates. Eigenstates, comprised of Eigenvector and Eigenvalue pairs, can be the basis for repeated movement patterns that create robotic locomotion. Eigenstate-based movement patterns are dynamic, inherently rejecting certain impulsive disturbances, and hence are appropriate for robotic platforms [6]. It may be possible to specify Eigenstates by leveraging both control and physical parameters in order to access a large design

space, even for under-actuated systems. In this sense, existing approaches for specifying Eigenstates, including methods that solely modify passive dynamics (through changes to mechanical design, for instance) or that rely solely on active feedback control can be bridged.

Passive parameter tuning has been implemented by many engineering disciplines. Modifying passive dynamics is heavily studied in applications such as creating earthquake resistant structures that dampen out vibrations, or harvesting energy from vibration environments [66, 67]. Tuning passive physical parameters of a system, such as element masses or stiffnesses, can also allow alteration of mode shape, resonant frequency, and decay rate [29]. However, all physical parameters are subject to inequality constraints that are either imposed by fundamental physics or by design requirements. For example, lengths are constrained by the physical volume in which the system must fit and mass must be positive. These constraints severely limit the mode shapes possible for a system. As we consider in this chapter, the addition of active control could extend the ability to specify system responses, or Eigenstates.

As an alternative to tuning passive system parameters, feedback control methods have also been defined to modify the Eigenstates of a system. It has long been known that Eigenvalues can be placed for controllable linear state-space systems. This ability has been leveraged to modify system responses using methods including but not limited to LQR and direct pole placement [30, 68-70]. Recently, control over both Eigenvectors and Eigenvalues has been considered. This area of controls is sometimes referred to as modal control or Eigenstructure assignment [28, 31, 71-73]. Modal control offers the ability to fully specify a system's mode shapes by pre-determining Eigenvalues and Eigenvectors [74, 75]. Existing Eigenstructure assignment methods using

control parameter selection has been done for fixed plant designs; tuning passive dynamic and active feedback parameters together for the purposes of Eigenstructure assignment remains an open research topic.

New methods for Eigenstructure assignment are particularly motivated by applications to the control of under-actuated systems, in which a relatively small number of actuators must be used to control a higher number of degrees-of-freedom. Typically, controllers for under-actuated systems rely upon strong inertial coupling between actuated and un-actuated system states [7]. In general, under-actuated systems are controlled by performing partial feedback linearization and then placing Eigenvalues of the system [8]. Although under-actuated systems have been extensively studied [8-10], existing methods do not allow full Eigenstate specification (including the assignment of Eigenvectors).

The emerging discipline of co-design offers the potential to address the limitations of existing passive design, Eigen-structure placement, and under-actuated control design methods applied to mode shaping problems for under-actuated systems. Co-design means the simultaneous design of a systems physical parameters as well as control parameters in order to achieve desired system goals [11, 12]. Co-design techniques have recently been developed to select control and physical architectures that reduce energy inputs or create minimal changes to physical structures while accomplishing previously impossible tasks for a given system [13, 14]. Although co-design has not yet been explored in detail for modal control or under-actuated systems, it has potential to expand in this area because the framework of co-design allows for altered cost functions to be solved based on optimization methods.

This chapter introduces a new generalization of co-design tuned for creating repeated motion patterns for under-actuated systems. The problem introduced in this chapter is a variant of that previously described in Chapter V (Dual-Domain Eigenstate Factorization). Dual-Domain Eigenstate Factorization allows for the alteration of physical and control parameters with the aim of creating desired Eigenstructures, even for under-actuated systems. Combining together tools for tuning passive dynamic parameters and feedback control parameters to achieve desired Eigenstates is a compelling area for new research, especially if systems can be under-actuated (see Figure 28).

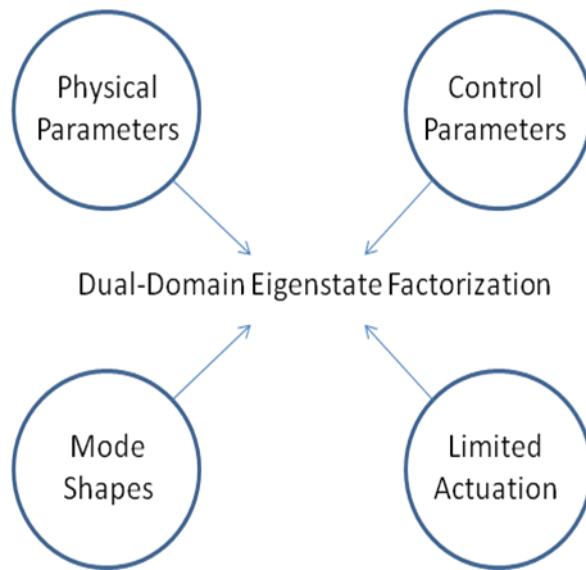


Figure 28: Research Topics Covered by Dual-Domain Eigenstate Factorization

The remainder of the chapter is organized as follows. The next section of the chapter describes a pair of solution approaches to an optimization based variant of the Dual-Domain Eigenstate Factorization problem. A model three-link pendulum system is then introduced that is used in

simulation to compare the two solution approaches. Section IV analyzes the results of those simulations. A brief discussion and conclusion follows.

II. Dual-Domain Eigenstate Factorization Optimization Variant

An optimization variant of Dual-Domain Eigenstate Factorization is used to solve the same MEDFRD problem described by Chapter III. The major difference between the optimization variant of Dual-Domain Eigenstate Factorization and Dual-Domain Eigenstate Factorization is the manner in which unwanted system dynamics are mitigated and the inclusion of inequality constraints. If only a subset of Eigenstates are desired to be specified then the unwanted Eigenstates can be mitigated by making their Eigenvalues large and negative. This is different than the full Eigenvalue specification methods given by Dual-Domain Eigenstate Factorization, allowing for partial Eigenvalue specifications and an odd number of states. Additionally, because Eigenstate specification is phrased as an optimization problem the inclusion of inequality constraints on system parameters is allowed.

When some Eigenstates are unspecified by the designer, it is important that these Eigenstates not have slow or unstable dynamics, which might swamp specified Eigenstates. One way to achieve this is to ensure that Eigenvalues of unspecified dynamics are as small as possible. In this sense a designer might constrain the worst case Eigenvalue λ_k to be less than ϕ .

$$\max_{k \in K} (\text{real}(\lambda_k)) < \phi \quad (89)$$

Here the set K includes only the unspecified Eigenstates.

The resulting constraint is non-convex. This variant formulation of Dual-Domain Eigenstate Factorization can be written as follows.

$$\begin{aligned}
& \min J_1 && (90) \\
\text{s.t.} & \max_{k \in K} (\text{real}(\lambda_k)) < \phi \\
& \mathbf{f}(\tilde{\mathbf{A}}) < 0 \\
& \mathbf{g}(\tilde{\mathbf{A}}) = 0 \\
& J_1 = \|\tilde{\mathbf{A}}\tilde{\mathbf{X}} - \tilde{\mathbf{X}}\tilde{\mathbf{\Lambda}}\|_F
\end{aligned}$$

In this problem, the free parameters are the elements of $\tilde{\mathbf{A}}$; the matrices $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{\Lambda}}$ are potentially rectangular constant matrices that contain the Eigenstates specified by a design engineer. Note that the Eigenvalue and Eigenvector matrices are no longer guaranteed to be square, because only a subset of modes are desired to be specified. It may be possible to know how many modes are feasibly specifiable. A description of determining the number of specifiable Eigenstates can be found in Appendix F, along with an example in Appendix G.

Because this optimization problem is non-convex, it is not immediately clear that a robust solution exists. As such, I will investigate the quality of solutions to the optimization problem in a subsequent section, by application to a model dynamic system. Furthermore, to provide a basis for comparison, a second optimization problem, an alternative version of (90) that relaxes the non-convex constraint on the unspecified Eigenvalues is also formulated.

The alternative problem formulation shifts the non-convexity from the set of constraints into the cost function. The modified cost function is the following.

$$J_2 = \|\tilde{\mathbf{A}}\mathbf{X} - \mathbf{X}\mathbf{\Lambda}\|_F \quad (91)$$

This cost function differs from that of (90) in that all Eigenstates are included in the cost function, both specified and unspecified. The unspecified Eigenstates and the elements of the unspecified Eigenvectors are left as free parameters. Thus, the cost function represents a quadratic relationship among the free parameters. In concept, this cost function allows a computer to specify the Eigenstates that are not explicitly defined by the design engineer. The non-convex Eigenvalue constraint of (89) can thereby be replaced with a linear constraint. The resulting is the following problem formulation.

$$\begin{aligned} & \min J_2 & (92) \\ \text{s.t. } & \max_{k \in K} (\text{real}(\lambda_k)) < \phi \\ & \mathbf{f}(\tilde{\mathbf{A}}) < 0 \\ & \mathbf{g}(\tilde{\mathbf{A}}) = 0 \\ & J_2 = \|\tilde{\mathbf{A}}\mathbf{X} - \mathbf{X}\mathbf{\Lambda}\|_F \end{aligned}$$

The unspecified Eigenstate constraint is now a linear (rather than polynomial) constraint, because the values λ_k are themselves tunable parameters (along the diagonal of $\mathbf{\Lambda}$) and not a nonlinear function of tunable parameters (e.g., the Eigenvalues of $\tilde{\mathbf{A}}$ as in the previous formulation). However, the actual Eigenvalues of $\tilde{\mathbf{A}}$ are no longer strictly constrained; rather they are simply optimized to approximate the tuned Eigenvalues λ_k (which are constrained).

It should be noted that the optimization does not enforce the constraint that the tunable Eigenvalues λ_k , if complex, must have a complex conjugate pair. Instead, the tunable Eigenvalues are simply required to be real.

III. Simulation Based Performance Assessment

This section applies the two problem formulations developed in the previous section to a model dynamic system (a three-link pendulum). Through this application how reliably these non-convex programs can be solved will be investigated, while also assessing the pros and cons of the two formulations.

In this comparison study, the first minimization problem, described by equation (90), will be labeled the *strict formulation*; the second minimization problem, described by equation (92), will be labeled the *relaxed formulation*. The second formulation is relaxed in the sense that the real part of the Eigenvalues of $\tilde{\mathbf{A}}$ do not strictly lie below ϕ . Because the relaxed formulation allows for a larger space of possible solutions, it is anticipated that the relaxed formulation will offer solutions with better matching to specified Eigenstates than the strict formulation.

In this comparison study, two different solution methodologies will also be tested. Specifically, I will compute solutions to the strict and relaxed formulations using both (1) a gradient-descent solver and (2) pattern search. It is anticipated that pattern search will provide better matching to specified Eigenstates than gradient-based local solvers, but that gradient-descent will provide solutions in a more reliable, consistent manner.

1. Optimization Solvers

Two different types of optimization solvers were implemented to compute solutions for the strict and relaxed formulations of the optimization variant of Dual-Domain Eigenstate Factorization: pattern search [76] and gradient descent [77]. Pattern search is a global search strategy; the method explores the parameter space in a manner that allows identification of multiple local minima for non-convex optimization problems. By comparison, gradient descent is a local strategy; gradient-descent solvers search only the basin of attraction associated with a single local minimum.

Matlab solvers were used for both the gradient-based and pattern search tests. The gradient-based solver was implemented using `fmincon` in Matlab [78]. Matlab was also used to implement pattern search. Pattern search was selected as an alternative to other global solvers, such as genetic algorithms, because it is neither gradient-based nor randomized.

2. Model System

a. Three-Link Pendulum

The simulation case-study focused on designing physical and control parameters for a three-link pendulum with feedback-actuation at each joint. The pendulum was selected as a simple mechanical analog to a robotic system, an analog featuring predominantly linear dynamics. (It is assumed that the links do not rotate more than 15 degrees, such that model nonlinearities need not be considered.) An illustration of the three-link pendulum system can be seen in Figure 29.

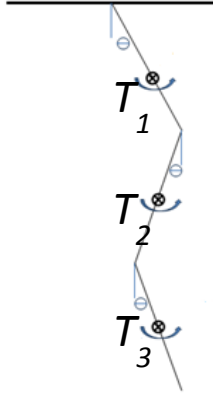


Figure 29: Three-Link Pendulum; with tunable passive dynamical parameters (length l_i , mass M_i , rotational spring constant k_i , and inertia I_i) and feedback-controlled joint torques (T_i).

For the three-link pendulum system $\tilde{\mathbf{A}} \in \mathbb{R}^{6 \times 6}$. Half of the rows of the $\tilde{\mathbf{A}}$ matrix are kinematic equations relating angle to angular speed; the other half of the rows are moment balances. The kinematic equations cannot be influenced through tuning of physical or feedback parameters; by contrast, parameter tuning is possible for each of the moment balance equations. This distinction is made clear by writing out the general form of the linearized moment balance and kinematic equations for each link. The kinematic equations are represented in (93) and do not include any physical or control parameters. Here θ_i represents the angle of i^{th} link of the pendulum chain relative to vertical.

$$\frac{d}{dt} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (93)$$

The moment balance equations represented by (94) depends on both physical parameters in the mass matrix \mathbf{M} and the state-forcing matrix \mathbf{D} , as well as control parameters in the torque vector \mathbf{u} .

$$\mathbf{M} \frac{d}{dt} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \mathbf{D} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} + \mathbf{u} \quad (94)$$

For this system, the parameter matrices (\mathbf{M} , \mathbf{D}) are constructed as follows. The physical parameters of the links include mass M_i , length l_i , rotational spring constant k_i , and inertia I_i .

$$\mathbf{M} = \begin{bmatrix} I_1 + (l_1/2)^2 M_1 + l_1^2 (M_2 + M_3) & M_2 (l_2/2) l_1 + M_3 l_1 l_2 & M_3 (l_3/2) l_1 \\ M_2 (l_2/2) l_1 + M_3 l_1 l_2 & I_2 + (l_2/2)^2 M_2 + l_2^2 M_3 & M_3 (l_3/2) l_2 \\ M_3 (l_3/2) l_1 & M_3 (l_3/2) l_2 & I_3 + M_3 (l_3/2)^2 \end{bmatrix} \quad (95)$$

$$\mathbf{D} = \begin{bmatrix} -M_1 g (l_1/2) - (M_2 + M_3) g l_1 - k_1 - k_2 & k_2 & 0 \\ k_2 & -M_2 g (l_2/2) - M_3 g l_2 - k_2 - k_3 & k_3 \\ 0 & k_3 & -M_3 g (l_3/2) - k_3 \end{bmatrix} \quad (96)$$

The vector \mathbf{u} represents the summation of torques acting on each link, computed in terms of a matrix of selected control gains \mathbf{K} and the states of the system \mathbf{z} .

$$\mathbf{u} = -\mathbf{K} \mathbf{z} \quad (97)$$

The states are assembled in the \mathbf{z} vector as follows.

$$\mathbf{z} = [\dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3 \quad \theta_1 \quad \theta_2 \quad \theta_3]^T \quad (98)$$

Using a state vector of this form, the dynamic equations (93) and (94) can be converted into a state-space representation. The corresponding $\tilde{\mathbf{A}}$ matrix that combines together physical and feedback control parameters is

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \mathbf{M}^{-1}\mathbf{D} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\mathbf{M}^{-1}\mathbf{K} \\ \mathbf{0} \end{bmatrix}. \quad (99)$$

The values of the first three rows can be set to any arbitrary value, by setting values of the \mathbf{M} , \mathbf{D} , and \mathbf{K} matrices appropriately. (In fact, redundant solutions are possible because more design parameters are available than independent coefficients in each of the first three rows of $\tilde{\mathbf{A}}$). The identity matrix \mathbf{I} is not tunable; hence there are no tunable parameters in the final three rows (e.g., the kinematic equations), a fact that must be considered carefully in specifying a desired Eigenstate for the system.

In order to simulate the design of an under-actuated system, it was assumed in the case study that the third actuator could not be controlled independently of the other two. More specifically, it was assumed that the third row of the \mathbf{K} matrix consisted of all zeros. An under-actuated system was chosen for the study, because conventional design methodologies (passive dynamic tuning, full-state modal control) do not apply to under-actuated systems.

b. Baseline Eigenstate Specification

To obtain a wave like progression of moving linkages, an oscillating mode shape for the three-link pendulum was selected. This oscillating mode shape was defined by an Eigenvalue, which determined the frequency of oscillation of the three links, and by an Eigenvector, which determined the amplitude of the oscillation of each link as well as the relative phasing of each link's swing. In this example it was chosen that the links oscillate at a frequency of $\pi/2$ rad/sec with a damping ratio of 0.063. The Eigenvalue for this case was $\lambda = -0.1 + \frac{\pi}{2}j$.

Because the kinematic equations of state contain no free parameters, it is possible to select arbitrary values for only three of the six elements of the target Eigenvector. Accordingly, only the elements of the Eigenvector associated with the angular position of each pendulum link were assigned; this specified portion of the target Eigenvector is \mathbf{x}_s .

$$\mathbf{x}_s = \begin{bmatrix} 1 \\ e^{\pi j/4} \\ e^{\pi j/2} \end{bmatrix} \quad (100)$$

In this vector, the third link angle lags the first by one-quarter of a cycle ($\frac{\pi}{2}$ rad), and the second link angle lags the first by only 1/8 cycle ($\frac{\pi}{4}$ rad).

Through partitioning (described in chapter III), the three additional elements of the Eigenvector, corresponding to angular velocities, are computed directly in terms of \mathbf{x}_s .

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{kd} \\ \mathbf{x}_s \end{bmatrix} = \begin{bmatrix} -.1 + \pi/2j \\ -1.18 + 1.04j \\ -\frac{\pi}{2} - .1j \\ 1 \\ (1+j)/\sqrt{2} \\ j \end{bmatrix} \quad (101)$$

To obtain oscillations, two Eigenstates must be specified together. In this case, the repeated Eigenvector and complementary (complex conjugate) Eigenvalue comprise the second target Eigenstate.

c. Baseline Constraints

Inequality constraints were introduced to ensure stability of unspecified Eigenstates and to require reasonable values for system physical parameters. Specifically, stability of unspecified modes was ensured by requiring all unspecified Eigenvalues to have a real part below $\phi = -0.4$. This requirement not only ensures stability; it also achieves fast decay of the unspecified Eigenmodes (because the real parts of the unspecified Eigenvalues are located at least four times farther from the imaginary axis than the specified Eigenvalues). Thus, the specified Eigenstates can be considered to be the dominant modes for the system.

Inequality constraints are also a key characteristic of the optimization variant of Dual-Domain Eigenstate Factorization. Two sets of inequality constraints on physical parameter values were considered. For a first set of constraints, labeled the *tight* constraints, upper and lower bounds for each parameter are summarized in Table II. For a second set of constraints, labeled the *loose* constraints, lower bounds are identical to those of the tight constraints, but upper bounds were not applied.

Table II. Tight Inequality Constraints on the $\tilde{\mathbf{A}}$ Matrix

Mass	Length	Inertia	Spring Constant	Control Gains
[0.01, 0.05]	[0.01,0.05]	[0.01,0.011]	[0.01,0.05]	[-0.1,0.1]

3. Monte Carlo Trials

Because the strict and relaxed problem formulations are non-convex, their solutions may depend on initial values. Local search algorithms (e.g., gradient descent) generally find a local optimum close to the initial value. Global search algorithms may locate local optima much farther from the initial value. To reduce sensitivity to initial value, a Monte Carlo procedure was used with randomly selected initial parameter values, drawn from the range defined by Table II. In all, 100 random trials were considered for each of eight configurations. The test matrix of eight configurations spans the two formulations (strict/relaxed), the two solvers (global/gradient-based), and the two parameter constraint configurations (Tight/Loose).

IV. Results

1. Representative Solution

Before compiling the results of the randomized Monte Carlo trials, it is instructive to first examine a representative solution in detail. In particular, a representative trial was considered in which the $\tilde{\mathbf{A}}$ matrix was computed using the strict formulation, the gradient-based solver, and the loose constraint configuration.

For the sixth-order pendulum system, an exact match to the desired Eigenstates is not generally possible unless all three joints are fully actuated. In the case study scenario, because independent actuation is applied only at the first two joints, the solution is necessarily approximate. The quality of this approximation can be assessed by comparing the optimized system Eigenstates to the target Eigenstates. Figure 30 illustrates the specified Eigenstates (red stars) next to the computed Eigenstates (blue circles) resulting from the solution of the strict formulation.

The process of matching Eigenstates involves both matching Eigenvalues and also matching Eigenvectors. These two steps are illustrated in Figure 30. Figure 30A, which plots Eigenvalues on the complex plane, demonstrates the two Eigenvalues for the computed solution lie extremely close to the two specified Eigenvalues. Moreover, the real parts of the remaining four Eigenvalues of the solution matrix $\tilde{\mathbf{A}}$ are all below the ϕ constraint (solid green vertical line). Figure 30B plots the amplitude and phase of each Eigenvector component relative to the first component. Since the magnitude of an Eigenvector is arbitrary, one component of the Eigenvector (the first in this case) can be matched exactly. As for the second and third components of the computed Eigenvector (associated with links 2 and 3), the relative phasing is matched within .1 radians for link 2 and .5 radians for link 3. The relative amplitude for link 2 is off by 12% while the relative amplitude for link 3 is off by 91%. It is of note that, in this representative example, the Eigenvalues were matched more closely than the Eigenvectors.

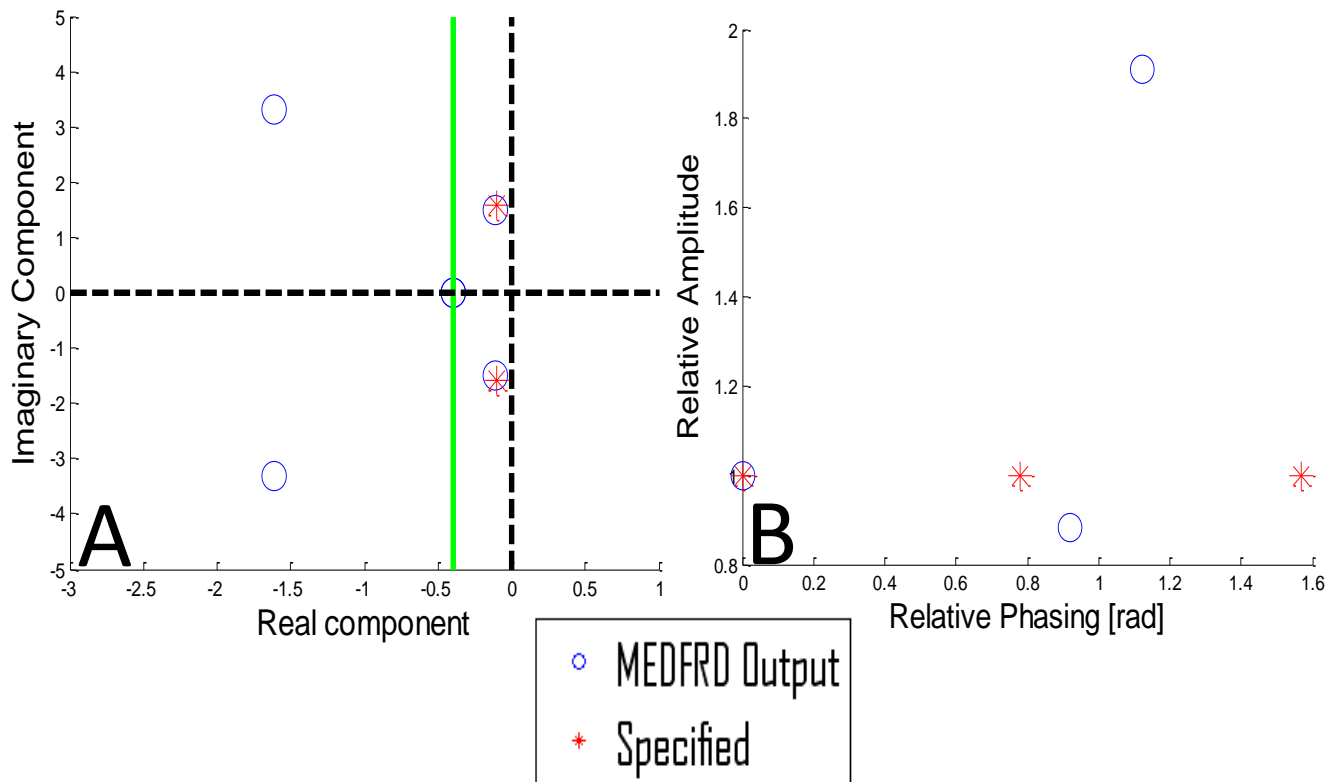


Figure 30: Representative Solution to the Optimization variant of Dual-Domain Eigenstate Factorization; a variant of the MEDFRD problem

2. Results of Monte Carlo Comparison Study

To more thoroughly examine the proposed solution methodologies, Monte Carlo simulations were employed to explore the sensitivity of the methods to initial conditions and to the introduction of physical design constraints. Solutions were considered for eight cases, comprising a 2x2x2 space of solution techniques, with one axis consisting of strict and relaxed solutions, another of global and gradient-based solvers, and the final axis of tight and loose parameter constraints.

In order to determine the effectiveness of the strict and relaxed formulations a metric must be selected that fairly compares solutions. For this purpose equation (29), which is here referred to as J_{comp} , was adopted. The value of J_{comp} is characteristic of the level of fit between the specified Eigenstates and actual Eigenstates for the system (with a J_{comp} value of 0 indicating an exact match). Other alternative norms for equation (29) could have been used, but as is demonstrated in Appendix B, the selection of matrix norm to determine J_{comp} is irrelevant.

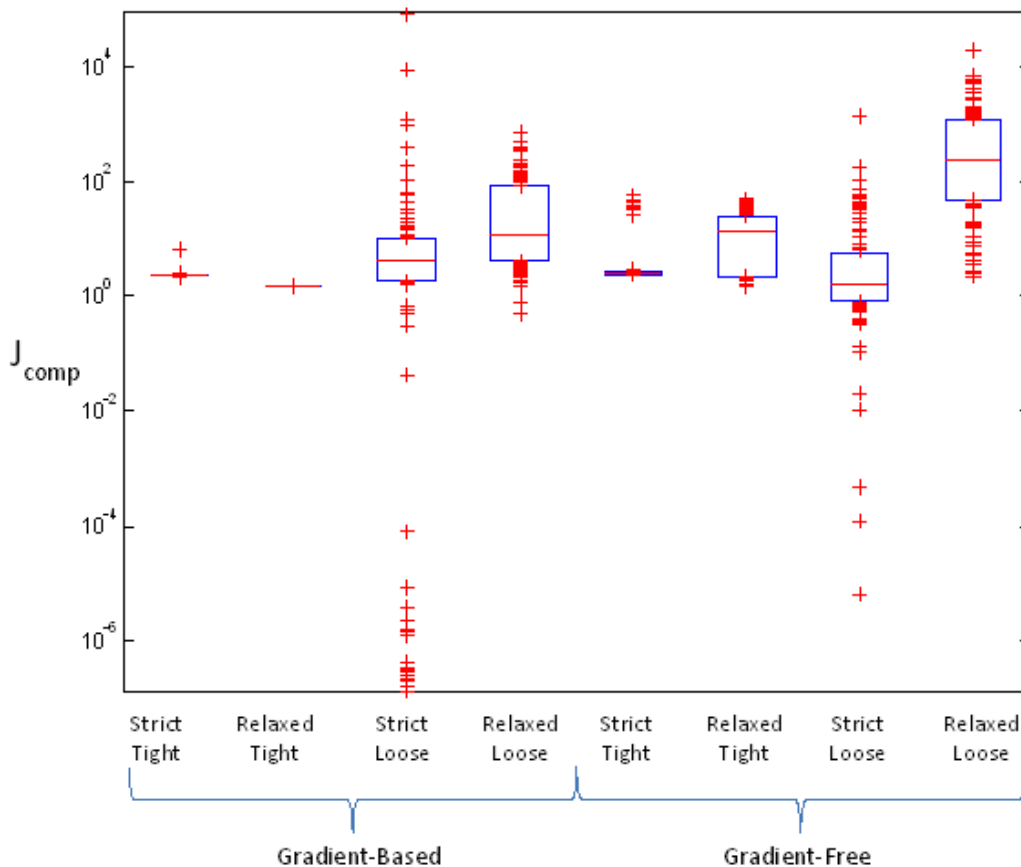


Figure 31: Box Plots of J_{comp} Values Achieved

For each of the eight cases configurations, a Monte Carlo simulation was run over 100 randomized initial conditions. Figure 31 indicates the results of all 800 solutions. For each of the eight configurations tested, a box plot was created that indicates the median J_{comp} value (red horizontal line inside box) and the bounds of the middle two quartiles (upper and lower edges of blue box). The remaining two quartiles are plotted as (red) crosses.

Other characteristics of the solution are also relevant in assessing solution quality. For instance, it is also important to consider how often the desired Eigenvalue limit (for unspecified Eigenvalues) is met. Table III indicates the percentage of trials in which maximum unspecified Eigenvalue limit ϕ was met. Table III also indicates the percentage of trials when the maximum unspecified Eigenvalue limit is not met. These cases are further divided into two subcategories; trials for which the solution has converged and trials for which the solution did not converge, either because the solution procedure did not run for enough iterations (Timed Out) or due to an ill-conditioned problem (Diverged).

Table III. Maximum Unspecified Eigenvalue Limit

	Strict Tight Gradient- Based	Relaxed Tight Gradient- Based	Strict Loose Gradient- Based	Relaxed Loose Gradient- Based	Strict Tight Gradient- Free	Relaxed Tight Gradient- Free	Strict Loose Gradient- Free	Relaxed Loose Gradient- Free
ϕ constraint met	100	82	99	32	94	14	88	22
ϕ constraint NOT met	0	9	0	43	0	33	0	9
Timed Out OR Diverged	0	9	1	25	6	53	12	69

V. Discussion

In analyzing the simulation data presented in Figure 31 and Table III, problem configurations which performed most favorably can be identified, at least for the model pendulum problem considered.

Matching performance, as quantified by J_{comp} , appears to favor the strict formulation whenever the gradient-free solver is used and the relaxed formulation whenever a gradient-based solver is

used, at least for the model pendulum problem. According to Figure 31 the strict formulation is favorable whenever a gradient-free optimization method is used because the median J_{comp} values are lower for the strict formulation than the relaxed formulation ([2.60, 1.68] vs. [13.43, 252.91] respectively), and the maximum J_{comp} outliers for the relaxed formulation are equal to or larger than that of the strict formulation. In other words, the strict formulation offers both more consistency and a better solution quality. However, when gradient-based optimization techniques are used, the relaxed formulation is more favorable than the strict formulation because the strict formulation has comparable median values to the relaxed formulation ([2.44, 4.43] vs. [1.49, 12.21] respectively), but is substantially more inconsistent. In particular, notice that the outliers for the strict formulation are very large (even on a log scale) when using a gradient-based solver for the case of loose constraints.

According to Table III it is clear that the solutions obtained using the strict formulation adhere to the maximum unspecified Eigenvalue constraint on a much more consistent basis than the relaxed formulation. Thus, the cases from Figure 31 in which the relaxed formulation was deemed favorable must be reevaluated. Considering the gradient-based cases, the strict formulation is able to meet the maximum unspecified Eigenvalue constraint on average 42.50% more often than the relaxed formulation. In fact, the strict formula never converges on a solution in which the maximum unspecified Eigenvalue limit is not observed. Alternatively, the relaxed formulation converges on a large number of solutions that do not adhere to the maximum unspecified Eigenvalue limit (average of 26.00% for gradient-based optimization cases), and has a substantial number of non-convergences of the optimization routine (average of 17.00% for

gradient-based optimization cases). If it is indeed important to meet the unspecified Eigenvalue constraint, using the strict formulation would thus be preferred over the relaxed formulation.

A reasonable balance of these diverse criteria is obtained when using the strict formulation with the gradient-free (i.e. pattern search) solver. The number of convergent solutions (that meet the ϕ constraint on the maximum unspecified Eigenvalue) is very high, though not 100%, when solving the strict formulation with pattern search. The median values for this configuration are also comparable or better than for other configurations, and the outliers are less severe. With the caveat that these results are based specifically on our study of a model three-link pendulum system, it is recommended to start with using pattern search to solve the strict formulation in applying the optimization variant of Dual-Domain Eigenstate Factorization to other dynamic systems.

VI. Conclusion

In this chapter, a design problem in which an engineer specifies a small set of desired modes for an under-actuated dynamic system was introduced. This problem was dubbed the optimization variant to Dual-Domain Eigenstate Factorization. Solutions to the optimization variant of Dual-Domain Eigenstate Factorization problem are obtained through co-design, in which both control and physical parameters are tuned to match the modal dynamics defined by the design engineer. This variant allows for inequalities on physical and control parameters to also be considered. For this chapter two proposed algorithms which attempt to minimize all non-specified Eigenvalues, were introduced.

Two different optimization algorithms we introduced to solve the optimization variant of Dual-Domain Eigenstate Factorization; one involving a non-convex constraint and the other a non-convex cost function. Gradient-based optimization and gradient-free optimization methods were applied to the two formulations. Combinations of the two formulations and the two solvers were assessed for a model dynamic system (a three-link pendulum). At least for this model system, applying pattern search to solve the strict formulation provided the best balance of Eigenstate matching and solution convergence.

Chapter 7: Conclusion

I. Overview

Each chapter in this thesis has developed the concept of Eigenstate specification, with the goal of creating periodic oscillations in under-actuated linear systems. This thesis has covered the motivation, specification, and implementation of periodic oscillations in systems. A depiction of the contributions of each chapter of this thesis, and how they tie into the overall goal of embedding periodic oscillations into natural dynamics, can be seen in Figure 6.

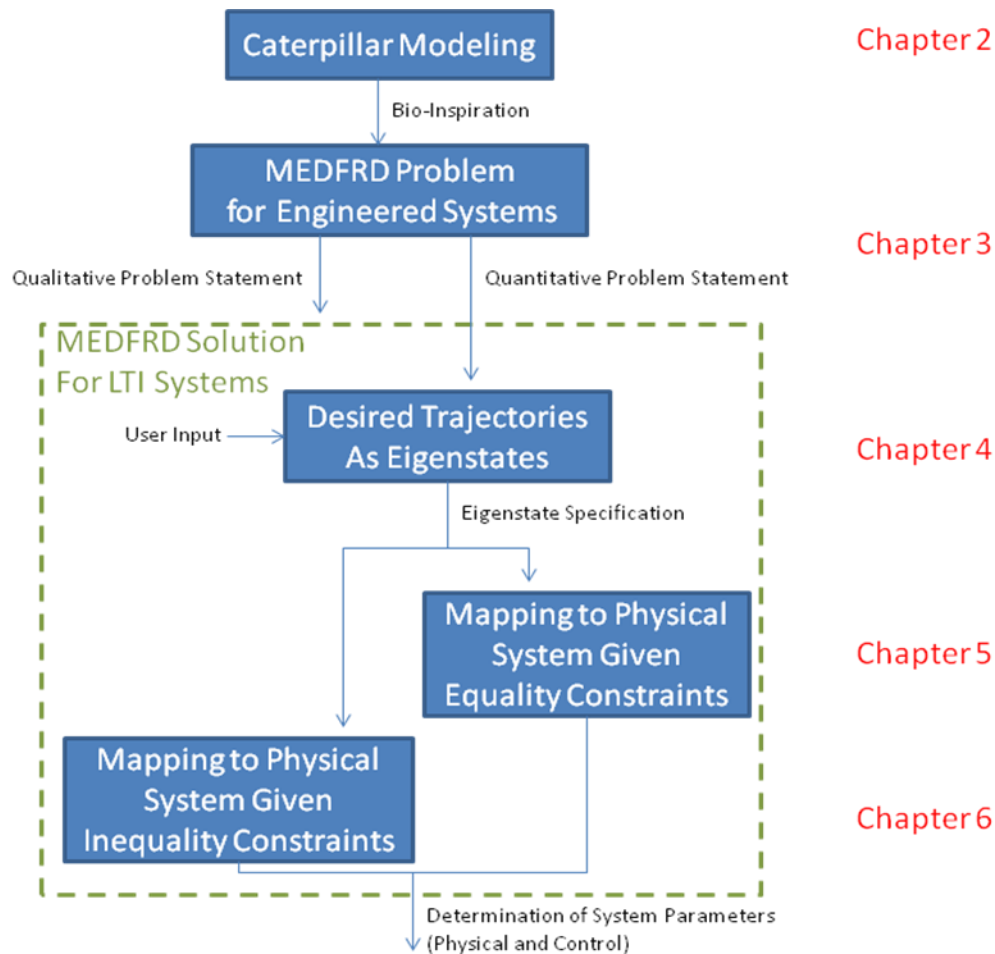


Figure 6: Thesis Overview

Each chapter in this thesis covered essential components toward a thorough understanding of the impact of physical and control parameters on system Eigenstates, and how to tune those Eigenstates to elicit desired dynamic behaviors. As is demonstrated in Figure 6, chapter II offered a thorough examination of an under-actuated model system, the *Manduca sexta* caterpillar, in order to gain insights into the mechanisms of periodic wave creation in natural systems. This motivation was used to determine a strict definition of Eigenstate matching, qualitatively and quantitatively, which is discussed in Chapter III. The quantitative description of the Eigenstate specification problem, referred to as Modal/Eigenstate Determination for Reoccurring Dynamics, was then used in Chapter IV to develop a method of converting desired oscillating trajectories into Eigenstates for linear systems. Finally, Chapter V and VI offered alternative solution methodologies for matching system Eigenstates to desired Eigenstates. Based on the methodologies described throughout this thesis design engineers can now translate desired periodic system trajectories into system Eigenstates, and embed those Eigenstates into a system as closely as possible through the alteration of physical and control parameters. The contributions of this thesis can potentially allow for under-actuated and constrained systems to generate periodic oscillations of system states in a designed manner.

II. Contributions

In this section the major contributions of this thesis will be reiterated and expanded upon. The major contributions pertain to biological inspiration for the generation of periodic oscillations in under-actuated systems, the embedding of desired system responses into Eigenstates, and the alteration of system parameters to match desired natural dynamics.

1. Bio-Inspiration

The first contribution of this thesis was:

1. Obtained model of internal forces within a crawling *Manduca sexta* caterpillar by applying inverse-kinematic model to experimentally acquire motion capture data.

- *Developed extensible-link model of caterpillar locomotion*
- *Generated first model of internal forces within a crawling caterpillar, forces which cannot otherwise be measured directly but that are relevant to*
 - *Engineering inspiration for crawling robots*
 - *Biological understanding of the “Environmental Skeleton” model of caterpillar locomotion*

The first contribution of this thesis document pertains to understanding the fundamentals of caterpillar locomotion. This section will reiterate the biologically inspired contributions of this document and expand upon each sub-contribution by referencing results demonstrated throughout the thesis document.

1. Obtained model of internal forces within a crawling *Manduca sexta* caterpillar by applying inverse-kinematic model to experimentally acquired motion capture data.

In order to understand how traveling waves are created by the caterpillar a novel extensible-link model was created, as can be seen in Figure 7. A dynamic model of *Manduca sexta* locomotion has never been accomplished previously, perhaps due to the animal’s highly compliant nature. The model generated in this thesis was based on inverse-dynamics and inverse-kinematics of animal tracking data as is demonstrated in sections II.3.a. and II.3.b. of chapter II. Simulated ground reaction force values determined using the caterpillar model were compared against

measured values, and found to be representative of those recorded (as is demonstrated in Figure 11 and 12). Given this confirmation of the model, internal forces were analyzed and the importance of tension in the animal during crawling was demonstrated. Internal force data has previously never been recorded or predicted.

- *Developed extensible-link model of caterpillar locomotion*

Developing an extensible link model is challenging because link lengths are allowed to change; shifting center of mass locations, moment arms, and consequently internal and external forces and torques. Extensible link models also require more tracking point locations to determine link orientations and displacements when applying inverse-kinematics (as is demonstrated by equation 4 through equation 9). Consequently, conventional rigid dynamic models generally do not allow for link length alterations. For this reason, high degrees of freedom systems are typically represented with continuum models. However, the model configuration demonstrated in section II.1. of chapter II can be adapted for a wide range of structures with high degrees of freedom. An extensible link model is an effective alternative to continuum models for highly flexible and deformable systems, in that they can bend and compress or extend. If a rigid model cannot capture the flexibility of a system and a continuum model offers unnecessary complexity, an extensible-link model may be a sufficient alternative. For this reason, an extensible-link model was applied to the *Manduca sexta* caterpillar. No dynamic model of *Manduca sexta* locomotion has previously been created and verified against recorded animal data.

- *Generated first internal force data (since direct measurements are not possible) which is relevant to:*

Internal force data has previously never been recorded or predicted. It is currently not possible to record internal force profiles during locomotion due to the number of muscles and the technical

challenge of embedding force sensing elements into a caterpillar. Consequently, the internal force profiles demonstrated in Figure 13 are the first prediction of *Manduca sexta* internal forces ever created.

- *Engineering inspiration for crawling robots*

It was observed that the periods of liftoff for each caterpillar segment are similar in terms of duration, and are phased from the anterior to the posterior, as is demonstrated in Figure 15. This indicated that the caterpillar uses phased periodic displacements of segments in order to generate locomotion and motivates the need for designed periodic oscillations in robotic systems (described in chapter III). The means in which the caterpillar locomotes through the propagation of a compressive wave, in which localized regions of tension are adjacent to compression (displayed in Figure 13) has not been previously implemented as a means of robot locomotion. Effectively, if the internal and external force profiles demonstrated in chapter II could be replicated in a robotic hardware, then caterpillar-like motion could occur.

- *Biological understanding of the “Environmental Skeleton” model of caterpillar locomotion*

Based on the model system demonstrated in chapter II it has been concluded that *Manduca sexta* uses its environment as a necessary component to its locomotion; using the substrate to create tension in body segments which aid in the propagation of a traveling wave. This determination is backed by Figure 13, which shows the importance of tension inside of the caterpillar during the propagation of a compressive wave during locomotion. Although external force recordings have indicated that tension in the caterpillar may play a critical role to creating its locomotion, the results demonstrated in chapter II have confirmed that this is true based on internal force predictions. This idea is referred to as the “Environmental Skeleton” hypothesis.

2. Eigenstate Specification

The second contribution of this thesis was:

2. Invented a new approach for specifying the modal dynamics of a linear system to match periodic state trajectories as closely as possible.

- *Proposed concept of embedding desired trajectories into closed-loop system modes*
- *Proposed Fourier-Eigenstate specification, a new method that sets Eigenvalues to multiples of a base frequency as a means to map a Fourier Series Description of a trajectory into an Eigenstate description*
- *Adapted Eigenstate specification to apply to not only to undamped systems but also to weakly damped systems*

The second contribution of this thesis document pertains to embedding desired periodic oscillations of system states into system Eigenstates (Eigenvalue and Eigenvector pairs). This section will reiterate the second contribution of this thesis and expand upon each sub-contribution by referencing results demonstrated throughout the thesis document.

2. Invented a new approach for specifying the modal dynamics of a linear system to match periodic state trajectories as closely as possible.

A methodology to generate periodic oscillatory movements in an under-actuated system, similar to those observed in the caterpillar, was investigated. A qualitative and quantitative problem statement involving the generation of periodic oscillations in under-actuated systems, through the manipulation of modal dynamics by physical and control parameters, was offered in chapter III. This problem definition has previously not been explored, and is an interesting new area for research. Based on the problem definition of chapter III, referred to as the *Modal/Eigenstate Determination for Reoccurring Dynamics* problem, it was decided to utilize the natural dynamics

of a system to create desired periodic trajectories of states when implementing a periodic forcing function. However, in order to recreate desired periodic displacements of system states it became necessary to be able to specify resonance modes of systems which illicit the desired responses. In order to achieve this goal a novel methodology of converting desired periodic state trajectories into mode shapes of a system was introduced in chapter IV, and is referred to as *Fourier-Eigenstate Specification*. Fourier-Eigenstate specification offers the first comprehensive description of embedding desired periodic oscillations into system Eigenstates.

- *Proposed concept of embedding desired trajectories into closed-loop system modes*

A qualitative and quantitative problem statement involving the generation of periodic oscillations in under-actuated systems through the manipulation of mode shapes was offered in chapter III. Section II.1. of chapter III laid out a complete qualitative description of the problem of embedding periodic oscillations into under-actuated and/or constrained system's Eigenstates. Section II.2 then methodically converted each qualitative description into a quantitative representation. This problem has previously not been explored, and is an interesting new area of research. The problem definition of chapter III was referred to as the Modal/Eigenstate Determination for Reoccurring Dynamics (MEDFRD) problem. The MEDFRD problem definition is particularly interesting to under-actuated systems, such as soft and hyper-redundant robots, because it may offer a way of generating periodic displacements that result in locomotion through tuned physical and control parameters; potentially allowing for fewer actuators and less mechanical complexity.

- *Proposed Fourier-Eigenstate specification, a new method that sets Eigenvalues to multiples of a base frequency as a means to map a Fourier Series Description of a trajectory into an Eigenstate description*

A methodology of converting desired periodic oscillations into Eigenstates of a system was introduced in chapter IV, and is referred to as Fourier-Eigenstate specification. No method to convert complex phased periodic oscillations of states into Eigenstates has previously been investigated. Fourier-Eigenstate Specification is capable of indicating what Eigenvalues (demonstrated by equation 54) and Eigenvectors (demonstrated by equation 46) each mode should be comprised of in order to generate desired periodic oscillations with designer specified amplitudes, relative phasing, frequency, and shape of oscillations. This methodology was demonstrated on a model system, a six cavity oscillatory platform which can be seen in Figure 17. The methodology was able to prescribe desired periodic oscillations into the Eigenstates of the cavity system and then implement a periodic forcing function to elicit desired system responses, as is demonstrated in Figure 20.

- *Adapted Eigenstate specification to apply to not only to undamped systems but also to weakly damped systems*

For all real systems some amount of damping must be present. Consequently the effect of damping in the Eigenvalues of a system on the modal response of a system was investigated in section II of chapter IV. It was demonstrated in section II of chapter IV that a single periodic forcing function can be constructed out of a set of amplified and phased sine waves that can compensate for modal response alterations for lightly damped structures. A quantitative description for correctly specifying forcing functions and the effects of damping can be observed in equation 47 through equation 55. This contribution is significant because it allows designers to have weakly damped systems (which are feasible) and to input forcing functions in a particular way as to elicit desired behaviors (as is demonstrated in Figure 20).

3. Eigenstate Matching

The third contribution of this thesis was:

3. Identified and solved a Dual-Domain Eigenstate problem as a means of converting dynamic system specifications into a realizable set of system parameters.

- *Identified that it is necessary to solve a Dual-Domain Eigenstate problem (a problem in which some values of state-update matrix are known and in which some of the Eigenvectors and Eigenvalues of that matrix are known, and in which the remaining matrix values and Eigenstates must be computed) in order to map desired modes into state-space description of system*
- *Proposed solving a set of non-linear algebraic equations in order to address the Dual-Domain problem involving equality constraints*
- *Adapted solutions to optimization problem format in order to include inequality constraints*

The third contribution of this thesis document pertains to tuning physical and control parameters of a system in order to match system Eigenstates to desired Eigenstates, thus creating the desired system dynamics to elicit periodic oscillations. This section will reiterate the third contribution of this thesis and expand upon each sub-contribution by referencing results demonstrated throughout the thesis document.

3. Identified and solved a Dual-Domain Eigenstate problem as a means of converting dynamic system specifications into a realizable set of system parameters.

With a firm grasp on what Eigenstates are needed to generate periodic oscillations of system states it was then important to understand how to create those Eigenstates in a system. The problem of specifying mode shapes becomes increasingly complex for systems with few actuators and constrained physical parameters. A methodology for tuning a system's passive

dynamic and active feedback parameters in order to achieve desired mode shapes, even for systems that are under-actuated and/or constrained, was investigated in chapter V and VI and is referred to as the *Dual-Domain Eigenstate Factorization* problem. The identification and definition of the problem described in section II of chapter V, is unique and has not been previously addressed. Two novel approaches for solving the Dual-Domain Eigenstate Factorization problem were thoroughly discussed in chapters V and VI, and were verified with two model systems (Figure 24 and Figure 29).

- *Identified that it is necessary to solve a Dual-Domain Eigenstate problem (a problem in which some values of state-update matrix are known and in which some of the Eigenvectors and Eigenvalues of that matrix are known, and in which the remaining matrix values and Eigenstates must be computed) in order to map desired modes into state-space description of system*

A unique problem of matching system Eigenstates to desired Eigenstates was identified as Dual-Domain Eigenstate Factorization in section II of chapter V. As is demonstrated in Figure 22, Dual-Domain Eigenstate Factorization involves tuning both system parameters as well as Eigenstate elements in order to match system Eigenstates as close as possible to desired Eigenstates. By opening up the search space to include physical, control and Eigenstate parameters physical constraints as well as desired Eigenstate specifications can both be accounted. Designers may thus be able to generate physically realizable systems that have Eigenstates as close as possible to desired specifications.

- *Proposed solving a set of non-linear algebraic equations in order to address the Dual-Domain problem involving equality constraints*

Chapter V offers a novel methodology for solving the Dual-Domain Eigenstate Factorization problem when only equality constraints on physical and control parameters are considered. No

method has been previously generated that attempts to solve Dual-Domain Eigenstate Factorization, with the goal of generating modal responses that create periodic oscillations. The methodology was verified on a model three-mass spring and damper system, and it was demonstrated (as can be seen in Figure 27 and Table 1) that physically realizable systems that adhere to all system constraints can be generated that elicit desired dynamic responses. Designers may be able to use the methodology offered in chapter V to generate periodic oscillations for under-actuated and constrained linear systems.

- *Adapted solutions to optimization problem format in order to include inequality constraints*

For some systems a physically realizable system may not be possible when only equality constraints are considered (e.g. masses may need to be positive). For this reason a novel variant of Dual-Domain Eigenstate Factorization based on optimization was introduced in Chapter VI. This methodology allows for both equality and inequality constraints, partial Eigenvalue specifications, and any number of states to be feasible (as is described in equations 90 and 92). A designer can implement the optimization variant of Dual-Domain Eigenstate Factorization without concern that the system solution is physically realizable, as was demonstrated by a three-link pendulum case study. The optimization variant of Dual-Domain Eigenstate Factorization offered in chapter VI can be used to generate desired dynamic responses in a system to a prescribed forcing function through the alteration of control and physical parameters, regardless of the level of actuation.

III. Future Work

In this thesis no methodology was offered to determine unique physical and control parameter combinations encompassed in a system's state-space representation. For all of the algorithms and methodologies described in this thesis the state-space representation combined physical and control parameters together into representative elements. Thus, once a desired state-space representation of a system was generated, the corresponding physical and control parameters that make that desired specification must be determined. In all of the methods discussed, all equality and inequality constraints imposed on physical and control parameters were mapped into constraints on the state-space representation. Thus, it is feasible to generate at least one set of physical and control parameters that match the desired state-space representation. However, no method was discussed for cases in which multiple physical and control parameter combinations could be encompassed by the state-space representation of a system. For example, it may be desirable to limit actuation (reduce control gains) or to limit physical system alterations from some base form. The selection of unique physical and control parameters from a given state-space representation is an area for future work.

Future work should address the "reality gap." That is, all methodologies are theoretically sound and justified in simulation, but variability found in practical application is an interesting region to explore. Hardware demonstrations and robustness analysis can be used to determine the sensitivity of embedding desired Eigenstates into systems. Also, linearizing non-linear systems works under specific conditions (e.g. small angle assumption). It would be interesting to determine the extent to which the Eigenstate specification methods introduced in this document would hold up for highly non-linear systems linearized about a reference point. Finally, in many

applications only discrete component sizes are possible. The effect of discrete components on the ability to match Eigenstates must be addressed if working hardware is to be generated. These topics remain as areas for future work.

The optimization variant of Dual-Domain Eigenstate specification given in chapter VI is a non-convex quadratically constrained quadratic problem. This is a type of optimization problem that is often difficult to solve, and generally impossible to know that a global optimal has been reached. Future work should try to identify convex relaxations of this optimization problem in order to achieve a globally optimal solution.

Robot design based on the optimization of simulated system performance can apply principles from this document to potentially improve design outputs and algorithm efficiency. Specifically, genetic algorithms using a dynamic simulation utility to determine a robot's fitness at achieving a particular task can be addressed. It has been demonstrated that the fitness landscape, while concurrently evolving physical and control parameters, is extremely spiky and difficult to navigate with a genetic algorithm based on a dynamic simulation utility [58]. Additionally, although many researchers have used algorithms to generate robotic platforms using dynamic simulation utilities, most do not locomote in intuitive manners [79 - 81]. The robots output from a genetic algorithm (or other random search routine) often do not operate in intuitive manners because they are judged on fitness criteria that are often highly specific (e.g. traveling the greatest distance) and do not consider manufacturability. Finally, designs output from simulation utilities are often not explicitly laid out in terms of control gains and physical parameter sets. However, it is clear that physical parameter and control input selections have dramatic effects on

robot performance in simulation [82]. Consequently, it is often up to a designer to make structures that are representative of the solutions output from simulation [54].

By leveraging a designer's desired periodic pattern, and restricting design choices to have specified mode shapes, the search space for an optimal design can be greatly reduced. This may make the searching for a global optimum easier and less computationally costly for a genetic algorithm. Also, the resulting motion will be constrained to be the designer's specified pattern. For some applications this may not be desirable, while for others an intuitive gait patterns is desirable (e.g. if snake-like motion is desired). Finally, by explicitly describing each mode with a combination of physical and control parameters, a designer can more easily and naturally convert a simulated design into hardware. The interaction between Fourier-Eigenstate Specification and the Dual-Domain Eigenstate Specification problem with genetic algorithms and dynamic simulation utilities is an interesting area that warrants further investigation.

IV. Summary

A novel methodology for generating periodic oscillations of system states was investigated in this document, from inspiration to motivation. It has been shown that even for systems that are physically limited and under-actuated it is feasible to create designed periodic phased oscillations of system states. The way in which the oscillations are captured by the natural dynamics is novel, and the methodologies of matching the natural resonant dynamics of the systems to prescribed dynamics is also a novel contribution. The contributions of this thesis can potentially be applied to any under-actuated and constrained linear time-invariant system in which it is desired to create desired periodic oscillations of system states.

Appendix A: *Manduca sexta* Model Parameters

Table IV: Manduca Model Parameters

Segment	Mass [kg]	Inertia[kgm ²]	Length [m]
12	0.0003836	3.708E-09	0.0084
11	0.0003836	3.708E-09	0.0084
10	0.00041102	3.973E-09	0.0084
9	0.00041102	3.973E-09	0.0077
8	0.00041102	3.973E-09	0.0077
7	0.00041102	3.973E-09	0.0077
6	0.0003836	3.708E-09	0.0085
5	0.0003836	3.708E-09	0.0085
4	0.00031168	2.915E-09	0.0085
3	0.00031168	2.915E-09	0.0085
2	0.00019763	2.059E-09	0.0085
1	0.00024763	2.588E-09	0.0085

Appendix B: Matrix Norm Investigation

Executive Summary- An exploration of alternative Matrix norms for the strict formulation of the optimization variant of Dual-Domain Partial Eigenstate Factorization (found in chapter VI) will be performed with the aid of a model system (a multi-link pendulum). The model system will demonstrate the effectiveness of alternative cost functions at achieving approximate and exact solutions to the Dual-Domain Partial Eigenstate Factorization optimization problem subject to a variety of constraint and actuation scenarios.

I. Analysis

For all of the analysis in this appendix the same model system of chapter VI, a three-link pendulum will be used in which rotational springs and not dampers are available. For a detailed description of the state-space system model of the three-link pendulum please refer to Chapter VI.

1. Specified Eigenstates

In order to determine the appropriate matrix norm when trying to get as close as possible to desired Eigenstates, I will prescribe an oscillating mode shape for the three-link pendulum. This oscillating mode shape is comprised of an Eigenvalue which determines the frequency of oscillation of the three-links, and an Eigenvector which describes the amplitude of the oscillations of each link as well as the relative phasing of each link's swing. In specifying Eigenstates for the pendulum, I will partition the assigned Eigenvectors, and assign only the elements of the Eigenvector associated with the angular position of each pendulum link. Through

partitioning, the elements of the Eigenvector corresponding to angular velocities are computed directly from those elements corresponding to position.

In this example it was chosen that the links oscillate at a frequency of $\pi/2$ rad/sec with a damping ratio of 0.063. The eigenvalue for this case is $\lambda = -0.1 + \frac{\pi}{2}j$. Furthermore, it was desired that the third link angle lag the first by one-quarter of a cycle ($\frac{\pi}{2}$ rad) and that the second link angle lag the first by only 1/8 cycle ($\frac{\pi}{4}$ rad). The resulting partitioned values for the eigenvector are thus:

$$\mathbf{x}_s = \begin{bmatrix} 1 \\ e^{\pi j/4} \\ e^{\pi j/2} \end{bmatrix} \quad (102)$$

Computing the remaining elements of the eigenvector based on kinematic relationships the full eigenvector is the following.

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_u \\ \mathbf{x}_s \end{bmatrix} = \begin{bmatrix} -.1 + \pi/2j \\ -1.18 + 1.04j \\ -\frac{\pi}{2} - .1j \\ 1 \\ (1 + j)/\sqrt{2} \\ j \end{bmatrix} \quad (103)$$

The complementary (complex conjugate) eigenvalue and eigenvector were specified in the same manner.

2. Scenarios to be Evaluated

For Eigenstate specification I will consider two actuation scenarios, in which two or three joint torques can be commanded via feedback control. The removal of a single joint torque can cause

the exact specification of the desired Eigenstate to become impossible. For this analysis, our baseline case we will set $\lambda_{\max, \text{unsp}} = -.5$. This will be done to ensure that the unspecified Eigenstate poles are located at least 4 times farther from the neutral axis than the specified Eigenstate poles (ensuring mitigation of unwanted dynamics), and that the optimization problem is single rather than multi-objective (resulting in a single solution rather than Pareto front). Inequality constraints will be arbitrarily selected to demonstrate the concept of exact and approximate solutions to the optimization variant of Dual-Domain Partial Eigenstate Factorization. The baseline system parameters were set tightly bounded, such that inequality constraints significantly restrict Eigenstate assignment, causing Eigenstate specification to have greater errors. The constraints used are summarized in Table II.

3. Alternative Matrix Norms

The ability to specify the desired Eigenstates given four matrix norms (Largest column sum, Largest row sum, Frobenius, Largest singular value) was analyzed to determine the effectiveness of the Dual-Domain Eigenstate Factorization variant at designing mode shapes for the three-link pendulum system. The strict Dual-Domain Eigenstate Factorization variant, described in chapter VI, was used to drive the system as close to the specified Eigenstates as possible under two levels of actuation.

II. Results

To determine if matrix norm selection has a significant impact on the Dual-Domain Eigenstate Factorization optimization solution, the Dual-Domain Eigenstate Factorization optimization problem was applied to each of the four matrix norms. The ability of the selected Dual-Domain

Eigenstate Factorization optimization variant cost function to match specified Eigenstates under inequality constraints and varying levels of actuation was investigated. The solutions to the Dual-Domain Eigenstate Factorization optimization method for both actuation cases and with each matrix norm cost function can be seen in Figures 32, 33, 34 and 35. Figure 32 and 33 depict the specified Eigenvalue, and the corresponding solutions output by the optimization method for each level of actuation and for each proposed matrix norm. Figure 34 and 35 indicates the relative phasing of the second and third links from the first link, given the specified Eigenvector for each actuation and matrix norm scenario. Figure 34 and 35 also depict the relative amplitudes of the degrees of freedom output by the optimization variant of Dual-Domain Eigenstate Factorization for the varying actuation levels and matrix norms.

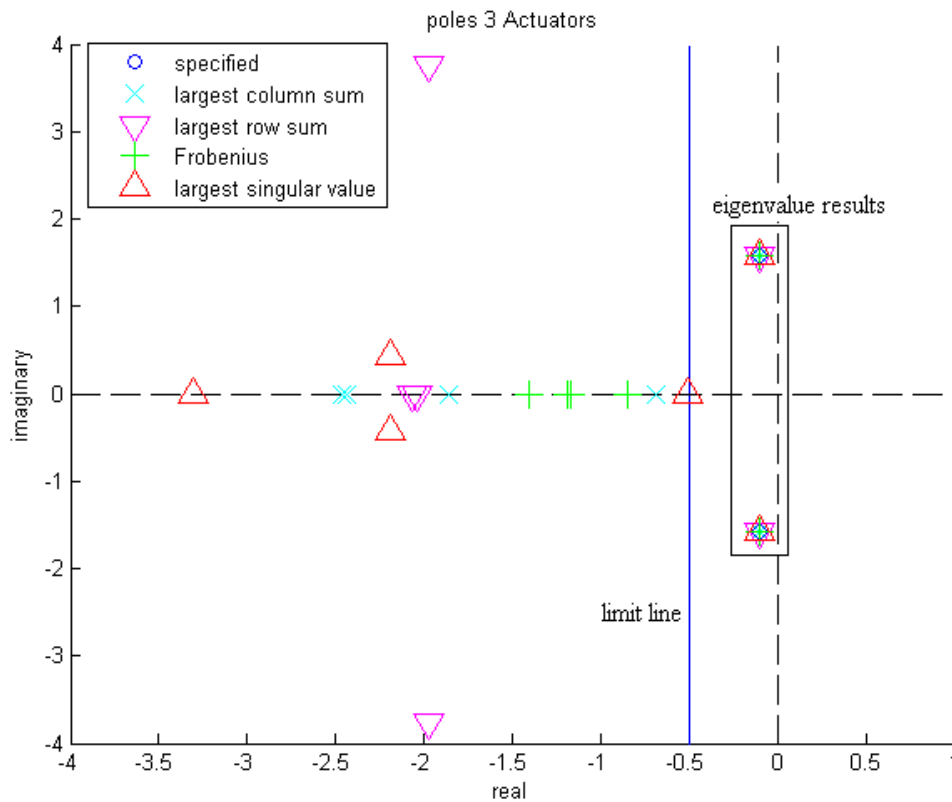


Figure 32: Frequency Matching 3 Actuators

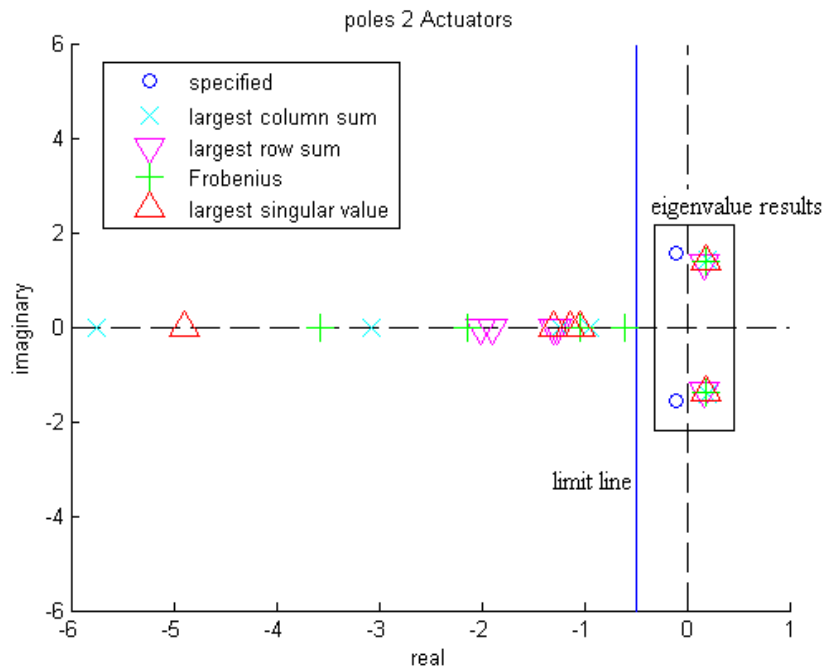


Figure 33: Frequency Matching 2 Actuators

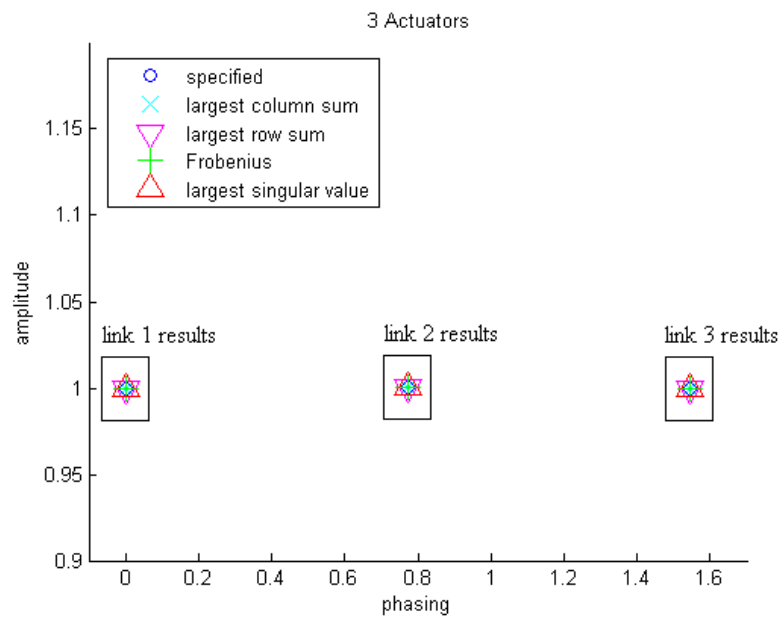


Figure 34: Amplitude/Phasing of Links 3 Actuators

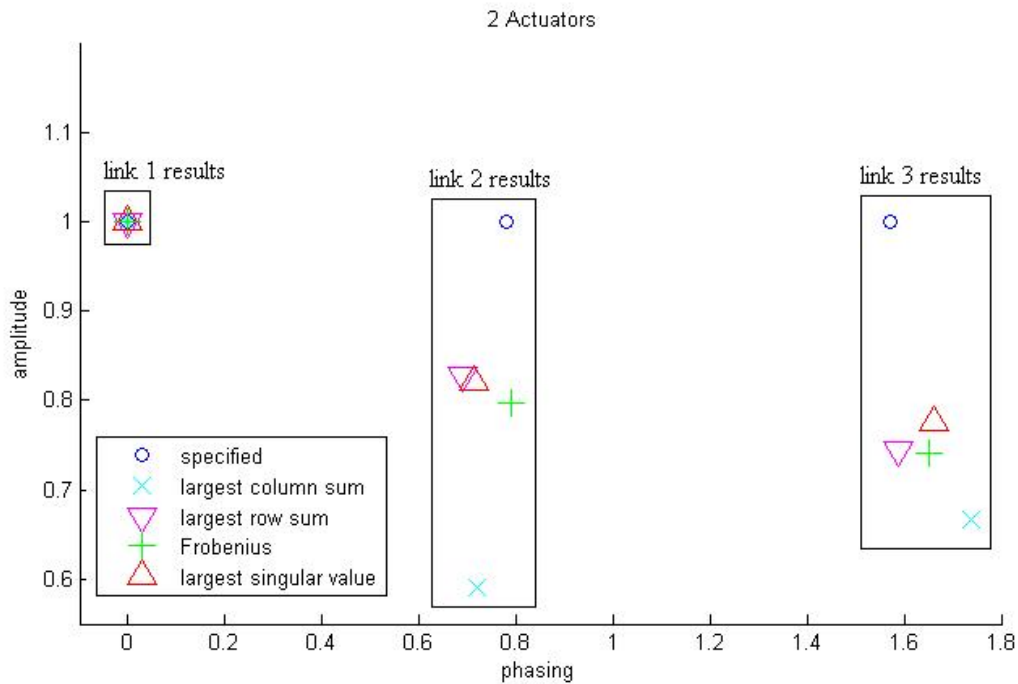


Figure 35: Amplitude/Phasing of Links 2 Actuators

III. Discussion

Figure 32, 33, 34 and 34 indicate that as the level of actuation reduces so does the ability to match the specified Eigenvalues and Eigenvectors. Figure 32 and 33 demonstrate that for all matrix norms the specified Eigenvalues seem to resemble the desired value while unspecified Eigenvalues are all less than $\lambda_{\max, \text{unsp}}$. Figure 32 depicts all matrix norms exactly achieving the desired Eigenvalues, while Figure 33 depicts all matrix norms attaining inexact solutions of equal error. Figure 34 and 35 show that as actuation reduces the ability to create desired phasing and amplitude reduces. Unlike Eigenvalue degradation it seems that Eigenvector amplitude error grows a relatively large amount with the absence of actuation. However, the phasing of the system is relatively preserved even when an actuator is removed. Of course alternative

constraints, selected Eigenvalues, and maximum unspecified Eigenvalue bounds can result in more or less accurate results in regards to matching Eigenvectors and Eigenvalues to desired ones. For the particular case study, however, based on Figures 32, 33, 34, and 35 the selected matrix norm for use in trying to achieve desired Eigenstates all result in similar solutions. It seems that the ability to specify desired Eigenstates is independent of the selected matrix norm, and is can be selected simply on ease of calculation.

IV. Conclusion

As actuators are removed from the three-link pendulum system, the ability to exactly specify the given Eigenstate is reduced. The strict formulation of the Mitigating Method seems to output solutions with comparable errors independent of the matrix norm selected. In this regard any proposed matrix norm cost function can be considered justified.

Appendix C: Modal Transfer Function Derivation

Given the state-space representation of a system

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u}.$$

Replacing \mathbf{A} with the Eigenvalue and Eigenvector matrices derived from Fourier-Eigenstate specification gives:

$$\dot{\mathbf{z}} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}\mathbf{z} + \mathbf{B}\mathbf{u}. \quad (104)$$

And if it is assumed that $\mathbf{g} = \mathbf{X}^{-1}\mathbf{z}$, then

$$\dot{\mathbf{g}} = \mathbf{\Lambda}\mathbf{g} + \mathbf{B}'\mathbf{u}. \quad (105)$$

Where $\mathbf{B}' = \mathbf{X}^{-1}\mathbf{B}$. Using this simplification the modal system response to a periodic forcing function can be described in Laplace form as:

$$\mathbf{g}(s) = (\mathbf{I}s - \mathbf{\Lambda})^{-1}\mathbf{B}'\mathbf{u}(s). \quad (106)$$

Thus, the transition vector between the system modal responses and the input forcing function can be described as:

$$\mathbf{G}(s) = (\mathbf{I}s - \mathbf{\Lambda})^{-1}\mathbf{B}'. \quad (107)$$

Appendix D: State-Space Representation of Cavity System

A state-space representation to the system can be created based on two primary assumptions.

1. First we assume that cavity pressure (P) is linearly related to cavity volume (V) such that:

$$\begin{aligned} P_1 &= \hat{A}V_1 \\ P_2 &= \hat{B}V_2 \\ P_3 &= \hat{C}V_3 \\ P_4 &= \hat{D}V_4 \\ P_5 &= \hat{E}V_5 \\ P_6 &= \hat{F}V_6 \end{aligned} \quad (108)$$

2. I also assume that the pressure drop due to fluid flow through the tubing connecting each cavity and exiting from the last cavity are related to the length of the tube (L), the radius of the tube (R) and the friction characteristics associated with the inlet and outlet of the tubes such that:

$$\begin{aligned} \alpha_1 &= \frac{\pi R_1^4}{128\eta L_1} + \frac{\pi R_1^3}{128\eta 55.54} \\ \alpha_2 &= \frac{\pi R_2^4}{128\eta L_2} + \frac{\pi R_2^3}{128\eta 55.54} \\ \alpha_3 &= \frac{\pi R_3^4}{128\eta L_3} + \frac{\pi R_3^3}{128\eta 55.54} \\ \alpha_4 &= \frac{\pi R_4^4}{128\eta L_4} + \frac{\pi R_4^3}{128\eta 55.54} \\ \alpha_5 &= \frac{\pi R_5^4}{128\eta L_5} + \frac{\pi R_5^3}{128\eta 55.54} \\ \alpha_6 &= \frac{\pi R_6^4}{128\eta L_6} + \frac{\pi R_6^3}{128\eta 55.54} \end{aligned} \quad (109)$$

Where η is equal to the dynamic viscosity of water and $\left(\frac{\pi R_i^3}{128\eta 55.54}\right)$ is equal to the fictional losses due to the tubing inlet and outlet for each tube i .

Given these two assumptions I can construct a state-space representation for the change in volume of each cavity with respect to time. The full state space representation for the system is given as:

$$\dot{\mathbf{V}} = \mathbf{A}\mathbf{V} + \mathbf{B}\mathbf{u} \quad (110)$$

$$\mathbf{u} = -\mathbf{K}\mathbf{V}$$

$$\mathbf{A} = \begin{bmatrix} -\alpha_1\hat{A} & \alpha_1\hat{B} & 0 & 0 & 0 & 0 \\ \alpha_1\hat{A} & -\alpha_1\hat{B} - \alpha_2\hat{B} & \alpha_2\hat{C} & 0 & 0 & 0 \\ 0 & \alpha_2\hat{B} & -\alpha_2\hat{C} - \alpha_3\hat{C} & \alpha_3\hat{D} & 0 & 0 \\ 0 & 0 & \alpha_3\hat{C} & -\alpha_3\hat{D} - \alpha_4\hat{D} & \alpha_4\hat{E} & 0 \\ 0 & 0 & 0 & \alpha_4\hat{D} & -\alpha_4\hat{E} - \alpha_5\hat{E} & \alpha_5\hat{F} \\ 0 & 0 & 0 & 0 & \alpha_5\hat{E} & -\alpha_5\hat{F} - \alpha_6\hat{F} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} K_1 & K_2 & K_3 & K_4 & K_5 & K_6 \\ K_7 & K_8 & K_9 & K_{10} & K_{11} & K_{12} \\ K_{13} & K_{14} & K_{15} & K_{16} & K_{17} & K_{18} \\ K_{19} & K_{20} & K_{21} & K_{22} & K_{23} & K_{24} \\ K_{25} & K_{26} & K_{27} & K_{28} & K_{29} & K_{30} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} \end{bmatrix}$$

$$\mathbf{V} = [V_1 \ V_2 \ V_3 \ V_4 \ V_5 \ V_6]^T$$

Where K_1 - K_{36} represents control gains present in the \mathbf{K} matrix, and V_1 - V_6 represents the volume of each cavity.

Appendix E: State-Space of Mass-Spring-Damper System

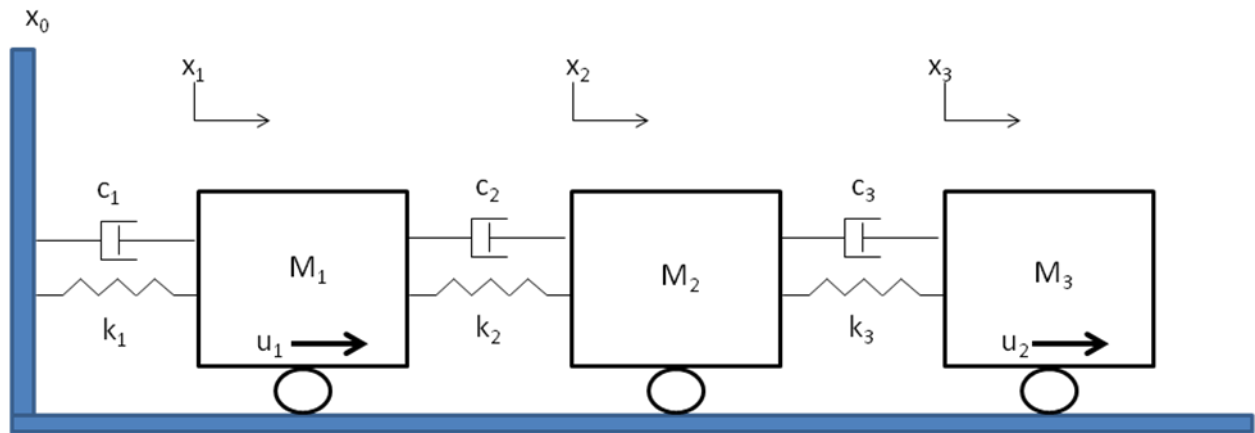


Figure 24: Mass-Spring-Damper Configuration

For this model it is assumed that $x_3 > x_2 > x_1$ and $\dot{x}_3 > \dot{x}_2 > \dot{x}_1$.

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (111)$$

$$\mathbf{u} = - \begin{bmatrix} K_1 & K_2 & K_3 & K_4 & K_5 & K_6 \\ K_7 & K_8 & K_9 & K_{10} & K_{11} & K_{12} \end{bmatrix} \mathbf{x}$$

$$\dot{\mathbf{x}} = \tilde{\mathbf{A}}\mathbf{x}$$

$$\tilde{\mathbf{A}} = \begin{bmatrix} \frac{-c_1 - c_2 - K_1}{m_1} & \frac{c_2 - K_2}{m_1} & \frac{-K_3}{m_1} & \frac{-k_1 - k_2 - K_4}{m_1} & \frac{k_2 - K_5}{m_1} & \frac{-K_6}{m_1} \\ \frac{c_2}{m_2} & \frac{-c_2 - c_3}{m_2} & \frac{c_3}{m_2} & \frac{k_2}{m_2} & \frac{-k_2 - k_3}{m_2} & \frac{k_3}{m_2} \\ \frac{-K_7}{m_3} & \frac{c_3 - K_8}{m_3} & \frac{-c_3 - K_9}{m_3} & \frac{-K_{10}}{m_3} & \frac{k_3 - K_{11}}{m_3} & \frac{-k_3 - K_{12}}{m_3} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Where K_i are the control gains, k_i are the linear springs, c_i are the linear dampers, and m_i are the masses.

Appendix F: Determining the Number of Specifiable Eigenstates for Linear Dynamic Systems

Executive Summary- This appendix introduces a metric called degree of shapeability to determine whether or not it is possible to assign a set of desired Eigenstates to a particular linear dynamic system. Specifying an Eigenstate, which consists of an Eigenvalue and its associated Eigenvector, can be essential in designing certain vibrating structures and undulating mechanisms. In general it may be desirable to specify more than one Eigenstate, or even a complete set of Eigenstates as in with Fourier-Eigenstate specification. The number of Eigenstates that can be specified for a particular linear system is not always immediately evident, however, because of physical and kinematic constraints, as well as actuator placement. To quantify the potential for active feedback and/or passive parameter tuning to influence a system's Eigenstates, I identify a metric, the degree of shapeability, that relates the structure of the linear system to the number of Eigenstates that a designer may fully define. I investigate the effect of system equality and inequality constraints on the ability to specify Eigenstates for linear dynamic systems.

I. Introduction

A system's *Eigenstates*, comprised of both Eigenvectors and Eigenvalues, are the basis for repeated movement patterns [6]. Eigenstate-based movement patterns are dynamic, inherently rejecting certain environmental disturbances, and hence are appropriate for applications such as the locomotion of robotic platforms in rough terrain. For many robotic platforms it is difficult to

*The material in this appendix was published in Proc. ASME Dynamic Systems and Control Conference 2011

generate dynamic, repeated movement patterns because of the physical constraints associated with design. This is especially true for *under-actuated* systems in which the number of degrees of freedom exceeds the number of actuators. For these cases, no method exists that allows for full specification of Eigenstates.

Caterpillar-inspired robots are one example of a robotic system that locomotes using a repeated motion pattern. A pneumatically actuated caterpillar-like robot, constructed with silicon elastomer, is illustrated in Figure 36. Such a robot has potential to squeeze through small spaces and traverse complex terrain. Caterpillar robots experience certain nonlinear forces (e.g. ground contact forces); however, I hypothesize that linearized models of such systems may be sufficient in designing robust, dynamic gaits. As such, this appendix will focus on linearized dynamic systems and specifically on the impact of design constraints and under-actuation on Eigenstate assignment.

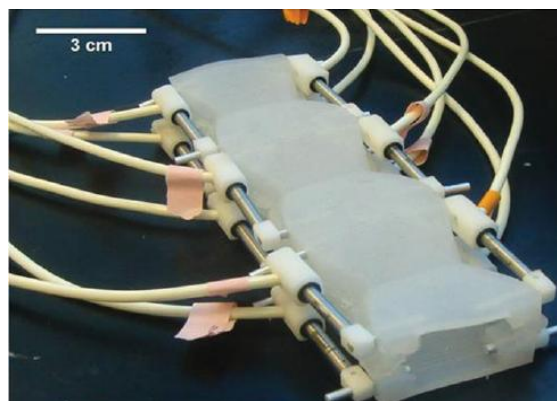


Figure 36: Pneumatically Actuated Deformable Robot

Eigenstates for a system can be altered by either tuning passive physical parameters or by incorporating active feedback control just as was demonstrated with Fourier-Eigenstate

Specification. Tuning passive physical parameters of a system, such as element masses or stiffnesses, can allow alteration of Eigenmode shape, resonant frequency, and decay rate [29]. When altering physical parameters, constraints often limit feasible solutions. These constraints are generally the result of kinematic coupling, size requirements, or fundamental physics (e.g. mass must be positive). In order to explore a wider design space given these constraints, it is desirable to incorporate feedback control.

Introducing feedback control can effectively modify passive parameters to assist in achieving a desired set of Eigenstates. The ability of feedback control laws to place Eigenvalues is well understood [30, 70, 83]. Control over eigenvectors (in addition to Eigenvalues) is a relatively new concept, referred to as *modal control* [28, 31, 74, 75]. Conventional modal control design methods focus on altering the control gains for full-state feedback of linear systems, but neglect the tuning of physical parameters.

Under-actuation is the major challenge in using feedback control to specify Eigenstates. As an example of an under-actuated system, again consider the pneumatic robot from Figure 36. The robot system is inherently under-actuated since the soft elastomer structure has, effectively, an infinite number of degrees of freedom. The system becomes more severely under-actuated if selected pneumatic pistons are removed. Eventually, if all of the actuators are removed from the robot, the robot's Eigenstates are purely a function of its physical parameters.

This appendix addresses an important gap in the existing literature. My approach is to characterize linear systems by quantifying system *shapeability*, which refers to the number of

Eigenstates which can be assigned given design constraints and under-actuation. The remainder of this appendix will develop the concept of degree of shapeability and explore the differing impacts of equality and inequality constraints on Eigenstate assignment. The appendix will conclude with a brief discussion of the capabilities and limitations of the proposed methodologies.

II. Eigenstate Specification

1. Degree of Shapeability

When trying to specify Eigenstates it is useful to the designer to know how many Eigenstates can be specified. To answer this question in a formal, quantitative manner, a *degree of shapeability* metric is defined. This metric might be used for example to determine the number of actuators required to achieve a particular undulating gait for the caterpillar-like robot seen in Figure 36.

Definition 1: A system's *degree of shapeability* is the number of Eigenstates that can be fully specified.

Here fully specified means that all Eigenstate parameters (the Eigenvalue and all elements of the Eigenvector) can be set independently of one another and of all other system Eigenstates.

The degree of shapeability is affected by equality and inequality constraints acting on the system. Equality constraints strictly limit the number and location of free parameters in the $\tilde{\mathbf{A}}$ matrix, and hence determine the maximum number of assignable Eigenstates. Inequality constraints do not necessarily limit the number of assignable Eigenstates unless an inequality constraint becomes

active. To understand the role of inequality constraints on the shapeability of a system, an example is helpful. Suppose a system has no active inequality constraints for a selected Eigenstate; then the degree of shapeability is simply equal to that determined by the equality constraints. On the other hand suppose all inequality constraints become active. There may be no feasible way to attain any independent Eigenstate in this situation and the degree of shapeability may drop to zero. Effectively, the number of active inequality constraints and the location of those constraints in the $\tilde{\mathbf{A}}$ matrix creates a *range of shapeability*, varying from the shapeability determined by equality constraints alone, to the shapeability achievable when all physical and control constraints are held constant (generally zero).

Definition 2: The *range of shapeability* is the closed set of values that the degree of shapeability may take, depending on the number of active inequality constraints.

2. Equality Constraints

The degree of shapeability of a system without inequality constraints can be determined based on the following.

Lemma 1: Degree of shapeability (S) of a system with only equality constraints is equal to the minimum number of free parameters in any one row of the $\tilde{\mathbf{A}}$ matrix.

A proof of this lemma is provided at the end of this appendix. Clearly, the number of assignable Eigenstates cannot exceed the total number of Eigenstates for a system. Hence, the degree of shapeability S cannot exceed the dimension D of the $\tilde{\mathbf{A}}$ matrix (i.e. $S \leq D$).

Practical application of Lemma 1 is straightforward if all equality constraints apply to individual parameters or to a linear combination of parameters in the same row. The number of free parameters in a row i is equal to the dimension of the row D , reduced by the number of independent equality constraints N_i . Considering the most constraining row, the following is true if no equality constraints apply across rows of $\tilde{\mathbf{A}}$.

$$S_\omega = D - \max_i(N_i) \quad (112)$$

Here S_ω is the degree of shapeability for a particular set of equality constraints. If all equality constraints apply either to individual elements of $\tilde{\mathbf{A}}$ or to multiple elements within a single row of $\tilde{\mathbf{A}}$, then no ambiguity exists in associating constraints with the rows of $\tilde{\mathbf{A}}$, and hence $S = S_\omega$. However, when an equality constraint applies to a linear combination of elements across multiple rows of $\tilde{\mathbf{A}}$, then the constraint may be associated with any one of those rows. As such, there may be multiple ways of associating equality constraints with the rows of $\tilde{\mathbf{A}}$. The full set of allowable associations is defined to be Ω . Considering all possible associations ($\omega \in \Omega$), the degree of shapeability is based on the association that allows the largest number of Eigenstates to be assigned.

$$S = \max_{\omega \in \Omega}(S_\omega) \quad (113)$$

The designer can choose to assign any number of Eigenstates up to degree of shapeability S .

A visual representation of the effect of shapeability, for a system with no inequality constraints, can be seen in Figure 37. The figure illustrates the approximate Pareto optimal curves (J_1 as a function of J_2) for a system with a variable number of actuators, much like the robotic system of Figure 36. As the number of actuators increases, the designer has more ability to specify the Eigenstates of the dynamic system. Hence, the system is more shapeable with more actuators, and less shapeable with fewer actuators. When the number of actuators is not sufficient to achieve the desired Eigenstates exactly, the value of J_1 cannot be set to zero. This limitation is represented in the figure when, as the number of available actuators decreases, the approximate Pareto optimal curve lifts away from the horizontal axis, indicating that J_1 is never zero and hence that an exact solution cannot be achieved. Even when an exact solution can be achieved (e.g. when the approximate Pareto optimal front intersects the horizontal axis with J_1 equal zero), additional actuators may still help in mitigating unspecified dynamics. In other words, additional actuators may aid in reducing the maximum unspecified Eigenvalue J_2 for a given value of J_1 . In the illustration, this effect can be identified by noting that the approximate Pareto fronts typically shift downward and to the left as the number of actuators is increased.

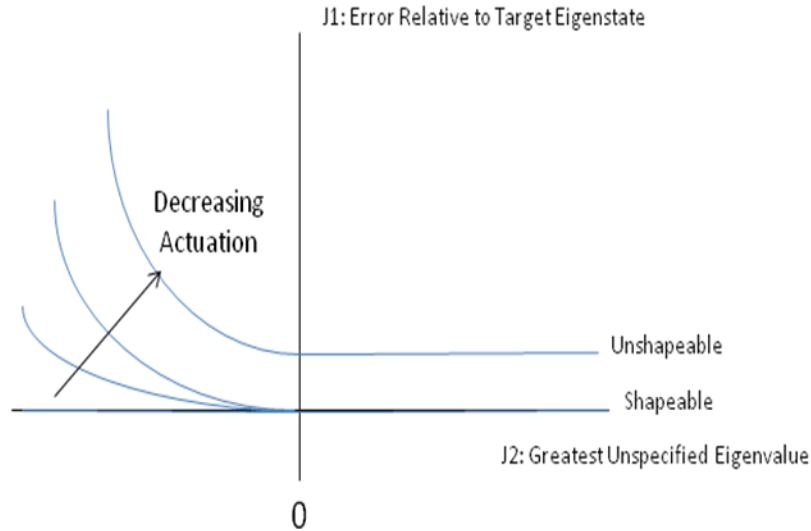


Figure 37: The Effect of Shapeability on the Solution of the MEDFRD Problem; The zero location on the J_2 axis indicates the location of neutral stability.

When the degree of shapeability S becomes less than the number of specified Eigenstates Q , then an exact solution for the specified Eigenstates is not possible. Any system in which $Q > S$ will be referred as *unshapeable*. Otherwise, the system will be referred to as *shapeable*.

3. Inequality Constraints

When the system has inequality constraints the degree of shapeability can reduce from that determined purely by equality constraints. A range of shapeability can be generated and is described by Corollary 1.

Corollary 1: The shapeability S of a system with inequality constraints is between δ and 0, where the upper bound δ is computed using equation (113).

This corollary is essentially a loose bound on the range of shapeability. The upper bound δ is based on the notion that the number of free parameters in the $\tilde{\mathbf{A}}$ matrix can never be higher than in the case when no inequality constraint is active. In this limiting case, shapeability is set only by equality constraints. As more inequality constraints become active, the number of free parameters in the system decreases, and shapeability can only decrease. In some cases, the number of inequality constraints may be sufficiently small that the lower end of the range of shapeability is a positive integer; however, it is likely that every parameter in the $\tilde{\mathbf{A}}$ matrix is subject to some practical engineering limitations (such that every element in the matrix is subject to at least one inequality constraint). In this limiting case, it is possible to specify certain Eigenstates which will cause inequality constraints on all matrix parameters to become active, thus pushing shapeability to zero. For these reason, it is justifiable to state that shapeability must lie between δ and 0 for any system with inequality constraints.

Corollary 1 emphasizes that inequality constraints can have significant effects on the ability to specify Eigenstates for a system. The degree of shapeability determined by equality constraints should not be over-interpreted. Because of inequality constraints, it may be possible to assign a particular Eigenstate, but it still may not be possible to assign that state to a desired target value.

If inequality bounds on variables of $\tilde{\mathbf{A}}$ are wide then an exact specification of Eigenstates may be possible. However, as inequality constraints tighten, exact Eigenstate specification can be made impossible, even for a fully shapeable system according to equality constraints, in which δ is equal to D . The addition of active inequality constraints effectively increases the number of

equality constraints on the system, and can reduce the shapeability of a system. Typical Pareto optimal curves for tightened inequality constraints are demonstrated in Figure 38.

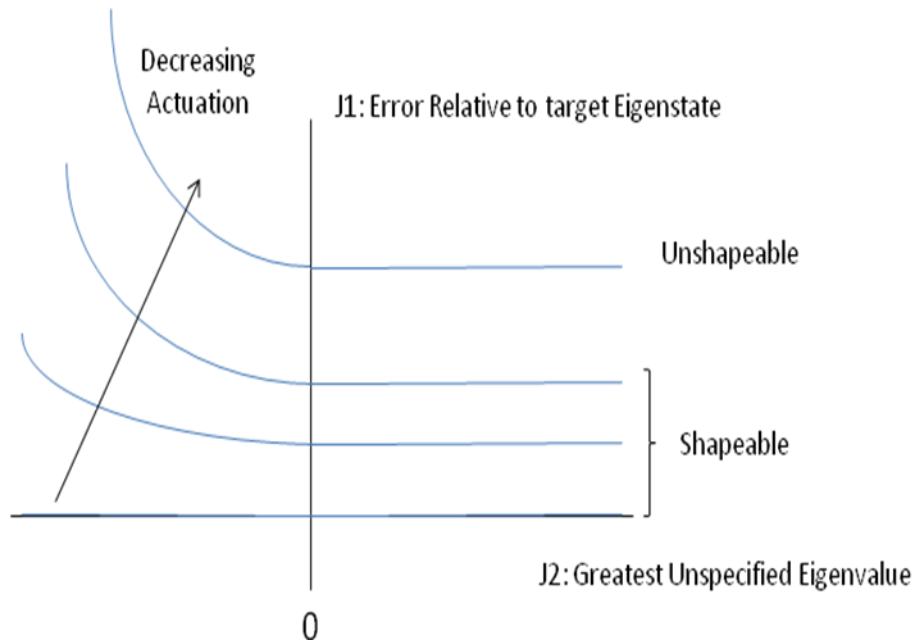


Figure 38: The Effect of Tightening Inequality Constraints of $\tilde{\mathbf{A}}$

Figure 38 is a visual representation of the effect of inequality constraints on the ability to specify Eigenstates. With the introduction of tight inequality constraints, an exact solution to J_1 may be made infeasible. This is in contrast to Figure 37, in which any shapeable system without inequality constraints can achieve a J_1 value of zero. The loss in ability to specify Eigenstates exactly, based on tight inequality bounds, is shown by the movement of approximate Pareto fronts away from the J_2 axis. Even for cases in which an exact solution is not possible, increasing actuation does increase the ability to approximate Eigenstates more closely. This is demonstrated in Figure 38 by the fact that as the number of actuators increases the approximate Pareto fronts

move downward and to the left. The effect of inequality constraints on the ability to specify Eigenstates exactly is highly dependent upon the number of free parameters and the form of the state-space representation of the system. Thus, degree of shapeability provides precise information about how many actuators are needed to specify a certain number of Eigenstates, but only qualitative information about whether a particular set of Eigenstates are achievable given a particular set of inequality constraints.

III. Conclusion

This appendix introduced a new metric for determining the number of Eigenstates which can be specified for a linear dynamic system. This metric, called degree of shapeability, provides insight into Eigenstate specification by considering both the actuation, morphology, and design requirements for the system. Degree of shapeability is particularly important in analyzing under-actuated cases. In concept, a designer could use the degree of shapeability metric to provide a preliminary indication of the number of actuators needed to achieve a desired set of Eigenstates, when complete actuation is not feasible. Degree of shapeability is defined crisply (at the upper limit) when only equality constraints are present. When inequality constraints are present in a systems matrix, the degree of shapeability is confined to a closed set of values called the range of shapeability.

Lemma 1 Proof:

Lemma 1: Degree of shapeability is equal to the minimum number of free parameters in any one row of the $\tilde{\mathbf{A}}$ matrix.

Proof: First consider the case in which only one Eigenstate is assigned. If the Eigenstate consists of the Eigenvalue λ_a and the Eigenvector \mathbf{x}_a , then the following equation must be true.

$$\tilde{\mathbf{A}}\mathbf{x}_a = \lambda_a\mathbf{x}_a \quad (114)$$

Each row of this vector equation may be written as follows, where $\tilde{\mathbf{A}}_i$ is the i^{th} row of the $\tilde{\mathbf{A}}$ matrix and where $x_{a,i}$ is the i^{th} element of the \mathbf{x}_a vector.

$$\tilde{\mathbf{A}}_i\mathbf{x}_a = \lambda_a x_{a,i} \quad (115)$$

In order to satisfy equation (115) for each i , given that the Eigenvector and Eigenvalue are fixed, at least one element of each row vector $\tilde{\mathbf{A}}_i$ must be a free parameter. Hence, specifying one arbitrary Eigenstate for a system requires that at least one element in each row of $\tilde{\mathbf{A}}$ is a free parameter.

Next consider the case when a second arbitrary Eigenstate is specified (consisting of Eigenvalue λ_b and Eigenvector \mathbf{x}_b). Applying (115) for each Eigenstate, the following must be true.

$$\tilde{\mathbf{A}}[\mathbf{x}_a \quad \mathbf{x}_b] = [\mathbf{x}_a \quad \mathbf{x}_b] \begin{bmatrix} \lambda_a & 0 \\ 0 & \lambda_b \end{bmatrix} \quad (116)$$

This equation puts two constraints on each row of the $\tilde{\mathbf{A}}$ matrix.

$$\begin{aligned}\tilde{\mathbf{A}}_i \mathbf{x}_a &= \lambda_a x_{a,i} \\ \tilde{\mathbf{A}}_i \mathbf{x}_b &= \lambda_b x_{b,i}\end{aligned}\tag{117}$$

Equation (117) represents a set of two linearly independent constraints on the elements of $\tilde{\mathbf{A}}_i$. This is true because the set of eigenvectors must be linearly independent for an Eigenvalue transformation to exist. Hence, the two Eigenvectors are linearly independent and so are the two constraints of (117). To satisfy the two constraints of (117), a solution is only possible if $\tilde{\mathbf{A}}_i$ contains two free parameters. Hence, specifying two Eigenstates for a system requires that at least two elements in each row of the transition matrix $\tilde{\mathbf{A}}$ are free parameters.

An extension of this logic makes possible a proof by induction. Consider the case in which S Eigenstates are specified. The generalization of (117) places S linearly independent constraints on each row vector $\tilde{\mathbf{A}}_i$. Hence, specifying S Eigenstates for a system requires at least S free parameters in each row of $\tilde{\mathbf{A}}$.

Appendix G: Shapeability Example

Executive Summary- Chapter V introduced the Dual-Domain Partial Eigenstate Factorization Method and examined a model system to demonstrate appropriate solution techniques. However, the effects of shapeability of the system on the solution quality can be investigated in further detail. This appendix will use the same model system described in Chapter VI, a three-link pendulum, to demonstrate the effects of under-actuation on system shapeability (which determines the number of Eigenstates that can feasibly be specified). This appendix will first reintroduce the model system and Dual-Domain Eigenstate Factorization Methods briefly. Sections on loss of actuations effect on shapeability will then follow.

I. MEDFRD Problem Overview

1. Exact Solution

Under some conditions, it is possible to exactly realize a desired set of Eigenstates. This section summarizes the case in which an exact solution is possible. The exact solution is based on the dynamics of a state-space system, which are written as:

$$\dot{\mathbf{z}} = \mathbf{Az} + \mathbf{Bu} \quad (118)$$

In this section, we seek to achieve desired Eigenstates by tuning passive and active parameters. In some cases, solutions are not unique, and several combinations of passive dynamic or active

*The material in this appendix was published in Proc. ASME Dynamic Systems and Control Conference 2011

feedback parameters achieve the desired Eigenstate. To remove this ambiguity, it is assumed full-state feedback control ($\mathbf{u} = -\mathbf{Kz}$) which simplifies the dynamic system as follows.

$$\begin{aligned}\dot{\mathbf{z}} &= \tilde{\mathbf{A}}\mathbf{z} \\ \tilde{\mathbf{A}} &= \mathbf{A} - \mathbf{BK}\end{aligned}\tag{119}$$

For a fully actuated system with no design constraints, the problem of specifying a desired set of Eigenstates is trivial. Simply, choose a matrix of Eigenvectors \mathbf{X} and a matrix of Eigenvalues Λ . The elements of the transition matrix $\tilde{\mathbf{A}}$ can be computed by the following equation.

$$\tilde{\mathbf{A}} = \mathbf{X}\Lambda\mathbf{X}^{-1}\tag{120}$$

Eigenstate assignment becomes more complex when constraints are added. Inequality constraints are associated with limitations on allowable physical parameters and also on the range of allowable control parameters (to avoid saturation, for example). Equality constraints may be associated with rigid kinematic constraints or with parameters that appear in multiple locations in the system due to dynamic coupling. Equality constraints are particularly important in under-actuated systems. Together, the sets of equality and inequality constraints are described below by two vector equations (\mathbf{f} and \mathbf{g}). Here inequalities are applied in an element-wise fashion. The variables \mathbf{c}_{ineq} and \mathbf{c}_{eq} are vectors of constants that characterize the equality and inequality constraints.

$$\begin{aligned}\tilde{\mathbf{A}} &= \mathbf{X}\Lambda\mathbf{X}^{-1} \\ \mathbf{f}(\tilde{\mathbf{A}}) &< \mathbf{c}_{\text{ineq}} \\ \mathbf{g}(\tilde{\mathbf{A}}) &= \mathbf{c}_{\text{eq}}\end{aligned}\tag{121}$$

2. Approximate Solution

If an exact solution is not feasible, then it may still be possible to tune system parameters to approximate a desired Eigenstate, which is the basis of. In this case, the exact equation (121) must be replaced with an optimization problem. This optimization problem has the following form, where the cost function J depends on the variable elements of the $\tilde{\mathbf{A}}$ matrix and the specified (fixed) Eigenstate parameters of the \mathbf{X} and Λ matrices. For the approximate solution, any inexact solution attained may carry along undesirable system dynamics.

$$\begin{aligned} \min_{\tilde{\mathbf{A}}} J \quad & J = \mathbf{h}(\tilde{\mathbf{A}}; \mathbf{X}, \Lambda) \\ & \mathbf{f}(\tilde{\mathbf{A}}) < \mathbf{c}_{\text{ineq}} \\ & \mathbf{g}(\tilde{\mathbf{A}}) = \mathbf{c}_{\text{eq}} \end{aligned} \quad (122)$$

The solution to this optimization problem depends on the specific choice of the cost function, \mathbf{h} . Various cost functions might reasonably be considered. In this section, as is consistent with previous chapters, we select a representative case: a cost function based on the Frobenius norm of the difference between the actual and desired system transition matrices.

$$\mathbf{h}(\tilde{\mathbf{A}}; \mathbf{X}, \Lambda) = \|\tilde{\mathbf{A}} - \mathbf{X}\Lambda\mathbf{X}^{-1}\|_{\text{F}} \quad (123)$$

The Frobenius matrix norm is defined as the root-sum-of-squares of the elements of a matrix.

II. Model System

The following example will be used in conjunction with the optimization variant of Dual-Domain Eigenstate Factorization to illustrate shapeability in relation to degree of actuators. The

example concerns the specification of Eigenstates for the stable three-link pendulum of chapter VI. For this system the $\tilde{\mathbf{A}}$ matrix is in $\mathfrak{R}^{6 \times 6}$. Half of the rows of the $\tilde{\mathbf{A}}$ matrix are kinematic constraints; the other half of the rows (representing the pendulum moment balances) consists of tunable parameters. To obtain intuition for this system, it is useful to consider a simpler single link pendulum, for which one row (kinematic constraint) has no tunable parameters and for which the other row (moment balance) comprises a set of tunable parameters. Here the term m is mass, l is pendulum length, I is inertia, and T is the joint torque for the pendulum. A joint rotational spring (k) and damper (c) have also been added. The dynamic equation for this system is the following.

$$\begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{-c}{I} & \frac{-mgl-k}{I} \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ I \\ 0 \end{bmatrix} T \quad (124)$$

It is assumed that the links do not rotate more than 15 degrees, such that model nonlinearities need not be considered. The $\tilde{\mathbf{A}}$ derived from the state-space representation has the form seen below, where K_1 and K_2 represent components of the control gain matrix.

$$\tilde{\mathbf{A}} = \begin{bmatrix} \frac{-c}{I} - \frac{1}{I}K_1 & \frac{-mgl-k}{I} - \frac{1}{I}K_2 \\ 1 & 0 \end{bmatrix} \quad (125)$$

By setting feedback parameters appropriately, the values of the first row of (125) can be set to any arbitrary value. The values of the second row are fixed, independent of all tunable parameters, active or passive. The three-link pendulum is simply a generalization of (125), in

which all parameters in the first three (moment-balance) rows are tunable, if torques at all joints can be controlled, and in which the parameters in the final three (kinematic) rows are not tunable.

III. Shapeability Investigation

In this section we investigate the utility of the degree of shapeability metric in characterizing the feedback-actuated, three-link pendulum. In this section, *strict formula* of the optimization variant off Dual-Domain Eigenstate Factorization (found in chapter VI) is used to assign Eigenstates to the three-link pendulum for several scenarios in which different numbers of joint-torque actuators are available and in which different sets of equality and inequality constraints are active. These results are compared to the degree of shapeability metric, to assess the utility of this metric in determining the number of actuators needed to realize a particular desired set of Eigenstates.

1. Specified Eigenstate

An oscillating mode shape for the three-link pendulum was specified. For the shapeability experiment, it was chosen arbitrarily that the links oscillate at a frequency of $\pi/2$ rad/sec with a damping ratio of 0.063. The eigenvalue for this case is $\lambda = 0.1 + \frac{\pi}{2}j$. Furthermore, it was chosen arbitrarily that the second link angle θ_2 should lag the first θ_1 by one-quarter of a cycle ($\frac{\pi}{2}$ rad), and that the third link angle θ_3 should lag the first by only 1/8 cycle ($\frac{\pi}{4}$ rad). The resulting eigenvector (accounting for coupling between angular velocity and angle) is:

$$\mathbf{x} = \begin{bmatrix} 1 \\ j \\ (1+j)/\sqrt{2} \\ \pi/2 \\ -\pi/2j \\ \pi/2(1-j)\sqrt{2} \end{bmatrix} \quad (126)$$

It must be noted that in order to obtain a periodic robot gait, it is necessary to perturb the system by inputting a periodic forcing function (for stable or marginally stable systems) or by entraining a limit cycle (for unstable systems).

The complementary Eigenvalue and Eigenvector (of the oscillatory pair) were specified as the complex conjugates of the primary Eigenvalue λ and the primary eigenvector \mathbf{x} . Hence, for this system, the total number of specified Eigenstates, Q , is two. According to shapeability analysis, we expect that system shapeability, S , must be at least two in order to specify these Eigenstates exactly.

2. Experimental Procedure

The Eigenstate specification problem was considered for a series of progressively less actuated scenarios, in which three, two, one, or zero joint torques can be commanded via feedback control. More specifically, removal of joint torques progresses downward from the constrained end of the three-link chain. As joint torques are removed, the associated rows of $\tilde{\mathbf{A}}$ have fewer tunable parameters as more equality constraints become active. Equality constraints are introduced when an element of the $\tilde{\mathbf{A}}$ matrix equals zero. As actuation is removed, some elements of the $\tilde{\mathbf{A}}$ matrix can no longer be shifted by feedback and hence become constrained to

equal zero. This removal of torques causes the system's degree of shapeability to change. For designs with a smaller number of actuators, the number of free variables in $\tilde{\mathbf{A}}$ decreases. When the shapeability S is smaller than the specified number of Eigenstates Q , it is anticipated that the system is unshapeable, and hence the desired Eigenstates cannot be obtained exactly.

This test was repeated twice, each time considering a different set of inequality constraints on the system's design parameters. The inequality constraints were arbitrarily selected to demonstrate contrasting extremes of constraint scenarios and demonstrate the range of shapeability concept, and are consistent with those used in chapter VI. In the first case, the system parameters are only very loosely bounded, such that inequality constraints play only a very small role in Eigenstate assignment. In this case, we expect the shapeability to approach its upper limit ($S \rightarrow \delta$). In the second case, the system parameters are much more tightly bounded, such that inequality constraints significantly restrict Eigenstate assignment, effectively reducing the degree of shapeability and causing Eigenstate specification to have greater errors. In other words, the system's shapeability approaches its lower limit ($S \rightarrow 0$) as constraints tighten. A comparison of the two cases emphasizes the role that inequality constraints play in assigning a system's Eigenstates. The constraints used in each of the two cases are summarized in Table II.

IV. Shapeability Experiment Results

To characterize the three-link pendulum system, the strict formula of the optimization variant of Dual-Domain Eigenstate Factorization was applied to each of eight instances: two cases of different inequality constraints on design parameters (Table II), each featuring four scenarios with different numbers of joint-torque actuators available (from three down to zero). Each of these eight instances is best characterized by a set of results (an approximate Pareto optimal

front) and not by a single result. To provide a simpler basis for comparison, however, we considered only a single point for each approximate Pareto optimal curve, the point where the value of J_2 was set to zero. By holding J_2 constant the degree of shapeability and inequality constraints could be analyzed independent of the unspecified dynamics.

In solving the strict Dual-Domain Eigenstate Factorization formulation, a pattern search algorithm was used rather than a gradient descent method because the cost function J_2 is not convex, and the results of chapter VI suggest that this approach is most favorable. The results of these computations are summarized in Table V. Values of J_1 for tests in which the loose bounds were applied are in the column labeled “Loose.” Values of J_1 when tight bounds were applied are in the column labeled “Tight.”

Table V: Varying Constraints and Shapeability

Actuators	δ	J1 (Loose)	J1 (Tight)
0	1	0.032	0.84
1	3	0	0.11
2	4	0	0.036
3	6	0	0

The first and last lines of Table V represent two extremes of Eigenstate assignment; these extremes are covered by existing methods, which predated Dual-Domain Eigenstate Factorization. The first line of Table V corresponds to the scenario in which no actuators are

present. In this case, dynamics are purely passive, and so Eigenstate placement would be possible using passive dynamic methods. The last line of Table V corresponds to the scenario in which all joints are actuated. In this scenario, it is always possible to specify all Eigenstates. The two middle lines of Table V are particularly interesting in that they exercise the capabilities of Dual-Domain Eigenstate Factorization, allowing for exact or approximate Eigenstate assignment, even for under-actuated systems, while mitigating unwanted system dynamics.

As actuators are removed from the three-link pendulum system, the degree of shapeability of the system changes. The upper bound on the degree of shapeability, δ , for each scenario is listed in the second column of Table V. In the Loose Constraint case, it is immediately evident that the shapeability limit indicates whether or not an exact solution is possible. When δ is larger than the number of Eigenstates specified ($Q = 2$), in the second and subsequent rows of the table, then the value of the cost function J_I can be pushed to its minimum value of zero, which indicates the desired Eigenvalues are obtained exactly. When δ is less than Q , as occurs in the first line of the table, then an exact solution is not feasible. Thus, when inequality constraints are loose, degree of shapeability provides a clear indication of whether or not an exact solution can be computed.

The role of δ is less crisp when inequality constraints are tight. In effect, the tight inequality constraints make the feasible space of Eigenstates smaller and smaller, until their distinction from equality constraints blurs. For the case of tight inequality constraints, the only exact solution was achieved by the fully actuated system (last line of Table V). The fully actuated system has a shapeability equal to the dimension of the $\tilde{\mathbf{A}}$ matrix ($\delta=D$), and since there are a large number of control parameters available for tuning, the inequality constraints have relatively

little impact on system design. For all scenarios with two or fewer actuators, an exact solution was not possible with the tight inequality constraints. This limitation was true even for cases in which δ was greater than Q . Thus, in the case of tight inequality constraints, the δ metric generated by equality constraints provides only partial information about whether or not a particular Eigenstate can be assigned to a dynamic system. Effectively, the degree of shapeability was reduced toward zero, as expected in the case of tight inequality constraints. Nonetheless, the upper bound, δ , nominally associated with equality constraints, was still indicative of the quality of the approximate solution, even in the case of tight inequality constraints. Specifically, a better approximation was achieved for partially actuated systems when δ was larger.

V. Conclusion

The number of fully assignable Eigenstates to a designer is a critical number that must be determined before assigning Eigenstates. In concept, a designer could use the degree of shapeability metric, rather than running a full design method (such as the optimization variant of Dual-Domain Eigenstate Factorization), to provide a preliminary indication of the number of actuators needed to achieve a desired set of Eigenstates, or the appropriate number of Eigenstates to specify. Degree of shapeability is defined crisply (at the upper limit δ) when only equality constraints are present. When inequality constraints are present in a systems $\tilde{\mathbf{A}}$ matrix, the degree of shapeability is confined to a closed set of values called the range of shapeability.

References

- [1] Chen, L., Ma, S., Wang, Y., Li, B. and Duan, D. (2007). "Design and modeling of a snake robot in traveling wave locomotion." *Mechanism and Machine Theory* 42 1632–1642.
- [2] McGeer, T. (1990). "Passive Dynamic Walking." *The International Journal of Robotics Research*, vol. 9 no. 2 pp. 62-82.
- [3] Srinivasan, M., Holmes P. (2008). "How well can spring-mass-like telescoping leg models fit multi-pedal sagittal-plane locomotion data?" *Journal of Theoretical Biology*, vol. 255, issue 1, pp. 1-7.
- [4] Pfeifer, R. et al. (2007). "Self-Organization, Embodiment, and Biologically Inspired Robotics." *Science* 318: 1088-1093.
- [5] Blich, J. (1996). "Artificial Intelligence Technologies for Robot Assisted Urban Search and Rescue." *Expert Systems With Applications*, Vol. 11, No. 2, pp. 109--124.
- [6] B. Kwon and M. Youn (1987). "Eigenvalue-Generalized Eigenvector Assignment by Output Feedback" *IEEE Transactions on Automatic Control*, VOL. AC-32. NO. 5.
- [7] Spong, M.W. (1994). "Partial feedback linearization of underactuated mechanical systems" *International Conference on Intelligent Robots and Systems '94*. Vol. 1 pg. 314-321.
- [8] Reyhanoglu, M. and McClamroch, N., (1999), "Dynamics and Control of a Class of Under-actuated Mechanical Systems." *IEEE Transactions on Automatic Control*, VOL. 44.
- [9] Gu, Y. (1993), "A Direct Adaptive Control Scheme for Under-Actuated Dynamic Systems." *Proceedings of the 32nd Conference on Decision and Control*, San Antonio, Texas.
- [10] Olfati-Saber, R. (2001), "Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles." MIT Ph. D. Dissertation.

- [11] Hosam K. Fathy , Julie A. Reyer , Panos Y. Papalambros and A. Galip Ulsoy. (2001) “On the Coupling between the Plant and Controller Optimization Problems.” *Proceedings of the American Control Conference* Pg. 1864-1869.
- [12] Abid, M., Chenguel, A., and Jerraya, A., (1996), “Exploration of hardware/software design space through a codesign of robot arm controller.” *Proceedings from Design Automation Conference*.
- [13] Allison, J. (2012), “Plant-Limited Co-Design of an Energy Efficient Counterbalanced Robotic Manipulator.” *In the Proceedings of the 2012 ASME Design Engineering Technical Conference*, DETC2012-71108, ASME.
- [14] Allison, J. and Nazar, S., (2010), “Combined Plant and Controller Design Using Decomposition-Based Design Optimization and the Minimum Principle.” *In the Proceedings of the 2010 ASME Design Engineering Technical Conference*, DETC2010-28887, ASME.
- [15] Lin, HT and Trimmer, BA. (2010). “Substrate as a skeleton: Ground reaction forces from a soft-bodied legged animal.” *J. Expt. Biol.* 213 1133-1142.
- [16] Brackenbury, J. (1999). “Fast Locomotion in Caterpillars.” *Journal of Insect Physiology* 45: 525-533.
- [17] Trimmer, B.A. et al. (2006). “Caterpillar Locomotion: A new Model for Soft-Bodied Climbing and Burrowing Robots.” *7th International Symposium on Technology and the Mine Problem*. Monterey, CA, May 2-5.
- [18] Woods, W.A. Fusillo, S.J. Trimmer, B.A. (2008). “Dynamic Properties of a Locomotory Muscle of the Tobacco Hornworm *Manduca Sexta* During Strain Cycling and Simulated Natural Crawling.” *The Journal of experimental Biology* 211: 873-882.

- [19] Trivedi, D. Rahn, C. Kier, W. Walker, I. (2008). "Soft robotics: Biological inspiration, state of the art, and future research." *Applied Bionics and Biomechanics* Vol. 5, No. 3, September 2008, 99–117.
- [20] Otake, M. Inaba, M. and Inoue, H. (1999). "Development of a Gel Robot made of Electro-Active Polymer PAMPS Gel." *IEEE SMC conference proceedings*, Oct. 12-15, Tokyo, Japan.
- [21] Cuttino, J.F. Van Dijck, B. Brown, A.M. (2005). "Design and Development of a Toroidal Flexure for Extended Motion Applications." *Precision Engineering* 29: 135-145.
- [22] Keller, J.B. and Falkovitz, M.S. (1983). "Crawling of worms." *Journal of theoretical Biology* 104: 417-442.
- [23] Chen, G. Pham, M.T., Redarce, T. (2009). "Sensor-based guidance control of a continuum robot for a semi-autonomous colonoscopy." *Robotics and Autonomous Systems* 57 712-722.
- [24] Walker, I. et al. (2005). "Continuum Robot Arms Inspired by Cephalopods." *Unmanned Ground Vehicle Technology VII*. Proceedings from SPIE Vol. 5804.
- [25] Liang, Y. McMeeking, R.M. Evans, A.G. (2006). "A Finite Element Simulation Scheme for Biological Muscular Hydrostats." *Journal of Theoretical Biology* 242: 142-150.
- [26] Jones, B. Walker, I (2006). "Kinematics for Multisection Continuum Robots." *IEEE Transactions on Robotics*, VOL. 22, NO. 1.
- [27] Otsuka, K. and Wayman, C.M. (1998). "Shape Memory Materials." Cambridge University Press, United Kingdom.
- [28] Zhang, J., Modi , V.J., and de Silva, C.W., (2002). "Modal Control with Intelligent Tuning for a Deployable Manipulator." *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, Pgs. 154 – 159.

- [29] Starkey, J.M. (1984). "Eigenvector Modification through Design Changes." *American Control Conference*, pg 1013 – 1018.
- [30] Franklin, G. and Powell, J. (2006). "Feedback Control of Dynamic Systems." Pearson Prentice Hall, New Jersey.
- [31] Ueda, J., and Yoshikaw, T. (2004). "Mode Shape Compensator for Improving Robustness of Manipulator Mounted on Flexible Base." *IEEE Transactions on Robotics and Automation*. Volume: 20, Issue: 2 ,Page(s): 256 – 268.
- [32] Van Griethuijsen, L. I. and Trimmer, B. A. (2009). "Kinematics of Horizontal and Vertical Caterpillar Crawling." *J. Exp. Biol.* 212,1455.
- [33] VanGriethuijsen L, Trimmer B. (2010). "Caterpillar crawling over irregular terrain: anticipation and local sensing." *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 2010; 196: 397-406.
- [34] Lin, H.-T., Slate, D., Paetsch, C., Dorfmann, L. and Trimmer, B. (2011). "Ontogenetic scaling of caterpillar body properties and its biomechanical implications on the use of hydrostatic skeletons." *Journal of Experimental Biology* 214, pg. 1194-1204.
- [35] Eaton, J. L. (1988). "*Lepidopteran Anatomy*:" Wiley New York.
- [36] Levine RB, Truman JW. (1985). "Dendritic reorganization of abdominal motoneurons during metamorphosis of the moth, *Manducasexta*." *J. Neurosci.*, 1985; 5: 2424-31.
- [37] Sandstrom DJ, Weeks JC. (1996). "Novel dual innervation of a larval proleg muscle by two similar motoneurons in the tobacco hornworm *Manducasexta*." *J. Exp. Biol.*, 1996; 199: 775-91.
- [38] Snodgrass RE. (1961). "The Caterpillar and the Butterfly." *Smithsonian Miscellaneous Collections*, 1961; 143: 51.

- [39] Weeks J, Ernst-Utzschneider K. (1989). “Respecification of larval proleg motoneurons during metamorphosis of the tobacco hornworm, *Manduca sexta*: segmental dependence and hormonal regulation.” *J. Neurobiol.*, 1989; 20: 569-92.
- [40] Belanger JH, Trimmer BA. (2000). “Combined kinematic and electromyographic analyses of proleg function during crawling by the caterpillar *Manduca sexta*.” *J Comp Physiol [A]*, 2000; 186: 1031-9.
- [41] Simon MA, Fusillo SJ, Colman K, Trimmer BA. (2010). “Motor patterns associated with crawling in a soft-bodied arthropod,” *J. Exp. Biol.*, 2010; 213: 2303-9.
- [42] Dorfmann A, Trimmer BA, Woods WA, Jr. (2007). “A constitutive model for muscle properties in a soft bodied arthropod.” *J. R. Soc. Interface*, 2007; 4: 257-69.
- [43] Dorfmann A, Woods Jn WA, Trimmer BA. (2008). “Muscle performance in a soft-bodied terrestrial crawler: constitutive modelling of strain-rate dependency.” *J. R. Soc. Interface*, 2008; 5: 349-62.
- [44] Lin HT, Dorfmann AL, Trimmer BA. (2009). “Soft-cuticle biomechanics: A constitutive model of anisotropy for caterpillar integument.” *J. Theor. Biol.*, 2009; 256: 447-57.
- [45] Trimmer, B.A. and Issberner, J. (2007). “Kinematics of Soft-Bodied, Legged Locomotion in *Manduca sexta* Larvae.” *The Biology Bulletin* 212: 130-142.
- [46] Ijspeert, A.J. (2001). “A Connectionist Central Pattern Generator for the Aquatic and Terrestrial Gaits of a Simulated Salamander.” *Biological Cybernetics* 84: 331-348.
- [47] Ostrowski, J., and Burdick, J. (1996). “Gait Kinematics for a Serpentine Robot.” *IEEE International Conference on Robotics and Automation*. Volume 2:1294-1299.

- [48] Ray, T. and Buller, A. (2005). "Automated Evolutionary Design, Robustness, and Adaptation of Side winding Locomotion of a Simulated Snake-Like Robot." *IEEE Transactions on Robotics*, Vol. 21, No. 4, August 2005.
- [49] Hochner and Tamar Flash, Yoram Yekutieli, Roni Sagiv-Zohar, Ranit Aharonov, Yaakov Engel, Binyamin (2005). "Dynamic Model of the Octopus Arm. I. Biomechanics of the Octopus Reaching Movement." *J Neurophysiol* 94:1443-1458, 2005.
- [50] Full, R., and Koditschek, D. (1999). "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, 202(23): 3325-3332.
- [51] Saranli, U. and Koditschek, D. (2003). "Template based control of hexapedal running." *International Conference on Robotics and Automation*, p. 1374-1379.
- [52] Kim, S., Clark, J.E. and M.R. Cutkosky. (2006). "iSprawl: design and tuning for high-speed autonomous open-loop running." *International Journal of Robotics Research*. 25: p. 903-912.
- [53] Carerra, E. and Serna, M. (1996). "Inverse Dynamics of Flexible Robots." *Mathematics and Computers in Simulation* 41 485-508.
- [54] Saunders, F. et al. (2009). "Experimental Verification of Soft-Robot Gaits Evolved Using a Lumped Dynamic Model." *Robotica*. Vol. 29, Issue 6.
- [55] Hedrick, T. (Jan. 2009. Jan. 2010). The Hedrick Lab at UNC Chapel Hill
<<http://www.unc.edu/~thedrick/software1.html>>
- [56] Craig, J. (1989). "Introduction to Robotics: Mechanics and Control, 2nd Edition." Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
- [57] Simon MA, Trimmer BA. (2009). "Movement encoding by a stretch receptor in the soft-bodied caterpillar, *Manduca sexta*." *J. Exp. Biol.*, 2009; 212: 1021-31.

- [58] Saunders, F., Rieffel, J. and Rife, J. (2009). "A method of accelerating convergence for genetic algorithms evolving morphological and control parameters for a biomimetic robot." *Proc. International Conference on Autonomous Robots and Agents (ICARA)*.
- [59] Khalil, H. (2001). "Nonlinear Systems (3rd edition)." Prentice Hall, 2001.
- [60] Clark, J., Cham, J., Bailey, S., Froehlich, E., Nahata, P., Full, R. and Cutkosky, M. (2001). "Biomimetic Design and Fabrication of a Hexapedal Running Robot." *Proceedings of the 2001 IEEE International Conference on Robotics*, Seoul, Korea. May 21-26, 2001.
- [61] Saunders, F. (2009). "Simulation of Soft Bodied Robots: Methods, Benefits, and Potential Applications" Master's thesis, Tufts University.
- [62] Bachelier, O., LAII, Poitiers, Bosche, J. and Mehdi, D. (2006). "On Pole Placement via Eigenstructure Assignment Approach." *IEEE Transactions on Automatic Control* Vol 51. Pg. 1554-1558.
- [63] Yunyun, D. (2008). "Energy-Based Control for a Class of Under-Actuated Mechanical Systems" Congress on Image and Signal Processing, 2008. CISP '08. Vol. 3 pg. 139/143.
- [64] Saunders, F. and Rife, J. (submitted, 2012). "Modal Eigenstate Determination for Reoccurring Dynamics." *International Journal of Controls*.
- [65] Bretscher, O. (2009). "Linear Algebra with Applications." Pearson.
- [66] Alhan, C., Gavin, H. and Aldemir, U., (2006), "Optimal Control: Basis for Performance Comparison of Passive and Semiactive Isolation Systems." *Journal of Engineering Mechanics*, Vol. 132, No. 7, pp. 705-713.
- [67] Zhu, D., Tudor, J. and Beeby, S., (2010), "Strategies for increasing the operating frequency range of vibration energy harvesters: a review." *Measurement Science and Technology*, 21, (2), 022001.

- [68] Hung, M. et al., (1997), "A New Eigenvalue Placement Method for Linear Dynamic Systems." *Proceedings of the 36th Conference on Decision & Control*, Pg 4286-4287.
- [69] Badran, S.M. and Choudhry, M.A., (1993), "Design of an Optimal controller with Prescribed Eigenvalues for Power System." *System Theory*, pg 140 – 144.
- [70] Olbrot, A.W., (1994), "Arbitrary Robust Eigenvalue Placement by a Static-State Feedback", *IEEE Transactions on Automatic control*, VOL. 41, NO. 8.
- [71] Duan, G. and Yu, H., (2006), "Complete Eigenstructure Assignment in High-Order Descriptor Linear Systems Via Proportional Plus Derivative State Feedback." *Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China.*
- [72] Wang, G., Lv, Q. and Duan, G., (2008), "PID Eigenstructure Assignment in Second-order Dynamic Systems: A Parametric Method." *Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China.*
- [73] Choi, J., Lee, J., Kim, Y. and Kang, T. (1995), "A design methodology using concurrent eigenstructure assignment." *Proceedings of the 34th Conference on Decision and Control*, New Orleans, LA.
- [74] S. Umar. (1986). "Eigenvalue/Eigenvector Assignment Using Output Feedback" *IEEE Transactions on Automatic Control*. Volume: 23 , Issue: 1 . Page(s): 79 - 81.
- [75] Tran, M. and Sawan, T., (1986), "Controller Design with Eigenvalue and Eigenvector Insensitivity to Parameter Variations." *American Control Conference*, pg 1845.
- [76] Hooke, R., Jeeves, T.A., (1961), "Direct search solution of numerical and statistical problems." *Journal of the Association for Computing Machinery*, (ACM) 8 (2): 212–229.
- [77] Papalambros, P. and Wilde, D., (2000), *Principles of Optimal Design*, Cambridge University Press.

[78] www.mathworks.com

[79] Lipson, H. (2005) "Evolutionary Design and Evolutionary Robotics", *Biomimetics*, CRC Press (Bar Cohen, Ed.) pp. 129-155.

[80] Lipson, H. and Pollack J. B. (2000). "Automatic Design and Manufacture of Artificial Lifeforms", *Nature* 406, pp. 974-978.

[81] Rieffel, J., Valero-Cuevas, F. and Lipson, H. (2010). "Morphological Communication: Exploiting Coupled Dynamics in a Complex Mechanical Structure to Achieve Locomotion". *J. R. Soc. Interface*.

[82] Knox, D., and Rieffel, J (2011). "Scalable co-evolution of soft robot properties and gaits". *Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems (ECAL)*. MIT Press. pg. 416-422.

[83] Hung, M. and Sehitoglu, H. (1997). "A New Eigenvalue Placement Method for Linear Dynamic Systems." *Proceedings of the 36th Conference on Decision & Control*. Pg 4286-4287.