

**THE EFFECTS OF SPATIAL AND TEMPORAL PROPERTIES ON A  
VISCOELASTIC MODEL OF THE DYSSYNCHRONOUS  
CANINE HEART**

A Thesis  
Submitted to the Graduate Faculty  
of the  
North Dakota State University  
of Agriculture and Applied Science

By

Cody Michael Satterlee

In Partial Fulfillment of the Requirements  
for the Degree of  
MASTER OF SCIENCE

Major Department:  
Electrical and Computer Engineering

April 2011

Fargo, North Dakota

North Dakota State University  
Graduate School

---

Title

The Effects of Spatial and Temporal Properties

---

On a Viscoelastic Model of the Dyssynchronous Canine Heart

---

By

Cody Satterlee

---

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

**MASTER OF SCIENCE**

---

## North Dakota State University Libraries Addendum

To protect the privacy of individuals associated with the document, signatures have been removed from the digital version of this document.

## ABSTRACT

Satterlee, Cody Michael, MS, Department of Electrical and Computer Engineering, College of Engineering and Architecture, North Dakota State University, April 2011. The Effects of Spatial and Temporal Properties on a Viscoelastic Model of the Dyssynchronous Canine Heart. Major Professor: Dr. Dan Ewert.

In this study, lumped parameter cardiovascular modeling has been used to understand the influence of muscle properties on mechanical dyssynchrony (MD) as well as general muscle dynamics. Incorporating viscous influence into the model allowed for an expanded view when analyzing muscle parameter response to MD. A unique method of ventricle segmentation was introduced that allowed fast analysis of regional and global ventricular properties. This segmentation process produced a ventricle with four identical sections each consisting of separately tunable muscle properties in the form of minimum and maximum elastance, elastance waveform delay, and myocardial viscous friction, yet these regional sections remained globally dependent. Elastance waveform delay proved to be the most influential property on MD as measured by internal flow fraction (IFF), followed by regional elastance magnitude, and finally regional viscosity influence. Due to the unique segmentation of this model, two metrics for IFF were derived; 1) the “true” IFF (IFF-4seg) and 2) the IFF as would be measured by an ideal conductance catheter (IFF-CC). The results of IFF-CC versus IFF-4seg show that conductance catheters are not capable of measuring IFF during a side-to-side volume transfer within the stacked cylinder under measurement. Finally, unique energetic situations were observed with this model that point to likely myocardium remodeling situations.

## **ACKNOWLEDGMENTS**

I would like to express my sincere gratitude to all of the people who have helped and inspired me during my master's study.

Foremost, I want to thank my advisor, Dr. Dan Ewert, for his enthusiastic approach to teaching, which I was initially exposed to several years before my master's study began. His patience, motivation, and willingness to help all of his students went far beyond what is to be expected. Just as noteworthy was his accessibility for advising despite his acceptance of a position at another university. As a result, my research was as smooth as can be expected.

I would like to thank the rest of my advisory committee: Dr. Larry Mulligan, Dr. Jake Glower, and Dr. Mark Sheridan. Dr. Mulligan was instrumental in my research through his knowledge of physiology which helped focus my attention and ensure the accuracy in my writings. Early in my research and testing, Dr. Glower suggested a new approach in my work that ended up saving me countless hours of trial and error. My enjoyment and understanding of physiology would not have been possible without the instruction of Dr. Sheridan.

My deepest gratitude goes to my family for their strong support and unconditional love throughout my life. Their unflappable encouragement and leadership cannot be overstated. Their generosity and understanding in times of need has helped propel me to new objectives in life.

My internship at Covidien in Boulder, CO, introduced me to many new ideas and processes in the engineering field that found their mark in my thesis. I thank Jim Orszulak

for hiring me and demonstrating remarkable teaching ability; his friendship and technical guidance are valued.

I would also like to thank Laura Dallmann and Priscilla Schlenker for their kind and gracious help in the ECE office. They are always willing to go the extra mile to help students.

Last but not least, I give thanks to God for my healthy life and motivation to continue pursuing my dreams.

# TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGMENTS.....	iv
LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix
INTRODUCTION.....	1
METHODS.....	6
Model Overview.....	6
Circulatory Model.....	7
Hill Muscle Model.....	8
Mechanics of Purely Elastic Ventricle.....	10
Mechanics of a Visco-Elastic Ventricle.....	16
Simulation and Data Analysis.....	20
Integration Error.....	23
RESULTS.....	25
Model Validation.....	25
Variations in Segmental Elastance.....	28
Elastance Delay.....	30
Multiple Segment Elastance Variations.....	32
Regional Ventricular Remodeling.....	33
Monte Carlo Simulation.....	36
Influence of Muscle Properties on IFF and Hemodynamics.....	39
DISCUSSION.....	50

Conductance Catheter Response.....	53
Limitations.....	54
CONCLUSION.....	55
REFERENCES.....	56
APPENDIX A: ADDITIONAL FIGURE.....	60
APPENDIX B: MATLAB CODE.....	61
Code Generating Single Simulation Waveforms.....	61
Code Generating Multiple Simulation Tests for Elastance and Delay.....	71
Code Generating Multiple Simulation Tests for Elastance and 'k'.....	86
Code Generating Monte Carlo Simulation Data.....	101

# LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Muscle Values.....	21
2. Base Waveform Results.....	27
3. Values Resulting from a Weak Segment A1.....	30
4. 10ms Delay Initiated in Segment A1.....	32
5. Multi-Segment Elastance Variations.....	34
6. Ventricle Remodeling Due to Muscle Parameters.....	36
7. Multiple Simulation Data.....	37
8. Monte Carlo Parameters.....	38
9. Parameters from 10 Lowest Resulting IFF-C.C.....	42
10. Parameters from 10 Highest Resulting IFF-C.C.....	43



# LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Kerckhoffs et al. 2008.....	4
2. Four-Segment Model.....	7
3. Modified Hill Muscle Model (Ewert, 2004).....	9
4. Free-Body Diagram.....	10
5. Illustration of a Purely Elastic Ventricle.....	11
6. Visco-Elastic Ventricle Model.....	17
7. Integration Error.....	24
8. Sample of Baseline Waveforms for LVP, MF, AoF, and VT.....	26
9. PV Loop and Inferior Vena Caval Occlusion.....	28
10. Waveform with Weak Contractility.....	29
11. 10ms Elastance Delay.....	31
12. Multi-Segment Elastance Variations.....	33
13. Waveforms Showing Elastance and Delay Variations.....	35
14. Monte Carlo IFF Results.....	39
15. Monte Carlo Results for EF, $P_{loss}$ , EFF, and $EFF_{mYO}$ .....	40
16. Scatter Plot of IFF-C.C. vs. IFF-4seg.....	41
17. Internal Flow Fraction.....	44
18. Cardiac Output and Ejection Fraction.....	45
19. External Work and Myocardium Work.....	46
20. PVA and $PVA_{mYO}$ .....	47
21. Efficiency and Myocardium Efficiency.....	48
22. Power Loss and Stroke Volume.....	49

23. Six-Segment Heart Model.....	60
----------------------------------	----

# INTRODUCTION

Cardiovascular disease (CVD) is a leading cause of death in the industrialized world. Medical research has led to the development of devices designed to attenuate the impact of CVD. These include ventricular assist devices (VAD), pacemakers, artificial hearts, and defibrillators. Current efforts to improve the impact of these devices on patients continue. Methods to evaluate these devices include human studies, animal studies, and modeling of various system and intracellular structures. However, animal studies are difficult, expensive, and lack the ability to easily manipulate parameters and can have large variation in effects that may be seen due to various biological differences. Previously, cardiovascular modeling has been approached from both an electrical and mechanical standpoint. In 1998, Nebot et al. described a quantitative approach to cardiovascular and central nervous system (CNS) modeling (Nebot, 1998). Others have modeled various aspects of the circulatory system by deriving an electrical analog (Sagawa, 1988; Beneken, 1996; Abdolrazaghi, 2010). Hunter et al. (Hunter, 2003) created a computational framework of the heart to describe the integrated effects of individual myocytes. Nielsen et al. (Nielsen, 1991) constructed a geometric model based on a truncated ellipsoid. Furthermore, transmural geometric modeling has surfaced as a very useful tool with the ability to mathematically describe the epicardial and endocardial surfaces of the heart.

One area of interest within the field of modeling is ventricular mechanical dyssynchrony (VMD). Mechanical dyssynchrony is widely believed to be a strong predictor

of heart failure and a basis for selecting patients for cardiac resynchronization therapy (CRT) (Steendijk, 2003; Lang, 2006; Helm, 2005; Cleland, 2006). VMD leads to reduced cardiac output due to ventricular inefficiencies originating at the whole heart and cellular level. QRS widening has been the key variable used to select patients for CRT. Several publications point to the lack of precision this variable provides (Vollmann, 2006; Gorcsan, 2007). However, CRT has shown to be remarkably successful in treating patients with LBBB and HF, even with the imperfections that come with using QRS. QRS widening can be caused by cardiac remodeling or causes the remodeling: the net effect of mechanical dyssynchrony, hence various responses to CRT can be found (Tang, 2008).

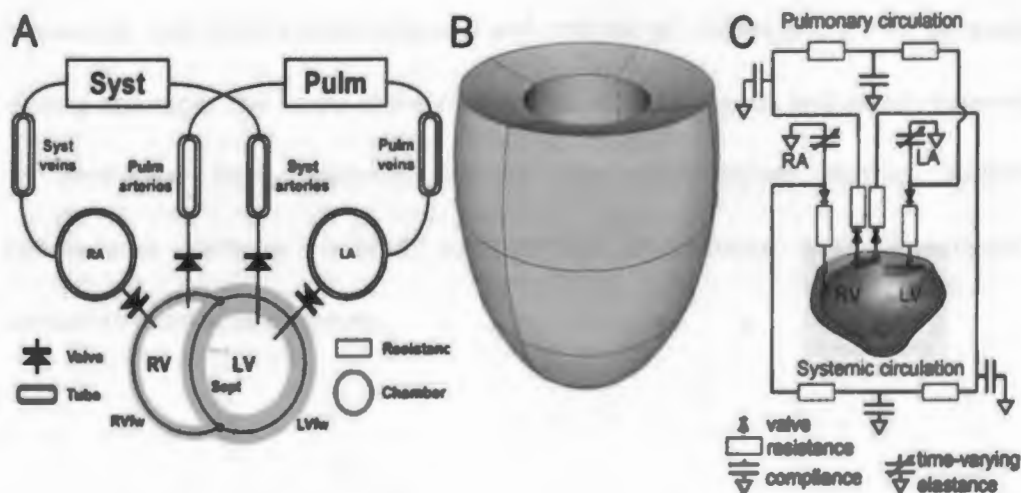
Cardiac resynchronization therapy (CRT) is a form of ventricular pacing for NYHA class II-IV heart failure patients that has shown to restore heart function when a conduction disturbance is present (Cleland, 2006). It has been previously reported that approximately 30% of patients do not respond to conventional CRT (McAlister, 2007). However, response is not a binary variable. To classify a patient as a non-responder due to the absence of improvement in NYHA, EF or quality of life while they are alive at 2 years post implant is short sighted. It is possible that response is lower in patients with ischemic HF or the presence of a myocardial infarct. In summary, the impact of mechanical dyssynchrony on ventricular function continues to be an area of interest and the use of modeling may lead to improvements in the response rate for all patients.

Several methods of analyzing ventricular mechanical dyssynchrony exist. Methods of quantifying mechanical dyssynchrony include tissue Doppler imaging (TDI), magnetic resonance imaging (MRI), speckle tracking, and conductance catheter. Additionally, three

proven indices of mechanical dyssynchrony have been developed for use with conductance catheter-derived data.(Steendijk, 2003) These indices utilize the conductance catheter-derived volume signals based on short-axis slices of the heart and exhibit a radial shortening effect that is quantified by comparative analysis over global volume. According to Steendijk et al. (Steendijk, 2003), these metrics of mechanical dyssynchrony are highly correlated with tissue Doppler-derived septal-to-lateral delay, yet produce an easily quantifiable result.

Previous computational models of mechanical dyssynchrony have been developed. Kerckhoffs et al. (Kerckhoffs R. P., 2003) constructed a three-dimensional finite element model of a canine heart using mechanics based on passive tissue stress, strain, and balancing momentum components. Kerckhoffs' model neglected inertial forces and utilized three-element Windkessel models for the arterial and venous sides. From this model, electromechanics were studied to better understand pacing site timing/myocardial function interactions. Separately, Kerckhoffs et al. (Kerckhoffs R. P., 2003) used a modified geometrical ellipsoid, as shown in Figure 1B, to simplify the myofiber orientation of the above finite element model. The result was used to investigate depolarization delay patterns, for which it was determined that "electromechanical delay times are heterogeneously distributed, such that a contraction in a normal heart is more synchronous than depolarization". Later, a more complete three dimensional finite element model of ventricular mechanics was developed by Kerckhoffs et al. (Kerckhoffs R. M., 2006), as shown in Figure 1C, in which a method for integrating a ventricular model within a complete model of the circulatory system was implemented.

Ventricular interactions were implemented through the use of a time varying elastance model that replaced the finite element model in successive steps, thereby creating a novel multi-scale model of cardiac mechanics. Lumens et al. (Lumens, 2008) published a lumped element model of ventricular mechanics that was implemented within a closed-loop cardiovascular system model coined “CircAdapt” as shown in Figure 1A. Independently, the ventricle was modeled on the basis of three wall segments interacting utilizing balancing laws to define equilibrium at each intersection. The referenced balancing deals with the solution to ventricular pressure and septal geometry, for which was determined through computational sarcomere myofiber stress equations. This model was used to simulate ventricular wall segment mechanics of left bundle branch block (LBBB).



**Figure 1. Kerckhoffs et al. 2008**

When it comes to modeling of biological tissue, visco-elastic properties should be taken into account. This variable has been the source of much controversy when

evaluating the relative importance of viscous influence on muscle dynamics. Authors such as Huxley and Hiedergerke(Huxley A. R., 1954) and Huxley and Hanson(Huxley H. E., 1954) suggests that viscous influence can be omitted due to its negligible effects. However, significant research points to the contrary (Ewert, 2004; Hill, 1938; Goldsmith, 2002; Granzier, 2004; Zhang, 2005). For liquids, Newton's Law may be used to determine the hydrodynamic properties, while Hooke's Law may be used for solid elastic materials. Biological tissue, however, can exhibit characteristics of both materials indicating a visco-elastic approach must be utilized. Ewert et al. (Ewert, 2004) contends that "viscosity estimates have been too readily discounted based on too simplistic a model."

This thesis introduces a ventricular computational model with the capability to analyze the effects of various heart parameters on mechanical dyssynchrony. Regional ventricular interactions were analyzed with the use of viscous and elastic parameters to ensure accuracy. The scope of the model is to allow for a quick and visual representation of ventricular flow, pressure, volume, and dyssynchrony metrics. Additionally, conductance catheter derived dyssynchrony calculations were investigated for comparative analysis purposes.

# METHODS

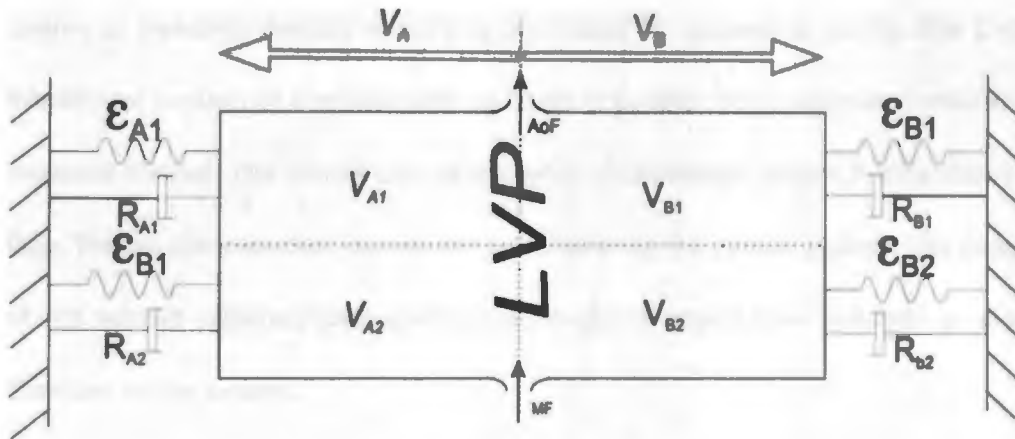
## Model Overview

The left ventricular model, consistent with a lumped parameter approach, is illustrated in Figure 2. The model is constructed on the basis of force balancing at the ventricular wall to analyze mechanical properties of the left ventricle (LV). Previously developed mathematical relationships for hydromotive pressure (HMP), elastance ( $\epsilon$ ), myocardial resistance (R), left ventricular pressure (LVP), LV volume (LVV), and aortic flow (AoF) were used to develop the force generation characteristics of the left ventricle (Ewert, 2004). The left ventricular geometry is mathematically separated into several compartments that are sensitive to regional muscle parameters as well as global heart function. These compartments do not act independent of global LV function, but rather allow for regional analysis of LV mechanics versus global function by creating an imaginary centerline and force balancing at each juncture.

Consistent with the pressure, elastance, volume relationship of a purely elastic ventricle as expressed in Eq. 1, a time-varying elastance waveform is generated consisting of a systolic phase governed by a predefined maximum elastance value and a diastolic phase governed by a predefined minimum elastance value. The elastance waveform can be mapped over any multiple of cycles, resulting in the simulation of the resulting number of cardiac cycles. For this reason, perfect initial values were unnecessary as the model self corrects on each successive loop and eventually reaches steady state.

$$LVP(t) = \epsilon_1(t) * \{V_1(t) - V_0\} \quad (1)$$





**Figure 2. Four-Segment Model.** Consistent with a visco-elastic approach, the model simulates regional and global cardiovascular mechanics using time-varying elastance and myocardial resistance. Letters 'A' and 'B' indicate left and right model segments respectively, while numbers '1' and '2' indicate short axis slices from base to apex respectively.

By separating the LV into compartments, mechanical dyssynchrony (a strong indication of inefficient pump performance) can be investigated by comparing regional to global LV volume using previously established indices influenced by radial contraction (Steendijk, 2003). Although each compartment or segment is self supportive by prescribed muscle parameters, the ventricle is characterized by the interaction of these muscle dynamics with global pressures, volumes, and flows.

### Circulatory Model

Cardiovascular system mechanics were modeled via a two-element Windkessel producing left atrial pressure (LAP) and aortic pressure (AoP) waveforms. For dyssynchrony testing purposes, LAP was kept constant in order to remove an additional

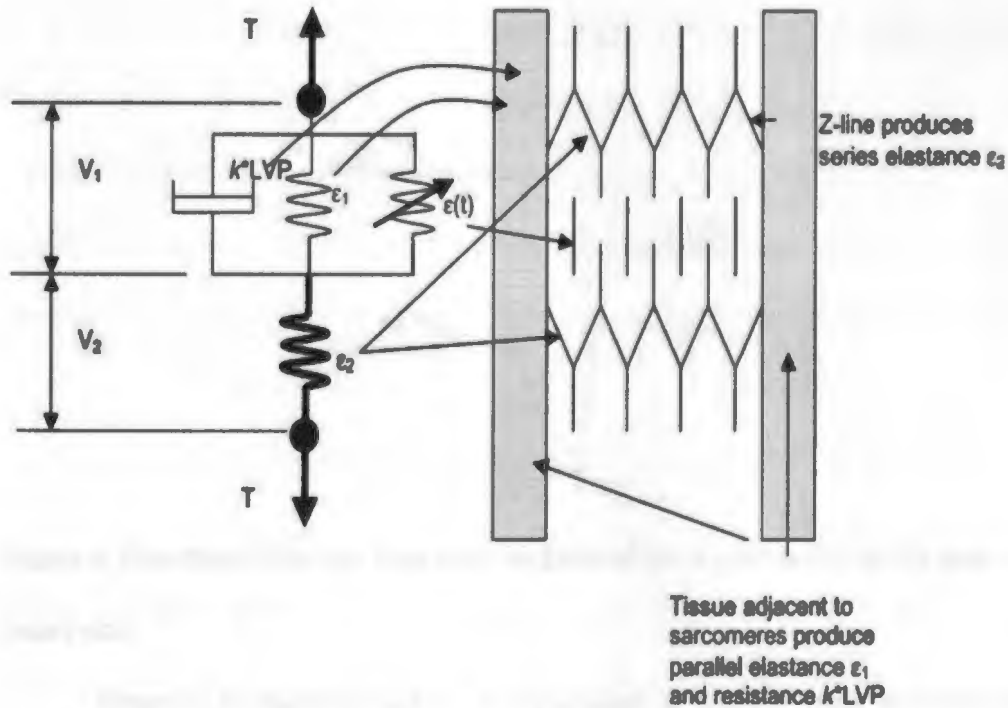
degree of freedom, thereby simplifying the model for conceptual clarity. The 2-element Windkessel consists of a resistor and capacitor in parallel. Total peripheral resistance was modeled through the Ohm's Law relationship of resistance where  $R$  determines blood flow. The parallel capacitor models the compliance of the venous system. The compliance of the venous system allows continuous peripheral blood flow and acts as a storage chamber during systole.

LV valve mechanics for the mitral and aortic valve were modeled by using an exponential variable resistor to simulate the restriction of blood flow during periods of valve closure. Finally, the complete system dynamics were computationally solved in Simulink (MATLAB R2009b) implicitly using the included dormand-prince ODE45 solver.

### **Hill Muscle Model**

The three-element Hill muscle model, popular for its application to skeletal muscle, simulates lumped parameter muscle mechanical response. Several investigators have shown that the Hill model can be adapted to cardiac muscle (Ewert, 2004; Glantz, 1975; Landesberg, 1994). Although there are variations in the modifications used to adapt Hill's muscle model to cardiac muscle, the variation used by (Ewert, 2004) depicted in Figure 3 is implemented in the LV model due to its incorporation of Thevenin equivalence and a variable myocardial friction element.

The benefit of using the modified Hill model is twofold. First, by transforming the mechanical Hill model to an electrical analog, the Thevenin equivalence can combine Sunagawa's impedance relationships, which incorporates kinetic energy loss due to

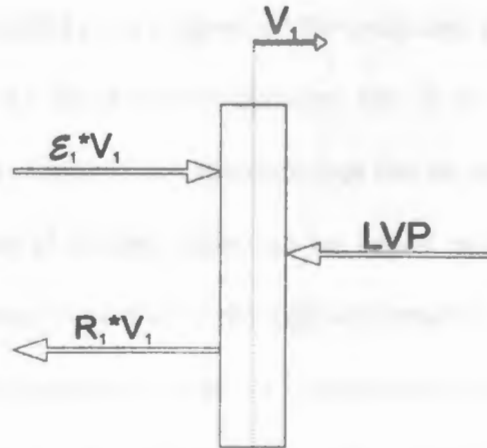


**Figure 3. Modified Hill Muscle Model (Ewert, 2004). Myocardial friction element denoted by a dashpot of resistance  $k \cdot LVP$  indicating a variable resistance of sarcomere tissue.**

elastance and potential energy loss due to myocardial resistance (Sunagawa, 1980). Second, the relationship between HMP and LVP can be defined as the loss due to the combined source impedance from tissue compression and viscous losses. The free-body diagram of Figure 4 represents the sum of force per unit area on the surface of the myocardium.

The end result from balancing these forces is the mathematical relationship in Eq. 2 defined by Ewert et al. (Ewert, 2004).

$$HMP - LVP = \epsilon_1 \cdot V_{EJ} + k \cdot LVP \cdot A_o F \quad (2)$$



**Figure 4. Free-Body Diagram.** Free-body diagram of force interaction at the ventricular heart wall

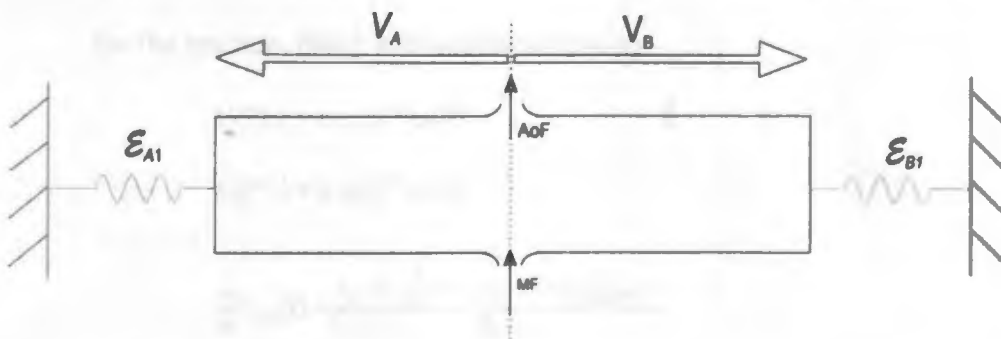
Where  $\epsilon_1$  is elastance and  $V_E$  is the volume of blood ejected from the ventricle during a cardiac cycle. This simplified equation is the result of assuming  $\epsilon_2 \gg \epsilon_1$ , thereby eliminating all terms with  $\epsilon_2$ . For further explanation, the reader is referred to literature (Ewert, 2004).

#### **Mechanics of a Purely Elastic Ventricle**

For clarity purposes, viscous influence on ventricular force dynamics is initially ignored. This allows for focus on force balancing of the ventricle without dealing with computationally expensive viscous influence, thereby simplifying the proof of concept. The relationship between a time-varying elastance  $\epsilon(t)$ , effective volume  $V$ , and left ventricular cavity pressure (LVP) of a purely elastic ventricle is described by Equation 3.

$$LVP(t) = \epsilon(t) * V(t) \quad (3)$$

For conceptual purposes, a diagram of the proposed purely elastic ventricle is shown below in Figure 5. For derivaton purposes one of the stacked slices has been removed to simplify the mathematical approach, but has no other visual or mechanical differences. Any number of stacked slices can be added by mathematically inserting additional muscle elements in parallel to the existing elements. By removing the viscous drag force, a further simplified model is constructed that utilizes well known mathematical relationships to define ventricular mechanics. A direct solution for this one slice model is derived and shown in the proceeding text. Briefly, the mechanics of the ventricle are separated into three phases. An ejection phase is described by equation 4 whereby the change in left ventricular volume  $\{LVV(t)\}$  is directly proportional to aortic flow  $\{AoF(t)\}$ , a filling phase is described by Equation 5 whereby  $LVV(t)$  is directly proportional to mitral flow, and an isovolumic phase that represents isovolumic contraction and isovolumic relaxation of the left ventricle.



**Figure 5. Illustration of a Purely Elastic Ventricle**

The three phases of a purely elastic ventricle are broken down as follows:

$$\text{Ejection: } -\frac{d}{dt}LVV = AoF, \quad \frac{LVP(t) - AoP(t)}{R} = AoF(t) = -\frac{d}{dt}LVV(t) \quad (4)$$

$$\text{Filling: } \frac{LAP - LVP(t)}{R} = \frac{d}{dt}LVV(t) \quad (5)$$

Isovolumic (contraction and relaxation) :

$$\begin{aligned} \frac{d}{dt}LVV(t) &= 0, \quad LVV(t) = V_{A1}(t) + V_{B1}(t), \\ 0 &= \frac{d}{dt}V_{B1}(t) + \frac{d}{dt}V_{A1}(t) \end{aligned} \quad (6)$$

From the diagram in Figure 5, individualized elastance waveforms may be selected to represent regional changes in myocardium parameters resulting from alterations such as heart failure or remodeling. Unbalanced elastance waveforms have the net effect of inversely altering the regional volume waveform as LVP is determined globally. Equation 3 can be manipulated to represent regional parameters as shown below. The derivatives of the two resulting regional equations are subsequently shown below and are useful for solving the dynamics for the ejection, filling, and isovolumic phases.

$$LVP(t) = \epsilon_{A1}(t) * V_{A1}(t) \quad (7)$$

$$LVP(t) = \epsilon_{B1}(t) * V_{B1}(t) \quad (8)$$

$$\frac{d}{dt}V_{A1}(t) = \frac{\epsilon_{A1}(t) * \frac{d}{dt}LVP(t) - LVP(t) * \frac{d}{dt}\epsilon_{A1}(t)}{\epsilon_{A1}^2(t)} \quad (9)$$

$$\frac{d}{dt}V_{B1}(t) = \frac{\epsilon_{B1}(t) * \frac{d}{dt}LVP(t) - LVP(t) * \frac{d}{dt}\epsilon_{B1}(t)}{\epsilon_{B1}^2(t)} \quad (10)$$

## Ejection

The ejection phase consists of the time period when LVP exceeds AoP. The pressure gradient forces the aortic valve open, allowing flow out of the ventricle. When LVP drops below AoP, the aortic valve closes and ejection ceases. In this model, the after-load of the ventricle is produced from a two-element Windkessel model of the arterial system described later in this section. Mathematically, ejection can be related to an electrical analog whereby LVP(t) and AoP(t) are considered voltages and  $\frac{d}{dt}LVV(t)$  is flow. R is the combined resistance of the aortic valve and arterial system during ejection only. Ohm's Law states that  $V/R = I$ ; where V is voltage, R is resistance, and I is current. The flow as a function of time for both sides of the equation must be the same as in the electrical sense, following the law of conservation of energy. The right side of the equation is negative because ejection produces a decreasing left ventricular volume.

$$\text{From above: } \frac{LVP(t) - AoP(t)}{R} = - \frac{d}{dt}LVV(t) \quad (4)$$

Substitute Equations 7  $\rightarrow$  10 into the right hand side of eq. 4 and simplify by solving for  $\frac{d}{dt}LVP(t)$ .

yields  
 $\longrightarrow$

$$\frac{d}{dt}LVP_{\text{ejection}}(t) = \frac{LVP(t) \cdot \left\{ \frac{\frac{d}{dt}\epsilon_{A1}(t)}{\epsilon_{A1}^2(t)} + \frac{\frac{d}{dt}\epsilon_{B1}(t)}{\epsilon_{B1}^2(t)} - \frac{1}{R} \right\} + \frac{AoP(t)}{R}}{\frac{1}{\epsilon_{A1}(t)} - \frac{1}{\epsilon_{B1}(t)}} \quad (11)$$

Integrate equation 11

yields  
 $\longrightarrow$

$$LVP_{\text{ejection}}(t) = \int \frac{LVP(t) \cdot \left\{ \frac{\frac{d}{dt}\epsilon_{A1}(t)}{\epsilon_{A1}^2(t)} + \frac{\frac{d}{dt}\epsilon_{B1}(t)}{\epsilon_{B1}^2(t)} - \frac{1}{R} \right\} + \frac{AoP(t)}{R}}{\frac{1}{\epsilon_{A1}(t)} - \frac{1}{\epsilon_{B1}(t)}} + C \quad (12)$$

Where C is the initial ejection pressure, in this case LVP = AoP at the onset of ejection. Equation 4 is valid during ejection.

### Filling

The filling phase consists of the time period when LVP is less than LAP. In the case of filling, a negative pressure gradient with respect to the ventricle forces the mitral valve open, allowing flow into the ventricle. Similar to the ejection phase, the filling phase can be represented through a relationship between pressure, flow, and resistance. The pressure gradient of LAP – LVP is acting on a mitral valve resistance R. The electrical analog of flow on the left side of Equation 5 must be equal to the change in left ventricular volume  $\{\frac{d}{dt}LVV(t)\}$ . This relationship is only valid during the filling phase.

$$\text{From above: } \frac{LAP - LVP(t)}{R} = \frac{d}{dt}LVV(t) \quad (5)$$

Substitute Equations 7 → 10 into the right hand side of Equation 5 and simplify by solving for  $\frac{d}{dt}LVP(t)$ .

yields  
→

$$\frac{d}{dt}LVP_{\text{filling}}(t) = \frac{LVP(t) \cdot \left\{ \frac{1}{R} - \frac{\frac{d}{dt}\epsilon_{A1}(t)}{\epsilon_{A1}^2(t)} + \frac{\frac{d}{dt}\epsilon_{B1}(t)}{\epsilon_{B1}^2(t)} \right\} - \frac{LAP}{R}}{\frac{1}{\epsilon_{A1}(t)} + \frac{1}{\epsilon_{B1}(t)}} \quad (13)$$

Integrate Equation 13

yields  
→

$$LVP_{\text{filling}}(t) = \int \frac{LVP(t) \cdot \left\{ \frac{1}{R} - \frac{\frac{d}{dt}\epsilon_{A1}(t)}{\epsilon_{A1}^2(t)} + \frac{\frac{d}{dt}\epsilon_{B1}(t)}{\epsilon_{B1}^2(t)} \right\} - \frac{LAP}{R}}{\frac{1}{\epsilon_{A1}(t)} + \frac{1}{\epsilon_{B1}(t)}} + C \quad (14)$$

Where C = LAP, in this case LAP is a steady state preload.



## Isovolumic

The two isovolumic phases consist of the time periods when  $LAP < LVP \leq AoP$  for isovolumic contraction and  $LAP \leq LVP < AoP$  for isovolumic relaxation. During the isovolumic phases, the mitral valve and aortic valve are closed indicating no net flow in or out of the ventricle. The volume within the ventricle is at a maximum during the isovolumic phases, also known as the end diastolic volume. During the isovolumic phases, regional flow back and forth may occur but no net external flow takes place. This is described by Equation 6.

$$\text{From above: } 0 = \frac{d}{dt}LVV(t) = \frac{d}{dt}V_{B1}(t) + \frac{d}{dt}V_{A1}(t) \quad (6)$$

Substitute Equations 7 → 10 into the right hand side of Equation 6 and simplify by solving for  $\frac{d}{dt}LVP(t)$

yields  
→

$$\frac{d}{dt}LVP_{\text{isovol}}(t) = \frac{-LVP(t) \cdot \left\{ \frac{\frac{d}{dt}\epsilon_{A1}(t)}{\epsilon_{A1}^2(t)} + \frac{\frac{d}{dt}\epsilon_{B1}(t)}{\epsilon_{B1}^2(t)} \right\}}{\frac{1}{\epsilon_{A1}(t)} + \frac{1}{\epsilon_{B1}(t)}} \quad (15)$$

Integration of Equation 15

yields  
→

$$LVP_{\text{isovol}}(t) = \int \frac{-LVP(t) \cdot \left\{ \frac{\frac{d}{dt}\epsilon_{A1}(t)}{\epsilon_{A1}^2(t)} + \frac{\frac{d}{dt}\epsilon_{B1}(t)}{\epsilon_{B1}^2(t)} \right\}}{\frac{1}{\epsilon_{A1}(t)} + \frac{1}{\epsilon_{B1}(t)}} + C \quad (16)$$

Where  $C = LAP$  for Isovolumic contraction (or)  $C = AoP$  for Isovolumic relaxation

Add the LVP from 3 phases described above:

$$LVP(t) = LVP_{\text{ejection}}(t) + LVP_{\text{filling}}(t) + LVP_{\text{isovol}}(t)$$

The purely elastic ventricle as described above can be easily constructed in Simulink requiring minimal computational power. The equations are correctly solved by cycling through the correct phase and turning off the unneeded phases according to the variables LAP, LVP, and AoP as defined above for each phase.

### **Mechanics of a Visco-Elastic Ventricle**

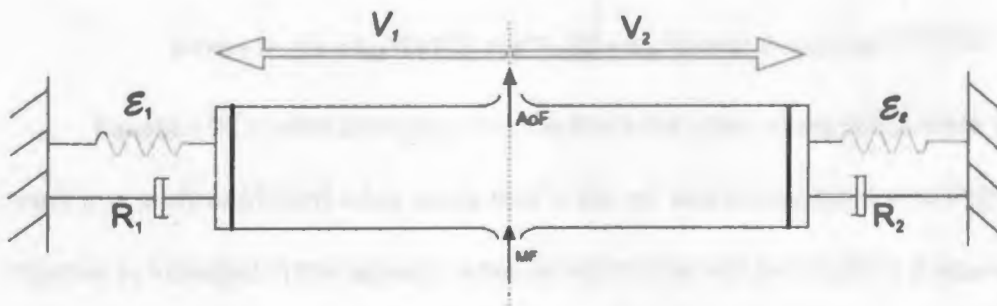
In addition to creating a computational model of the left ventricle with the capability of analyzing mechanical dyssynchrony, this work also aims at addressing the effects of viscous influence on mechanical dyssynchrony. Although computationally expensive, adding viscous influence allows the researcher a more detailed and accurate picture of ventricular dynamics. The relationship between pressure, volume, elastance, and viscous resistance can be represented by Equation 17 (Ewert, 2004).

$$LVP(t) = \epsilon_1(t) * V_1(t) + R_1(t) * \frac{d}{dt} V_1(t) \quad (17)$$

Where  $R_1(t) = K_1 * LVP(t)$ ,  $K_1$  = viscous resistance constant (ml/sec)

A diagram of a visco-elastic ventricle is shown below in Figure 6. Similar to the purely elastic ventricle previously discussed, the visco-elastic ventricle is separated into compartments by an imaginary line that serves the purpose of allowing regional analysis of ventricular dynamics. The addition of the dashpots  $R_1$  and  $R_2$  on either side represent the viscous influence. The dashpots and elastic elements are attached to the heart wall or piston on one side and an immobile wall on the other. As the ventricle fills and ejects, the elastic elements and dashpots are compressed and expanded creating the necessary dynamics to model the proposed equations. Via the diagram, flow enters and leaves as

indicated in the form of mitral flow (MF) and aortic flow (AoF).  $V_1$  and  $V_2$  represent the regional volumes of each respective side that is divided by the proposed centerline. In theory, the centerline is always fixed, however flow from one region to another may occur that does not result in global ejection. This aspect of mechanical dyssynchrony is further analyzed in the discussion as it relates to information obtained by conductance catheter derived data which can not quantify this type of heart movement.



**Figure 6. Visco-Elastic Ventricle Model**

Unlike the purely elastic ventricle described earlier, the visco-elastic ventricle is best solved using a variable resistance for the mitral valve and aortic valve. In this way, one set of equations can be written that describes all three phases of ventricular dynamics. During isovolumic and filling phases, the aortic valve will be represented by a high resistance, while during ejection the aortic valve will be represented by a low resistance that progressively decreases simulating an opening valve. The mitral valve is represented by a high resistance during isovolumic phases and during ejection, while during filling, the mitral valve is represented by a low resistance of progressively decreasing value.

Equations governing a visco-elastic ventricle:

$$LVV(t) = V_{A1}(t) + V_{B1}(t) \quad (18)$$

$$LVP(t) = \epsilon_{A1}(t) * V_{A1}(t) + R_{A1}(t) * \frac{d}{dt} V_{A1}(t) \quad (19)$$

$$LVP(t) = \epsilon_{B1}(t) * V_{B1}(t) + R_{B1}(t) * \frac{d}{dt} V_{B1}(t) \quad (20)$$

$$MF(t) - AoF(t) = \Delta LVV(t) \quad (21a)$$

Where  $R_{A1}(t) = K_{A1} * LVP(t)$  and  $R_{B1}(t) = K_{B1} * LVP(t)$  in units of  $\frac{\text{mmHg} \cdot \text{sec}}{\text{cc}}$

Equation 21 is valid throughout the cardiac cycle since during filling mitral flow will result in a positive  $\Delta LVV(t)$  while aortic flow is cut off and during ejection  $\Delta LVV(t)$  will be negative as indicated in the equation whereas mitral flow will be cut off. It is interesting to note that while there is no global volume change during isovolumic phases, inter-segmental volume changes can occur as indicated from Equation 18. Equation 21a now becomes the basis for governing all flow characteristics of the ventricular model as well as forming the foundation for simultaneously solving Equations 18-21.

Solve:

$$MF(t) - AoF(t) = \frac{d}{dt} LVV(t) \quad (21b)$$

$$\frac{LAP - LVP(t)}{R_{mit}(t)} - \frac{LVP(t) - AoP(t)}{R_{Ao}(t)} = \frac{d}{dt} V_{A1}(t) + \frac{d}{dt} V_{B1}(t) \quad (22)$$

$$\frac{LAP - LVP(t)}{R_{mit}(t)} - \frac{LVP(t) - AoP(t)}{R_{Ao}(t)} = \frac{LVP(t) - \epsilon_{A1}(t) * V_{A1}(t)}{R_{A1}(t)} + \frac{LVP(t) - \epsilon_{B1}(t) * V_{B1}(t)}{R_{B1}(t)} \quad (23)$$

Implicitly solve for  $LVP(t)$  and simplify:

$\xrightarrow{\text{yields}}$

$$LVP(t) = \frac{\frac{LAP}{R_{mit}(t)} + \frac{AoP(t)}{R_{Ao}(t)} + \frac{\epsilon_{A1}(t) \cdot V_{A1}(t)}{R_{A1}(t)} + \frac{\epsilon_{B1}(t) \cdot V_{B1}(t)}{R_{B1}(t)}}{\frac{R_{A1}(t) \cdot R_{mit}(t) \cdot R_{Ao}(t) + R_{B1}(t) \cdot R_{mit}(t) \cdot R_{Ao}(t) + R_{mit}(t) \cdot R_{A1}(t) \cdot R_{B1}(t) + R_{A1}(t) \cdot R_{B1}(t) \cdot R_{Ao}(t)}{R_{A1}(t) \cdot R_{B1}(t) \cdot R_{mit}(t) \cdot R_{Ao}(t)}} \quad (24)$$

Where  $R_{mit}(t)$  is defined as a decaying exponential function:  $R_{mit}\{LAP-LVP(t)\}$  and  $R_{Ao}(t)$  is defined as a decaying exponential function:  $R_{Ao}\{LVP(t)-AoP(t)\}$

$$R_{mit}(t) = 0.05 * e^{-\left(\frac{LAP-LVP}{10}\right)} + 0.1 \quad (25)$$

$$R_{Ao}(t) = 0.05 * e^{-\left(\frac{LVP-AoP}{10}\right)} + 0.1 \quad (26)$$

One of the main goals of this work is to investigate the viability of using a conductance catheter to extract data for assessing mechanical dyssynchrony. A conductance catheter separates the ventricle into several stacked cylinders from apex to base. To accurately represent this feature, the model as described in Figure 6 is insufficient. An additional “layer” or slice must be added to the model to accurately represent the stacked cylinder functionality of conductance catheter derived data as simply as possible. Such a model was previously shown in Figure 2. Similar to Equations 18-21, but varying in parameter values are Equations 27-29 shown below:

$$LVV(t) = V_{A1}(t) + V_{B1}(t) + V_{A2}(t) + V_{B2}(t) \quad (27)$$

$$LVP(t) = \epsilon_{A2}(t) * V_{A2}(t) + R_{A2}(t) * \frac{d}{dt} V_{A2}(t) \quad (28)$$

$$LVP(t) = \epsilon_{B2}(t) * V_{B2}(t) + R_{B2}(t) * \frac{d}{dt} V_{B2}(t) \quad (29)$$

While not directly provided here, the solution set to the model described in Figure 2 closely relates to Equation 24 previously derived, with the difference being two

additional sets of equations (Equation 28 and Equation 29) that are combined in parallel with the previous model. As before, Equation 21 is manipulated to include all four segments of the new model. The end result is a ventricular model with four uniquely independent chambers that all contribute equally to global performance. From this point forward, all reference will be based on the model represented in Figure 2.

### **Simulation and Data Analysis**

The complete system dynamics were computationally solved in Simulink (MATLAB R2009b) implicitly using the included dormand-prince ODE45 solver. Maximum integration step size was set at  $10^{-5}$ . The multi-segment visco-elastic model was simulated on a 64-bit Windows based platform with an AMD Phenom II X3 720 processor and 8.0GB of RAM. Average simulation time including all analysis computations was 1.1 min for a total of 9 consecutive beats. In comparison, the coupled FE model constructed by Kerckhoffs et al. required a simulation time of 156h and 25 min to complete 30 beats of data using a Linux based platform and 3.2 GHz Intel Pentium 4 processor (Kerckhoffs R. M., 2006). Baseline muscle parameters used in simulation are listed in Table 1.

Various hemodynamic and two mechanical dyssynchrony metrics were incorporated through Matlab to form the basis for this analysis. Cardiac output (CO), ejection fraction (EF), external work (EW), pressure volume area (PVA), efficiency (EW/PVA), stroke volume (SV), and internal flow fraction calculated in two variations where the standard metrics. Three new global metrics were incorporated to illustrate key concepts related to mechanical dyssynchrony. Those key metrics include work done by

the myocardium ( $W_{myo}$ ), power lost by the myocardium ( $P_{loss}$ ), pressure volume area of the myocardium ( $PVA_{myo}$ ), and myocardium efficiency ( $EW/PVA_{myo}$ ).

**Table 1. Muscle Values**

Variable	Value	Units
Ea1max	20	mmHg/ml
Eb1max	20	mmHg/ml
Ea2max	20	mmHg/ml
Eb2max	20	mmHg/ml
Ea1min	0.85	mmHg/ml
Eb1min	0.85	mmHg/ml
Ea2min	0.85	mmHg/ml
Eb2min	0.85	mmHg/ml
ka1	0.006	s/L
kb1	0.006	s/L
ka2	0.006	s/L
kb2	0.006	s/L

All of the initial eight basic metrics with the exception of internal flow fraction are self explanatory from conventional cardiovascular research methods. Internal flow is a novel index of mechanical dyssynchrony first identified by Steendijk et al. (Steendijk, 2003) to describe the “non-uniform contraction and filling associated with ineffective shifting of blood volume within the LV”. Internal flow fraction (IFF) is calculated by integrating Equation 30 over the cardiac cycle and dividing by the integrated total flow (AoF + MF).

$$IFF(t) = \frac{\left| \sum \frac{dV_{seg,i}(t)}{dt} \right| - \left| \frac{dV_{LV}(t)}{dt} \right|}{2} \quad (30)$$

As indicated earlier, IFF was defined through two variations. As a conductance catheter segments the left ventricle into several stacked cylinders, the same approach

was utilized to closely mimic the results of such analysis and such calculations will be referred to as IFF-C.C. In reference to Figure 2, volumes  $V_{A1}$  and  $V_{B1}$  were summed to account for “segment 1” of a conductance catheter approach, and volumes  $V_{A2}$  and  $V_{B2}$  were summed to account for “segment 2”. The end result mimics a stacked cylinder approach for analyzing the volume waveforms for a cardiac cycle. The 2<sup>nd</sup> variation of IFF included treating all four volume compartments as individual segments and will be referred to as IFF-4seg in this work for clerical purposes. Both variations utilize Equation 30 and are otherwise identical.

$W_{myo}$  is calculated by integrating  $LVP \cdot \frac{dV_{seg,i}}{dt}$  and summing together all segments for only time points where that segment was ejecting as indicated by a segmental volume derivative  $< 0$ . The result closely mimics EW, but incorporates work due to internal flows between segments and eliminates diastolic work.  $P_{loss}$  is defined as the power lost due to myocardium losses and can be correlated in the electrical sense to  $I^2 \cdot R$  losses. All calculations for  $W_{myo}$ ,  $P_{loss}$ , and  $PVA_{myo}$  considered a four-segment analysis.

$$P_{loss} = \sum_{i=1}^{B2} \int_{LVP > AOP}^{AOP \geq LVP} k_{seg,i} \cdot LVP(t) \cdot \left\{ \frac{dV_{seg,i}}{dt}(t) \right\}^2 \quad (31)$$

$PVA_{myo}$  is calculated through the following equation:

$$PVA_{myo} = PW + P_{loss} + W_{myo} \quad (32)$$

,where PW is defined as potential work.



### Integration Error

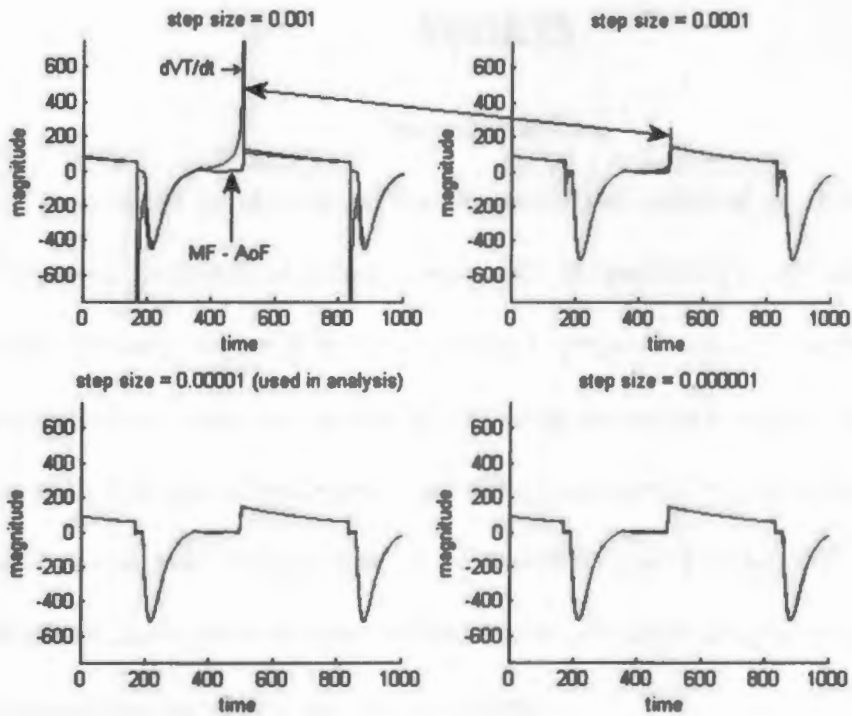
Figure 7 illustrates the integration error inherent in the Simulink model for varying step sizes of the Dormand-Prince solver. The waveform in green represents the sum of all four segmental time derivatives:

$$\text{Green waveform} = \sum_{i=A1}^{B2} \frac{dV_{seg.i}(t)}{dt} = \frac{dVT(t)}{dt} \quad (29)$$

The embedded blue waveform represents the calculated flow as shown in Equation 30 and previously derived from the left hand side of Equation 21:

$$\text{Blue waveform} = \text{MF} - \text{AoF} \quad (30)$$

The lower two plots represent a decreased step size of sufficient magnitude to indicate symmetry and data tracking of the blue waveform by the green waveform. In the top two plots, inflection points can be seen during transition between mitral flow and aortic flow which could account for undesirable manipulation of dyssynchrony and hemodynamic metrics. For all analysis, a step size of  $10^{-5}$  was used to strike a balance between acceptable flow tracking and computation time as seen in the lower left plot of Figure 7.



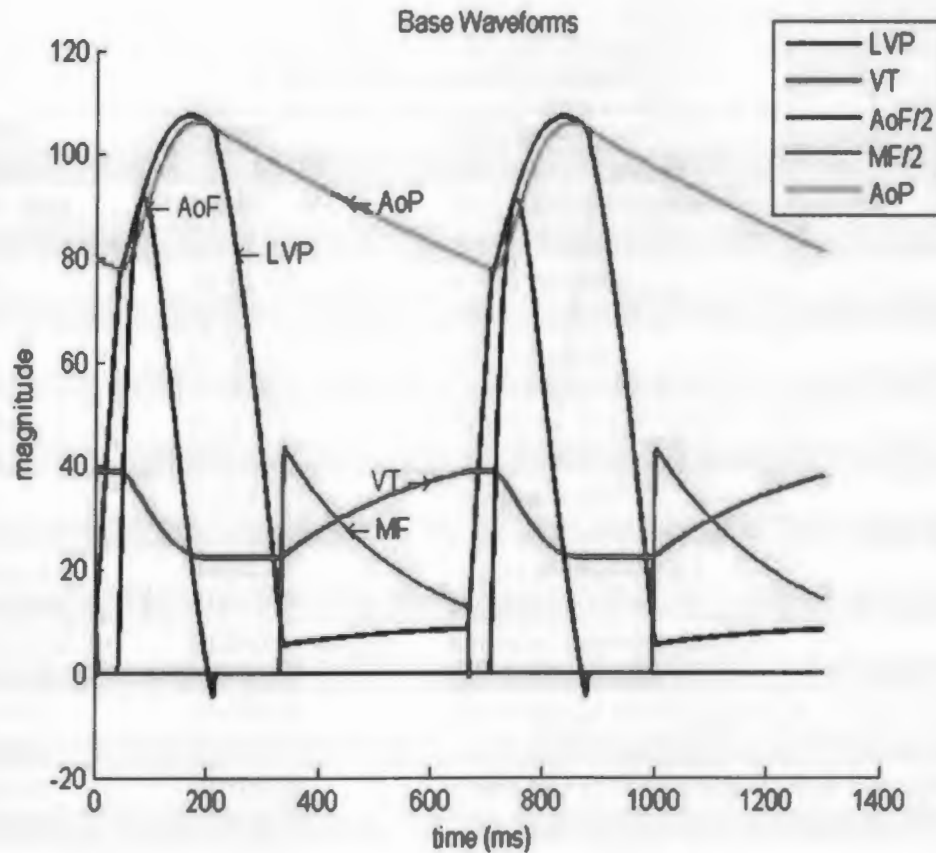
**Figure 7. Integration Error. Progression of step size in Simulink model showing Integration error of the dormand-prince solver**

# RESULTS

## Model Validation

As outlined earlier, the ventricular model was designed to mimic the general hemodynamic properties of a canine heart, with the exception of arterial preload which was kept constant. Figure 8 shows an excerpt of pressure, volume, and flow waveforms that represent an “ideal” canine ventricle of perfectly matched regional elastance values. While atrial kick has not been accounted for in this model, a very small and brief flow reversal can be seen causing aortic valve closure. This small flow reversal is the direct result of the aortic valve equation defined earlier and could possibly be manipulated to simulate varying degrees of valvular regurgitation.

For analysis purposes, initial starting values for pressure and volume were appropriately matched and tuned to the heart model to ensure steady state was reached with sufficient accuracy. Six seconds of data was recorded per simulation at a heart rate of 90bpm, indicating nine beats of data. This standardization was carried throughout testing to ensure consistency between results. A bootstrapping technique of setting a single regional elastance maximum and minimum to zero while all other values remain standardized, and cycling through all four ventricle regions was implemented to ensure consistency of results and accuracy in model calculations. The desired result of the bootstrapping test is to show that all four segments perform identical under similar loading conditions, for which was validated by this test.



**Figure 8. Sample of Baseline Waveforms for LVP, MF, AoF, and VT**

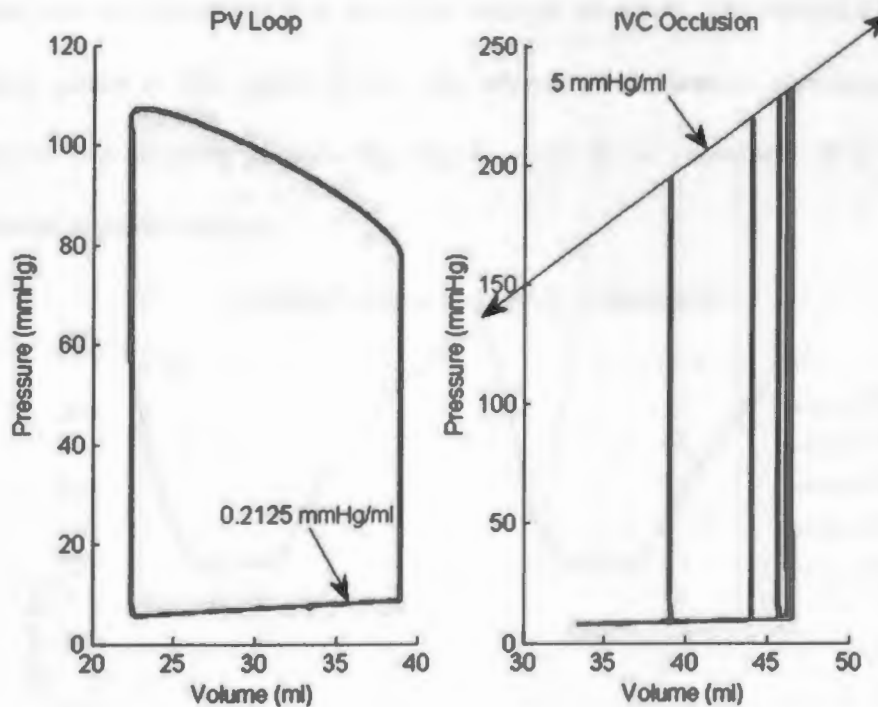
Table 2 presents corresponding data to Figure 8 that lists hemodynamic values calculated from the simulation taken at 90 bpm. Baseline efficiency can be calculated as 57% by the division of external work and pressure volume area. Taking into account myocardium work and power loss, the efficiency drops to 50.5% by the division of external work and  $PVA_{myo}$ . Baseline data represents ideal and perfectly synchronous waveforms using the muscle parameters provided in Table 1. Data collected from animal studies will differ by not containing perfect symmetry of beat-to-beat waveforms and volume at EDV and ESV.

**Table 2. Base Waveform Results**

Metric	Value	Description
IFF - C.C.	0.00%	C.C. modeled Internal Flow Fraction
IFF - 4seg	0.00%	4seg modeled Internal Flow Fraction
CO	1.50 L/min	cardiac output
SV	16.63 ml	stroke volume
EF	42.70%	ejection fraction
EW	0.1997 J	external work
PVA	0.3498 J	pressure volume area
PVAmyo	0.3957 J	PVAmyo = Wmyo + PW + Ploss
Wmyo	0.2164 J	work done by the myocardium
PW	0.1501 J	potential work
Ploss	0.0438 W	power loss of the myocardium
WmyoA1	0.0541 J	Wmyo of segment A1
WmyoB1	0.0541 J	Wmyo of segment B1
WmyoA2	0.0541 J	Wmyo of segment A2
WmyoB2	0.0541 J	Wmyo of segment B2
PlossA1	0.0109 W	power loss of the myocardium in segment A1
PlossB1	0.0109 W	power loss of the myocardium in segment B1
PlossA2	0.0109 W	power loss of the myocardium in segment A2
PlossB2	0.0109 W	power loss of the myocardium in segment B2

A corresponding PV loop and IVC occlusion is shown below in Figure 10. Extrapolation of the ESPVR line shows a slope of 5, correctly deriving the maximum elastance value of this model, as indicated by the parallel combination of  $20 \frac{\text{mmHg}}{\text{ml}}$  for each of four segments. Diastolic elastance is the parallel combination of minimum elastance values, and was found to be  $0.2125 \frac{\text{mmHg}}{\text{ml}}$  as depicted in Figure 10. This value directly relates to the minimum elastance value specified in the Simulink model.

Furthermore, visual inspection of Figure 8 and Figure 9 are in agreement with basic cardiac mechanics. The data collected in Table 2 is agreeable with accepted canine heart behavior, hence, the model is considered valid.

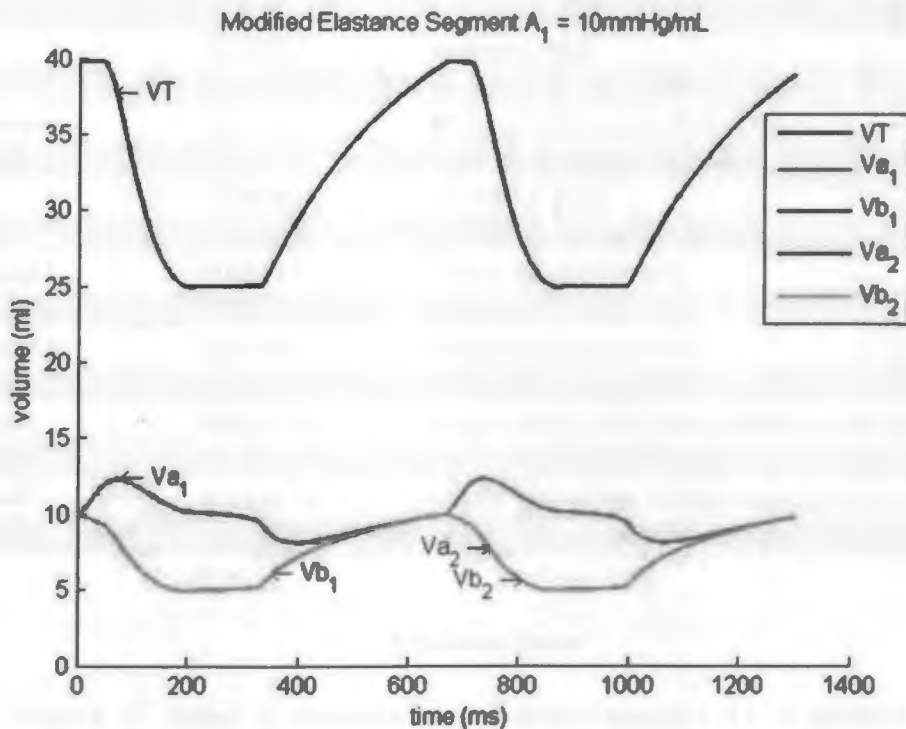


**Figure 9. PV Loop and Inferior Vena Caval Occlusion. Maximum and minimum elastance values can be calculated from the ESPVR and EDPVR lines respectively, and indicates that the model behaves as it should.**

### Variations in Segmental Elastance

Figure 10 shows the resulting elastance and volume waveforms of a dyssynchronous event. The elastance value of segment A1 is set to a value of  $\frac{1}{2}$  that of the other three segments. In this case,  $\epsilon_{A1, \max} = 10 \frac{\text{mmHg}}{\text{ml}}$ , while the remaining three segments have a maximum elastance value of  $20 \frac{\text{mmHg}}{\text{ml}}$ . The diastolic elastance was left uniform at  $.85 \frac{\text{mmHg}}{\text{ml}}$  for all segments. As the elastance element of segment A1 is weaker than the other three, regional ejection of that segment occurs later than the remaining

three due to it receiving flow from the stronger elements. Mechanical dyssynchrony is clearly visible in this region before the volume of the weaker elastance segment A1 returns to a matching path during diastole as would be expected due to the matching diastolic elastance values.



**Figure 10. Waveform with Weak Contractility. Showing segment A1 with a weaker elastance than the other three elements. Upon ejection, the weaker segment initially receives volume until the myofibers are stretched enough to eject itself.**

Values compiled in Table 3 represent the hemodynamic response to the dyssynchronous event shown in Figure 10. In this example, efficiency is reduced to 51.2%, while a further reduction to 42.7% is seen if myocardium work and power losses are taken into account.

**Table 3. Values Resulting from a Weak Segment A1**

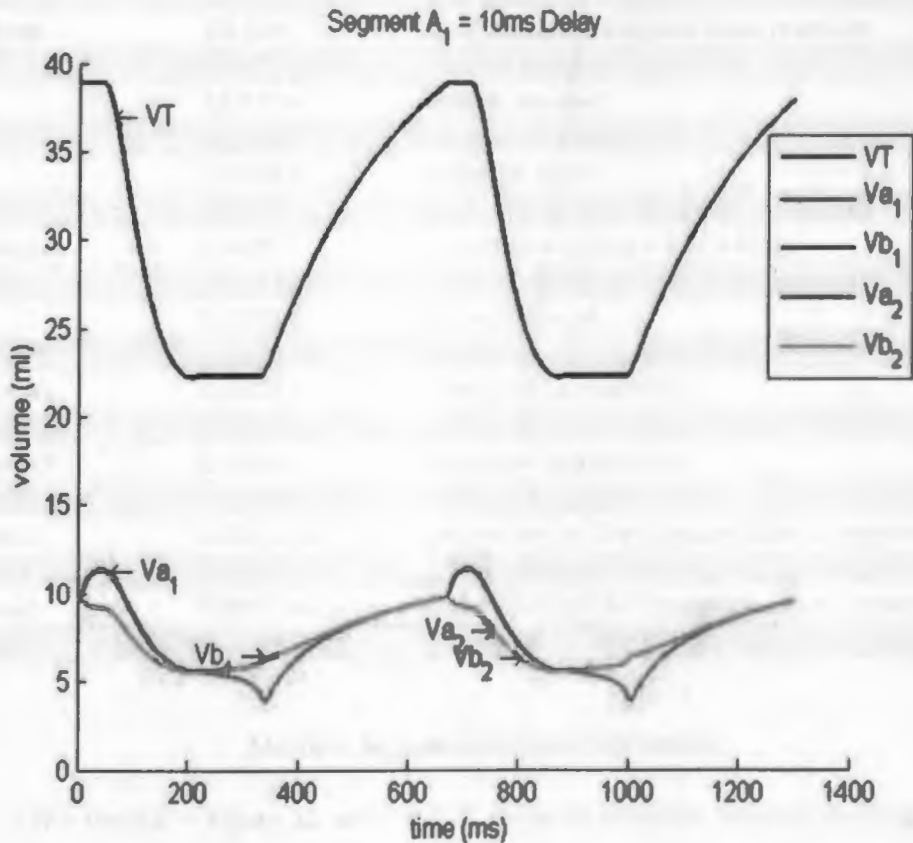
<b>Metric</b>	<b>Value</b>	<b>Description</b>
<b>IFF - C.C.</b>	<b>6.24%</b>	<b>C.C. modeled Internal Flow Fraction</b>
<b>IFF - 4seg</b>	<b>14.79%</b>	<b>4seg modeled Internal Flow Fraction</b>
<b>CO</b>	<b>1.33 L/min</b>	<b>cardiac output</b>
<b>SV</b>	<b>14.8 ml</b>	<b>stroke volume</b>
<b>EF</b>	<b>37.20%</b>	<b>ejection fraction</b>
<b>EW</b>	<b>0.1555 J</b>	<b>external work</b>
<b>PVA</b>	<b>0.3039 J</b>	<b>pressure volume area</b>
<b>PVAmYo</b>	<b>0.3642 J</b>	<b>PVAmYo = Wmyo + PW + Ploss</b>
<b>Wmyo</b>	<b>0.1913 J</b>	<b>work done by the myocardium</b>
<b>PW</b>	<b>0.1484 J</b>	<b>potential work</b>
<b>Ploss</b>	<b>0.0369 W</b>	<b>power loss of the myocardium</b>
<b>WmyoA1</b>	<b>0.0311 J</b>	<b>Wmyo of segment A1</b>
<b>WmyoB1</b>	<b>0.0534 J</b>	<b>Wmyo of segment B1</b>
<b>WmyoA2</b>	<b>0.0534 J</b>	<b>Wmyo of segment A2</b>
<b>WmyoB2</b>	<b>0.0534 J</b>	<b>Wmyo of segment B2</b>
<b>PlossA1</b>	<b>0.0035 W</b>	<b>power loss of the myocardium in segment A1</b>
<b>PlossB1</b>	<b>0.0111 W</b>	<b>power loss of the myocardium in segment B1</b>
<b>PlossA2</b>	<b>0.0111 W</b>	<b>power loss of the myocardium in segment A2</b>
<b>PlossB2</b>	<b>0.0111 W</b>	<b>power loss of the myocardium in segment B2</b>

### **Elastance Delay**

Figure 11 shows a unique situation where segment A1 is delayed by 10ms compared to the elastance waveforms of the remaining segments. The result is similar to the previous example in that due to the slower increase in elastance of segment A1, volume is initially received as ejection is initiated in the other three segments. The point at which segment A1 ejects represents the inflection point for which the sarcomere has stretched enough via the Frank-Starling relationship to balance the force produced by the other segments to the point where all segments now contribute to global ejection.



Although not apparent by inspection, this waveform behavior is demonstrated through Equation 24.



**Figure 11. 10ms Elastance Delay.** A 10ms delay in the elastance waveform of segment A1 results in dyssynchrony during isovolumic contraction, ejection, and during part of the isovolumic filling phase.

The hemodynamic results of Figure 11 are shown in Table 4. Efficiency is calculated to be 57.1%, while 48.6% is realized if myocardium work and power losses are included. Segment A1 produced substantial increases in both  $P_{loss}$  and  $W_{myo}$  due to the delay initiated in that segment.

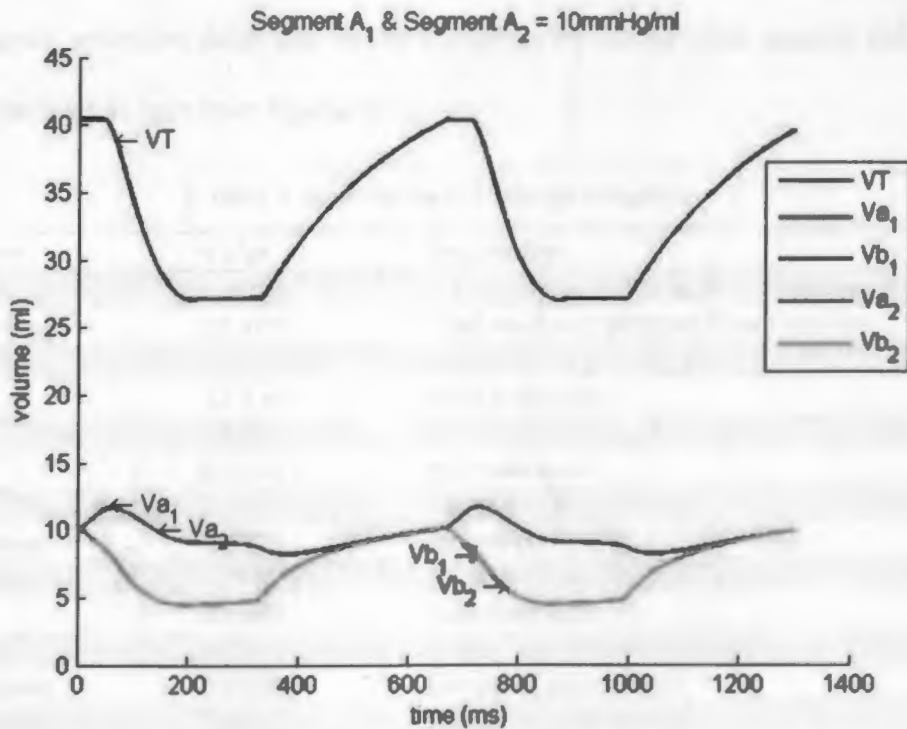
**Table 4. 10 ms Delay Initiated in Segment A1**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
IFF-C.C.	6.34%	C.C. modeled Internal Flow Fraction
IFF-4seg	10.20%	4seg modeled Internal Flow Fraction
CO	1.49 L/min	cardiac output
SV	16.58 ml	stroke volume
EF	42.60%	ejection fraction
EW	0.1985 J	external work
PVA	0.3475 J	pressure volume area
PVAmYo	0.4087 J	PVAmYo = Wmyo + PW + Ploss
Wmyo	0.2294 J	work done by the myocardium
PW	0.1491 J	potential work
Ploss	0.0454 W	power loss of the myocardium
WmyoA1	0.0825 J	Wmyo of segment A1
WmyoB1	0.0489 J	Wmyo of segment B1
WmyoA2	0.0489 J	Wmyo of segment A2
WmyoB2	0.0489 J	Wmyo of segment B2
PlossA1	0.0201 W	power loss of the myocardium in segment A1
PlossB1	0.0085 W	power loss of the myocardium in segment B1
PlossA2	0.0085 W	power loss of the myocardium in segment A2
PlossB2	0.0085 W	power loss of the myocardium in segment B2

### **Multiple Segment Elastance Variations**

The results in Figure 12 and Table 5 show an effective internal sloshing of blood that occurs when one side of the heart is substantially weaker than the other side. The elastance of segment A1 and A2 were initialized to a maximum value of  $10 \frac{\text{mmHg}}{\text{ml}}$ , while the elastance values of B1 and B2 on the right side were kept at  $20 \frac{\text{mmHg}}{\text{ml}}$ . The component of viscous friction,  $k$ , was kept consistent with other simulations at a value of 6 s/L. The situation here is different than the previous simulations in that the top slice of the ventricle containing segments A1 and B1 produces an identical volume waveform to the bottom slice containing segments A2 and B2. In this case, IFF-C.C. was 0.0% while IFF-4seg

was 18.7%. Clearly dyssynchrony exists by viewing the individual segmental volume waveforms as shown in Figure 12.



**Figure 12. Multi-Segment Elastance Variations.**

### Regional Ventricular Remodeling

The situation of Figure 13 and the corresponding data in Table 6 illustrate the possibility of ventricular remodeling in the presence of substantial dyssynchrony. The initial values for maximum elastance of segment A1 was set to  $10 \frac{\text{mmHg}}{\text{ml}}$  as well as being delayed by 40 ms. The combined effects of these two modifications reveals a situation where the sarcomere force produced by segment A1 is in danger of not being able to counterbalance the required pressure to eject. Normally, the heart has many compensatory mechanisms that may prevent the situation in Figure 13 from being

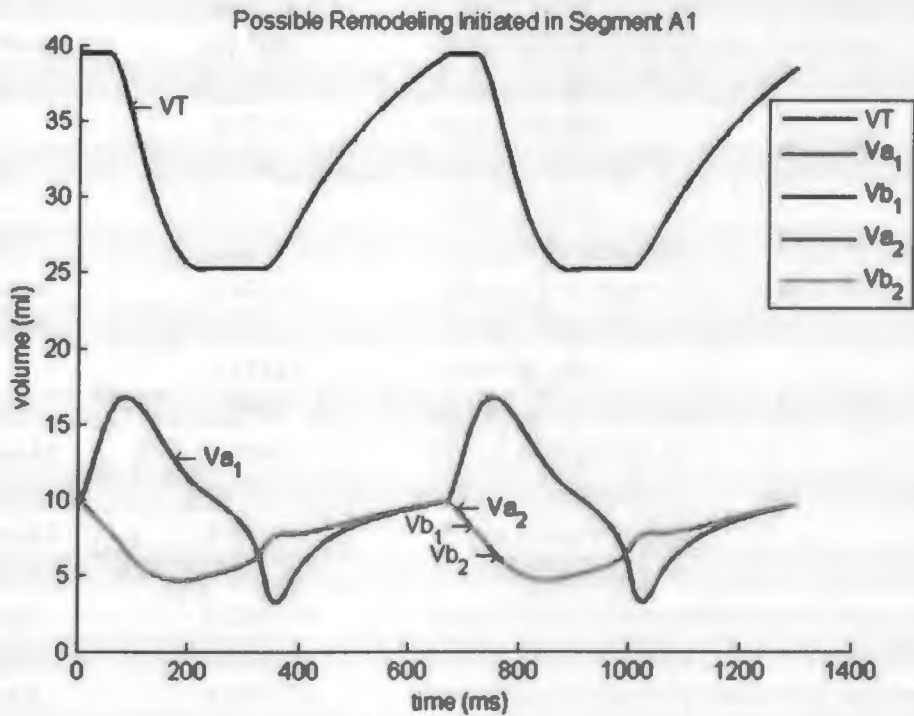
realized before a counterbalancing force is presented or severe heart dysfunction ensues. Regardless, this situation is meaningful to observe because it illustrates the effects of elastance waveform delay and can be compared to various other spacial and temporal relationships as seen from Figures 10-12.

**Table 5. Multi-Segment Elastance Variations**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
<b>IFF - C.C.</b>	0.00%	C.C. modeled Internal Flow Fraction
<b>IFF - 4seg</b>	18.70%	4seg modeled Internal Flow Fraction
<b>CO</b>	1.20 L/min	cardiac output
<b>SV</b>	13.3 ml	stroke volume
<b>EF</b>	32.90%	ejection fraction
<b>EW</b>	0.1241 J	external work
<b>PVA</b>	0.2685 J	pressure volume area
<b>PVAmyo</b>	0.3239 J	$PVA_{myo} = W_{myo} + PW + Ploss$
<b>Wmyo</b>	0.1610 J	work done by the myocardium
<b>PW</b>	0.1443 J	potential work
<b>Ploss</b>	0.0278 W	power loss of the myocardium
<b>WmyoA1</b>	0.0302 J	Wmyo of segment A1
<b>WmyoB1</b>	0.0503 J	Wmyo of segment B1
<b>WmyoA2</b>	0.0302 J	Wmyo of segment A2
<b>WmyoB2</b>	0.0503 J	Wmyo of segment B2
<b>PlossA1</b>	0.0037 W	power loss of the myocardium in segment A1
<b>PlossB1</b>	0.0102 W	power loss of the myocardium in segment B1
<b>PlossA2</b>	0.0037 W	power loss of the myocardium in segment A2
<b>PlossB2</b>	0.0102 W	power loss of the myocardium in segment B2

Interestingly, despite substantial mechanical dyssynchrony, the global waveform of the volume signal from Figure 13 very closely resembles the global waveform of the baseline waveforms from Figure 8. For this isolated event, this observation shows that in theory, large alterations in segmental elastance delay and elastance magnitude does not inhibit global ejection or filling. Once again, this situation is unlikely to occur in nature without triggering substantial feedback mechanisms that alter venous and arterial resistive or

elastic properties, or potentially trigger mechanisms that alter heart muscle dynamics all together.



**Figure 13. Waveforms Showing Elastance and Delay Variations. Segment A1 is initiated with a max elastance of  $10 \frac{\text{mmHg}}{\text{ml}}$  as well as delayed 40 ms from the remaining three segments. Despite large amounts of mechanical dyssynchrony, the global volume waveform follows a consistent path. Global Isovolumic contraction is slightly longer in duration than previous simulations. Remodeling is likely to occur in segment A1.**

Below, Table 7 shows the results of the simulations previously discussed and illustrated from Figures 8-13. The most notable trends were seen when modifying segmental elastance as substantial drops in cardiac output and stroke volume were seen. In each of these cases, a considerable increase in IFF-C.C. was seen over baseline values.

**Table 6. Ventricle Remodeling Due to Muscle Parameters**

<b>Metric</b>	<b>Value</b>	<b>Description</b>
<b>IFF - C.C.</b>	<b>32.50%</b>	<b>C.C. modeled Internal Flow Fraction</b>
<b>IFF - 4seg</b>	<b>52.70%</b>	<b>4seg modeled Internal Flow Fraction</b>
<b>CO</b>	<b>1.28 L/min</b>	<b>cardiac output</b>
<b>SV</b>	<b>14.2 ml</b>	<b>stroke volume</b>
<b>EF</b>	<b>36.00%</b>	<b>ejection fraction</b>
<b>EW</b>	<b>0.1423 J</b>	<b>external work</b>
<b>PVA</b>	<b>0.2776 J</b>	<b>pressure volume area</b>
<b>PVAmyo</b>	<b>0.4064 J</b>	<b>PVAmyo = Wmyo + PW + Ploss</b>
<b>Wmyo</b>	<b>0.2411 J</b>	<b>work done by the myocardium</b>
<b>PW</b>	<b>0.1352 J</b>	<b>potential work</b>
<b>Ploss</b>	<b>0.0452 W</b>	<b>power loss of the myocardium</b>
<b>WmyoA1</b>	<b>0.1053 J</b>	<b>Wmyo of segment A1</b>
<b>WmyoB1</b>	<b>0.0452 J</b>	<b>Wmyo of segment B1</b>
<b>WmyoA2</b>	<b>0.0452 J</b>	<b>Wmyo of segment A2</b>
<b>WmyoB2</b>	<b>0.0452 J</b>	<b>Wmyo of segment B2</b>
<b>PlossA1</b>	<b>0.0240 W</b>	<b>power loss of the myocardium in segment A1</b>
<b>PlossB1</b>	<b>0.0071 W</b>	<b>power loss of the myocardium in segment B1</b>
<b>PlossA2</b>	<b>0.0071 W</b>	<b>power loss of the myocardium in segment A2</b>
<b>PlossB2</b>	<b>0.0071 W</b>	<b>power loss of the myocardium in segment B2</b>

### **Monte Carlo Simulation**

A Monte Carlo simulation of random sampling of muscle parameters was implemented (n=2000). The uniform random sampling included 16 degrees of freedom including four each of maximum elastance (Ea1max, Ea2max, Eb1max, Eb2max), minimum elastance (Ea1min, Ea2min, Eb1min, Eb2min), elastance waveform delay (Da1, Da2, Db1, Db2), and myocardium viscous friction (ka1, ka2, kb1, kb2) selectively representing the 4 individual regional segments. A uniform random distribution was chosen for each variable

with individual range as follows from Table 8. A heart rate of 90 bpm, LAP of  $10 \frac{\text{mmHg}}{\text{ml}}$ , and arterial Windkessel TPR of 3.7 and compliance of 0.45 were used for all 2000 samples.

**Table 7. Multiple Simulation Data**

Metric	Base Waveform	Ea1 delay 10ms	Ea1max 10mmHg/mL	Ea1max Ea2max 10mmHg/mL	remodeled segment	Units
IFF-EC	0.00%	6.34%	6.24%	0.00%	32.50%	%
IFF-4seg	0.00%	10.20%	14.79%	18.70%	52.70%	%
CO	1.5	1.49	1.33	1.2	1.28	L/min
SV	16.63	16.58	14.8	13.3	14.2	ml
EF	42.70%	42.60%	37.20%	32.90%	36.00%	%
EW	0.200	0.199	0.156	0.124	0.142	J
PVA	0.350	0.3475	0.304	0.269	0.278	J
PVAmyo	0.396	0.409	0.364	0.324	0.406	J
Wmyo	0.216	0.229	0.191	0.161	0.241	J
PW	0.150	0.149	0.148	0.144	0.135	J
Ploss	0.044	0.045	0.037	0.028	0.045	W
WmyoA1	0.054	0.083	0.031	0.030	0.105	J
WmyoB1	0.054	0.049	0.053	0.050	0.045	J
WmyoA2	0.054	0.049	0.053	0.030	0.045	J
WmyoB2	0.054	0.049	0.053	0.050	0.045	J
PlossA1	0.011	0.020	0.004	0.004	0.024	W
PlossB1	0.011	0.009	0.011	0.010	0.007	W
PlossA2	0.011	0.009	0.011	0.004	0.007	W
PlossB2	0.011	0.009	0.011	0.010	0.007	W

The Monte Carlo simulation included 2,000 repeated samples across 16 degrees of freedom as outlined in Table 8. The frequency of various hemodynamic results and the distribution of mechanical dyssynchrony defined by internal flow fraction are listed in Figures 14-16. Specifically, the mechanical dyssynchrony metrics were assessed with the

hypothesis that due to the limitations of the conductance catheter, potentially important or valuable information relating to heart health and functionality would be missing.

**Table 8. Monte Carlo Parameters**

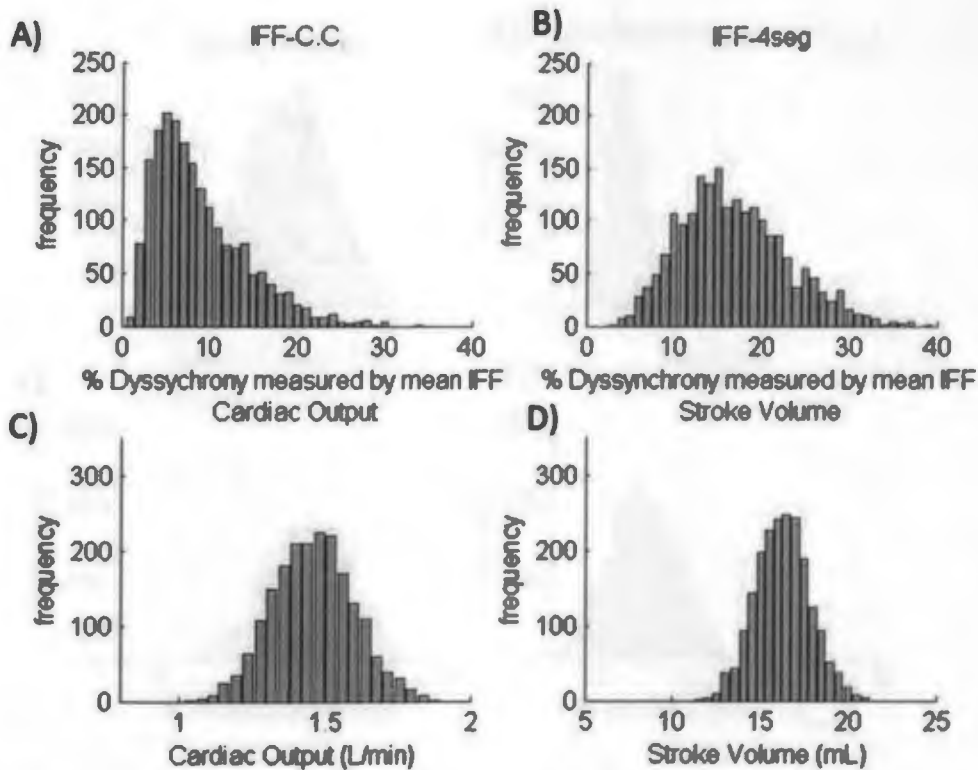
Variable	Distribution	Minimum	Maximum	Units
Ea1max	Uniform	10	30	mmHg/ml
Eb1max	Uniform	10	30	mmHg/ml
Ea2max	Uniform	10	30	mmHg/ml
Eb2max	Uniform	10	30	mmHg/ml
Ea1min	Uniform	0.6	1	mmHg/ml
Eb1min	Uniform	0.6	1	mmHg/ml
Ea2min	Uniform	0.6	1	mmHg/ml
Eb2min	Uniform	0.6	1	mmHg/ml
Da1	Uniform	0	20	ms
Db1	Uniform	0	20	ms
Da2	Uniform	0	20	ms
Db2	Uniform	0	20	ms
ka1	Uniform	3	15	s/L
kb1	Uniform	3	15	s/L
ka2	Uniform	3	15	s/L
kb2	Uniform	3	15	s/L

Figure 16 illustrates the downfall of measuring mechanical dyssynchrony through the use of a conductance catheter. By plotting IFF-4seg vs. IFF-C.C., it becomes very clear that IFF-C.C. incorrectly represents mechanical dyssynchrony for a large portion of the data set.

The 10 lowest and 10 highest IFF-CC readings were individually recorded along with the corresponding muscle parameters and IFF-4seg values in Table 9 and 10 respectively. The collection of data illustrates multiple trends across the spatial and



temporal properties of the myocardium for low and high values of mechanical dyssynchrony.

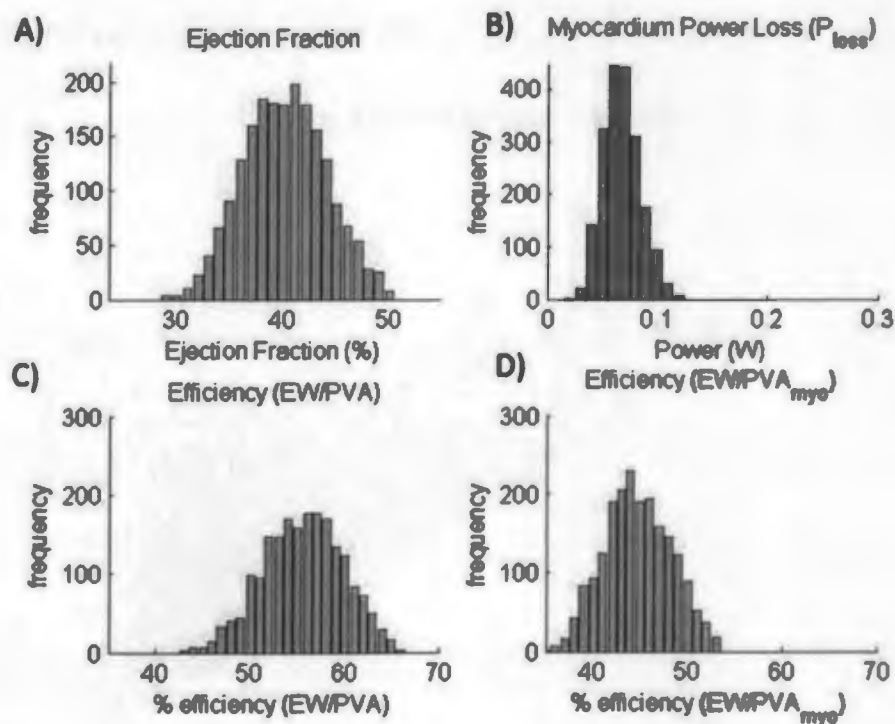


**Figure 14. Monte Carlo IFF Results. Distribution plots showing the frequency of simulations over a specified range for IFF-C.C., IFF-4seg, CO, and SV. (n=2000)**

#### **Influence of Muscle Properties on IFF and Hemodynamics**

In order to understand the specific influence of a wide range in elastance, delay, and myocardial friction on mechanical dyssynchrony and other hemodynamics, the model was run through a series of computational loops which recorded unique results for each progressively increasing variable. In reference to Figure 17, internal flow fraction was recorded for progressive alterations in regional elastance and regional viscous friction

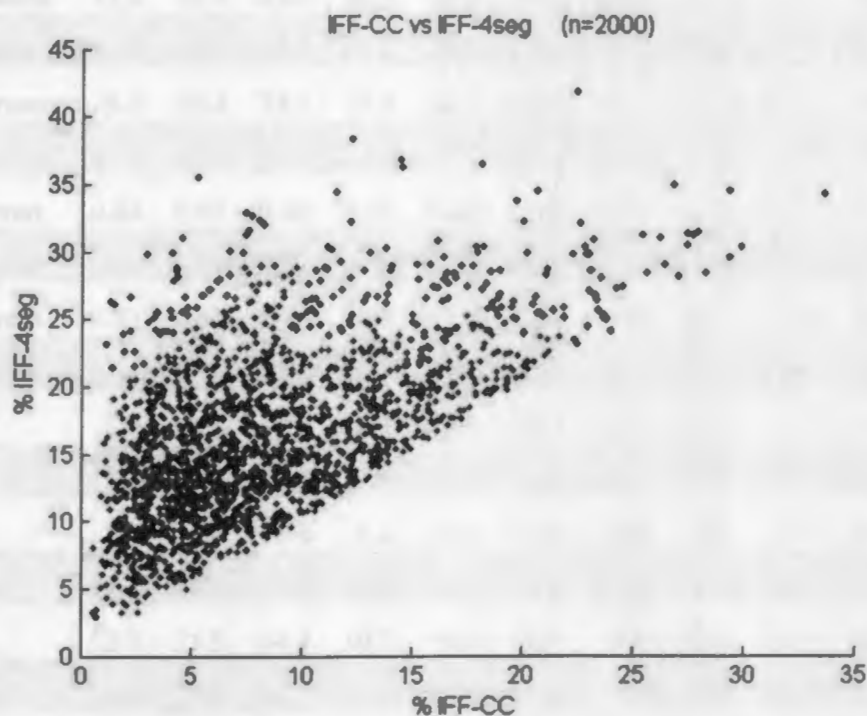
while all other muscle parameters for the remaining three segments represented physiologically healthy muscle tissue as previously outlined.



**Figure 15. Monte Carlo Results for EF,  $P_{loss}$ , EFF, and EFF<sub>myo</sub>. Distribution plots showing the frequency of simulations over a specified range for EF,  $P_{loss}$ , EFF, and EFF<sub>myo</sub>. (n=2000)**

The results of 5000 simulations were plotted three-dimensionally to better realize strong influencing parameters and various inflection points if any. Regional elastance in segment A1 was progressively varied from 10 to 30  $\frac{\text{mmHg}}{\text{ml}}$ , while viscous friction was varied from 0.0015 to 0.0105  $\frac{\text{sec}}{\text{ml}}$ . Figures 17-22 show the relationship of alterations in elastance and viscous friction on IFF-C.C., IFF-4seg, cardiac output (CO), ejection fraction (EF), external work (EW), work of the myocardium ( $W_{myo}$ ), pressure volume area (PVA),

pressure volume area inclusive to myocardium work and power losses ( $PVA_{myo}$ ), efficiency (EFF), myocardium efficiency ( $EFF_{myo}$ ), power losses of the myocardium due to viscous friction ( $P_{loss}$ ), and stroke volume (SV).



**Figure 16. Scatter Plot of IFF-C.C. vs. IFF-4seg.**

Similar to the alterations in muscle parameters seen in Figures 10-13, Figures 17-22 show dyssynchrony and hemodynamic results for a wide range of elastance values and delays represented three dimensionally. Similar to before, mechanical dyssynchrony was represented through internal flow fraction that defines the segmental volumes in two separate ways. First, by combining segmental volumes  $V_{A1}$  and  $V_{b1}$  the model produces a stacked cylinder that closely represents data obtainable from a conductance catheter.

Table 9. Parameters from 10 Lowest Resulting IFF-C.C.

IFF-C.C.	0.60	0.61	0.68	0.69	0.91	0.92	0.96	1.00	1.01	1.05	Units %
IFF-4seg	8.1	3.2	2.9	3.5	5.1	11.9	7.1	13.6	6.0	8.5	%
Ea1max	13.2	20.4	23.0	18.5	24.4	26.6	24.8	20.0	28.6	14.1	mmHg/ml
Eb1max	28.7	26.8	27.5	25.2	24.9	29.7	14.9	15.8	28.6	23.1	mmHg/ml
Ea2max	28.0	26.3	24.5	17.7	22.2	17.3	22.1	12.3	21.8	15.6	mmHg/ml
Eb2max	18.5	19.4	30.0	19.6	17.5	29.5	28.6	18.7	25.7	23.6	mmHg/ml
Ea1min	0.84	0.69	0.85	0.70	0.82	0.91	0.71	0.98	0.87	0.98	mmHg/ml
Eb1min	0.64	0.87	0.92	0.92	0.88	0.85	0.67	0.75	0.89	0.70	mmHg/ml
Ea2min	0.82	0.92	0.76	0.73	0.63	0.66	0.85	0.99	0.64	0.64	mmHg/ml
Eb2min	0.66	0.72	0.85	0.71	0.76	0.88	0.98	0.63	0.85	0.97	mmHg/ml
Da1	5.0	13.8	7.1	3.2	13.7	15.3	16.0	14.3	7.1	11.5	ms
Db1	10.7	19.7	3.7	7.9	8.1	3.5	12.3	10.1	12.5	18.2	ms
Da2	8.9	15.4	6.4	5.7	11.1	4.8	17.8	6.6	9.5	12.8	ms
Db2	7.1	16.6	5.1	4.9	12.4	18.1	10.4	15.1	12.5	19.4	ms
ka1	12.3	11.5	14.3	10.7	8.0	11.8	8.6	13.0	10.7	10.6	s/L
kb1	13.6	10.1	13.7	14.5	13.1	13.5	6.0	6.0	13.7	12.6	s/L
ka2	11.8	12.0	14.9	5.2	9.6	6.2	12.1	9.4	3.9	13.6	s/L
kb2	7.9	9.0	14.4	13.9	12.8	13.2	10.1	8.2	10.8	14.5	s/L

In the second method, the regional volume segments are considered separately in their contribution to internal flow, thereby potentially producing results that vary from the first method. Throughout this analysis, the changing variables were limited to one segment, while the remaining three segments were kept at constant values that simulate a healthy canine heart. The end result shows what happens to a heart with regional disease or myocardial abnormalities. The elastance of segment A1 was varied from 10 to 30  $\frac{\text{mmHg}}{\text{ml}}$ , while delay of segment A1 was varied from 0 to 40ms delay.

**Table 10. Parameters from 10 Highest Resulting IFF-C.C.**

IFF-C.C.	33.6	29.8	29.3	29.4	28.3	27.9	27.7	27.5	27.4	27.0	Units
											%
<b>IFF-4seg</b>	<b>34.4</b>	<b>30.3</b>	<b>29.7</b>	<b>34.5</b>	<b>28.4</b>	<b>31.4</b>	<b>31.2</b>	<b>31.4</b>	<b>30.4</b>	<b>28.2</b>	<b>%</b>
<b>Ea1max</b>	<b>20.3</b>	<b>11.5</b>	<b>17.0</b>	<b>11.1</b>	<b>19.2</b>	<b>27.3</b>	<b>10.9</b>	<b>26.6</b>	<b>26.7</b>	<b>20.5</b>	<b>mmHg/ml</b>
<b>Eb1max</b>	<b>22.9</b>	<b>15.1</b>	<b>10.1</b>	<b>12.7</b>	<b>45.4</b>	<b>26.9</b>	<b>14.9</b>	<b>20.8</b>	<b>20.4</b>	<b>14.4</b>	<b>mmHg/ml</b>
<b>Ea2max</b>	<b>11.2</b>	<b>19.4</b>	<b>16.8</b>	<b>23.9</b>	<b>35.8</b>	<b>20.5</b>	<b>23.8</b>	<b>15.4</b>	<b>11.2</b>	<b>27.2</b>	<b>mmHg/ml</b>
<b>Eb2max</b>	<b>12.9</b>	<b>16.1</b>	<b>23.0</b>	<b>25.7</b>	<b>22.1</b>	<b>11.4</b>	<b>24.7</b>	<b>11.1</b>	<b>25.6</b>	<b>22.3</b>	<b>mmHg/ml</b>
<b>Ea1min</b>	<b>0.65</b>	<b>0.93</b>	<b>0.95</b>	<b>0.81</b>	<b>0.64</b>	<b>0.97</b>	<b>0.90</b>	<b>0.73</b>	<b>0.60</b>	<b>0.77</b>	<b>mmHg/ml</b>
<b>Eb1min</b>	<b>0.67</b>	<b>0.67</b>	<b>0.88</b>	<b>0.75</b>	<b>0.46</b>	<b>0.62</b>	<b>0.78</b>	<b>0.63</b>	<b>0.75</b>	<b>0.94</b>	<b>mmHg/ml</b>
<b>Ea2min</b>	<b>0.82</b>	<b>0.69</b>	<b>0.72</b>	<b>0.65</b>	<b>0.52</b>	<b>0.75</b>	<b>0.74</b>	<b>0.74</b>	<b>0.75</b>	<b>0.94</b>	<b>mmHg/ml</b>
<b>Eb2min</b>	<b>0.75</b>	<b>0.80</b>	<b>0.97</b>	<b>0.62</b>	<b>0.50</b>	<b>0.70</b>	<b>0.76</b>	<b>0.78</b>	<b>0.89</b>	<b>0.91</b>	<b>mmHg/ml</b>
<b>Da1</b>	<b>4.0</b>	<b>16.9</b>	<b>17.9</b>	<b>16.3</b>	<b>0.5</b>	<b>1.2</b>	<b>13.7</b>	<b>11.2</b>	<b>2.4</b>	<b>19.1</b>	<b>ms</b>
<b>Db1</b>	<b>0.0</b>	<b>16.5</b>	<b>19.9</b>	<b>17.3</b>	<b>2.3</b>	<b>4.6</b>	<b>14.1</b>	<b>19.4</b>	<b>6.5</b>	<b>18.4</b>	<b>ms</b>
<b>Da2</b>	<b>16.2</b>	<b>1.1</b>	<b>1.5</b>	<b>3.9</b>	<b>9.4</b>	<b>18.7</b>	<b>8.8</b>	<b>1.8</b>	<b>17.3</b>	<b>7.7</b>	<b>ms</b>
<b>Db2</b>	<b>18.1</b>	<b>2.9</b>	<b>2.6</b>	<b>12.1</b>	<b>8.1</b>	<b>16.1</b>	<b>0.0</b>	<b>0.0</b>	<b>13.5</b>	<b>3.3</b>	<b>ms</b>
<b>ka1</b>	<b>8.3</b>	<b>6.1</b>	<b>14.8</b>	<b>15.0</b>	<b>1.2</b>	<b>3.5</b>	<b>7.0</b>	<b>5.8</b>	<b>5.9</b>	<b>12.6</b>	<b>s/L</b>
<b>kb1</b>	<b>7.4</b>	<b>13.7</b>	<b>4.1</b>	<b>3.5</b>	<b>4.9</b>	<b>14.6</b>	<b>8.1</b>	<b>3.8</b>	<b>3.3</b>	<b>4.4</b>	<b>s/L</b>
<b>ka2</b>	<b>7.4</b>	<b>6.0</b>	<b>11.2</b>	<b>4.7</b>	<b>4.7</b>	<b>14.2</b>	<b>6.2</b>	<b>9.0</b>	<b>6.2</b>	<b>4.9</b>	<b>s/L</b>
<b>kb2</b>	<b>6.5</b>	<b>7.5</b>	<b>14.1</b>	<b>5.7</b>	<b>1.2</b>	<b>3.6</b>	<b>5.4</b>	<b>5.8</b>	<b>12.0</b>	<b>7.3</b>	<b>s/L</b>

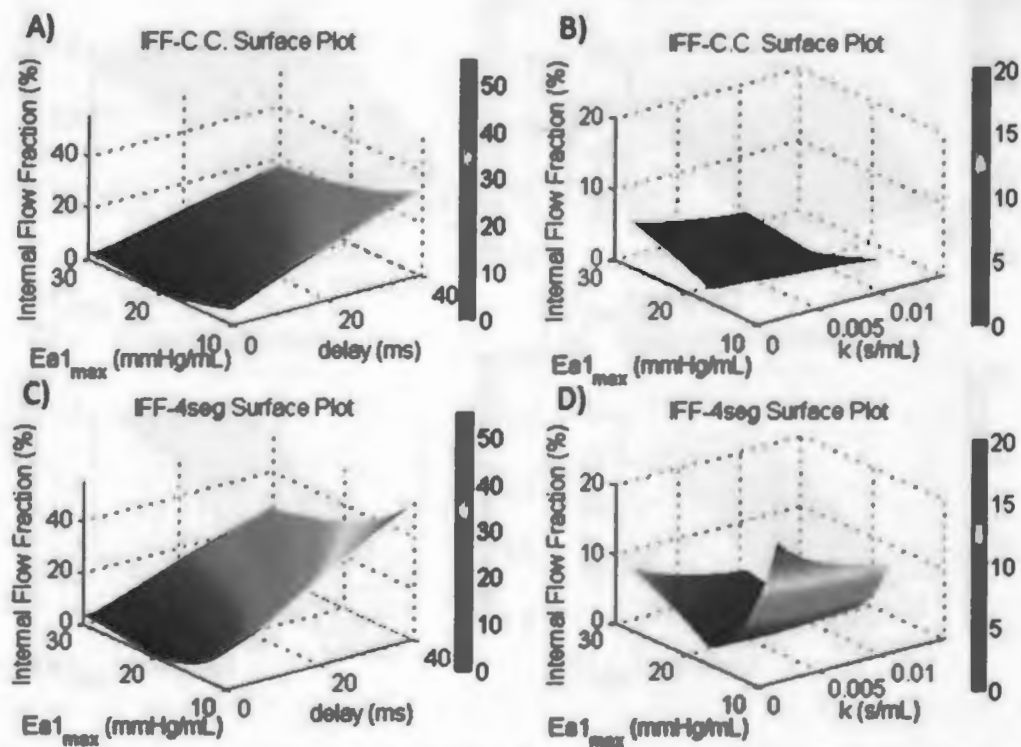


Figure 17. Internal Flow Fraction. The top two plots display IFF-C.C. while the bottom two plots display IFF-4seg. In part A)  $Ea1_{max}$  is varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while delay of that elastance waveform is varied from 0-40ms. The graph shows IFF-C.C. for all combinations. In part B)  $Ea1_{max}$  is varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while  $k$  is varied from .0015 to .0105  $\frac{\text{sec}}{\text{ml}}$  with IFF-C.C. on the z-axis. In part C)  $Ea1_{max}$  is once again varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while delay of that elastance waveform is varied from 0-40ms. IFF-4seg is along the z-axis. In part D)  $Ea1_{max}$  is varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while  $k$  is varied from .0015 to .0105  $\frac{\text{sec}}{\text{ml}}$  with IFF-4seg along the Z-axis.

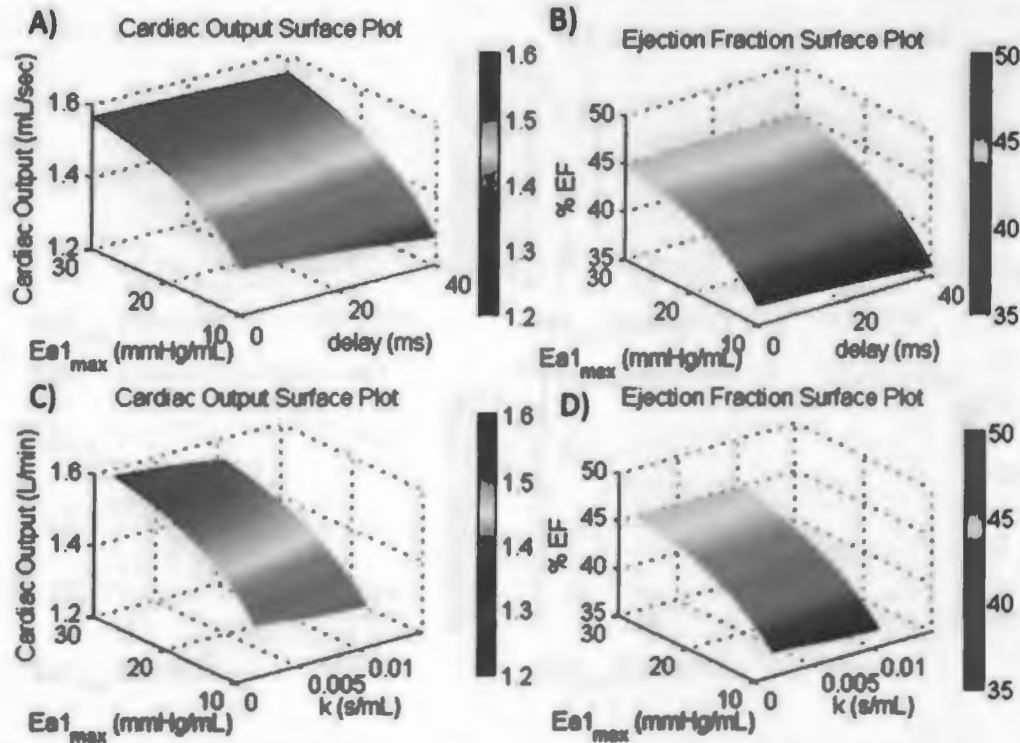


Figure 18. Cardiac Output and Ejection Fraction. The top two plots have  $Ea1_{max}$  and  $Ea1_{max}$  delay along the X and Y- axis while the bottom two plots have  $Ea1_{max}$  and  $k$  along the X and Y-axis. In part A)  $Ea1_{max}$  is varied from 10 -30  $\frac{mmHg}{ml}$  while delay of that elastance waveform is varied from 0-40ms. The graph shows CO for all combinations. In part B)  $Ea1_{max}$  is varied from 10 -30  $\frac{mmHg}{ml}$  while delay of that elastance waveform is varied from 0-50ms. The graph shows EF for all combinations. In part C)  $Ea1_{max}$  is once again varied from 10 -30  $\frac{mmHg}{ml}$  while  $k$  is varied from .0015 to .0105  $\frac{sec}{ml}$  with CO along the z-axis. In part D)  $Ea1_{max}$  is varied from 10 -30  $\frac{mmHg}{ml}$  while  $k$  is varied from .0015 to .0105  $\frac{sec}{ml}$  with EF along the Z-axis.

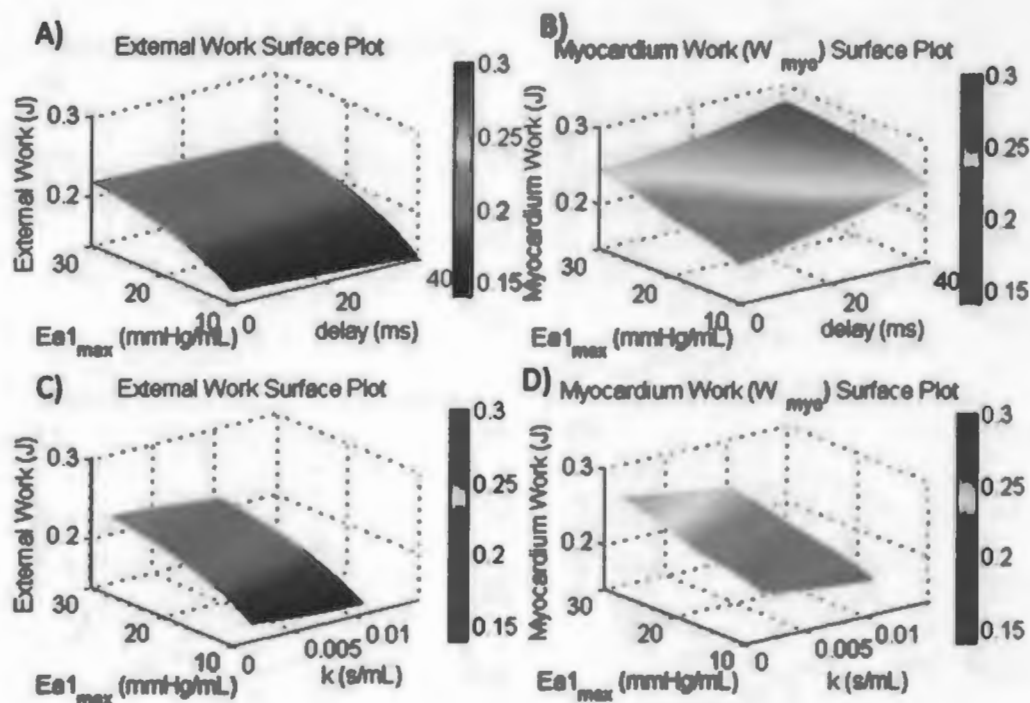
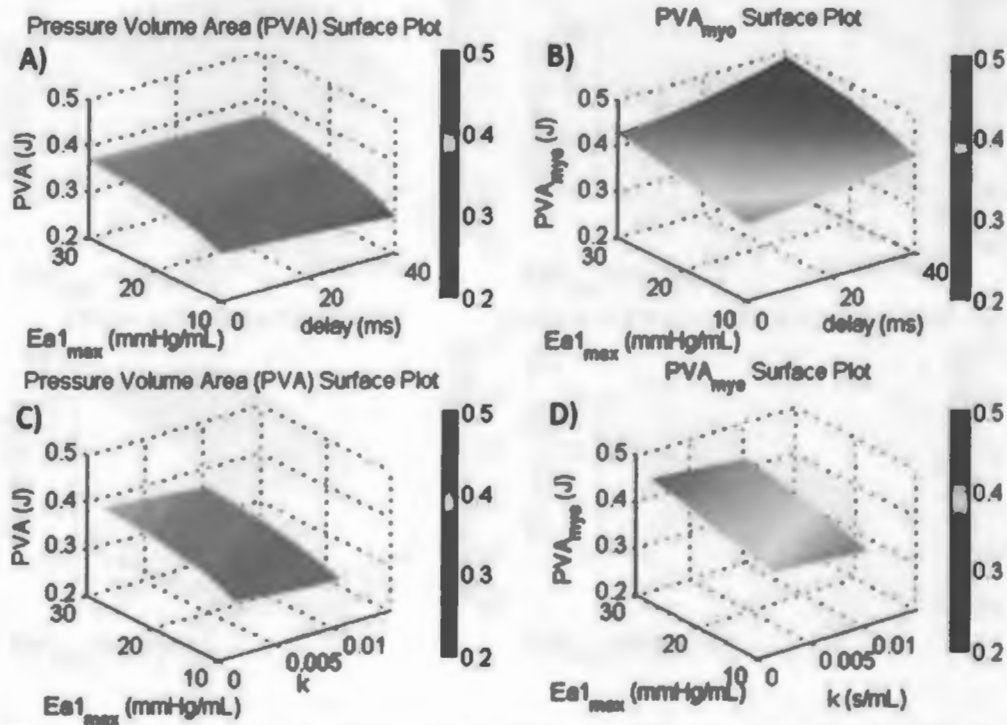
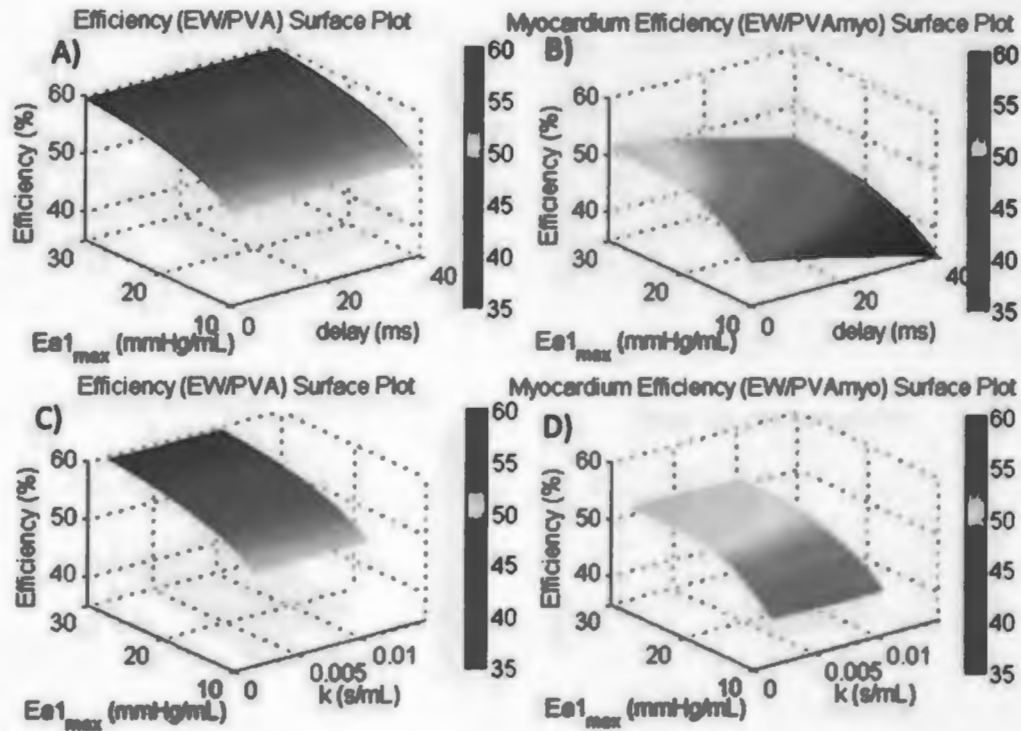


Figure 19. External Work and Myocardium Work. The top two plots have  $Ea1_{max}$  and  $Ea1_{max}$  delay along the X and Y-axis while the bottom two plots have  $Ea1_{max}$  and  $k$  along the X and Y-axis. In part A)  $Ea1_{max}$  is varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while delay of that elastance waveform is varied from 0-40ms. The graph shows EW for all combinations. In part B)  $Ea1_{max}$  is varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while delay of that elastance waveform is varied from 0-40ms. The graph shows  $W_{myo}$  for all combinations. In part C)  $Ea1_{max}$  is once again varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while  $k$  is varied from .0015 to .0105  $\frac{\text{sec}}{\text{ml}}$  with EW along the z-axis. In part D)  $Ea1_{max}$  is varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while  $k$  is varied from .0015 to .0105  $\frac{\text{sec}}{\text{ml}}$  with  $W_{myo}$  along the Z-axis.

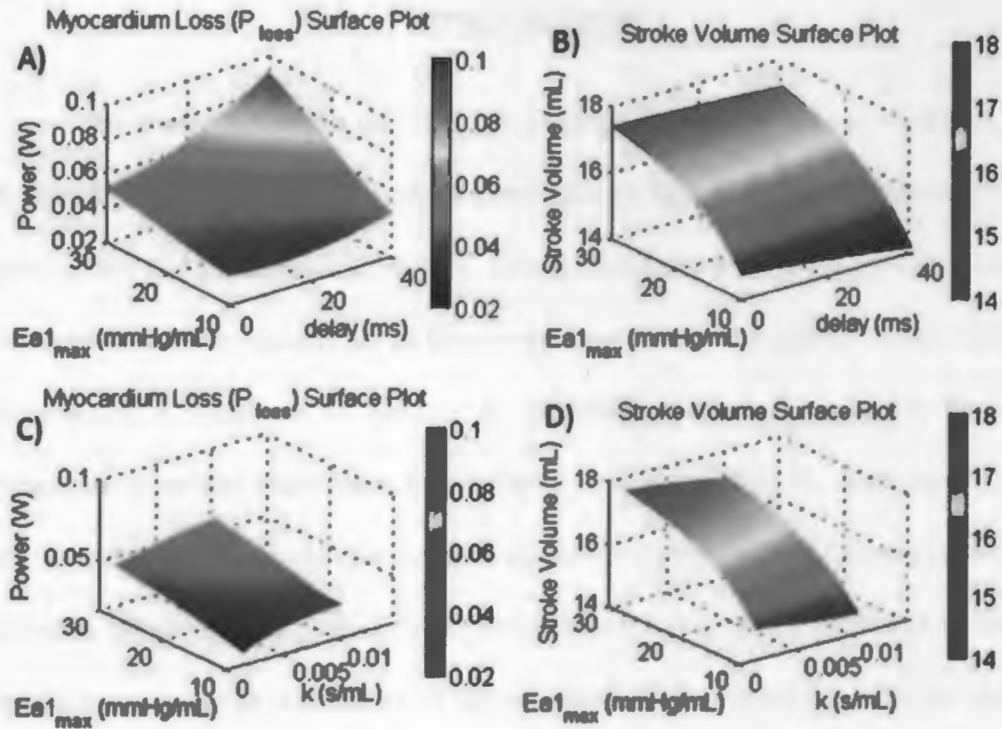




**Figure 20. PVA and PVA<sub>myoe</sub>.** The top two plots have  $Ea1_{max}$  and  $Ea1_{max}$  delay along the X and Y- axis while the bottom two plots have  $Ea1_{max}$  and  $k$  along the X and Y-axis. In part A)  $Ea1_{max}$  is varied from 10 -30  $\frac{mmHg}{ml}$  while delay of that elastance waveform is varied from 0-40ms. The graph shows PVA for all combinations. In part B)  $Ea1_{max}$  is varied from 10 -30  $\frac{mmHg}{ml}$  while delay of that elastance waveform is varied from 0-40ms. The graph shows PVA<sub>myoe</sub> for all combinations. In part C)  $Ea1_{max}$  is once again varied from 10 -30  $\frac{mmHg}{ml}$  while  $k$  is varied from .0015 to .0105  $\frac{sec}{ml}$  with PVA along the z-axis. In part D)  $Ea1_{max}$  is varied from 10 -30  $\frac{mmHg}{ml}$  while  $k$  is varied from .0015 to .0105  $\frac{sec}{ml}$  with PVA<sub>myoe</sub> along the Z-axis.



**Figure 21. Efficiency and Myocardium Efficiency.** The top two plots have  $Ea1_{max}$  and  $Ea1_{max}$  delay along the X and Y- axis while the bottom two plots have  $Ea1_{max}$  and k along the X and Y-axis. In part A)  $Ea1_{max}$  is varied from 10 -30  $\frac{mmHg}{ml}$  while delay of that elastance waveform is varied from 0-40ms. The graph shows efficiency for all combinations. In part B)  $Ea1_{max}$  is varied from 10 -30  $\frac{mmHg}{ml}$  while delay of that elastance waveform is varied from 0-40ms. The graph shows myocardium efficiency for all combinations. In part C)  $Ea1_{max}$  is once again varied from 10 -30  $\frac{mmHg}{ml}$  while k is varied from .0015 to .0105  $\frac{sec}{ml}$  with efficiency along the z-axis. In part D)  $Ea1_{max}$  is varied from 10 -30  $\frac{mmHg}{ml}$  while k is varied from .0015 to .0105  $\frac{sec}{ml}$  with myocardium efficiency along the Z-axis.



**Figure 22. Power Loss and Stroke Volume.** The top two plots have  $Ea1_{max}$  and  $Ea1_{max}$  delay along the X and Y- axis while the bottom two plots have  $Ea1_{max}$  and  $k$  along the X and Y-axis. In part A)  $Ea1_{max}$  is varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while delay of that elastance waveform is varied from 0-40ms. The graph shows  $P_{loss}$  for all combinations. In part B)  $Ea1_{max}$  is varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while delay of that elastance waveform is varied from 0-40ms. The graph shows SV for all combinations. In part C)  $Ea1_{max}$  is once again varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while  $k$  is varied from .0015 to .0105  $\frac{\text{sec}}{\text{ml}}$  with  $P_{loss}$  along the z-axis. In part D)  $Ea1_{max}$  is varied from 10 -30  $\frac{\text{mmHg}}{\text{ml}}$  while  $k$  is varied from .0015 to .0105  $\frac{\text{sec}}{\text{ml}}$  with SV along the Z-axis.

## DISCUSSION

This thesis describes a technique of lumped parameter ventricular modeling that is designed to look at the complex nature of mechanical dyssynchrony as it relates to muscle parameters and hemodynamic metrics. Computation power and model complexity can often times inhibit researchers as they investigate trends in modeled muscle dynamics. As a result, a multitude of solutions to modeling whole heart dynamics have been proposed by various researchers. Kerckhoffs et al. (Kerckhoffs R. M., 2006) constructed a 3D finite element model that was coupled to a lumped parameter circulatory system in an effort to obtain cardiac phase independence, which assumes the net effect of enabling multiple cardiac cycle simulations by conservation of blood mass and also the ability to simulate valvular or septal pathologies. Rolle et al. (Rolle, 2007) utilized a tissue-level approach and an identification algorithm that maps the strain energy function over an ellipsoid. Dou et al. (Dou, 2009) constructed a ventricular model utilizing the dynamic cell model first proposed by Winslow et al. (Winslow, 1999) through a finite element method for the analysis of bundle branch block. Completely encompassing the complex nature of heart mechanics through a computational model is exceedingly difficult, if not impossible, due to the fact that the heart is a load dependent source and receives input from a multitude of receptors and feedback loops for which their influences are still not fully understood.

A lumped parameter system such as the one in this study allows for the integration of viscous influence without substantially inhibiting computation viability.

Myocardial viscous influence is left unaccounted for in several recent models such as those proposed by Kerckhoffs et al. (Kerckhoffs R. A., 2009) and Dou et al. (Dou, 2009), most likely due to the additional computational expense. The debate of finite element versus lumped parameter lies completely with the researcher's requirement for detail and computation speed requirements. In 1931, scientist and philosopher Alfred Korzybski famously expressed that "the map is not the territory" (Korzybski, 1933). In cardiovascular modeling we must construct representations of how the heart functions based on our perception of reality and not necessarily what is actually happening; else we would be working on a live heart. Extending further, you can never have a perfect map nor would it be beneficial to have one analogous to the idea of using a satellite imagery map to navigate a densely populated city.

By isolating certain load dependencies, attention can be focused in a specific area of interest; in this case mechanical dyssynchrony. For this reason, a reactive preload was eliminated from this model by using a steady state left atrial pressure. While a reactive afterload was implemented to ensure integrity of flow dynamics, a 2-element Windkessel diminished the reactive effects of multiple element circulatory systems used by other authors (Arai, 2010; Kerckhoffs R. M., 2006).

By examining the results obtained from Figure 17.A and 17.B, it becomes clear that the properties most influential to mechanical dyssynchrony are elastance segmental delay, elastance magnitude, and myocardial viscous friction, in that order. From Figure 17.A, a delay in segment A1 of 40ms produced an IFF-C.C. value of over 30% compared to a maximum excursion of about 7% through large alterations in segment A1 maximum

elastance, and from Figure 17.B the component of viscous friction accounted for negligible amounts of dyssynchrony in comparison at roughly 3%.

Due to the relationship between CO and SV, and the fact that HR remained constant throughout all simulations, one can expect similar results to emerge. Indeed, Figure 18.A and 18.C compare favorably with 22.B and 22.D. While no parameter showed to have strong influence on CO and SV, the myocardial viscous friction component ( $k$ ) possibly accounts for the greatest degree of change due to the its inhibitory effects of excess energy waste for high  $k$  values, preventing the myocardium from reaching its true end diastolic volume.

In terms of ejection fraction, no clear property was magnified, very similar to the results seen with CO and SV. This is likely due to the relatively small changes in global EDV and ESV for varying parameters even in such cases as high mechanical dyssynchrony. Besides Figures 18.B and 18.D, this relatively stable nature of EDV and ESV for substantial mechanical dyssynchrony can be seen in Figures 10-13.

When analyzing EW and PVA as a function of elastance, delay, and viscous friction, an overall lack of response consistent with high degrees of mechanical dyssynchrony may indicate that these standard metrics are ill fit for reliably measuring myocardium health in an ideal situation. The problem is that while the heart may be producing adequate external work, the amount of energy required to produce that work in the form of oxygen consumption has gone up significantly. This is certainly the case for the simulation in Figure 13 which produced IFF-C.C. of 32.5% and IFF-4seg of 52.7%. The solution taken in this work was to calculate work of the myocardium ( $W_{myo}$ ) and  $PVA_{myo}$ . In particular,  $W_{myo}$

takes into account the work by each segment individually. In the case where internal flows of blood do not directly result in ejection, work may normally go unaccounted for. Defined by equation 31,  $P_{loss}$  accounts for any power lost by each segment from friction due to tissue interactions at the sarcomere level. The response between EW, PVA,  $W_{myo}$ ,  $PVA_{myo}$ , as well as  $P_{loss}$  can be seen in Figures 19, 20, and 22.A and C. A strong example of unaccounted for work is the simulation shown in Figure 13. Further consulting table 6 reveals that  $P_{loss}$  and  $W_{myo}$  for segment A1 ( $P_{lossA1}$  and  $W_{myoA1}$  respectively) account for over 53% of losses and 44% of myocardium work of the whole ventricle, clearly unsustainable amounts!

### Conductance Catheter Response

By observation of Figure 12 and Table 5, it is shown that segmental conductance catheter signals effectively do not measure some forms of abnormal regional wall motion contained within a particular short axis volume slice. Even though clear mechanical dyssynchrony is visible in Figure 12, IFF-C.C. reported 0.0% dyssynchrony while IFF-4seg revealed 18.7% dyssynchrony. One of the reasons why conductance catheters can produce differing results stems from their design and the contraction of a heart. The conductance catheter segments the left ventricle into multiple short-axis slices and determines volume of this slice through parallel conductance. While the catheter can detect a change in volume of the slice, it does not have the ability to determine to what extent the slice moved back and forth without volume change. This is the situation that has been successfully simulated in Figure 12. Steendijk et al. (Steendijk, 2003) came to a similar conclusion when developing three novel indices of mechanical dyssynchrony as

well as indicating that the proposed metrics “may underestimate phase changes obtained by comparing regional lateral and septal wall motions”.

### **Limitations**

This cardiovascular model is based on a simplified version of a canine heart that lacks many of the complexities found in various finite element models. Furthermore, a very simple approach towards system mechanics was used in a 2-element Windkessel afterload and a steady-state preload. For researchers exploring other pathways of heart failure outside of mechanical dyssynchrony, changes to the system mechanics and broad model overview may be necessary.

This model examines mechanical dyssynchrony based on a radial wall motion approach. While this view of the heart encompasses the majority of heart contraction, recent research has expanded focus on various methods of breaking down the anatomy of a contraction such as longitudinal contraction and left ventricular twist (Sade, 2008; Gorcsan, 2007). Analyzing longitudinal dyssynchrony may be of little value as these metrics generally have a very tight dynamic range, unlike that of radial motion (Sade, 2008). LV twist however, has been shown to be a major component of contraction (Sonnenblick, 1967) and beneficial for early detection of heart failure (Wang, 2008). The importance of LV twist lies within the structural changes that can occur for imbalances of the torque generated by the myocardial fibers of the endocardium and epicardium (Wang, 2008). There could be strong potential for this analysis to be implemented in a model of the heart.



## CONCLUSION

In this study, a method of lumped parameter cardiovascular modeling has been developed. The method allows for extremely fast computation of muscle dynamics and a broad spectrum of tests that are not feasible with other models due to time restrictions. The mechanical property of elastance delay was the most influential property that was tested in response to mechanical dyssynchrony in the form of IFF-C.C. and IFF-4seg. Alterations in the magnitude of the elastance waveform and the myocardial component of viscous friction produced successively less influence on mechanical dyssynchrony through IFF calculations for both IFF-C.C. and IFF-4seg. Additionally, the model has demonstrated that conductance catheters tend to filter out regional wall motion when an offsetting wall motion is realized within the same circumferential slice. Unanticipated results were found in a variety of cases where high mechanical dyssynchrony did not necessarily equate to significant curtailment in output metrics such as CO, EW, and PVA. This was largely due to the increased burden taken on by one segment and represents scenarios where remodeling is almost certain to occur if the heart condition was left unabated.

## REFERENCES

- Abdolrazaghi, M. M. (2010). Mathematical Modelling and Electrical Analog Equivalent of the Human Cardiovascular System.
- Adelstein, E. S. (2007). Scar burden by myocardial perfusion imaging predicts echocardiographic response to cardiac resynchronization therapy in ischemic cardiomyopathy. *Am heart J* (153), 105-112.
- Arai, T. K. (2010, Sept 4). Cardiac Output and Stroke Volume Estimation Using a Hybrid of Three Windkessel Models. 32nd Annual International Conference of the IEEE EMBS , 4971-4974.
- Beneken, J. V. (1996). The use of multiple models in cardiovascular system studies: transport and perturbation methods. 15.
- Cleland, J. (2006). Longer-term effects of cardiac resynchronization therapy on mortality in heart failure. (27), 1928-1932.
- Dou, J. L. (2009). Mechanical analysis of congestive heart failure caused by bundle branch block based on an electromechanical canine heart model. *Physics In Medicine And Biology* , 54, 353-371.
- Ewert, D. B. (2004). The Effect of Heart Rate, Preload, and Afterload on the Viscoelastic Properties of the Swine Myocardium. 32 (9), 1211-1222.
- Glantz, S. (1975). A Three-element model describes excised cat papillary muscle elasticity. *American Journal of Physiology* , 1, 284-294.
- Goldsmith, E. T. (2002). The dynamic interaction of the extracellular matrix in cardiac remodeling. S314-S318.
- Gorcsan, J. (2007). Combined Longitudinal and Radial Dyssynchrony Predicts Ventricular Response After Resynchronization Therapy. (50), 1476-1483.
- Granzier, H. a. (2004). The giant protein titin: A major player in myocardial mechanics, signaling, and disease. (94), 284-295.
- Helm, R. (2005). Cardiac Dyssynchrony Analysis Using Circumferential Versus Longitudinal Strain. (111), 2760-2767.

- Hill, A. (1938). The heat of shortening and the dynamic constants of muscle. (126), 136-195.
- Hunter, P. A. (2003). Modeling Total Heart Function. 5, 147-177.
- Huxley, A. R. (1954). Structural changes in muscle during contraction; interference microscopy of living muscle fibers. (173), 971-973.
- Huxley, H. E. (1954). Changes in the cross-striations of muscle during contraction and stretch of their structural interpretation. (173), 973-976.
- Jansen, A. F. (2007). The influence of myocardial scar and dyssynchrony on reverse remodeling in cardiac resynchronization therapy. *European Journal of Echocardiography* .
- Kerckhoffs, R. A. (2009). Effects of biventricular pacing and scar size in a computational model of the failing heart with left bundle branch block. *Med Image Anal* (13), 362-369.
- Kerckhoffs, R. M. (2006). Coupling of a 3D Finite Element Model of Cardiac Ventricular Mechanics to Lumped Systems Models of the Systemic and Pulmonic Circulation. *Annals of Biomedical Engineering* , 35 (1), 1-18.
- Kerckhoffs, R. P. (2003). Homogeneity of Cardiac Contraction Despite Physiological Asynchrony of Depolarization: A Model Study. *Annals of Biomedical Engineering* , 31, 536-547.
- Kerckhoffs, R. P. (2003). Intra- and interventricular asynchrony of electromechanics in the ventricularly paced heart. *Journal of Engineering Mathematics* , 201-216.
- Korzybski, A. (1933). *Science and Sanity: An Introduction to Non-Aristotelian Systems and General Semantics*.
- Landesberg, A. S. (1994). Mechanical regulation of cardiac muscle by coupling calcium kinetics with cross-bridge cycling: a dynamic model. *American Journal of Physiology* , 267 (2), H779-H795.
- Lang, R. (2006). Three-Dimensional Echocardiography: The Benefits of the Additional Dimension. (48), 2053-2069.
- Lumens, J. T. (2008). Modeling Ventricular Interaction: A Multiscale Approach from Sarcomere Mechanics to Cardiovascular System Hemodynamics. *Pacific Symposium of Biocomputing* , 378-389.

McAlister, F. J. (2007). Cardiac resynchronization therapy for patients with left ventricular systolic dysfunction: a systematic review. *JAMA* (297), 2502-2514.

Nebot, A. F. (1998). Mixed quantitative/qualitative modeling of the cardiovascular system and simulation of the cardiovascular system. *Computer Methods and Programs in Biomedicine* , 127-155.

Nielsen, P. G. (1991). Mathematical model of geometry and fibrous structure of the heart. 260, H1365-78.

Rolle, V. L. (2007). A Tissue-Level Electromechanical Model of the Left Ventricle: Application to the Analysis of Intraventricular Pressure. *Computers in Cardiology* , 34, 669-672.

Sade, L. E. (2008). Effect of Mechanical Dyssynchrony and Cardiac Resynchronization Therapy on Left Ventricular Rotational Mechanics. *American Journal of Cardiology* , 101, 1163-1169.

Sagawa, K. L. (1988). Cardiac Contraction and the Pressure-Volume Relationship.

Sonnenblick, E. J. (1967). The ultrastructure of the heart in systole and diastole. *Circulation Research* , 21, 423-431.

Steendijk, P. (2003). Quantification of left ventricular mechanical dyssynchrony by conductance catheter in heart failure patients. (286), H723-H730.

Strum, D. M. (2000). Modeling of Asynchronous Myocardial Contraction by Effective Stroke Volume Analysis. *Cardiovascular Anesthesia* , 90, 243-251.

Sunagawa, K. A. (1980). Estimation of the hydromotive source pressure from ejecting beats of the left ventricle. *IEEE Trans. Biomed. Eng.* (27), 299-305.

Tang, W. W. (2008). Relation of mechanical dyssynchrony with underlying cardiac structure and performance in chronic heart failure: implications of clinical response to cardiac resynchronization. *Europace* (10), 1370-1374.

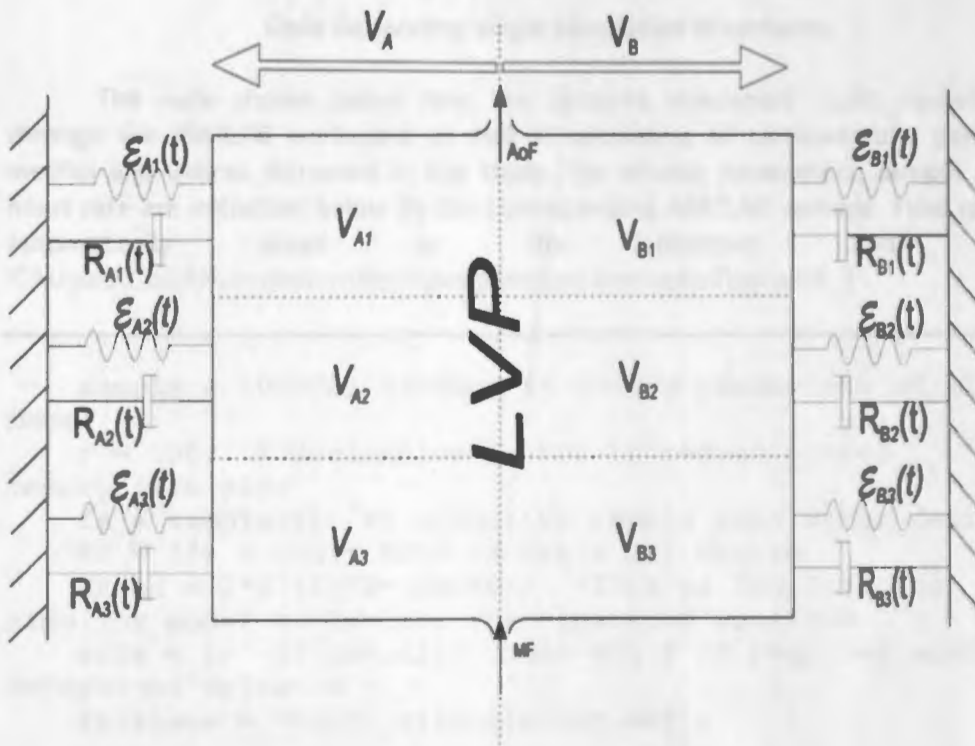
Vollmann, D. L.-B. (2006). Biventricular Pacing Improves the Blunt Force-Frequency Relation Present During Univentricular Pacing in Patients With Heart Failure and Conduction Delay. *Circulation* , 113, 953-959.

Wang, J. D.-A. (2008). Preserved left ventricular twist and circumferential deformation, but depressed longitudinal and radial deformation in patients with diastolic heart failure. *European Heart Journal* , 29, 1283-1289.

Winslow, R. J. (1999). Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, II model studies. *Circulation Research* , 84, 571-586.

Zhang, G. (2005). Evaluating the viscoelastic properties of biological tissues in a new way. *5* (1), 85-90.

## APPENDIX A: ADDITIONAL FIGURE



**Figure 23. Six-Segment Heart Model**

## APPENDIX B: MATLAB CODE

### Code Generating Single Simulation Waveforms

The code shown below runs the Simulink simulation 'multi\_viscoelastic.mdl' through the MATLAB workspace as well as calculating all cardiovascular performance metrics and indices discussed in this study. The muscle parameters, sample rate, and heart rate are initialized below by the corresponding MATLAB variable. Final results are automatically saved to the directory and filename 'C:\Users\Cody\Documents\thesis\programs\paramswEEP4\param4\_1'.

---

```
sample = 100000; %%check in config parameters of simulink
model
r = 100; % decimation factor to reduce stored
memory/file size
fs = sample/r; %% effective sample rate after decimation
HR = 90; % Heart Rate in Beats Per Minute
HRrad = 2*3.14159*(HR/60); %This is fed into the
simulink model to control the elastance waveform
size = 1; %% manually count the # of required sims
determined below-->
filename = 'multi_viscoelastic.mdl';

final_matrix = zeros(9,5); %% preallocate to increase
speed
matrix1 = zeros(1,size);
matrix1_total = zeros(1,size);
x_values = zeros(1,size);

for n = 1:size

    Eal_max = 36+.1*(n-1);
    Eb1_max = 36;
    Ea2_max = 36;
    Eb2_max = 36;
    Eal_min = .56;
    Eb1_min = .56;
    Ea2_min = .56;
    Eb2_min = .56;
    td1 = 0.01;
    td2 = 0;
    td3 = 0;
    td4 = 0;
    k1 = .005;
```

```

k2 = .0015;
k3 = .0015;
k4 = .0015;

Eeq = 1/Ea1_max + 1/Eb1_max + 1/Ea2_max + 1/Eb2_max;
Eeq = 1/Eeq;

simOut = sim(filename);

base = 'visco';
numframe = num2str(n);

visco.(genvarname(['values' numframe])) = [Ea1_max
Ea1_min Eb1_max Eb1_min Ea2_max Ea2_min Eb2_max Eb2_min td1
td2 td3 td4 k1 k2 k3 k4];
data(n).values = [Ea1_max Ea1_min Eb1_max Eb1_min Ea2_max
Ea2_min Eb2_max Eb2_min td1 td2 td3 td4 k1 k2 k3 k4];
visco.(genvarname(['lvp' numframe])) =
decimate(values.signals.values(:,1),r);
visco.(genvarname(['aop' numframe])) =
decimate(values.signals.values(:,2),r);
visco.(genvarname(['aof' numframe])) =
decimate(values.signals.values(:,3),r);
visco.(genvarname(['mf' numframe])) =
decimate(values.signals.values(:,4),r);
visco.(genvarname(['Val' numframe])) =
decimate(values.signals.values(:,5),r);
visco.(genvarname(['Vb1' numframe])) =
decimate(values.signals.values(:,6),r);
visco.(genvarname(['Va2' numframe])) =
decimate(values.signals.values(:,7),r);
visco.(genvarname(['Vb2' numframe])) =
decimate(values.signals.values(:,8),r);
visco.(genvarname(['VT' numframe])) =
decimate(values.signals.values(:,9),r);
visco.(genvarname(['Ea1' numframe])) =
decimate(values.signals.values(:,10),r);
visco.(genvarname(['Eb1' numframe])) =
decimate(values.signals.values(:,11),r);
visco.(genvarname(['Ea2' numframe])) =
decimate(values.signals.values(:,12),r);
visco.(genvarname(['Eb2' numframe])) =
decimate(values.signals.values(:,13),r);
visco.(genvarname(['DVT' numframe])) =
diff(visco.(genvarname(['VT' numframe])))*fs;
visco.(genvarname(['DVT' numframe]))(1:500) = 0;

```



```

        visco.(genvarname(['Dlvpa' numframe])) =
diff(visco.(genvarname(['lvp' numframe])));
        visco.(genvarname(['Dlvp' numframe])) =
diff(visco.(genvarname(['lvp' numframe])));
        visco.(genvarname(['Dlvp' numframe]))(1:7000) = 0;
        visco.(genvarname(['absDVT' numframe])) =
abs(visco.(genvarname(['DVT' numframe])));
        visco.(genvarname(['DVal' numframe])) =
diff(visco.(genvarname(['Val' numframe])))*fs;
        visco.(genvarname(['DVb1' numframe])) =
diff(visco.(genvarname(['Vb1' numframe])))*fs;
        visco.(genvarname(['DVa2' numframe])) =
diff(visco.(genvarname(['Va2' numframe])))*fs;
        visco.(genvarname(['DVb2' numframe])) =
diff(visco.(genvarname(['Vb2' numframe])))*fs;
        visco.(genvarname(['DVal' numframe]))(1:500) = 0;
        visco.(genvarname(['DVb1' numframe]))(1:500) = 0;
        visco.(genvarname(['DVa2' numframe]))(1:500) = 0;
        visco.(genvarname(['DVb2' numframe]))(1:500) = 0;
        visco.(genvarname(['Dlvp' numframe]))(1:500) = 0;
        visco.(genvarname(['absDVT' numframe]))(1:500) = 0;

```

%%%mitral flow beat index%%%%%%%%%

```

        visco.(genvarname(['mlindex' numframe])) =
find(visco.(genvarname(['mf'
numframe]))(400:length(visco.(genvarname(['mf'
numframe]))))>.1*max(visco.(genvarname(['mf'
numframe]))(1000:length(visco.(genvarname(['mf'
numframe]))))));
        visco.(genvarname(['mlindex' numframe])) =
visco.(genvarname(['mlindex' numframe])) + 400;
        visco.(genvarname(['m2index' numframe])) =
find(diff(visco.(genvarname(['mlindex' numframe])))-1);
        visco.(genvarname(['minindex' numframe])) =
visco.(genvarname(['mlindex'
numframe]))(visco.(genvarname(['m2index' numframe])) + 1);
        visco.(genvarname(['minindexEDV' numframe])) =
visco.(genvarname(['mlindex'
numframe]))(visco.(genvarname(['m2index' numframe])));

```

%%%end mitral flow beat index%%%

```

        visco.(genvarname(['EDV' numframe])) =
visco.(genvarname(['VT'
numframe]))(visco.(genvarname(['minindexEDV' numframe])) + 20);

```

```

        visco.(genvarname(['ESV' numframe])) =
visco.(genvarname(['VT'
numframe]))(visco.(genvarname(['mindex' numframe])) - 20);
        visco.(genvarname(['SV' numframe])) =
visco.(genvarname(['EDV' numframe])) -
visco.(genvarname(['ESV' numframe]));
        visco.(genvarname(['CO' numframe])) =
visco.(genvarname(['SV' numframe]))*HR/60;
        visco.(genvarname(['EF' numframe])) =
(visco.(genvarname(['SV'
numframe])))/visco.(genvarname(['EDV' numframe]))*100;

```

~~~~~aortic flow beat index~~~~~

```

        visco.(genvarname(['aoflindex' numframe])) =
find(visco.(genvarname(['aof'
numframe]))(800:length(visco.(genvarname(['aof'
numframe]))))>.05*max(visco.(genvarname(['aof'
numframe]))(1000:length(visco.(genvarname(['aof'
numframe]))))) );
        visco.(genvarname(['aoflindex' numframe])) =
visco.(genvarname(['aoflindex' numframe])) + 800;
        visco.(genvarname(['aof2index' numframe])) =
find(diff(visco.(genvarname(['aoflindex' numframe]))~1);
        visco.(genvarname(['aofindex' numframe])) =
visco.(genvarname(['aoflindex'
numframe]))(visco.(genvarname(['aof2index' numframe])) + 1);
        visco.(genvarname(['aofindexEDV' numframe])) =
visco.(genvarname(['aoflindex'
numframe]))(visco.(genvarname(['aof2index' numframe])));

```

~~~~~end aortic flow beat index~~~~~

```

        visco.(genvarname(['EWtop' numframe])) = -
trapz(visco.(genvarname(['VT'
numframe]))((visco.(genvarname(['aofindex' numframe]))(6)-
23):(visco.(genvarname(['aofindexEDV'
numframe]))(7)+27)),visco.(genvarname(['lvp'
numframe]))((visco.(genvarname(['aofindex' numframe]))(6)-
23):(visco.(genvarname(['aofindexEDV' numframe]))(7)+27)));
        visco.(genvarname(['EWbottom' numframe])) =
trapz(visco.(genvarname(['VT'
numframe]))((visco.(genvarname(['mindex' numframe]))(6)-
2):(visco.(genvarname(['mindexEDV'
numframe]))(7))+0),visco.(genvarname(['lvp'
numframe]))((visco.(genvarname(['mindex' numframe]))(6)-
2):(visco.(genvarname(['mindexEDV' numframe]))(7)+0)));

```

```

        visco.(genvarname(['EW' numframe])) =
visco.(genvarname(['EWtop' numframe])) -
visco.(genvarname(['EWbottom' numframe]));
        visco.(genvarname(['EW' numframe])) =
visco.(genvarname(['EW' numframe]))*133.322/1e6; %% convert
from mmHg*mL --> Joules

        visco.(genvarname(['numB' numframe])) =
visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindexEDV' numframe]))(6))
- Eeq*visco.(genvarname(['ESV' numframe]))(6);
        visco.(genvarname(['numX' numframe])) =
0:.001:visco.(genvarname(['ESV' numframe]))(6);
        visco.(genvarname(['numY' numframe])) =
Eeq*visco.(genvarname(['numX' numframe])) +
visco.(genvarname(['numB' numframe]));
        visco.(genvarname(['PW' numframe])) =
trapz(visco.(genvarname(['numX'
numframe])),visco.(genvarname(['numY' numframe])));
        visco.(genvarname(['PW' numframe])) =
visco.(genvarname(['PW' numframe]))*133.322/1e6; %% convert
from mmHg*mL --> Joules
        visco.(genvarname(['PVA' numframe])) =
visco.(genvarname(['PW' numframe])) + visco.(genvarname(['EW'
numframe]));
        visco.(genvarname(['PWseg' numframe])) =
visco.(genvarname(['PW' numframe]))/4;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Wmyo %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
visco.(genvarname(['WmyoVal' numframe])) = 0;
visco.(genvarname(['WmyoVb1' numframe])) = 0;
visco.(genvarname(['WmyoVa2' numframe])) = 0;
visco.(genvarname(['WmyoVb2' numframe])) = 0;
visco.(genvarname(['Wmyo' numframe])) = 0;
for w = visco.(genvarname(['minindexEDV'
numframe]))(6):visco.(genvarname(['minindexEDV' numframe]))(7)
%returns the data points for one cardiac cycle starting at
the 6th beat
        if(visco.(genvarname(['DVal' numframe]))(w) < 0)
            visco.(genvarname(['WmyoVal' numframe])) =
visco.(genvarname(['WmyoVal' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp' numframe]))(w+1))/2
            *abs((visco.(genvarname(['Val' numframe]))(w) -
visco.(genvarname(['Val' numframe]))(w+1)));
        end

```

```

    if(visco.(genvarname(['DVb1' numframe]))(w) < 0)
        visco.(genvarname(['WmyoVb1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Vb1'
numframe]))(w) - visco.(genvarname(['Vb1' numframe]))(w+1)));
    end
    if(visco.(genvarname(['DVa2' numframe]))(w) < 0)
        visco.(genvarname(['WmyoVa2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Va2'
numframe]))(w) - visco.(genvarname(['Va2' numframe]))(w+1)));
    end
    if(visco.(genvarname(['DVb2' numframe]))(w) < 0)
        visco.(genvarname(['WmyoVb2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Vb2'
numframe]))(w) - visco.(genvarname(['Vb2' numframe]))(w+1)));
    end
end

visco.(genvarname(['WlossA1' numframe])) =
sum(k1*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVal'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVal'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;
visco.(genvarname(['WlossB1' numframe])) =
sum(k2*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVb1'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVb1'
numframe]))(visco.(genvarname(['aofindex'

```

```

numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;
    visco.(genvarname(['WlossA2' numframe])) =
sum(k3*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['Dva2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['Dva2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;
    visco.(genvarname(['WlossB2' numframe])) =
sum(k4*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVb2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVb2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;
    visco.(genvarname(['Wloss' numframe])) =
visco.(genvarname(['WlossA1' numframe])) +
visco.(genvarname(['WlossB1' numframe])) +
visco.(genvarname(['WlossA2' numframe])) +
visco.(genvarname(['WlossB2' numframe]));

    visco.(genvarname(['WmyoVal' numframe])) =
visco.(genvarname(['WmyoVal' numframe]))*133.322/1e6; %
convert from mmHg*mL --> Joules
    visco.(genvarname(['WmyoVb1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe]))*133.322/1e6;
    visco.(genvarname(['WmyoVa2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe]))*133.322/1e6;
    visco.(genvarname(['WmyoVb2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe]))*133.322/1e6;
    %visco.(genvarname(['Wloss' numframe])) =
visco.(genvarname(['Wloss' numframe]))*133.322/1e6; %
convert from mmHg*mL --> Joules
    visco.(genvarname(['Wmyo' numframe])) =
visco.(genvarname(['WmyoVal' numframe])) +
visco.(genvarname(['WmyoVb1' numframe])) +

```

```

visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['WmyoVb2' numframe]));
    visco.(genvarname(['PVAmYo' numframe])) =
visco.(genvarname(['PW' numframe])) +
visco.(genvarname(['Wmyo' numframe])) +
visco.(genvarname(['Wloss' numframe]))*(60/HR); %%multiply
Wloss *2/3 because Wloss is in units of Watts (J/s), and one
cycle is 60/90

```

```

    visco.(genvarname(['PVAmYoA1' numframe])) =
visco.(genvarname(['WmyoVa1' numframe])) +
visco.(genvarname(['WmyoVa1' numframe])) +
visco.(genvarname(['PWseg' numframe]));
    visco.(genvarname(['PVAmYoB1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['PWseg' numframe]));
    visco.(genvarname(['PVAmYoA2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['PWseg' numframe]));
    visco.(genvarname(['PVAmYoB2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe])) +
visco.(genvarname(['WmyoVb2' numframe])) +
visco.(genvarname(['PWseg' numframe]));

```

```

    %%%%%%%%%%%%%%%%% end Wmyo %%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%Internal Flow modeled by a Conductance
Catheter%%%%%%%%%%%%%%%%%%%%%%%%

```

```

    visco.(genvarname(['V1_cc' numframe])) =
visco.(genvarname(['Va1' numframe])) +
visco.(genvarname(['Vb1' numframe]));
    visco.(genvarname(['V2_cc' numframe])) =
visco.(genvarname(['Va2' numframe])) +
visco.(genvarname(['Vb2' numframe]));
    visco.(genvarname(['DV1_cc' numframe])) =
diff(visco.(genvarname(['V1_cc' numframe]))).*fs;
    visco.(genvarname(['DV2_cc' numframe])) =
diff(visco.(genvarname(['V2_cc' numframe]))).*fs;
    visco.(genvarname(['DV1_cc' numframe]))(1:500) = 0;
    visco.(genvarname(['DV2_cc' numframe]))(1:500) = 0;

    visco.(genvarname(['Dvt_cc' numframe])) =
abs(visco.(genvarname(['DV1_cc' numframe])) +
abs(visco.(genvarname(['DV2_cc' numframe])));

```



```

        visco.(genvarname(['IF_cc' numframe])) =
        (visco.(genvarname(['Dvt_cc' numframe])) -
        visco.(genvarname(['absDVT' numframe])))/2;

        for j = 1:length(visco.(genvarname(['mindex' numframe])))
- 1
            dns.allIF(j,1) = trapz(visco.(genvarname(['IF_cc'
numframe]))(visco.(genvarname(['mindex'
numframe]))(j):visco.(genvarname(['mindex'
numframe]))(j+1))*1/fs);
            dns.allDVT(j,1) = trapz(visco.(genvarname(['absDVT'
numframe]))(visco.(genvarname(['mindex'
numframe]))(j):visco.(genvarname(['mindex'
numframe]))(j+1))*1/fs);
        end
        visco.(genvarname(['IFF' numframe])) =
        dns.allIF./dns.allDVT;
        visco.(genvarname(['IFFpercent' numframe])) =
        visco.(genvarname(['IFF' numframe])).*100;

        visco.(genvarname(['stdIFF' numframe])) =
        std(visco.(genvarname(['IFF' numframe])));
        visco.(genvarname(['meanIFF' numframe])) =
        mean(visco.(genvarname(['IFF' numframe])));
        visco.(genvarname(['meanIFFpercent' numframe])) =
        visco.(genvarname(['meanIFF' numframe])).*100;
        data(n).std = visco.(genvarname(['stdIFF' numframe]));
        data(n).meanIFF = visco.(genvarname(['meanIFF'
numframe]));
        data(n).meanIFFpercent =
        visco.(genvarname(['meanIFFpercent' numframe]));

        %%%end IFF by Conductance Catheter%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Internal Flow modeled by 4 segment
        dyssynchrony data%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        visco.(genvarname(['IF_total_cc' numframe])) =
        ((abs(visco.(genvarname(['DVa1' numframe])))) +
        abs(visco.(genvarname(['DVa2' numframe])))) +
        abs(visco.(genvarname(['DVb1' numframe])))) +
        abs(visco.(genvarname(['DVb2' numframe])))) -
        visco.(genvarname(['absDVT' numframe])))/2;

```

```

    for j = 1:length(visco.(genvarname(['mindex' numframe])))
- 1
        dns.allIF_total(j,1) =
trapz(visco.(genvarname(['IF_total_cc'
numframe]))(visco.(genvarname(['mindex'
numframe]))(j):visco.(genvarname(['mindex'
numframe]))(j+1))*1/fs);
        end
        visco.(genvarname(['IFF_total' numframe])) =
dns.allIF_total./dns.allDVT;
        visco.(genvarname(['stdIFF_total' numframe])) =
std(visco.(genvarname(['IFF_total' numframe])));
        visco.(genvarname(['meanIFF_total' numframe])) =
mean(visco.(genvarname(['IFF_total' numframe])));
        visco.(genvarname(['meanIFFpercent_total' numframe])) =
visco.(genvarname(['meanIFF_total' numframe])).*100;
        data(n).std_total = visco.(genvarname(['stdIFF_total'
numframe]));
        data(n).meanIFF_total =
visco.(genvarname(['meanIFF_total' numframe]));
        data(n).meanIFFpercent_total =
visco.(genvarname(['meanIFFpercent_total' numframe]));

n

    directory1 =
'C:\Users\Cody\Documents\thesis\programs\paramsweep4\param4_';
;
    directory2 = strcat(directory1, numframe);
    directory2 = char(directory2);
    save(directory2, '-struct', 'visco')

    matrix1(1,n) = data(n).meanIFFpercent;
    matrix1_total(1,n) = data(n).meanIFFpercent_total;

    x_values(1,n) = Eal_max - Eb2_max;
    data(n).COplot = visco.(genvarname(['CO' numframe]));
    data(n).SVplot = visco.(genvarname(['SV' numframe]));
    data(n).EFplot = visco.(genvarname(['EF' numframe]));
    data(n).EWplot = visco.(genvarname(['EW' numframe]));

    COplot(1,n) = visco.(genvarname(['CO' numframe]))(4);
    SVplot(1,n) = visco.(genvarname(['SV' numframe]))(4);
    EFplot(1,n) = visco.(genvarname(['EF' numframe]))(4);
    EWplot(1,n) = visco.(genvarname(['EW' numframe]));
    PVAplot(1,n) = visco.(genvarname(['PVA' numframe]));

```



```

clear visco
clear values
clear dns

end

```

---

### Code Generating Multiple Simulation Tests for Elastance and Delay

The code shown below runs the Simulink simulation 'multi\_viscoelastic.mdl' through the MATLAB workspace as well as calculating all cardiovascular performance metrics and indices discussed in this study. The code will run 2601 simulations with each representing unique combinations of maximum elastance and elastance delay of segment A1. This was used to construct the 3-Dimensional plots of Figures 18-23. The muscle parameters, sample rate, and heart rate are initialized below by the corresponding MATLAB variable. Final results are automatically saved to the directory and filename 'C:\Users\Cody\Documents\thesis\programs\paramsweep5\param5\_'.

---

```

sample = 100000; %%check in config parameters of simulink
model. IMPORTANT: IF you change sample rate, also change 'r'
by the same factor to keep program consistent throughout
    r = 100; % decimation factor to reduce stored
memory/file size
    fs = sample/r; %% effective sample rate after decimation
    HR = 90; % Heart Rate in Beats Per Minute
    HRrad = 2*3.14159*(HR/60); %This is fed into the
simulink model to control the elastance waveform
    size = 51; %% manually count the # of required sims
determined below-->

    filename = 'multi_viscoelastic.mdl';

    final_matrix = zeros(9,5); %%% preallocate to increase
speed
    matrix1 = zeros(1,size);
    matrix1_total = zeros(1,size);
    x_values = zeros(1,size);
    Z = zeros(51,51);
    ZZ = zeros(51,51);

```

```

CO = zeros(51,51);
SV = zeros(51,51);
EF = zeros(51,51);
EW = zeros(51,51);
PVA = zeros(51,51);
EFF = zeros(size,size);
EFFmyo = zeros(size,size);
Wmyo = zeros(size,size);
Wloss = zeros(size,size);
PVAmYo = zeros(size,size);
WmyoA1 = zeros(size,size);
WmyoB1 = zeros(size,size);
WmyoA2 = zeros(size,size);
WmyoB2 = zeros(size,size);
WlossA1 = zeros(size,size);
WlossB1 = zeros(size,size);
WlossA2 = zeros(size,size);
WlossB2 = zeros(size,size);
PVAmYoA1 = zeros(size,size);
PVAmYoB1 = zeros(size,size);
PVAmYoA2 = zeros(size,size);
PVAmYoB2 = zeros(size,size);
PW = zeros(size,size);

for n = 1:size
    for m = 1:size
        Ea1_max = 20+.6*(n-1);  %%% if this is changed,
change the 3d matrix size on line 262 for variable X
        Eb1_max = 36;
        Ea2_max = 36;
        Eb2_max = 36;
        Ea1_min = .56;
        Eb1_min = .56;
        Ea2_min = .56;
        Eb2_min = .56;
        td1 = 0 + .001*(m-1);  %%% if this is changed,
change the 3d matrix size on line 262 for variable X
        td2 = 0;
        td3 = 0;
        td4 = 0;
        k1 = .0015;
        k2 = .0015;
        k3 = .0015;
        k4 = .0015;
    end
end

```

```

Eeq = 1/Ea1_max + 1/Eb1_max + 1/Ea2_max +
1/Eb2_max; %%elastance elements are in parallel
Eeq = 1/Eeq; %% equivalent elastance

simOut = sim(filename);

base = 'visco';
numframe1 = num2str(n);
numframe2 = strcat('_', numframe1);
numframe3 = num2str(m);
numframe4 = strcat('_', numframe3);
numframe = strcat(numframe2, numframe4);

visco.(genvarname(['values' numframe])) =
[Ea1_max Ea1_min Eb1_max Eb1_min Ea2_max Ea2_min Eb2_max
Eb2_min td1 td2 td3 td4 k1 k2 k3 k4];
data(n,m).values = [Ea1_max Ea1_min Eb1_max
Eb1_min Ea2_max Ea2_min Eb2_max Eb2_min td1 td2 td3 td4 k1 k2
k3 k4];

visco.(genvarname(['lvp' numframe])) =
decimate(values.signals.values(:,1),r);
visco.(genvarname(['aop' numframe])) =
decimate(values.signals.values(:,2),r);
visco.(genvarname(['aof' numframe])) =
decimate(values.signals.values(:,3),r);
visco.(genvarname(['mf' numframe])) =
decimate(values.signals.values(:,4),r);
visco.(genvarname(['Val' numframe])) =
decimate(values.signals.values(:,5),r);
visco.(genvarname(['Vb1' numframe])) =
decimate(values.signals.values(:,6),r);
visco.(genvarname(['Va2' numframe])) =
decimate(values.signals.values(:,7),r);
visco.(genvarname(['Vb2' numframe])) =
decimate(values.signals.values(:,8),r);
visco.(genvarname(['VT' numframe])) =
decimate(values.signals.values(:,9),r);
visco.(genvarname(['Ea1' numframe])) =
decimate(values.signals.values(:,10),r);
visco.(genvarname(['Eb1' numframe])) =
decimate(values.signals.values(:,11),r);
visco.(genvarname(['Ea2' numframe])) =
decimate(values.signals.values(:,12),r);

```

```

        visco.(genvarname(['Eb2' numframe])) =
decimate(values.signals.values(:,13),r);
        visco.(genvarname(['DVT' numframe])) =
diff(visco.(genvarname(['VT' numframe]))*fs;
        visco.(genvarname(['DVT' numframe]))(1:500) = 0;
        visco.(genvarname(['Dlvp' numframe])) =
diff(visco.(genvarname(['lvp' numframe])));
        visco.(genvarname(['Dlvp' numframe])) =
diff(visco.(genvarname(['lvp' numframe])));
        visco.(genvarname(['Dlvp' numframe]))(1:7000) =
0;

        visco.(genvarname(['absDVT' numframe])) =
abs(visco.(genvarname(['DVT' numframe])));
        visco.(genvarname(['DVal' numframe])) =
diff(visco.(genvarname(['Val' numframe]))*fs;
        visco.(genvarname(['DVb1' numframe])) =
diff(visco.(genvarname(['Vb1' numframe]))*fs;
        visco.(genvarname(['DVal2' numframe])) =
diff(visco.(genvarname(['Va2' numframe]))*fs;
        visco.(genvarname(['DVb2' numframe])) =
diff(visco.(genvarname(['Vb2' numframe]))*fs;
        visco.(genvarname(['DVal' numframe]))(1:500) = 0;
        visco.(genvarname(['DVb1' numframe]))(1:500) = 0;
        visco.(genvarname(['DVal2' numframe]))(1:500) = 0;
        visco.(genvarname(['DVb2' numframe]))(1:500) = 0;
        visco.(genvarname(['Dlvp' numframe]))(1:500) = 0;

        %%%mitral flow beat index%%%%%%%%%%

        visco.(genvarname(['mlindex' numframe])) =
find(visco.(genvarname(['mf'
numframe]))(400:length(visco.(genvarname(['mf'
numframe]))))>.1*max(visco.(genvarname(['mf'
numframe]))(1000:length(visco.(genvarname(['mf'
numframe]))))) );
        visco.(genvarname(['mlindex' numframe])) =
visco.(genvarname(['mlindex' numframe])) + 400;
        visco.(genvarname(['m2index' numframe])) =
find(diff(visco.(genvarname(['mlindex' numframe]))~1);
        visco.(genvarname(['mindex' numframe])) =
visco.(genvarname(['mlindex'
numframe]))(visco.(genvarname(['m2index' numframe])) + 1);
        visco.(genvarname(['mindexEDV' numframe])) =
visco.(genvarname(['mlindex'
numframe]))(visco.(genvarname(['m2index' numframe])));

```

```

%      %%%%end mitral flow beat index%%%

        visco.(genvarname(['EDV' numframe])) =
visco.(genvarname(['VT'
numframe]))(visco.(genvarname(['minindexEDV' numframe])) + 20);
        visco.(genvarname(['ESV' numframe])) =
visco.(genvarname(['VT'
numframe]))(visco.(genvarname(['minindex' numframe])) - 20);
        visco.(genvarname(['SV' numframe])) =
visco.(genvarname(['EDV' numframe])) -
visco.(genvarname(['ESV' numframe]));
        visco.(genvarname(['CO' numframe])) =
visco.(genvarname(['SV' numframe]))*HR/60;
        visco.(genvarname(['EF' numframe])) =
(visco.(genvarname(['SV'
numframe]))./visco.(genvarname(['EDV' numframe])))*100;

%      %%%aortic flow beat index%%%%%%%%%%

        visco.(genvarname(['aoflindex' numframe])) =
find(visco.(genvarname(['aof'
numframe]))(800:length(visco.(genvarname(['aof'
numframe]))))>.05*max(visco.(genvarname(['aof'
numframe]))(1000:length(visco.(genvarname(['aof'
numframe]))))));
        visco.(genvarname(['aoflindex' numframe])) =
visco.(genvarname(['aoflindex' numframe])) + 800;
        visco.(genvarname(['aof2index' numframe])) =
find(diff(visco.(genvarname(['aoflindex' numframe]))~1);
        visco.(genvarname(['aofindex' numframe])) =
visco.(genvarname(['aoflindex'
numframe]))(visco.(genvarname(['aof2index' numframe])) + 1);
        visco.(genvarname(['aofindexEDV' numframe])) =
visco.(genvarname(['aoflindex'
numframe]))(visco.(genvarname(['aof2index' numframe])));

%      %%%%end aortic flow beat index%%%

        visco.(genvarname(['EWtop' numframe])) = -
trapz(visco.(genvarname(['VT'
numframe]))((visco.(genvarname(['aofindex' numframe]))(6)-
23):(visco.(genvarname(['aofindexEDV'
numframe]))(7)+27)),visco.(genvarname(['lvp'
numframe]))((visco.(genvarname(['aofindex' numframe]))(6)-
23):(visco.(genvarname(['aofindexEDV' numframe]))(7)+27)));
        visco.(genvarname(['EWbottom' numframe])) =
trapz(visco.(genvarname(['VT'

```

```

numframe]))((visco.(genvarname(['mindex' numframe]))(6)-
2):(visco.(genvarname(['mindexEDV'
numframe]))(7))+0),visco.(genvarname(['lvp'
numframe]))((visco.(genvarname(['mindex' numframe]))(6)-
2):(visco.(genvarname(['mindexEDV' numframe]))(7)+0)));
    visco.(genvarname(['EW' numframe])) =
visco.(genvarname(['EWtop' numframe])) -
visco.(genvarname(['EWbottom' numframe]));
    visco.(genvarname(['EW' numframe])) =
visco.(genvarname(['EW' numframe]))*133.322/1e6; %% convert
from mmHg*mL --> Joules

    visco.(genvarname(['numB' numframe])) =
visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindexEDV' numframe]))(6))
- Eeq*visco.(genvarname(['ESV' numframe]))(6);
    visco.(genvarname(['numX' numframe])) =
0:.001:visco.(genvarname(['ESV' numframe]))(6);
    visco.(genvarname(['numY' numframe])) =
Eeq*visco.(genvarname(['numX' numframe])) +
visco.(genvarname(['numB' numframe]));
    visco.(genvarname(['PW' numframe])) =
trapz(visco.(genvarname(['numX'
numframe])),visco.(genvarname(['numY' numframe])));
    visco.(genvarname(['PW' numframe])) =
visco.(genvarname(['PW' numframe]))*133.322/1e6; %% convert
from mmHg*mL --> Joules
    visco.(genvarname(['PVA' numframe])) =
visco.(genvarname(['PW' numframe])) + visco.(genvarname(['EW'
numframe]));
    visco.(genvarname(['PWseg' numframe])) =
visco.(genvarname(['PW' numframe]))/4;

##### Wmyo #####
visco.(genvarname(['WmyoVal' numframe])) = 0;
visco.(genvarname(['WmyoVb1' numframe])) = 0;
visco.(genvarname(['WmyoVa2' numframe])) = 0;
visco.(genvarname(['WmyoVb2' numframe])) = 0;
visco.(genvarname(['Wmyo' numframe])) = 0;
for w = visco.(genvarname(['mindexEDV'
numframe]))(6):visco.(genvarname(['mindexEDV' numframe]))(7)
%returns the data points for one cardiac cycle starting at
the 6th beat
    if(visco.(genvarname(['DVal' numframe]))(w) < 0)
        visco.(genvarname(['WmyoVal' numframe])) =
visco.(genvarname(['WmyoVal' numframe])) +

```

```

(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp' numframe]))(w+1))/2
*abs((visco.(genvarname(['Val' numframe]))(w) -
visco.(genvarname(['Val' numframe]))(w+1)));
end
if(visco.(genvarname(['DVb1' numframe]))(w) < 0)
    visco.(genvarname(['WmyoVb1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Vb1'
numframe]))(w) - visco.(genvarname(['Vb1' numframe]))(w+1)));
end
if(visco.(genvarname(['DVa2' numframe]))(w) < 0)
    visco.(genvarname(['WmyoVa2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Va2'
numframe]))(w) - visco.(genvarname(['Va2' numframe]))(w+1)));
end
if(visco.(genvarname(['DVb2' numframe]))(w) < 0)
    visco.(genvarname(['WmyoVb2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Vb2'
numframe]))(w) - visco.(genvarname(['Vb2' numframe]))(w+1)));
end
end

visco.(genvarname(['WlossA1' numframe])) =
sum(k1*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVal'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVal'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;
visco.(genvarname(['WlossB1' numframe])) =
sum(k2*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'

```



```

numframe]))(7)).*visco.(genvarname(['DVb1'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVb1'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;
    visco.(genvarname(['WlossA2' numframe])) =
sum(k3*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVA2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVA2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;
    visco.(genvarname(['WlossB2' numframe])) =
sum(k4*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVb2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVb2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;
    visco.(genvarname(['Wloss' numframe])) =
visco.(genvarname(['WlossA1' numframe])) +
visco.(genvarname(['WlossB1' numframe])) +
visco.(genvarname(['WlossA2' numframe])) +
visco.(genvarname(['WlossB2' numframe]));

    visco.(genvarname(['WmyoVal' numframe])) =
visco.(genvarname(['WmyoVal' numframe]))*133.322/1e6;
convert from mmHg*mL --> Joules
    visco.(genvarname(['WmyoVb1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe]))*133.322/1e6;
    visco.(genvarname(['WmyoVa2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe]))*133.322/1e6;
    visco.(genvarname(['WmyoVb2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe]))*133.322/1e6;

```



```

    visco.(genvarname(['Wloss' numframe])) =
visco.(genvarname(['Wloss' numframe]))*133.322/1e6; %%
convert from mmHg*mL --> Joules
    visco.(genvarname(['Wmyo' numframe])) =
visco.(genvarname(['WmyoVal' numframe])) +
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['WmyoVb2' numframe]));
    visco.(genvarname(['PVAmyo' numframe])) =
visco.(genvarname(['PW' numframe])) +
visco.(genvarname(['Wmyo' numframe])) +
visco.(genvarname(['Wloss' numframe]))*(60/HR); %%multiply
Wloss *2/3 because Wloss is in units of Watts (J/s), and one
cycle is 60/90

```

```

    visco.(genvarname(['PVAmyoA1' numframe])) =
visco.(genvarname(['WmyoVal' numframe])) +
visco.(genvarname(['WmyoVal' numframe])) +
visco.(genvarname(['PWseg' numframe]));
    visco.(genvarname(['PVAmyoB1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['PWseg' numframe]));
    visco.(genvarname(['PVAmyoA2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['PWseg' numframe]));
    visco.(genvarname(['PVAmyoB2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe])) +
visco.(genvarname(['WmyoVb2' numframe])) +
visco.(genvarname(['PWseg' numframe]));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% end Wmyo %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

-- %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Internal Flow modeled by a
Conductance Catheter%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

    visco.(genvarname(['V1_cc' numframe])) =
visco.(genvarname(['Val' numframe])) +
visco.(genvarname(['Vb1' numframe]));
    visco.(genvarname(['V2_cc' numframe])) =
visco.(genvarname(['Va2' numframe])) +
visco.(genvarname(['Vb2' numframe]));
    visco.(genvarname(['DV1_cc' numframe])) =
diff(visco.(genvarname(['V1_cc' numframe]))).*fs;

```

```

        visco.(genvarname(['DV2_cc' numframe])) =
diff(visco.(genvarname(['V2_cc' numframe]))).*fs;
        visco.(genvarname(['DV1_cc' numframe]))(1:500) =
0;
        visco.(genvarname(['DV2_cc' numframe]))(1:500) =
0;

        visco.(genvarname(['Dvt_cc' numframe])) =
abs(visco.(genvarname(['DV1_cc' numframe])))+
abs(visco.(genvarname(['DV2_cc' numframe])));
        visco.(genvarname(['IF_cc' numframe])) =
(visco.(genvarname(['Dvt_cc' numframe])) -
visco.(genvarname(['absDVT' numframe])))/2;

        for j = 1:length(visco.(genvarname(['mindex'
numframe])))-1
            dns.allIF(j,1) =
trapz(visco.(genvarname(['IF_cc'
numframe]))(visco.(genvarname(['mindex'
numframe]))(j):visco.(genvarname(['mindex'
numframe]))(j+1))*1/fs);
            dns.allDVT(j,1) =
trapz(visco.(genvarname(['absDVT'
numframe]))(visco.(genvarname(['mindex'
numframe]))(j):visco.(genvarname(['mindex'
numframe]))(j+1))*1/fs);
        end
        visco.(genvarname(['IFF' numframe])) =
dns.allIF./dns.allDVT;
        visco.(genvarname(['IFFpercent' numframe])) =
visco.(genvarname(['IFF' numframe])).*100;

        visco.(genvarname(['stdIFF' numframe])) =
std(visco.(genvarname(['IFF' numframe])));
        visco.(genvarname(['meanIFF' numframe])) =
mean(visco.(genvarname(['IFF' numframe])));
        visco.(genvarname(['meanIFFpercent' numframe])) =
visco.(genvarname(['meanIFF' numframe])).*100;
        data(n,m).std = visco.(genvarname(['stdIFF'
numframe]));
        data(n,m).meanIFF = visco.(genvarname(['meanIFF'
numframe]));
        data(n,m).meanIFFpercent =
visco.(genvarname(['meanIFFpercent' numframe]));

```

#####end IFF by Conductance Catheter#####

#####Internal Flow modeled by 4 segment  
dyssynchrony data#####

```
visco.(genvarname(['IF_total_cc' numframe])) =  
((abs(visco.(genvarname(['DVa1' numframe])))) +  
abs(visco.(genvarname(['DVa2' numframe])))) +  
abs(visco.(genvarname(['DVB1' numframe])))) +  
abs(visco.(genvarname(['DVB2' numframe])))) -  
visco.(genvarname(['absDVT' numframe])))/2;
```

```
for j = 1:length(visco.(genvarname(['mindex'  
numframe])) - 1  
    dns.allIF_total(j,1) =  
trapz(visco.(genvarname(['IF_total_cc'  
numframe]))(visco.(genvarname(['mindex'  
numframe]))(j):visco.(genvarname(['mindex'  
numframe]))(j+1))*1/fs);  
end  
visco.(genvarname(['IFF_total' numframe])) =  
dns.allIF_total./dns.allDVT;  
visco.(genvarname(['stdIFF_total' numframe])) =  
std(visco.(genvarname(['IFF_total' numframe])));  
visco.(genvarname(['meanIFF_total' numframe])) =  
mean(visco.(genvarname(['IFF_total' numframe])));  
visco.(genvarname(['meanIFFpercent_total'  
numframe])) = visco.(genvarname(['meanIFF_total'  
numframe])).*100;
```

```
data(n,m).std_total =  
visco.(genvarname(['stdIFF_total' numframe]));  
data(n,m).meanIFF_total =  
visco.(genvarname(['meanIFF_total' numframe]));  
data(n,m).meanIFFpercent_total =  
visco.(genvarname(['meanIFFpercent_total' numframe]));
```

numframe

```
directory1 =  
'C:\Users\Cody\Documents\thesis\programs\paramsweep5\param5';  
directory2 = strcat(directory1, numframe);  
directory2 = char(directory2);
```

```

save(directory2, '-struct', 'visco')

Z(n,m) = data(n,m).meanIFFpercent;
ZZ(n,m) = data(n,m).meanIFFpercent_total;
CO(n,m) = visco.(genvarname(['CO' numframe]))(4);
SV(n,m) = visco.(genvarname(['SV' numframe]))(4);
EF(n,m) = visco.(genvarname(['EF' numframe]))(4);
EW(n,m) = visco.(genvarname(['EW' numframe]));
PVA(n,m) = visco.(genvarname(['PVA' numframe]));
Wmyo(n,m) = visco.(genvarname(['Wmyo'
numframe]));
PVAmYo(n,m) = visco.(genvarname(['PVAmYo'
numframe]));
Wloss(n,m) = visco.(genvarname(['Wloss'
numframe]));
WmyoA1(n,m) = visco.(genvarname(['WmyoVa1'
numframe]));
WmyoB1(n,m) = visco.(genvarname(['WmyoVb1'
numframe]));
WmyoA2(n,m) = visco.(genvarname(['WmyoVa2'
numframe]));
WmyoB2(n,m) = visco.(genvarname(['WmyoVb2'
numframe]));
WlossA1(n,m) = visco.(genvarname(['WlossA1'
numframe]));
WlossB1(n,m) = visco.(genvarname(['WlossB1'
numframe]));
WlossA2(n,m) = visco.(genvarname(['WlossA2'
numframe]));
WlossB2(n,m) = visco.(genvarname(['WlossB2'
numframe]));
PVAmYoA1(n,m) = visco.(genvarname(['PVAmYoA1'
numframe]));
PVAmYoB1(n,m) = visco.(genvarname(['PVAmYoB1'
numframe]));
PVAmYoA2(n,m) = visco.(genvarname(['PVAmYoA2'
numframe]));
PVAmYoB2(n,m) = visco.(genvarname(['PVAmYoB2'
numframe]));
PW(n,m) = visco.(genvarname(['PW' numframe]));

clear visco
clear values
clear dns

```

```

end
end

save
C:\Users\Cody\Documents\thesis\programs\paramsweep5\workspace
data

Y = 20:.6:50;
X = 0:.001:.050;
X = X*1000;
subplot(2,1,1)
surfc(X,Y,Z)
xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Internal Flow Fraction
(%)'),title('conductance catheter dyssynchrony surface plot')
shading interp
colormap(jet)
colorbar
subplot(2,1,2)
surfc(X,Y,ZZ)
xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Internal Flow Fraction (%)'),title('full
dyssynchrony surface plot')
shading interp
colormap(jet)
colorbar
fh = figure(1);
set(fh, 'color', 'white');
figure
subplot(2,1,1)
surfc(X,Y,CO)
xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Cardiac Output (mL/sec)'),title('cardiac
output surface plot')
shading interp
colormap(jet)
colorbar
subplot(2,1,2)
surfc(X,Y,EF)
xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Ejection Fraction'),title('ejection
fraction surface plot')
shading interp

```

```

colormap(jet)
colorbar
fh = figure(2);
set(fh, 'color', 'white');
figure
subplot(2,1,1)
surfc(X,Y,EW)
xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('External Work (J)'),title('external work
surface plot')
shading interp
colormap(jet)
colorbar
subplot(2,1,2)
surfc(X,Y,PVA)
xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Pressure Volume Area
(J)'),title('puressure volume area (PVA) surface plot')
shading interp
colormap(jet)
colorbar
fh = figure(3);
set(fh, 'color', 'white');
figure
subplot(2,1,1)
surfc(X,Y,100*EW./PVA)
xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Efficiency (%)'),title('efficiency
(EW/PVA) surface plot')
shading interp
colormap(jet)
colorbar
subplot(2,1,2)
surfc(X,Y,SV)
xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Stroke Volume (mL)'),title('stroke volume
surface plot')
shading interp
colormap(jet)
colorbar
fh = figure(4);
set(fh, 'color', 'white');
figure
subplot(2,1,1)
surfc(X,Y,Wmyo)

```

```

xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Myocardium Work (J)'),title('Myocardium
Work (W_m_y_o) surface plot')
shading interp
colormap(jet)
colorbar
subplot(2,1,2)
surfc(X,Y,PVAm_yo)
xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Work (J)'),title('PVA_m_y_o surface
plot')
shading interp
colormap(jet)
colorbar
fh = figure(5);
set(fh, 'color', 'white');
figure
subplot(2,1,1)
surfc(X,Y,Wloss)
xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Power (W)'),title('Myocardium Loss
(W_l_o_s_s) surface plot')
shading interp
colormap(jet)
colorbar
subplot(2,1,2)
surfc(X,Y,100*EW./PVAm_yo)
xlabel('delay (ms)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Efficiency (%)'),title('Myocardium
efficiency surface plot')
shading interp
colormap(jet)
colorbar
fh = figure(6);
set(fh, 'color', 'white');

```

## Code Generating Multiple Simulation Tests for Elastance and 'K'

The code shown below runs the Simulink simulation 'multi\_viscoelastic.mdl' through the MATLAB workspace as well as calculating all cardiovascular performance metrics and indices discussed in this study. The code will run 2601 simulations with each representing unique combinations of maximum elastance and myocardial viscous friction 'K' of segment A1. This was used to construct the 3-Dimensional plots of Figures 18-23. The muscle parameters, sample rate, and heart rate are initialized below by the corresponding MATLAB variable. Final results are automatically saved to the directory and filename 'C:\Users\Cody\Documents\thesis\programs\paramswEEP7\param7\_'. All terminating workspace data is saved to the file and directory path 'C:\Users\Cody\Documents\thesis\programs\paramswEEP7\workspacedata'.

---

```
sample = 100000; %%check in config parameters of simulink
model
r = 100; % decimation factor to reduce stored
memory/file size
fs = sample/r; %% effective sample rate after decimation
HR = 90; % Heart Rate in Beats Per Minute
HRrad = 2*3.14159*(HR/60); %%This is fed into the
simulink model to control the elastance waveform
size = 51; %% manually count the # of required sims
determined below-->
filename = 'multi_viscoelastic.mdl';

final_matrix = zeros(9,5); %%% preallocate to increase
speed
matrix1 = zeros(1,size);
matrix1_total = zeros(1,size);
x_values = zeros(1,size);
Z = zeros(51,51);
ZZ = zeros(51,51);
CO = zeros(51,51);
SV = zeros(51,51);
EF = zeros(51,51);
EW = zeros(51,51);
PVA = zeros(51,51);
Wmyo = zeros(51,51);
Wloss = zeros(51,51);
PVAmYo = zeros(51,51);
```



```

EFF = zeros(51,51);
EFFmyo = zeros(51,51);
WmyoA1 = zeros(size,size);
WmyoB1 = zeros(size,size);
WmyoA2 = zeros(size,size);
WmyoB2 = zeros(size,size);
WlossA1 = zeros(size,size);
WlossB1 = zeros(size,size);
WlossA2 = zeros(size,size);
WlossB2 = zeros(size,size);
PVAmyoA1 = zeros(size,size);
PVAmyoB1 = zeros(size,size);
PVAmyoA2 = zeros(size,size);
PVAmyoB2 = zeros(size,size);
PW = zeros(size,size);

for n = 1:size
    for m = 1:size

        Eal_max = 20+.6*(n-1); %%% if this is changed,
change the 3d matrix size on line 262 for variable X
        Ebl_max = 36;
        Ea2_max = 36;
        Eb2_max = 36;
        Eal_min = .56;
        Ebl_min = .56;
        Ea2_min = .56;
        Eb2_min = .56;
        td1 = 0; %% + .001*(m-1); %%% if this is
changed, change the 3d matrix size on line 262 for variable X
        td2 = 0;
        td3 = 0;
        td4 = 0;
        k1 = .005 + .0009*(m-1);
        k2 = .0015;
        k3 = .0015;
        k4 = .0015;

        Eeq = 1/Eal_max + 1/Ebl_max + 1/Ea2_max +
1/Eb2_max;
        Eeq = 1/Eeq;

        simOut = sim(filename);

```

```

base = 'visco';
numframe1 = num2str(n);
numframe2 = strcat('_', numframe1);
numframe3 = num2str(m);
numframe4 = strcat('_', numframe3);
numframe = strcat(numframe2, numframe4);

visco.(genvarname(['values' numframe])) =
[Ea1_max Ea1_min Eb1_max Eb1_min Ea2_max Ea2_min Eb2_max
Eb2_min td1 td2 td3 td4 k1 k2 k3 k4];
data(n,m).values = [Ea1_max Ea1_min Eb1_max
Eb1_min Ea2_max Ea2_min Eb2_max Eb2_min td1 td2 td3 td4 k1 k2
k3 k4];
visco.(genvarname(['lvp' numframe])) =
decimate(values.signals.values(:,1),r);
visco.(genvarname(['aop' numframe])) =
decimate(values.signals.values(:,2),r);
visco.(genvarname(['aof' numframe])) =
decimate(values.signals.values(:,3),r);
visco.(genvarname(['mf' numframe])) =
decimate(values.signals.values(:,4),r);
visco.(genvarname(['Va1' numframe])) =
decimate(values.signals.values(:,5),r);
visco.(genvarname(['Vb1' numframe])) =
decimate(values.signals.values(:,6),r);
visco.(genvarname(['Va2' numframe])) =
decimate(values.signals.values(:,7),r);
visco.(genvarname(['Vb2' numframe])) =
decimate(values.signals.values(:,8),r);
visco.(genvarname(['VT' numframe])) =
decimate(values.signals.values(:,9),r);
visco.(genvarname(['Ea1' numframe])) =
decimate(values.signals.values(:,10),r);
visco.(genvarname(['Eb1' numframe])) =
decimate(values.signals.values(:,11),r);
visco.(genvarname(['Ea2' numframe])) =
decimate(values.signals.values(:,12),r);
visco.(genvarname(['Eb2' numframe])) =
decimate(values.signals.values(:,13),r);
visco.(genvarname(['DVT' numframe])) =
diff(visco.(genvarname(['VT' numframe])))*fs;
visco.(genvarname(['DVT' numframe]))(1:500) = 0;
visco.(genvarname(['Dlvp' numframe])) =
diff(visco.(genvarname(['lvp' numframe])));
visco.(genvarname(['Dlvp' numframe])) =
diff(visco.(genvarname(['lvp' numframe])));

```

```

0;
    visco.(genvarname(['Dlvp' numframe]))(1:7000) =
    visco.(genvarname(['absDVT' numframe])) =
abs(visco.(genvarname(['DVT' numframe])));
    visco.(genvarname(['DVal' numframe])) =
diff(visco.(genvarname(['Val' numframe])))*fs;
    visco.(genvarname(['DVb1' numframe])) =
diff(visco.(genvarname(['Vb1' numframe])))*fs;
    visco.(genvarname(['DVa2' numframe])) =
diff(visco.(genvarname(['Va2' numframe])))*fs;
    visco.(genvarname(['DVb2' numframe])) =
diff(visco.(genvarname(['Vb2' numframe])))*fs;
    visco.(genvarname(['DVal' numframe]))(1:500) = 0;
    visco.(genvarname(['DVb1' numframe]))(1:500) = 0;
    visco.(genvarname(['DVa2' numframe]))(1:500) = 0;
    visco.(genvarname(['DVb2' numframe]))(1:500) = 0;
    visco.(genvarname(['Dlvp' numframe]))(1:500) = 0;

```

%%%mitral flow beat index%%%%%%%%%

```

    visco.(genvarname(['mlindex' numframe])) =
find(visco.(genvarname(['mf'
numframe]))(400:length(visco.(genvarname(['mf'
numframe]))))>.1*max(visco.(genvarname(['mf'
numframe]))(1000:length(visco.(genvarname(['mf'
numframe]))))));
    visco.(genvarname(['mlindex' numframe])) =
visco.(genvarname(['mlindex' numframe])) + 400;
    visco.(genvarname(['m2index' numframe])) =
find(diff(visco.(genvarname(['mlindex' numframe])))-1);
    visco.(genvarname(['mindex' numframe])) =
visco.(genvarname(['mlindex'
numframe]))(visco.(genvarname(['m2index' numframe])) + 1);
    visco.(genvarname(['mindexEDV' numframe])) =
visco.(genvarname(['mlindex'
numframe]))(visco.(genvarname(['m2index' numframe])));

```

%%%end mitral flow beat index%%%

```

    visco.(genvarname(['EDV' numframe])) =
visco.(genvarname(['VT'
numframe]))(visco.(genvarname(['mindexEDV' numframe])) + 20);
    visco.(genvarname(['ESV' numframe])) =
visco.(genvarname(['VT'
numframe]))(visco.(genvarname(['mindex' numframe])) - 20);

```

```

        visco.(genvarname(['SV' numframe])) =
visco.(genvarname(['EDV' numframe])) -
visco.(genvarname(['ESV' numframe]));
        visco.(genvarname(['CO' numframe])) =
visco.(genvarname(['SV' numframe]))*HR/60;
        visco.(genvarname(['EF' numframe])) =
(visco.(genvarname(['SV'
numframe])))/visco.(genvarname(['EDV' numframe]))*100;

```

```

%%aortic flow beat index#####

```

```

        visco.(genvarname(['aoflindex' numframe])) =
find(visco.(genvarname(['aof'
numframe]))(800:length(visco.(genvarname(['aof'
numframe]))))>.05*max(visco.(genvarname(['aof'
numframe]))(1000:length(visco.(genvarname(['aof'
numframe]))))));
        visco.(genvarname(['aoflindex' numframe])) =
visco.(genvarname(['aoflindex' numframe])) + 800;
        visco.(genvarname(['aof2index' numframe])) =
find(diff(visco.(genvarname(['aoflindex' numframe]))~1);
        visco.(genvarname(['aofindex' numframe])) =
visco.(genvarname(['aoflindex'
numframe]))(visco.(genvarname(['aof2index' numframe])) + 1);
        visco.(genvarname(['aofindexEDV' numframe])) =
visco.(genvarname(['aoflindex'
numframe]))(visco.(genvarname(['aof2index' numframe])));

```

```

% #####end aortic flow beat index###

```

```

        visco.(genvarname(['EWtop' numframe])) = -
trapz(visco.(genvarname(['VT'
numframe]))((visco.(genvarname(['aofindex' numframe]))(6)-
23):(visco.(genvarname(['aofindexEDV'
numframe]))(7)+27)),visco.(genvarname(['lvp'
numframe]))((visco.(genvarname(['aofindex' numframe]))(6)-
23):(visco.(genvarname(['aofindexEDV' numframe]))(7)+27)));
        visco.(genvarname(['EWbottom' numframe])) =
trapz(visco.(genvarname(['VT'
numframe]))((visco.(genvarname(['mindex' numframe]))(6)-
2):(visco.(genvarname(['mindexEDV'
numframe]))(7))+0),visco.(genvarname(['lvp'
numframe]))((visco.(genvarname(['mindex' numframe]))(6)-
2):(visco.(genvarname(['mindexEDV' numframe]))(7)+0)));
        visco.(genvarname(['EW' numframe])) =
visco.(genvarname(['EWtop' numframe])) -
visco.(genvarname(['EWbottom' numframe]));

```

```

    visco.(genvarname(['EW' numframe])) =
visco.(genvarname(['EW' numframe]))*133.322/1e6; %% convert
from mmHg*mL --> Joules

    visco.(genvarname(['numB' numframe])) =
visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindexEDV' numframe]))(6))
- Eeq*visco.(genvarname(['ESV' numframe]))(6);
    visco.(genvarname(['numX' numframe])) =
0:.001:visco.(genvarname(['ESV' numframe]))(6);
    visco.(genvarname(['numY' numframe])) =
Eeq*visco.(genvarname(['numX' numframe])) +
visco.(genvarname(['numB' numframe]));
    visco.(genvarname(['PW' numframe])) =
trapz(visco.(genvarname(['numX'
numframe])),visco.(genvarname(['numY' numframe])));
    visco.(genvarname(['PW' numframe])) =
visco.(genvarname(['PW' numframe]))*133.322/1e6; %% convert
from mmHg*mL --> Joules
    visco.(genvarname(['PVA' numframe])) =
visco.(genvarname(['PW' numframe])) + visco.(genvarname(['EW'
numframe]));
    visco.(genvarname(['PWseg' numframe])) =
visco.(genvarname(['PW' numframe]))/4;

    %%% Wmyo %%%
visco.(genvarname(['WmyoVal' numframe])) = 0;
visco.(genvarname(['WmyoVb1' numframe])) = 0;
visco.(genvarname(['WmyoVa2' numframe])) = 0;
visco.(genvarname(['WmyoVb2' numframe])) = 0;
visco.(genvarname(['Wmyo' numframe])) = 0;
for w = visco.(genvarname(['mindexEDV'
numframe]))(6):visco.(genvarname(['mindexEDV' numframe]))(7)
%returns the data points for one cardiac cycle starting at
the 6th beat
    if(visco.(genvarname(['DVal' numframe]))(w) < 0)
        visco.(genvarname(['WmyoVal' numframe])) =
visco.(genvarname(['WmyoVal' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp' numframe]))(w+1))/2
*abs((visco.(genvarname(['Val' numframe]))(w) -
visco.(genvarname(['Val' numframe]))(w+1)));
    end
    if(visco.(genvarname(['DVb1' numframe]))(w) < 0)
        visco.(genvarname(['WmyoVb1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe])) +

```

```

(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Vb1'
numframe]))(w) - visco.(genvarname(['Vb1' numframe]))(w+1)));
    end
    if(visco.(genvarname(['DVA2' numframe]))(w) < 0)
        visco.(genvarname(['WmyoVa2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Va2'
numframe]))(w) - visco.(genvarname(['Va2' numframe]))(w+1)));
    end
    if(visco.(genvarname(['DVB2' numframe]))(w) < 0)
        visco.(genvarname(['WmyoVb2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Vb2'
numframe]))(w) - visco.(genvarname(['Vb2' numframe]))(w+1)));
    end
end

```

```

    visco.(genvarname(['WlossA1' numframe])) =
sum(k1*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVal'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVal'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;

```

```

    visco.(genvarname(['WlossB1' numframe])) =
sum(k2*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVb1'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVb1'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;

```



```

    visco.(genvarname(['WlossA2' numframe])) =
sum(k3*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVA2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVA2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;
    visco.(genvarname(['WlossB2' numframe])) =
sum(k4*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVB2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVB2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;
    visco.(genvarname(['Wloss' numframe])) =
visco.(genvarname(['WlossA1' numframe])) +
visco.(genvarname(['WlossB1' numframe])) +
visco.(genvarname(['WlossA2' numframe])) +
visco.(genvarname(['WlossB2' numframe]));
    visco.(genvarname(['WmyoVal' numframe])) =
visco.(genvarname(['WmyoVal' numframe]))*133.322/1e6;%%
convert from mmHg*mL --> Joules
    visco.(genvarname(['WmyoVb1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe]))*133.322/1e6;
    visco.(genvarname(['WmyoVa2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe]))*133.322/1e6;
    visco.(genvarname(['WmyoVb2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe]))*133.322/1e6;
    %visco.(genvarname(['Wloss' numframe])) =
visco.(genvarname(['Wloss' numframe]))*133.322/1e6; %%
convert from mmHg*mL --> Joules
    visco.(genvarname(['Wmyo' numframe])) =
visco.(genvarname(['WmyoVal' numframe])) +
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['WmyoVb2' numframe]));
    visco.(genvarname(['PVAmyo' numframe])) =
visco.(genvarname(['PW' numframe])) +

```

```

visco.(genvarname(['Wmyo' numframe])) +
visco.(genvarname(['Wloss' numframe]))*(60/HR); %%multiply
Wloss *2/3 because Wloss is in units of Watts (J/s), and one
cycle is 60/90

```

```

visco.(genvarname(['PVAmyoA1' numframe])) =
visco.(genvarname(['WmyoVa1' numframe])) +
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['PWseg' numframe]));
visco.(genvarname(['PVAmyoB1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['PWseg' numframe]));
visco.(genvarname(['PVAmyoA2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['PWseg' numframe]));
visco.(genvarname(['PVAmyoB2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe])) +
visco.(genvarname(['WmyoVb2' numframe])) +
visco.(genvarname(['PWseg' numframe]));

```

```

%%%%%%%% end Wmyo %%%%%%%%%

```

```

%%%%%%%%Internal Flow modeled by a
Conductance Catheter%%%%%%%%

```

```

visco.(genvarname(['V1_cc' numframe])) =
visco.(genvarname(['Va1' numframe])) +
visco.(genvarname(['Vb1' numframe]));
visco.(genvarname(['V2_cc' numframe])) =
visco.(genvarname(['Va2' numframe])) +
visco.(genvarname(['Vb2' numframe]));
visco.(genvarname(['DV1_cc' numframe])) =
diff(visco.(genvarname(['V1_cc' numframe]))).*fs;
visco.(genvarname(['DV2_cc' numframe])) =
diff(visco.(genvarname(['V2_cc' numframe]))).*fs;
visco.(genvarname(['DV1_cc' numframe]))(1:100) =
0;
visco.(genvarname(['DV2_cc' numframe]))(1:100) =
0;

visco.(genvarname(['Dvt_cc' numframe])) =
abs(visco.(genvarname(['DV1_cc' numframe])) +
abs(visco.(genvarname(['DV2_cc' numframe])));

```



```

        visco.(genvarname(['IF_cc' numframe])) =
(visco.(genvarname(['Dvt_cc' numframe])) -
visco.(genvarname(['absDVT' numframe])))/2;

        for j = 1:length(visco.(genvarname(['mindex'
numframe])) - 1
            dns.allIF(j,1) =
trapz(visco.(genvarname(['IF_cc'
numframe]))(visco.(genvarname(['mindex'
numframe]))(j):visco.(genvarname(['mindex'
numframe]))(j+1))*1/fs);
            dns.allDVT(j,1) =
trapz(visco.(genvarname(['absDVT'
numframe]))(visco.(genvarname(['mindex'
numframe]))(j):visco.(genvarname(['mindex'
numframe]))(j+1))*1/fs);
            end
            visco.(genvarname(['IFF' numframe])) =
dns.allIF./dns.allDVT;
            visco.(genvarname(['IFFpercent' numframe])) =
visco.(genvarname(['IFF' numframe])).*100;

            visco.(genvarname(['stdIFF' numframe])) =
std(visco.(genvarname(['IFF' numframe])));
            visco.(genvarname(['meanIFF' numframe])) =
mean(visco.(genvarname(['IFF' numframe])));
            visco.(genvarname(['meanIFFpercent' numframe])) =
visco.(genvarname(['meanIFF' numframe]))*100;
            data(n,m).std = visco.(genvarname(['stdIFF'
numframe]));
            data(n,m).meanIFF = visco.(genvarname(['meanIFF'
numframe]));
            data(n,m).meanIFFpercent =
visco.(genvarname(['meanIFFpercent' numframe]));

        %%%%end IFF by Conductance Catheter%%%%%%%%%%%%%%

        %%%%%%%%%%%%%Internal Flow modeled by 4 segment
dyssynchrony data%%%%%%%%%%%%%%

        visco.(genvarname(['IF_total_cc' numframe])) =
((abs(visco.(genvarname(['DVal' numframe])) +
abs(visco.(genvarname(['DVa2' numframe])) +
abs(visco.(genvarname(['DVb1' numframe])) +
abs(visco.(genvarname(['DVb2' numframe])))) -
visco.(genvarname(['absDVT' numframe])))/2;

```

```

        for j = 1:length(visco.(genvarname(['mindex'
numframe])) - 1
            dns.allIF_total(j,1) =
trapz(visco.(genvarname(['IF_total_cc'
numframe]))(visco.(genvarname(['mindex'
numframe]))(j):visco.(genvarname(['mindex'
numframe]))(j+1))*1/fs);
            end
            visco.(genvarname(['IFF_total' numframe])) =
dns.allIF_total./dns.allDVT;
            visco.(genvarname(['stdIFF_total' numframe])) =
std(visco.(genvarname(['IFF_total' numframe])));
            visco.(genvarname(['meanIFF_total' numframe])) =
mean(visco.(genvarname(['IFF_total' numframe])));
            visco.(genvarname(['meanIFFpercent_total'
numframe])) = visco.(genvarname(['meanIFF_total'
numframe])).*100;

            data(n,m).std_total =
visco.(genvarname(['stdIFF_total' numframe]));
            data(n,m).meanIFF_total =
visco.(genvarname(['meanIFF_total' numframe]));
            data(n,m).meanIFFpercent_total =
visco.(genvarname(['meanIFFpercent_total' numframe]));

            numframe

            directory1 =
'C:\Users\Cody\Documents\thesis\programs\paramsweep7\param7';
            directory2 = strcat(directory1, numframe);
            directory2 = char(directory2);
            save(directory2, '-struct', 'visco')

            Z(n,m) = data(n,m).meanIFFpercent;
            ZZ(n,m) = data(n,m).meanIFFpercent_total;
            CO(n,m) = visco.(genvarname(['CO' numframe]))(4);
            SV(n,m) = visco.(genvarname(['SV' numframe]))(4);
            EF(n,m) = visco.(genvarname(['EF' numframe]))(4);
            EW(n,m) = visco.(genvarname(['EW' numframe]));
            PVA(n,m) = visco.(genvarname(['PVA' numframe]));
            Wmyo(n,m) = visco.(genvarname(['Wmyo'
numframe]));
            Wloss(n,m) = visco.(genvarname(['Wloss'
numframe]));

```

```

        PVAmYo(n,m) = visco.(genvarname(['PVAmYo'
numframe])));
        WmyoA1(n,m) = visco.(genvarname(['WmyoVa1'
numframe])));
        WmyoB1(n,m) = visco.(genvarname(['WmyoVb1'
numframe])));
        WmyoA2(n,m) = visco.(genvarname(['WmyoVa2'
numframe])));
        WmyoB2(n,m) = visco.(genvarname(['WmyoVb2'
numframe])));
        WlossA1(n,m) = visco.(genvarname(['WlossA1'
numframe])));
        WlossB1(n,m) = visco.(genvarname(['WlossB1'
numframe])));
        WlossA2(n,m) = visco.(genvarname(['WlossA2'
numframe])));
        WlossB2(n,m) = visco.(genvarname(['WlossB2'
numframe])));
        PVAmYoA1(n,m) = visco.(genvarname(['PVAmYoA1'
numframe])));
        PVAmYoB1(n,m) = visco.(genvarname(['PVAmYoB1'
numframe])));
        PVAmYoA2(n,m) = visco.(genvarname(['PVAmYoA2'
numframe])));
        PVAmYoB2(n,m) = visco.(genvarname(['PVAmYoB2'
numframe])));
        PW(n,m) = visco.(genvarname(['PW' numframe]));

        clear visco
        clear values
        clear dns
    end
end
save
C:\Users\Cody\Documents\thesis\programs\paramsweep7\workspace
data
%%save(paramsweep5data)

Y = 20:.6:50;
X = 0.005:.0009:0.05;
%figure
subplot(2,1,1)
surfc(X,Y,Z)
xlabel('k (s/mL)'),ylabel('Ea1_m_a_x
(mmHg/mL)'),zlabel('Internal Flow Fraction
(%)'),title('conductance catheter dyssynchrony surface plot')

```

```

shading interp
colormap(jet)
colorbar
subplot(2,1,2)
surfc(X,Y,ZZ)
xlabel('k (s/mL)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Internal Flow Fraction (%)'),title('full
dyssynchrony surface plot')
fh = figure(1);
set(fh, 'color', 'white');
shading interp
colormap(jet)
colorbar
figure
subplot(2,1,1)
surfc(X,Y,CO)
xlabel('k (s/mL)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Cardiac Output (mL/sec)'),title('cardiac
output surface plot')
colorbar
subplot(2,1,2)
surfc(X,Y,EF)
xlabel('k (s/mL)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('Ejection Fraction'),title('ejection
fraction surface plot')
fh = figure(2);
set(fh, 'color', 'white');
shading interp
colormap(jet)
colorbar
figure
subplot(2,1,1)
surfc(X,Y,EW)
xlabel('k (s/mL)'),ylabel('Eal_m_a_x
(mmHg/mL)'),zlabel('External Work (J)'),title('external work
surface plot')
shading interp
colormap(jet)
colorbar
subplot(2,1,2)
surfc(X,Y,PVA)
xlabel('k'),ylabel('Eal_m_a_x (mmHg/mL)'),zlabel('Pressure
Volume Area (J)'),title('pressure volume area (PVA) surface
plot')
fh = figure(3);
set(fh, 'color', 'white');

```

```

figure
subplot(2,1,1)
surf(X,Y,100*EW./PVA)
xlabel('k (s/mL)'),ylabel('Ea1_m_a_x
(mmHg/mL)'),zlabel('Efficiency (%)'),title('efficiency
(EW/PVA) surface plot')
shading interp
colormap(jet)
colorbar
subplot(2,1,2)
surf(X,Y,SV)
xlabel('k (s/mL)'),ylabel('Ea1_m_a_x
(mmHg/mL)'),zlabel('Stroke Volume (mL)'),title('stroke volume
surface plot')
shading interp
colormap(jet)
colorbar
fh = figure(4);
set(fh, 'color', 'white');
figure
subplot(2,1,1)
surf(X,Y,Wmyo)
xlabel('k (s/mL)'),ylabel('Ea1_m_a_x
(mmHg/mL)'),zlabel('Myocardium Work (J)'),title('Myocardium
Work (W_m_y_o) surface plot')
shading interp
colormap(jet)
colorbar
subplot(2,1,2)
surf(X,Y,PVAm_yo)
xlabel('k'),ylabel('Ea1_m_a_x (mmHg/mL)'),zlabel('PVA_m_y_o
(J)'),title('PVA_m_y_o surface plot')
shading interp
colormap(jet)
colorbar
fh = figure(5);
set(fh, 'color', 'white');
figure
subplot(2,1,1)
surf(X,Y,100*EW./PVAm_yo)
xlabel('k (s/mL)'),ylabel('Ea1_m_a_x
(mmHg/mL)'),zlabel('Efficiency (%)'),title('Myocardium
efficiency (EW/PVAm_yo) surface plot')
shading interp
colorbar
subplot(2,1,2)

```



## Code Generating Monte Carlo Simulation Data

The code shown below runs the Simulink simulation 'multi\_viscoelastic.mdl' through the MATLAB workspace as well as calculating all cardiovascular performance metrics and indices discussed in this study. The code will run 2000 simulations as denoted by the MATLAB variable 'size' with each representing unique combinations of maximum elastance, minimum elastance, elastance waveform delay, and myocardial viscous friction 'K' for all 4 segments of the model. The code also automatically plots Figures 15 and 16. The muscle parameters, sample rate, and heart rate are initialized below by the corresponding MATLAB variable. Final results for each simulation are automatically saved to the directory and filename 'C:\Users\Cody\Documents\thesis\programs\paramswEEP15\param15\_'. All terminating workspace data is saved to the file and directory path 'C:\Users\Cody\Documents\thesis\programs\paramswEEP15\workspacedata'.

---

```
sample = 100000; %%check in config parameters of simulink
model to match!! This is very important!!!
r = 100; % decimation factor to reduce stored
memory/file size
fs = sample/r; %% effective sample rate after decimation
HR = 90; % Heart Rate in Beats Per Minute
HRrad = 2*3.14159*(HR/60); This is fed into the
simulink model to control the elastance waveform
size = 2000; %% manually count the # of required sims
determined below-->
```

```
filename = 'multi_viscoelastic.mdl';
```

```
final_matrix = zeros(20,5); %% preallocate to increase
speed
```

```
matrix1 = zeros(1,size);
matrix1_total = zeros(1,size);
CO = zeros(1,size);
SV = zeros(1,size);
EF = zeros(1,size);
EW = zeros(1,size);
PVA = zeros(1,size);
EFF = zeros(1,size);
EFFmyo = zeros(1,size);
Wmyo = zeros(1,size);
Wloss = zeros(1,size);
PVAmyo = zeros(1,size);
WmyoA1 = zeros(1,size);
WmyoB1 = zeros(1,size);
```

```

WmyoA2 = zeros(1,size);
WmyoB2 = zeros(1,size);
WlossA1 = zeros(1,size);
WlossB1 = zeros(1,size);
WlossA2 = zeros(1,size);
WlossB2 = zeros(1,size);
PVAmYoA1 = zeros(1,size);
PVAmYoB1 = zeros(1,size);
PVAmYoA2 = zeros(1,size);
PVAmYoB2 = zeros(1,size);
PW = zeros(1,size);

```

```

for n = 1:size

```

```

    Ea1_max = random('unif', 18, 54);
    Ea1_min = random('unif', .45, .7);
    Eb1_max = random('unif', 18, 54);
    Eb1_min = random('unif', .45, .7);
    Ea2_max = random('unif', 18, 54);
    Ea2_min = random('unif', .45, .7);
    Eb2_max = random('unif', 18, 54);
    Eb2_min = random('unif', .45, .7);
    td1 = random('unif', 0, .02);
    td2 = random('unif', 0, .02);
    td3 = random('unif', 0, .02);
    td4 = random('unif', 0, .02);
    k1 = random('unif', .001, .01);
    k2 = random('unif', .001, .01);
    k3 = random('unif', .001, .01);
    k4 = random('unif', .001, .01);

```

```

    Eeq = 1/Ea1_max + 1/Eb1_max + 1/Ea2_max + 1/Eb2_max;
    Eeq = 1/Eeq;

```

```

    simOut = sim(filename);

```

```

    base = 'visco';
    numframe = num2str(n);

```

```

    visco.(genvarname(['values' numframe])) = [Ea1_max
Ea1_min Eb1_max Eb1_min Ea2_max Ea2_min Eb2_max Eb2_min td1
td2 td3 td4 k1 k2 k3 k4];
    data(n).values = [Ea1_max Ea1_min Eb1_max Eb1_min Ea2_max
Ea2_min Eb2_max Eb2_min td1 td2 td3 td4 k1 k2 k3 k4];

```



```

        visco.(genvarname(['lvp' numframe])) =
decimate(values.signals.values(:,1),r);
        visco.(genvarname(['lvp' numframe])) =
decimate(values.signals.values(:,2),r);
        visco.(genvarname(['lvp' numframe])) =
decimate(values.signals.values(:,3),r);
        visco.(genvarname(['mf' numframe])) =
decimate(values.signals.values(:,4),r);
        visco.(genvarname(['Va1' numframe])) =
decimate(values.signals.values(:,5),r);
        visco.(genvarname(['Vb1' numframe])) =
decimate(values.signals.values(:,6),r);
        visco.(genvarname(['Va2' numframe])) =
decimate(values.signals.values(:,7),r);
        visco.(genvarname(['Vb2' numframe])) =
decimate(values.signals.values(:,8),r);
        visco.(genvarname(['VT' numframe])) =
decimate(values.signals.values(:,9),r);
        visco.(genvarname(['Ea1' numframe])) =
decimate(values.signals.values(:,10),r);
        visco.(genvarname(['Eb1' numframe])) =
decimate(values.signals.values(:,11),r);
        visco.(genvarname(['Ea2' numframe])) =
decimate(values.signals.values(:,12),r);
        visco.(genvarname(['Eb2' numframe])) =
decimate(values.signals.values(:,13),r);
        visco.(genvarname(['DVT' numframe])) =
diff(visco.(genvarname(['VT' numframe])))*fs;
        visco.(genvarname(['DVT' numframe]))(1:500) = 0;
        visco.(genvarname(['Dlvp' numframe])) =
diff(visco.(genvarname(['lvp' numframe])));
        visco.(genvarname(['Dlvp' numframe])) =
diff(visco.(genvarname(['lvp' numframe])));
        visco.(genvarname(['Dlvp' numframe]))(1:7000) = 0;
        visco.(genvarname(['absDVT' numframe])) =
abs(visco.(genvarname(['DVT' numframe])));
        visco.(genvarname(['DVA1' numframe])) =
diff(visco.(genvarname(['Va1' numframe])))*fs;
        visco.(genvarname(['DVB1' numframe])) =
diff(visco.(genvarname(['Vb1' numframe])))*fs;
        visco.(genvarname(['DVA2' numframe])) =
diff(visco.(genvarname(['Va2' numframe])))*fs;
        visco.(genvarname(['DVB2' numframe])) =
diff(visco.(genvarname(['Vb2' numframe])))*fs;
        visco.(genvarname(['DVA1' numframe]))(1:500) = 0;
        visco.(genvarname(['DVB1' numframe]))(1:500) = 0;

```

```

visco.(genvarname(['DVa2' numframe]))(1:500) = 0;
visco.(genvarname(['DVb2' numframe]))(1:500) = 0;
visco.(genvarname(['Dlvp' numframe]))(1:500) = 0;

%%%mitral flow beat index%%%%%%%%%

visco.(genvarname(['mlindex' numframe])) =
find(visco.(genvarname(['mf'
numframe]))(400:length(visco.(genvarname(['mf'
numframe]))))>.1*max(visco.(genvarname(['mf'
numframe]))(1000:length(visco.(genvarname(['mf'
numframe]))))));
visco.(genvarname(['mlindex' numframe])) =
visco.(genvarname(['mlindex' numframe])) + 400;
visco.(genvarname(['m2index' numframe])) =
find(diff(visco.(genvarname(['mlindex' numframe]))~1);
visco.(genvarname(['mindex' numframe])) =
visco.(genvarname(['mlindex'
numframe]))(visco.(genvarname(['m2index' numframe])) + 1);
visco.(genvarname(['mindexEDV' numframe])) =
visco.(genvarname(['mlindex'
numframe]))(visco.(genvarname(['m2index' numframe])));

%      %%%%end mitral flow beat index%%%

visco.(genvarname(['EDV' numframe])) =
visco.(genvarname(['VT'
numframe]))(visco.(genvarname(['mindexEDV' numframe])) + 20);
visco.(genvarname(['ESV' numframe])) =
visco.(genvarname(['VT'
numframe]))(visco.(genvarname(['mindex' numframe])) - 20);
visco.(genvarname(['SV' numframe])) =
visco.(genvarname(['EDV' numframe])) -
visco.(genvarname(['ESV' numframe]));
visco.(genvarname(['CO' numframe])) =
visco.(genvarname(['SV' numframe]))*HR/60;
visco.(genvarname(['EF' numframe])) =
(visco.(genvarname(['SV'
numframe])))/visco.(genvarname(['EDV' numframe]))*100;

%%%aortic flow beat index%%%%%%%%%

visco.(genvarname(['aoflindex' numframe])) =
find(visco.(genvarname(['aof'
numframe]))(800:length(visco.(genvarname(['aof'
numframe]))))>.05*max(visco.(genvarname(['aof'

```

```

numframe]))(1000:length(visco.(genvarname(['aof'
numframe]))))));
    visco.(genvarname(['aoflindex' numframe])) =
visco.(genvarname(['aoflindex' numframe])) + 800;
    visco.(genvarname(['aof2index' numframe])) =
find(diff(visco.(genvarname(['aoflindex' numframe]))~1);
    visco.(genvarname(['aofindex' numframe])) =
visco.(genvarname(['aoflindex'
numframe]))(visco.(genvarname(['aof2index' numframe])) + 1);
    visco.(genvarname(['aofindexEDV' numframe])) =
visco.(genvarname(['aoflindex'
numframe]))(visco.(genvarname(['aof2index' numframe])));

%      %%%end aortic flow beat index%%
    visco.(genvarname(['EWtop' numframe])) = -
trapz(visco.(genvarname(['VT'
numframe]))((visco.(genvarname(['aofindex' numframe]))(6)-
23):(visco.(genvarname(['aofindexEDV'
numframe]))(7)+27)),visco.(genvarname(['lvp'
numframe]))((visco.(genvarname(['aofindex' numframe]))(6)-
23):(visco.(genvarname(['aofindexEDV' numframe]))(7)+27)));
    visco.(genvarname(['EWbottom' numframe])) =
trapz(visco.(genvarname(['VT'
numframe]))((visco.(genvarname(['minindex' numframe]))(6)-
2):(visco.(genvarname(['minindexEDV'
numframe]))(7))+0),visco.(genvarname(['lvp'
numframe]))((visco.(genvarname(['minindex' numframe]))(6)-
2):(visco.(genvarname(['minindexEDV' numframe]))(7)+0)));
    visco.(genvarname(['EW' numframe])) =
visco.(genvarname(['EWtop' numframe])) -
visco.(genvarname(['EWbottom' numframe]));
    visco.(genvarname(['EW' numframe])) =
visco.(genvarname(['EW' numframe]))*133.322/1e6; %% convert
from mmHg*mL --> Joules

    visco.(genvarname(['numB' numframe])) =
visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindexEDV' numframe]))(6))
- Eeq*visco.(genvarname(['ESV' numframe]))(6);
    visco.(genvarname(['numX' numframe])) =
0:.001:visco.(genvarname(['ESV' numframe]))(6);
    visco.(genvarname(['numY' numframe])) =
Eeq*visco.(genvarname(['numX' numframe])) +
visco.(genvarname(['numB' numframe]));

```

```

        visco.(genvarname(['PW' numframe])) =
trapz(visco.(genvarname(['numX'
numframe])),visco.(genvarname(['numY' numframe])));
        visco.(genvarname(['PW' numframe])) =
visco.(genvarname(['PW' numframe]))*133.322/1e6; %% convert
from mmHg*mL --> Joules
        visco.(genvarname(['PVA' numframe])) =
visco.(genvarname(['PW' numframe])) + visco.(genvarname(['EW'
numframe]));
        visco.(genvarname(['PWseg' numframe])) =
visco.(genvarname(['PW' numframe]))/4;

        %%%%%%%%%%%%%%%%% Wmyo %%%%%%%%%%%%%%%%%%%%%%%%%
        visco.(genvarname(['WmyoVal' numframe])) = 0;
        visco.(genvarname(['WmyoVb1' numframe])) = 0;
        visco.(genvarname(['WmyoVa2' numframe])) = 0;
        visco.(genvarname(['WmyoVb2' numframe])) = 0;
        visco.(genvarname(['Wmyo' numframe])) = 0;
for w = visco.(genvarname(['minindexEDV'
numframe]))(6):visco.(genvarname(['minindexEDV' numframe]))(7)
%returns the data points for one cardiac cycle starting at
the 6th beat
        if(visco.(genvarname(['DVal' numframe]))(w) < 0)
            visco.(genvarname(['WmyoVal' numframe])) =
visco.(genvarname(['WmyoVal' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp' numframe]))(w+1))/2
*abs((visco.(genvarname(['Val' numframe]))(w) -
visco.(genvarname(['Val' numframe]))(w+1)));
            end
            if(visco.(genvarname(['DVb1' numframe]))(w) < 0)
                visco.(genvarname(['WmyoVb1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Vb1'
numframe]))(w) - visco.(genvarname(['Vb1' numframe]))(w+1)));
                end
                if(visco.(genvarname(['DVA2' numframe]))(w) < 0)
                    visco.(genvarname(['WmyoVa2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Va2'
numframe]))(w) - visco.(genvarname(['Va2' numframe]))(w+1)));
                    end

```

```

    if(visco.(genvarname(['DVb2' numframe]))(w) < 0)
        visco.(genvarname(['WmyoVb2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe])) +
(visco.(genvarname(['lvp' numframe]))(w) +
visco.(genvarname(['lvp'
numframe]))(w+1))/2*abs((visco.(genvarname(['Vb2'
numframe]))(w) - visco.(genvarname(['Vb2' numframe]))(w+1)));
    end
end

```

```

    visco.(genvarname(['WlossA1' numframe])) =
sum(k1*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVal'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVal'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;

```

```

    visco.(genvarname(['WlossB1' numframe])) =
sum(k2*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVb1'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVb1'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;

```

```

    visco.(genvarname(['WlossA2' numframe])) =
sum(k3*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVA2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVA2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;

```

```

    visco.(genvarname(['WlossB2' numframe])) =
sum(k4*visco.(genvarname(['lvp'
numframe]))(visco.(genvarname(['aofindex'

```

```

numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVB2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7)).*visco.(genvarname(['DVB2'
numframe]))(visco.(genvarname(['aofindex'
numframe]))(6):visco.(genvarname(['aofindexEDV'
numframe]))(7))./fs)*133.322/1e6;
    visco.(genvarname(['Wloss' numframe])) =
visco.(genvarname(['WlossA1' numframe])) +
visco.(genvarname(['WlossB1' numframe])) +
visco.(genvarname(['WlossA2' numframe])) +
visco.(genvarname(['WlossB2' numframe]));

    visco.(genvarname(['WmyoVal' numframe])) =
visco.(genvarname(['WmyoVal' numframe]))*133.322/1e6;%%
convert from mmHg*mL --> Joules
    visco.(genvarname(['WmyoVb1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe]))*133.322/1e6;
    visco.(genvarname(['WmyoVa2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe]))*133.322/1e6;
    visco.(genvarname(['WmyoVb2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe]))*133.322/1e6;
    %visco.(genvarname(['Wloss' numframe])) =
visco.(genvarname(['Wloss' numframe]))*133.322/1e6; %%
convert from mmHg*mL --> Joules
    visco.(genvarname(['Wmyo' numframe])) =
visco.(genvarname(['WmyoVal' numframe])) +
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['WmyoVb2' numframe]));
    visco.(genvarname(['PVAmyo' numframe])) =
visco.(genvarname(['PW' numframe])) +
visco.(genvarname(['Wmyo' numframe])) +
visco.(genvarname(['Wloss' numframe]))*(60/HR); %%multiply
Wloss *2/3 because Wloss is in units of Watts (J/s), and one
cycle is 60/90

    visco.(genvarname(['PVAmyoA1' numframe])) =
visco.(genvarname(['WmyoVal' numframe])) +
visco.(genvarname(['WmyoVal' numframe])) +
visco.(genvarname(['PWseg' numframe]));
    visco.(genvarname(['PVAmyoB1' numframe])) =
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['WmyoVb1' numframe])) +
visco.(genvarname(['PWseg' numframe]));

```

```

        visco.(genvarname(['PVAmYoA2' numframe])) =
visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['WmyoVa2' numframe])) +
visco.(genvarname(['PWseg' numframe]));
        visco.(genvarname(['PVAmYoB2' numframe])) =
visco.(genvarname(['WmyoVb2' numframe])) +
visco.(genvarname(['WmyoVb2' numframe])) +
visco.(genvarname(['PWseg' numframe]));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% end Wmyo %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Internal Flow modeled by a Conductance
Catheter%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

        visco.(genvarname(['V1_cc' numframe])) =
visco.(genvarname(['Va1' numframe])) +
visco.(genvarname(['Vb1' numframe]));
        visco.(genvarname(['V2_cc' numframe])) =
visco.(genvarname(['Va2' numframe])) +
visco.(genvarname(['Vb2' numframe]));
        visco.(genvarname(['DV1_cc' numframe])) =
diff(visco.(genvarname(['V1_cc' numframe]))).*fs;
        visco.(genvarname(['DV2_cc' numframe])) =
diff(visco.(genvarname(['V2_cc' numframe]))).*fs;
        visco.(genvarname(['DV1_cc' numframe]))(1:500) = 0;
        visco.(genvarname(['DV2_cc' numframe]))(1:500) = 0;

```

```

        visco.(genvarname(['Dvt_cc' numframe])) =
abs(visco.(genvarname(['DV1_cc' numframe])) +
abs(visco.(genvarname(['DV2_cc' numframe])));
        visco.(genvarname(['IF_cc' numframe])) =
(visco.(genvarname(['Dvt_cc' numframe])) -
visco.(genvarname(['absDVT' numframe])))/2;

```

```

        for j = 1:length(visco.(genvarname(['mindex' numframe])))
- 1
            dns.allIF(j,1) = trapz(visco.(genvarname(['IF_cc'
numframe]))(visco.(genvarname(['mindex'
numframe]))(j):visco.(genvarname(['mindex'
numframe]))(j+1))*1/fs);
            dns.allDVT(j,1) = trapz(visco.(genvarname(['absDVT'
numframe]))(visco.(genvarname(['mindex'
numframe]))(j):visco.(genvarname(['mindex'
numframe]))(j+1))*1/fs);
        end

```

```

        visco.(genvarname(['IFF' numframe])) =
dns.allIF./dns.allDVT;
        visco.(genvarname(['IFFpercent' numframe])) =
visco.(genvarname(['IFF' numframe])).*100;
        visco.(genvarname(['stdIFF' numframe])) =
std(visco.(genvarname(['IFF' numframe])));
        visco.(genvarname(['meanIFF' numframe])) =
mean(visco.(genvarname(['IFF' numframe])));
        visco.(genvarname(['meanIFFpercent' numframe])) =
visco.(genvarname(['meanIFF' numframe])).*100;
        data(n).std = visco.(genvarname(['stdIFF' numframe]));
        data(n).meanIFF = visco.(genvarname(['meanIFF'
numframe]));
        data(n).meanIFFpercent =
visco.(genvarname(['meanIFFpercent' numframe]));
        %%%end IFF by Conductance Catheter%%%%%%%%%%%%%%

        %%%%%%%%%%%%%%%Internal Flow modeled by 4 segment
dyssynchrony data%%%%%%%%%%%%%%

        visco.(genvarname(['IF_total_cc' numframe])) =
((abs(visco.(genvarname(['DVal' numframe])))) +
abs(visco.(genvarname(['DVA2' numframe])))) +
abs(visco.(genvarname(['DVB1' numframe])))) +
abs(visco.(genvarname(['DVB2' numframe])))) -
visco.(genvarname(['absDVT' numframe])))/2;

        for j = 1:length(visco.(genvarname(['minindex' numframe])))
- 1
            dns.allIF_total(j,1) =
trapz(visco.(genvarname(['IF_total_cc'
numframe]))(visco.(genvarname(['minindex'
numframe]))(j):visco.(genvarname(['minindex'
numframe]))(j+1))*1/fs);
            end
            visco.(genvarname(['IFF_total' numframe])) =
dns.allIF_total./dns.allDVT;
            visco.(genvarname(['stdIFF_total' numframe])) =
std(visco.(genvarname(['IFF_total' numframe])));
            visco.(genvarname(['meanIFF_total' numframe])) =
mean(visco.(genvarname(['IFF_total' numframe])));
            visco.(genvarname(['meanIFFpercent_total' numframe])) =
visco.(genvarname(['meanIFF_total' numframe])).*100;

```



```

data(n).std_total = visco.(genvarname(['stdIFF_total'
numframe]));
data(n).meanIFF_total =
visco.(genvarname(['meanIFF_total' numframe]));
data(n).meanIFFpercent_total =
visco.(genvarname(['meanIFFpercent_total' numframe]));

n

directory1 =
'C:\Users\Cody\Documents\thesis\programs\paramsweep15\param15
_';
directory2 = strcat(directory1, numframe);
directory2 = char(directory2);
save(directory2, '-struct', 'visco')

matrix1(1,n) = data(n).meanIFFpercent;
matrix1_total(1,n) = data(n).meanIFFpercent_total;
CO(1,n) = visco.(genvarname(['CO' numframe]))(8);
SV(1,n) = visco.(genvarname(['SV' numframe]))(8);
EF(1,n) = visco.(genvarname(['EF' numframe]))(8);
EW(1,n) = visco.(genvarname(['EW' numframe]));
PVA(1,n) = visco.(genvarname(['PVA' numframe]));
EFF(1,n) = visco.(genvarname(['EW'
numframe]))/visco.(genvarname(['PVA' numframe]));
EFFmyo(1,n) = visco.(genvarname(['EW'
numframe]))/visco.(genvarname(['PVAmYo' numframe]));
Wmyo(1,n) = visco.(genvarname(['Wmyo' numframe]));
Wloss(1,n) = visco.(genvarname(['Wloss' numframe]));
PVAmYo(1,n) = visco.(genvarname(['PVAmYo' numframe]));
WmyoA1(1,n) = visco.(genvarname(['WmyoVa1' numframe]));
WmyoB1(1,n) = visco.(genvarname(['WmyoVb1' numframe]));
WmyoA2(1,n) = visco.(genvarname(['WmyoVa2' numframe]));
WmyoB2(1,n) = visco.(genvarname(['WmyoVb2' numframe]));
WlossA1(1,n) = visco.(genvarname(['WlossA1' numframe]));
WlossB1(1,n) = visco.(genvarname(['WlossB1' numframe]));
WlossA2(1,n) = visco.(genvarname(['WlossA2' numframe]));
WlossB2(1,n) = visco.(genvarname(['WlossB2' numframe]));

```

```

    PVAmYoA1(1,n) = visco.(genvarname(['PVAmYoA1'
numframe])));
    PVAmYoB1(1,n) = visco.(genvarname(['PVAmYoB1'
numframe])));
    PVAmYoA2(1,n) = visco.(genvarname(['PVAmYoA2'
numframe])));
    PVAmYoB2(1,n) = visco.(genvarname(['PVAmYoB2'
numframe])));
    PW(1,n) = visco.(genvarname(['PW' numframe]));

    clear visco
    clear values
    clear dns
end

matrix2 = matrix1;
matrix2_total = matrix1_total;
matrix1 = sort(matrix1, 2);
matrix1_total = sort(matrix1_total, 2);
COsort = sort(CO, 2);
SVsort = sort(SV, 2);
EFsort = sort(EF, 2);
EWsort = sort(EW, 2);
PVAsort = sort(PVA, 2);
EFFsort = sort(EFF, 2);
Wmyosort = sort(Wmyo, 2);
Wlosssort = sort(Wloss, 2);
PVAmYosort = sort(PVAmYo, 2);
EFFmyosort = sort(EFFmyo, 2);
%%%%%%%%skewness%%%%%%%%

for i = 1:20
    [r1,c1,v1] = find(matrix2 == matrix1(1,i));
    [r1_t,c1_t,v1_t] = find(matrix2_total ==
matrix1_total(1,i));
    [r_co,c_co,v_co] = find(CO == COsort(1,i));
    [r_sv,c_sv,v_sv] = find(SV == SVsort(1,i));
    [r_ef,c_ef,v_ef] = find(EF == EFsort(1,i));
    [r_ew,c_ew,v_ew] = find(EW == EWsort(1,i));
    [r_pva,c_pva,v_pva] = find(PVA == PVAsort(1,i));
    [r_eff,c_eff,v_eff] = find(EFF == EFFsort(1,i));
    [r_wmyo,c_wmyo,v_wmyo] = find(Wmyo == Wmyosort(1,i));
    [r_wloss,c_wloss,v_wloss] = find(Wloss ==
Wlosssort(1,i));
    [r_pvamyo,c_pvamyo,v_pvamyo] = find(PVAmYo ==
PVAmYosort(1,i));

```

```
[r_effmyo,c_effmyo,v_effmyo] = find(EFFmyo ==  
EFFmyosort(1,i));
```

```
final_matrix(1,i) = matrix1(1,i);  
final_matrix(2,i) = c1;  
final_matrix(3,i) = matrix1_total(1,i);  
final_matrix(4,i) = c1_t;  
final_matrix(11,i) = COsort(1,i);  
final_matrix(12,i) = c_co;  
final_matrix(13,i) = SVsort(1,i);  
final_matrix(14,i) = c_sv;  
final_matrix(15,i) = EFsort(1,i);  
final_matrix(16,i) = c_ef;  
final_matrix(17,i) = EWsort(1,i);  
final_matrix(18,i) = c_ew;  
final_matrix(19,i) = PVAsort(1,i);  
final_matrix(20,i) = c_pva;  
final_matrix(21,i) = EFFsort(1,i);  
final_matrix(22,i) = c_eff;  
final_matrix(23,i) = Wmyosort(1,i);  
final_matrix(24,i) = c_wmyo;  
final_matrix(25,i) = Wlosssort(1,i);  
final_matrix(26,i) = c_wloss;  
final_matrix(27,i) = PVAsort(1,i);  
final_matrix(28,i) = c_pvamyo;  
final_matrix(29,i) = EFFmyosort(1,i);  
final_matrix(30,i) = c_effmyo;
```

```
end
```

```
for i = (size-20):size  
[r1,c1,v1] = find(matrix2 == matrix1(1,i));  
[r1_t,c1_t,v1_t] = find(matrix2_total ==  
matrix1_total(1,i));  
[r_co,c_co,v_co] = find(CO == COsort(1,i));  
[r_sv,c_sv,v_sv] = find(SV == SVsort(1,i));  
[r_ef,c_ef,v_ef] = find(EF == EFsort(1,i));  
[r_ew,c_ew,v_ew] = find(EW == EWsort(1,i));  
[r_pva,c_pva,v_pva] = find(PVA == PVAsort(1,i));  
[r_eff,c_eff,v_eff] = find(EFF == EFFsort(1,i));  
[r_wmyo,c_wmyo,v_wmyo] = find(Wmyo == Wmyosort(1,i));  
[r_wloss,c_wloss,v_wloss] = find(Wloss ==  
Wlosssort(1,i));  
[r_pvamyo,c_pvamyo,v_pvamyo] = find(PVAmyo ==  
PVAmyosort(1,i));  
[r_effmyo,c_effmyo,v_effmyo] = find(EFFmyo ==  
EFFmyosort(1,i));
```

```

final_matrix(6,i) = matrix1(1,i);
final_matrix(7,i) = c1;
final_matrix(8,i) = matrix1_total(1,i);
final_matrix(9,i) = c1_t;
final_matrix(11,i) = COsort(1,i);
final_matrix(12,i) = c_co;
final_matrix(13,i) = SVsort(1,i);
final_matrix(14,i) = c_sv;
final_matrix(15,i) = EFsort(1,i);
final_matrix(16,i) = c_ef;
final_matrix(17,i) = EWsort(1,i);
final_matrix(18,i) = c_ew;
final_matrix(19,i) = PVAsort(1,i);
final_matrix(20,i) = c_pva;
final_matrix(21,i) = EFFsort(1,i);
final_matrix(22,i) = c_eff;
final_matrix(23,i) = Wmyosort(1,i);
final_matrix(24,i) = c_wmyo;
final_matrix(25,i) = Wlosssort(1,i);
final_matrix(26,i) = c_wloss;
final_matrix(27,i) = PVAmyo(1,i);
final_matrix(28,i) = c_pvmyo;
final_matrix(29,i) = EFFmyosort(1,i);
final_matrix(30,i) = c_effmyo;
end

dysplot = zeros(round(max(matrix1)),1);    %% preallocate
dysplotCO = zeros(round(max(CO)),1);
dysplotSV = zeros(round(max(SV)),1);
dysplotEF = zeros(round(max(EF)),1);
dysplotEW = zeros(round(max(EW)*100),1);
dysplotPVA = zeros(round(max(PVA)*100),1);
dysplotEFF = zeros(round(max(EFF*100)),1); % *100 because
you want a percent. The matrix will be a 1x1 otherwise
dysplotWmyo = zeros(round(max(Wmyo)*100),1);
dysplotWloss = zeros(round(max(Wloss)*100),1);
dysplotPVAmyo = zeros(round(max(PVAmyo)*100),1);
pEW=.01:.01:length(dysplotEW)/100;
pPVA=.01:.01:length(dysplotPVA)/100;
pWmyo=.01:.01:length(dysplotWmyo)/100;
pWloss=.01:.01:length(dysplotWloss)/100;
pPVAmyo=.01:.01:length(dysplotPVAmyo)/100;
dysplotEFFmyo = zeros(round(max(EFFmyo*100)),1);

for i = 1:size

```

```

    for n = 1:round(max(matrix1))
        if(n-1 <= matrix1(1,i) && matrix1(1,i) < n)
            dysplot(n,1)= dysplot(n,1) +1;
        end
    end
end

dysplot_t = zeros(round(max(matrix1_total)),1);
preallocate

for i = 1:size
    for n = 1:round(max(matrix1_total))
        if(n-1 <= matrix1_total(1,i) && matrix1_total(1,i) <
n)
            dysplot_t(n,1)= dysplot_t(n,1) +1;
        end
    end
end

for i = 1:size
    for n = 1:round(max(CO))
        if(n-1 <= CO(1,i) && CO(1,i) < n)
            dysplotCO(n,1)= dysplotCO(n,1) +1;
        end
    end
end

for i = 1:size
    for n = 1:round(max(SV))
        if(n-1 <= SV(1,i) && SV(1,i) < n)
            dysplotSV(n,1)= dysplotSV(n,1) +1;
        end
    end
end

for i = 1:size
    for n = 1:round(max(EF))
        if(n-1 <= EF(1,i) && EF(1,i) < n)
            dysplotEF(n,1)= dysplotEF(n,1) +1;
        end
    end
end

for i = 1:size
    for n = 1:round(max(EW)*100)
        if(n-1 <= EW(1,i)*100 && EW(1,i)*100 < n)

```

```

        dysplotEW(n,1)= dysplotEW(n,1) +1;
    end
end
end

for i = 1:size
    for n = 1:round(max(PVA)*100)
        if(n-1 <= PVA(1,i)*100 && PVA(1,i)*100 < n)
            dysplotPVA(n,1)= dysplotPVA(n,1) +1;
        end
    end
end

for i = 1:size
    for n = 1:round(max(EFF*100))
        if(n-1 <= EFF(1,i)*100 && EFF(1,i)*100 < n)
            dysplotEFF(n,1)= dysplotEFF(n,1) +1;
        end
    end
end

for i = 1:size
    for n = 1:round(max(Wmyo)*100)
        if(n-1 <= Wmyo(1,i)*100 && Wmyo(1,i)*100 < n)
            dysplotWmyo(n,1)= dysplotWmyo(n,1) +1;
        end
    end
end

for i = 1:size
    for n = 1:round(max(Wloss)*100)
        if(n-1 <= Wloss(1,i)*100 && Wloss(1,i)*100 < n)
            dysplotWloss(n,1)= dysplotWloss(n,1) +1;
        end
    end
end

for i = 1:size
    for n = 1:round(max(PVAmYo)*100)
        if(n-1 <= PVAmYo(1,i)*100 && PVAmYo(1,i)*100 < n)
            dysplotPVAmYo(n,1)= dysplotPVAmYo(n,1) +1;
        end
    end
end

for i = 1:size

```

```

    for n = 1:round(max(EFFmyo*100))
        if(n-1 <= EFFmyo(1,i)*100 && EFFmyo(1,i)*100 < n)
            dysplotEFFmyo(n,1)= dysplotEFFmyo(n,1) +1;
        end
    end
end

save
C:\Users\Cody\Documents\thesis\programs\paramsweep15\workspac
edata

subplot(2,2,1)
bar(dysplot,'c')
title('IFF-C.C.')
xlabel('% Dyssynchrony measured by mean IFF')
ylabel('frequency')
%axis([0 25 0 max(dysplot)])
axis([0 40 0 350])
% fh = figure(1);
% set(fh, 'color', 'white');
set(gca,'Box','off');
%figure
subplot(2,2,2)
bar(dysplot_t,'c')
title('IFF-4seg')
xlabel('% Dyssynchrony measured by mean IFF')
ylabel(' frequency')
%axis([0 25 0 max(dysplot_t)])
axis([0 40 0 350])
% fh = figure(2);
% set(fh, 'color', 'white');
set(gca,'Box','off');
%figure
subplot(2,2,3)
bar(dysplotCO,'c')
title('Cardiac Output')
xlabel('Cardiac Output (mL/sec)')
ylabel('frequency')
axis([min(CO)-5 max(CO)+5 min(dysplotCO) max(dysplotCO)+50])
% fh = figure(3);
% set(fh, 'color', 'white');
set(gca,'Box','off');
%figure
subplot(2,2,4)
bar(dysplotSV,'c')
title('Stroke Volume')

```

```

xlabel('Stroke Volume (mL)')
ylabel('frequency')
axis([min(SV)-5 max(SV)+5 min(dysplotSV) max(dysplotSV)+100])
fh = figure(1);
set(fh, 'color', 'white');
set(gca, 'Box', 'off');

```

```

figure
subplot(2,2,1)
bar(dysplotEF, 'c')
title('Ejection Fraction')
xlabel('Ejection Fraction (%)')
ylabel('frequency')
axis([min(EF)-5 max(EF)+5 min(dysplotEF) max(dysplotEF)+20])
set(gca, 'Box', 'off');
subplot(2,2,2)
bar(pWloss, dysplotWloss, 'c')
title('Myocardium Power Loss (P_l_o_s_s)')
xlabel('Power (W)')
ylabel('frequency')
axis([0 .4 0 300])
set(gca, 'Box', 'off');
%figure
subplot(2,2,3)
bar(dysplotEFF, 'c')
title('Efficiency (EW/PVA)')
xlabel('% efficiency (EW/PVA)')
ylabel('frequency')
axis([45 90 0 350])
set(gca, 'Box', 'off');
subplot(2,2,4)
bar(dysplotEFFmyo, 'c')
title('Efficiency (EW/PVA_m_y_o)')
xlabel('% efficiency (EW/PVA_m_y_o)')
ylabel('frequency')
axis([45 90 0 350])
set(gca, 'Box', 'off');
fh2 = figure(2);
set(fh2, 'color', 'white');

```

```

figure
subplot(2,2,1)
bar(pEW, dysplotEW, 'c')
title('External Work (EW)')
xlabel('External Work (J)')
ylabel('frequency')

```



```

axis([.25 .9 0 300])
set(gca,'Box','off');
subplot(2,2,2)
bar(pWmyo,dysplotWmyo,'c')
title('Myocardium Work (W_m_y_o)')
xlabel('Work (J)')
ylabel('frequency')
axis([.25 .9 0 300])
set(gca,'Box','off');
subplot(2,2,3)
bar(pPVA,dysplotPVA,'c')
title('Pressure Volume Area (PVA)')
xlabel('PVA (J)')
ylabel('frequency')
axis([.25 .9 0 300])
set(gca,'Box','off');
subplot(2,2,4)
bar(pPVAmYo,dysplotPVAmYo,'c')
title('PVA_m_y_o')
xlabel('PVA (J)')
ylabel('frequency')
axis([.25 .9 0 300])
set(gca,'Box','off');
fh3 = figure(3);
set(fh3, 'color', 'white');

```