A Reduced Order Systems Approach to Prediction of Emergent Behaviors of Biological Systems

by

Michaëlle Ntala Mayalu

B.S., Mechanical Engineering, Massachusetts Institute of Technology, 2010
S.M., Mechanical Engineering, Massachusetts Institute of Technology, 2012

Submitted to the Department of Mechanical Engineering in partial fulfillment of requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

© Massachusetts Institute of Technology 2017. All right Reserved

Signature redacted

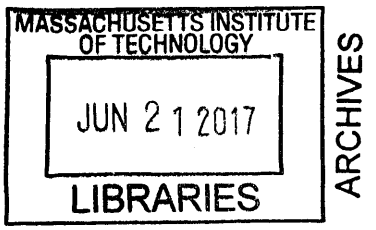
Signature of Author.....
Department of Mechanical Engineering
May 12, 2017

Signature redacted

Certified by.....
H. Harry Asada
Ford Professor Mechanical Engineering
Thesis Supervisor

Signature redacted

Accepted by.....
Rohan Abeyaratne
Chairman, Department Committee on Graduate Students



A Reduced Order Systems Approach to Prediction of Emergent Behaviors of Biological Systems

by

Michaëlle Ntala Mayalu

Submitted to the Department of Mechanical Engineering
On May 12, 2017, in partial fulfillment of the requirements of the degree of
Doctor of Philosophy

Abstract

One of the most fundamental questions in developmental biology and tissue engineering is how cells organize to form complex structures characterized by tissues, organs and whole organisms. The coordination of cells to form complex structures is facilitated by their communication via the surrounding gel (or extracellular matrix) where they live. In addition to answering questions in development, studying how cells communicate and coordinate over distance via the extracellular matrix (ECM) can give insight into pathological situations such as cancer metastasis, and wound healing. Although the exchange of molecular and biochemical signals is a key mechanism in cell to cell communication, cells can also communicate biomechanically through the ECM. Modeling mechanical interactions between cells and the ECM can advance understanding of biomechanical signaling during tissue formation.

Observation of the mechanisms for mechanical interaction between contractile cells within an extracellular matrix has resulted in detailed models that can describe single-cell migration and spreading on (and within) various of substrates. By incorporating sub-cellular behaviors (such as focal adhesion dynamics, cytoskeleton remodeling, actin motor activity and remodeling of the surrounding fibrous matrix), these models can integrate both the purely mechanical interaction within the surrounding matrix as well as the internal adaptive response to mechanical cues from the surrounding matrix. As a result, a vast amount of simulation data can be created from analyzing single-cell/matrix interactions numerically. In addition, numerous cell types and environmental conditions may be represented by varying multiple parameters within the model. However, complex and extensive mechanisms involved in emergent behavior of multiple interacting cells and surrounding matrix may become intractable due to mathematical and computational complexity.

This thesis will address how we can exploit simulation data describing the nonlinear dynamics of single-cell/matrix behavior to create a reduced-order linear state equation in latent variable space. Furthermore, in order to predict multi-cell emergent behavior, the reduced-order linear models of single cells are used as components in a comprehensive framework based on linear superposition of mutually shared matrix dynamics.

The linear latent state equation describing the nonlinear dynamics of a single-cell and surrounding matrix is created in three steps. First, using Bond Graph Theory, a set of independent state equations(derived from the bond graph) may be augmented by adding equations using auxiliary variables necessary to "sufficiently inform" the nonlinear dynamics. This creates an augmented state space where a linear description of the nonlinear system can be found. Second,

the augmented (auxiliary and state) variables are simulated for various initial conditions. Using the resulting simulated data, we perform Principal Component Analysis in order to approximate a lower dimensional linear manifold within the augmented space. Third, we transform the augmented state equation to latent space representation by orthogonal projection onto the basis defined within the lower dimensional linear manifold. While the resultant latent state equation is linear, complex nonlinearities are embedded in the compact model, leading to precise and global linearization of nonlinear dynamics.

Using the linear representation of single-cell/matrix dynamics we may perform linear operations such as projection, to isolate matrix dynamics of individual cells, and superposition, to combine matrix dynamics of individual cells and approximate a multi-cell matrix environment. Using these linear operations, we can effectively link single-cell models to predict multi-cell emergent behaviors. The hypothesis to prove (drawn from experimental evidence) is that multiple cells can effectively interact by transmitting force to neighboring cells through the shared matrix environment.

In this thesis, I consider two models describing the nonlinear dynamics of single-cell/ECM mechanics. The first model is a 1-D lumped parameter model created to explore the aspects of cell sensing over an elastic ECM. Although it is possible to reproduce bio-mechanical interactive behaviors, polarity is not considered within the 1-Dmodel. The second model is a highly detailed biophysical distributed parameter system describing cell/ECM mechanics based on previous works and can accurately reproduce experimental observations.

Thesis Supervisor: H. Harry Asada

Title: Ford Professor Mechanical Engineering

Acknowledgements

My success, in finishing my thesis and my graduate studies, is due to the support of my advisor, thesis committee, family, friends, and lab members. I could not have done it without them.

I would like to thank Professor Asada for his rigorous training in technical writing for grants, conference and journal papers, effective communication for seminars and presentations, and in how to develop significant and thorough research methodologies. He involved me in an interdisciplinary field and I got to work on exciting and new topics incorporating biology, control theory, machine learning and computation. It was truly a privilege to learn with him and work with him. I would like to thank Professor Kamm for his open invitation to use his lab for early experiments the last minute use of his high performance computers for finishing my thesis. I am truly appreciative and am grateful for his kindnesses and support through my years as a graduate student and an EBICS (Emergent behaviors of Cellular Systems) trainee. I would also like to thank Professor Kemp whose expertise in the biological and modeling field is truly an inspiration and I am honored to have her as part of my committee. Just as importantly, I would like to thank Professor Del Vecchio, for her wise career advice and support and her openness to explore the different control and machine learning methodologies I proposed in my thesis.

All of the students and postdocs in the Asada lab have played a large role in my learning and professional growth. I want to especially thank MinCheol Kim who has been collaborating with me for the past 7 years including masters and PHD. I couldn't have made as much progress without his mentorship, guidance, and encouragement. He was always there for me, providing the most valuable advice and information and help with my computations. I can't thank him enough for all his help. I would also like to thank Hyeon Yu Kim for all her expertise and support in biological experiments. I have learned so much from her. I would also like to thank my good friend Grace Lee who, even though having a busy schedule as a top surgical resident at MGH, found time to keep my company as a studied and support and encourage me.

I would like to thank my parents Alfred and Mary Mayalu, for supporting and encouraging me at each stage of my academic and personal development. Every decision and sacrifice they took from transferring me to private school, to facilitating the opportunity for me to take college

calculus and physics in high school has formed me in the person that I am today. They have been tremendously supportive in every way. I could not ask for better parents. I would like to thank my sister, Nancy, whose hard work ethic and perseverance at her job as GE executive it truly inspiring. I could be the person I am today without our long conversations and her sage advice. I would also like to thank my brother Alfred Jr who could understand the stresses and difficulties of being a graduate student (since he is currently a graduate student at Virginia tech). I also greatly enjoyed the summer we spent together while he interned for MIT Lincoln labs. He was always eager to learn new concepts and teach me about areas out of my expertize. His excitement for the pursuit of knowledge is truly an inspiration. I would like to conclude by thanking my undergraduate advisor and UROP supervisor, Professor Neville Hogan, who allowed me to explore and fall in love with the controls field. I am so grateful for the opportunity I had to work in his lab as an undergraduate student.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Biological Context for the Model

1.1.1 Cell–Cell Mechanical Interaction within an Extracellular Matrix Environment can drive Emergent Behavior

Biomechanical interaction between cells is a fundamental mechanism during the processes of development, cancer metastasis, and wound healing. For all these processes, cells must communicate over distance through the protein gel (extracellular matrix) in which they live[1]–[6]. It is known that there is a complex interplay between both biochemical and biomechanical signals to mediate the interaction among cells and matrix components [4], [7], [51], [53]. These interactions ultimately lead to coordinated movement of cells to produce more complex structures but are still not completely understood[5].

Remarkably Guo et al. revealed that mechanical force alone transmitted through the extracellular matrix (ECM), with the attenuation of diffusive factors secreted by cells, can initiate long-range traction forces to create cell migration and patterning[3]. This evidence suggests that matrix-mediated mechanical communication is critical for robust cell-cell interactions and pattern formation. Therefore, examining the mechanical aspect of intercellular communication through the ECM can give much insight into multi-cell emergent behaviors leading tissue formation and development [1], [8], [9]. Although the mechanisms driving multi-cell patterning and coordinated cell organization are still poorly understood, a wide range of likely mechanisms have been observed and inferred through experiments[3]–[6], [8]–[10]. Based on these inferences, mathematical and computational models may be created[1], [4], [5], [11]–[13].

1.1.2 Emergent Changes in ECM through Cooperative Mechanical Interaction of Cells

To better understand how mechanical cell-cell communication via the ECM can coordinate the self-organization of cells to form complex structures, some mechanisms to consider are the intermediate emergent behaviors arising within the matrix itself. The extracellular matrix(ECM) serves as the medium in which biochemical and molecular signals can be transmitted between cells to promote collective response[6], [54]. However, in general, these signals are short-lived and

move over short distances depending on molecular size[5]. Mechanical signals, in the form of tissue strains and stresses[1], [8], [9] [3], [14], can not only mediate short-range, mechanical cell-cell communication[6], but also act over long distances and integrate mechanical information over the whole tissue[3], [10].

Specifically Winer et al. showed that fibroblasts and human mesenchymal stem cells on fibrin deform the substrate by several microns up to five cell lengths away from their plasma membrane leading to long distance cell-cell communication [8]. Furthermore, Guo et al. observed long-range non-dispersed (i.e., confined in the central region connecting two cells) force transmission within collagen/matrigel mixtures though measurement of increased deformation velocity between cells. Non-dispersed force transmission is a crucial mechanism for the initiation and maintenance of long-scale multi-cell linear patterns[3]. Finally, Fernandez et. al showed that cells (osteoblasts and fibroblasts) embedded inside a 3-Dimensional collagen gel could spontaneously contract the entire gel volume via collective contractile activity, an important mechanism behind tissue formation. Furthermore, the authors showed that this behavior is dependent on the number and spacing (i.e. density) of cells within the gel [10]. These findings suggest that cell induced emergent mechanical changes within ECM are a critical step in further emergence leading to tissue formation[3], [6], [10].

This thesis will show that it is possible to predict the aforementioned intermediate emergent behaviors arising within the matrix using a comprehensive framework that integrates multiple linearized models describing single-cell/ECM mechanics. The key construct behind the approach is the superposition of mechanical forces propagated within the ECM by each individual cell model. Under the correct mathematical formulation we can show that the aforementioned construct is sufficient to reproduce:

1) Experimentally observed long-range non-dispersed force transmission between cells through measurement of increased deformation velocity between cells [3].
2) Experimentally observed global contraction of gel volume via collective cell-contractile activity (as opposed to local deformations of single cell embedded within the gel)[10].

Through our study of intermediate emergent behaviors arising within the matrix we can advance our understanding of biomechanical signaling mechanisms during tissue formation and multi-cellular patterning.

The cell induced emergent mechanical changes within the ECM described above suggest that individual cells interact mechanically through integration of the complicated strain fields propagating within the ECM network[10]. Therefore in our model, we consider the superposition of individual subsystems (describing single cell-ECM interactions) to these elucidate interactive behaviors. To this end we must construct the individual subsystems in such a way to facilitate superposition.

1.1.3 Fundamental Components of Single-Cell Matrix Interaction

As mentioned in the previous section, it is desired to model the integration of the complicated strain fields (induced by individual cells) propagating within the ECM network in order to reproduce emergent matrix behaviors. Therefore we first model then superpose individual subsystems (describing single cell-ECM interactions).

Let us consider the basic building block: an individual cell's interactions with the surrounding matrix environment. Experimental evidence suggests that cellular traction forces produce local strains in the matrix, which can affect the motility of nearby cells [6] . Thus along with the ability of the extracellular matrix to transmit stresses, cells can also internally modulate their state in response to mechanical input[6], [10]. The key concept is that cell-matrix interactions are fed back to the cell, which influences cell polarity, contractility, stiffness and strength of focal adhesions [14]–[16]. Specifically, Lo et al., observed experimentally that local change in substrate tension of 3T3 fibroblasts (on flexible polyacrylamide sheets coated with type I collagen) caused the cell to change its anterior-posterior polarization, and moved towards the stiffer local gradient[15].

These findings suggest that, when modeling single-cell/matrix mechanics it is important to consider the complex interplay between purely mechanical phenomena (resulting from crosstalk between cell contractility and matrix mechanics) and the cell's internal adaptive response (resulting from external mechanical cues). Therefore, we propose to model the individual components (describing single cell interactions) within our more complex modeling framework by using cell polarity to connect the cell's internal adaptive response to bi-directional cell-matrix mechanical interaction. Here, cell polarity is defined as asymmetry in cell shape, distinguishing the anterior-posterior regions of the cell. Given the strain characteristics of the surrounding matrix (influenced by bi-directional cell/matrix mechanical interactions), the cell modifies its polarity

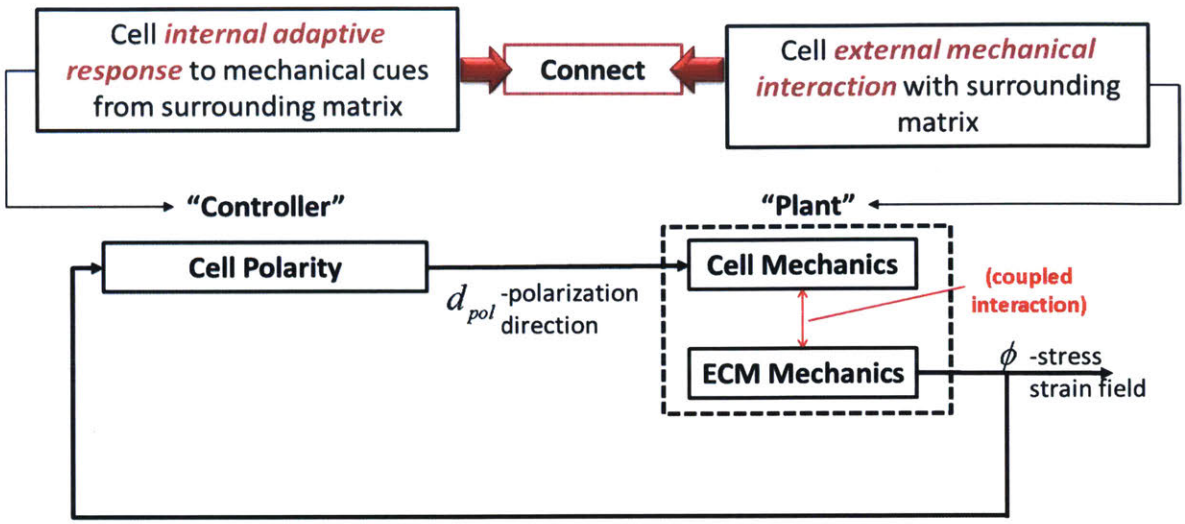


Figure 1: Block diagram showing the connection between purely mechanical phenomena (resulting from crosstalk between cell contractility and matrix mechanics) and the cell's internal adaptive response within the model.

which in turn changes the protrusion of pseudopodia and lamellipodia. This initiates formation of new adhesions, and the release of old adhesions on the ECM substrate, and development of traction[15]–[17]. As the ECM substrate, and development of traction[15]–[17]. As previously discussed, the traction forces propagated throughout the surrounding matrix. In order to predict the more complex collective behavior involved in emergent matrix interactions, our model superposes the propagation these forces for each individual cell.

1.2 Challenges of Modeling Matrix-Mediated Mechanical Interactions at the Population Scale

Current computational models that are based on mechanistic understanding at the single-cell and sub-cellular scale can be used to study interaction between two or three cells with sufficient detail[11]–[13]. Based on first principles, these computational models can predict dynamic behaviors of cell-ECM interactions. However, the number of cells capable of being modeled is limited by mathematical and computational complexity required to describe the sub-cellular mechanisms involved.

Traditionally, agent based modeling is used to study emergent phenomena because it can predict the adaptive behavior of individual components as a result of underlying rules [4]–[6], [18].

However, when simulating large populations, current agent-based models often require abstraction of details at the single-cell and sub-cellular scale in order to limit mathematical and computational complexity. In addition, describing the interactions between agents are often determined somewhat heuristically [18]. In the context of representing mechanical cell-cell communication via the ECM, Rehnart et al. used a 2-Dimesional hybrid cellular Potts and finite element model to reproduce observed single cell behavior, pairwise cell interactions and collective cell behavior[6]. However abstractions were necessary to keep the simulation tractable. These included isotropic and linear elasticity assumptions of the ECM and exclusion of cell-substrate adhesion from the model. As a result, nonlinear strain stiffening mechanisms could not be sufficiently modeled and the model contradicted several important experimental observations[6]. Furthermore in order to increase efficiency, Rehnart et al. modeled the ECM as a finite element model instead of a more realistic discrete fibrous network[6]. The fibrous nature of the extracellular matrix (ECM), and the presence of cross-linked fibers is critical for transmission large scale of forces [1], [9] and therefore a key component in model cell to cell mechanical interactions. As can be deduced, these simplifications may put limitations to the translational potential in clinical settings [19].

It is therefore necessary to develop a methodology that can take advantage of the modular architecture in agent-based models while retaining sufficient mechanistic detail with reduced complexity. In addition, we would like to develop a method that facilitates the integration of individual subsystems to represent complex collective behaviors. To this end, we have proposed a linearized agent-based framework comprised of linearized components based on previously simulated data obtained from detailed single-cell mechanistic computational models. The linearized formulation of each agent allows for the superposition of multiple agents to simulate multi-cell interactions. Using this method, computational expense and time are decreased significantly and sufficient mechanistic detail is retained in the simulation.

1.3 Prior Work on Representing Large Scale Dynamical Systems via Low-dimensional Projection and Feature Extraction of Simulated data

Previous works on model order reduction of nonlinear dynamic systems develop multi-variate statistical methods that employ order reduction via low-dimensional projection and feature extraction of simulated data[20]–[25]. Generally, these methods have applications in structural

dynamics, fluid mechanics, damage detection, and multibody systems. Within these applications, a very fine scale description is necessary compared to the size of the structure, and the finite element discretization of the underlying partial differential equations leads to large, potentially highly nonlinear (therefore requiring a fine discretization in time), numerical problem[23].

Within these previous methods, "snapshots" of the state are generated during preliminary simulations of the partial differential equations. These "snapshots" are collected in a matrix($\boldsymbol{X}$) and the Singular Value Decomposition of $\boldsymbol{X}$ is computed. Assuming that the first k singular values are ''large" with respect to the succeeding ones, $\boldsymbol{X}$ can be approximated by means of orthogonal projection of original states onto the k the left-singular vectors of $\boldsymbol{X}$ [22]. Using the results it is possible to re-write and approximate of the state and output equations of the original system within a lower dimensional space.

However, contrary to the proposed approach described in this thesis, the resulting reduced-order equations described in these works remain nonlinear and therefore would not work for our proposed linear superposition approach. This signifies that the reduced-order computation is still dependent on the original system order and nonlinearities contained within the original system. Furthermore, in previous works the original mechanistic system used to simulate data is treated as a "black box" and consequently the selected simulated variables lack physical meaning and are insufficient to describe dynamics if the dynamics lie within a nonlinear manifold [20], [22]. Although the reduced order equations may be linearized using Taylor series expansion, the resulting linearization produces a less accurate model that the proposed approach[26].

1.4 Contributions of this Thesis

The main objective of this thesis is predict emergent behaviors involving matrix-mediated mechanical interaction between populations of cells on an elastic matrix substrate. From the above discussion, it is clear that the proposed work can give insight not only to the study and control on interacting cells but also the general approach may be applied to systems of interacting nonlinear agents, which would otherwise be prohibitively complex to compute.

In the following chapters we will describe the multiphase approach to prediction of multi-cell systems:

Phase 1: Creation of Linear Latent State Equation Describing the Single-cell/matrix dynamics by:

a. Use of Bond Graph theory to augment the original independent state equations by adding auxiliary variables necessary to "sufficiently inform[26]" the nonlinear dynamics.
b. Creation of dataset through simulation of auxiliary and state variables for various initial conditions and application of Principal Component Analysis in order to approximate a lower dimensional linear manifold within the augmented space
c. Transformation of the augmented state equation to a latent space representation by orthogonal projection onto the basis defined within the lower dimensional linear manifold

Phase II: Linking single-cell models to predict multi-cell emergent behavior by:

a. Use of linear projection, to isolate matrix dynamics of individual cells in latent space
b. Use of linear superposition to combine matrix dynamics of individual cells and approximate a multi-cell environment

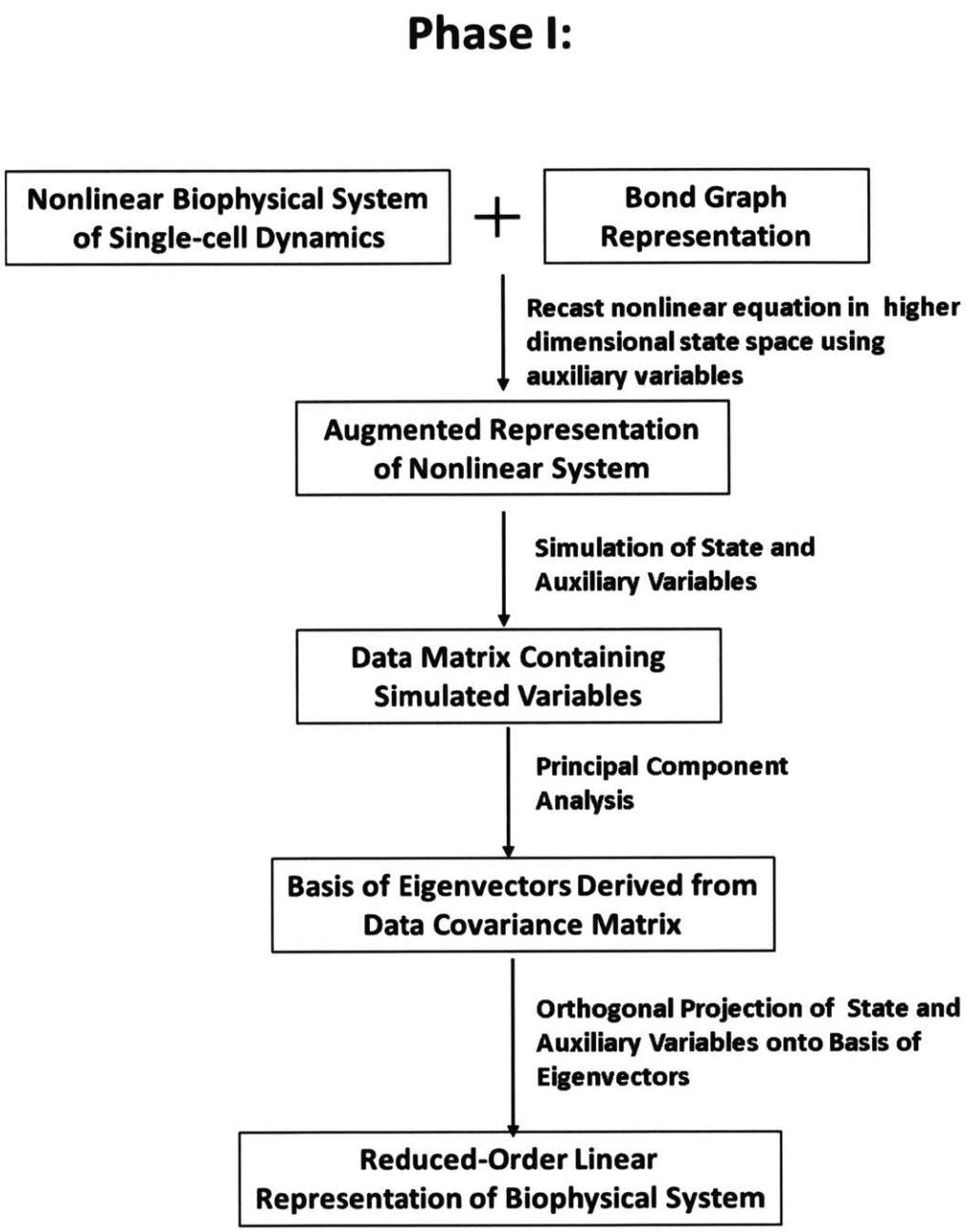


Figure 2: The approach consists of a two-phase system of first creating a reduced order linear model of a single cell. We first augment the system to a higher dimensional space and then use latent analysis to transform to the system to a linear latent space where we can truncate to lower order.

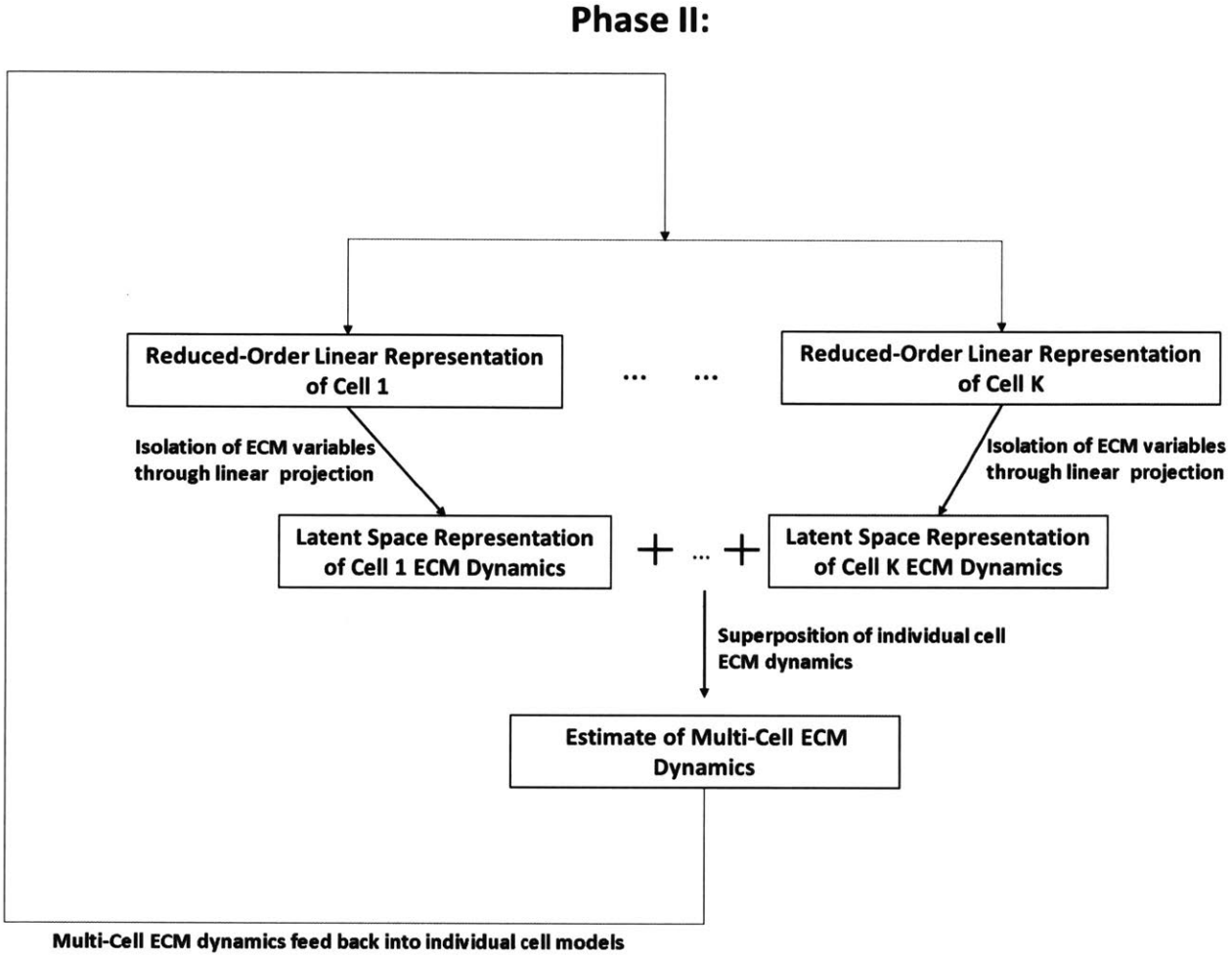


Figure 3: The approach consists of a two-phase system of second of which is linking single-cell models to predict multi-cell emergent behavior. First we isolate the matrix dynamics of each model representing single cell matrix mechanics. Then we superpose the ECM dynamics in order to create a estimate of the multi-cellular matrix environment.

In Chapter 2, the general methodology and underlying assumptions are discussed. Chapter 3 applies the approach to a 1-Dimensional lumped parameter case study capable of reproducing bio-mechanical interactive behaviors. In Chapter 4, applies the approach to a highly detailed biophysical distributed parameter system describing cell/ECM mechanics[13]. In chapters 2 and 3, the general approach is modified or extended as needed for each model. Chapter 5 is a discussion and preliminary analysis for extension of the approach to a fully 3-D biophysical distributed parameter system in which the cell is embedded within a 3-D fibrous matrix.

2. Methodology, Theory and Computation

2.1 Definition of Nonlinear System using Bond Graph Theory

2.1.1 Problem Statement

Consider nonlinear lumped or distributed (finite mode) parameter dynamical system described by state equation:

$$\frac{dx}{dt} = g(x) \in \mathbb{R}^{n\times 1} \tag{1}$$

Where x represents the $n \times 1$ state variable vector. We wish to use Bond Graphs to:

1. Graphically represent the bi-directional exchange of energy through the dynamical system
2. Identify energy storage and dissipation elements and how they connect

In addition we wish to use Bond Graph Theory to show the following:

1. That the equations describing the class of systems capable of being characterized by bond graphs may be written as a linear combination of nonlinear terms. These nonlinear terms represent effort and flow within the system.
2. That through augmentation of the system to include nonlinear effort and flow variable dynamics, we can effectively compensate for the nonlinear dynamics within the system

2.1.2 Bond Graph Representation

Bond graphs are used to describe lumped/distributed parameter multi-energy domain systems ranging from mechanical, electrical and hydraulic but can also seamlessly incorporate chemical, thermodynamic or biophysical domains within the same representation[27]–[29]. Bond graphs can even be used to describe biochemical networks[30], [31]. Because of these characteristics, it is the natural choice to represent the aforementioned biophysical system describing cell/ECM mechanics.

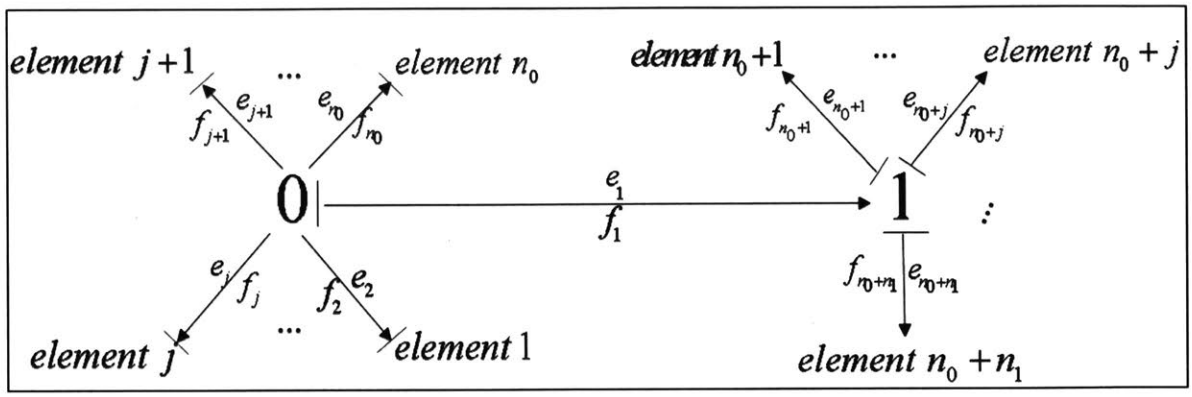


Figure 4: General Diagram of Bond Graph

Consider the bond graph shown in Fig. 4. Modeling elements (e.g. masses, springs, and dampers) representing energy storage and dissipation in the system are connected through lines, termed power bonds.

Two variables (e,f) (termed power variables) are associated with each bond, the product of which represents power flowing through the bond. In the mechanical domain, variables (e,f) represent the force (or effort) and velocity (or flow) respectively. The arrow of each bond represents assumed direction of positive energy flow.

One key feature of the bond is causality, which is represented by a vertical bar placed on one end of the bond. Causality explains which of the power variables are dependent and which are independent. Furthermore, causality analysis allows us to find causal relationships among all the elements and obtain a computable procedure for determining state transitions.

The 1's and 0's in the Bond Graph (termed junctions) represents Kirchhoff's Voltage Law (sum of efforts equals zero), and to Kirchhoff's Current Law (sum of flows equals zero) respectively. Using the generalized graph in figure 4 we may write the equations represented by the junctions as:

$$\begin{aligned} &\sum_{j=1}^{n_0} f_j = 0 && \sum_{j=n_0+1}^{n_0+n_1} e_j = 0 \\ &e_1 = e_j \ \ (j = 2,\ldots,n_0) && f_1 = f_j \ \ (j = n_0+1,\ldots,n_0+n_1) \end{aligned} \tag{2}$$

Similarly we may write equations for the elements characterized by constitutive laws that are functions of the original state variables (x):

$$\begin{aligned} e_j &= \Phi_{e,j}(x) \\ f_j &= \Phi_{f,j}(x) \end{aligned} \tag{3}$$

2.1.3 Deriving State Equations using Bond Graphs

The first step to derive the state equations, is to list all of the governing equations for the bond graph. The table below, summarizes the relationship between bonds and their governing equations:

	Bond w/ Causality	Governing equation
Modeling Element	$\rightarrow\vert\, element_j$	$f_j(t) = \Phi^e{}_{f,j}(e(x)) = \Phi_{f,j}(x(t))$
	$\vert\rightarrow element_j$	$e_j(t) = \Phi^f{}_{e,j}(f(x)) = \Phi_{e,j}(x(t))$
0 Junction	$\begin{array}{c}\downarrow \\ \rightarrow\vert\, 0 \,\leftarrow\vert \\ \uparrow\end{array}$	$\sum_{j=1}^{n_0} f_j = 0$ $e_1 = e_j \ (j = 2,\ldots,n_0)$ (n_0 *total # of bonds surrounding* 0 *junction*)
1 Junction	$\begin{array}{c}\downarrow \\ \vert\rightarrow 1 \vert\leftarrow \\ \uparrow\end{array}$	$\sum_{j=1}^{n_1} e_j = 0$ $e_1 = e_j \ (j = 2,\ldots,n_1)$ (n_1 *total # of bonds surrounding* 1 *junction*)

Table 1: Summary of relationship between bonds and their governing equations

The governing equations for the general graph given in figure 4 are the following:

$$\left.\begin{array}{l} 0\ \textit{junction equations}: \\ \textit{for } j = 1,\ldots,n_0 \\ 1.\ \sum_{j=1}^{n_0} f_j = 0 \\ 2.\ e_1 = e_j \\ 1\ \textit{junction equations} \\ \textit{for } j = n_0 + 1,\ldots,n_0 + n_1 \\ 3.\ \sum_{j=n_0+1}^{n_0+n_1} e_j = 0 \\ 4.\ f_1 = f_j \\ \textit{constitutive equations from elements} \\ \textit{for } j = 1,\ldots,n_0 + n_1 \\ e_j = \Phi_{e,j}(x) \\ f_j = \Phi_{f,j}(x) \end{array}\right\} \Rightarrow \frac{dx}{dt} = g(x) \qquad (4)$$

Here we use causality to isolate the dependent effort/flow variable within each junction equation. If we substitute the constitutive equations (describing the effort and flow variables) into the junction equations we can algebraically manipulate the equations to regain equation(1).

Consequently, the equations in (4) will provide the appropriate state equations governing the dynamics of the original system. It is important to note that manipulations depends on the causality and nature of modeling elements (i.e. dissipative or energy storage) of the system. An example of these manipulations can be seen when the approach is applied to the 3-Deimstional system in Chapter 4.

Expanding equation (1) to reveal each individual state variable $x_i\,(i=1,...,n)$ in state variable vector $x \in \mathbb{R}^{n\times 1}$:

$$\underbrace{\begin{bmatrix} \frac{dx_1}{dt} \\ \vdots \\ \frac{dx_i}{dt} \\ \vdots \\ \frac{dx_n}{dt} \end{bmatrix}}_{\frac{dx}{dt}} = \underbrace{\begin{bmatrix} w_{11} & \cdots & \cdots & w_{1n_\eta} \\ \vdots & & & \vdots \\ w_{i1} & \cdots & \cdots & w_{in_\eta} \\ \vdots & & & \vdots \\ w_{n1} & \cdots & \cdots & w_{nn_\eta} \end{bmatrix}}_{W_\eta} \underbrace{\begin{bmatrix} \vdots \\ \Phi_{e,j}(x) \\ \vdots \\ \Phi_{f,j}(x) \\ \vdots \end{bmatrix}}_{\eta} \triangleq g(x) \tag{5}$$

Here, $W_\eta \in \mathbb{R}^{n \times n_\eta}$ is a weight matrix and $\eta \in \mathbb{R}^{n_\eta \times 1}$ is a vector (termed auxiliary vector) containing effort and flow variables represented by their constitutive laws. Essentially we have written the rate of change of each state variable $\left(\frac{dx_i}{dt}\right)$ as a linear combination of the effort and flow variables represented by their constitutive laws $\left(\Phi_{f,j}(x), \Phi_{e,j}(x)\right)$. This follows from the junction equations where the effort and flow variables are linearly superposed. It should be noted that η_i is only a subset of the constitutive equations since some were used during algebraic manipulation to isolate the state variables.

Equation (5) reveals a linear representation of the system with respect to auxiliary variable vector η_i. Furthermore we augment the system by explicitly writing the dynamics of η_i as:

$$\begin{aligned} \frac{dx}{dt} &= \mathbf{W}_\eta \cdot \boldsymbol{\eta} \\ \frac{d\boldsymbol{\eta}}{dt} &= \mathrm{H}(x, \boldsymbol{\eta}) \end{aligned} \tag{6}$$

Since the auxiliary variables represent nonlinear output of the constitutive laws, there dynamics comprise a different nonlinear structure than that of the original state variable dynamics. Therefore, adding the dynamics to the original state equations provides a richer and more complete description of the nonlinear system. Here we do not explicitly differentiate η_i but only acknowledge that it will be some nonlinear function $\mathrm{H}(x, \eta)$ of the original state and the auxiliary variables. Equation (6) shows that by augmenting the system we may decouple the dynamics into

linear and nonlinear parts. Writing the equations in this way we explicitly describe and account for the nonlinear dynamics within the system.

2.2 Linear Reduced Order Latent Representation using Simulated Data

2.2.1Problem Statement

Consider the augmented system given by equation(6). We wish to create a reduced order representation of the augmented system in latent variable space denoted by equation:

$$\frac{dz(t)}{dt} \approx \mathbf{A}z \in \mathbb{R}^{m\times 1} \tag{7}$$

Here, $m \ll n$. In order to create the reduced-order latent variable model, we must find the appropriate transformation matrix $V \in \mathbb{R}^{(n+n_\eta)\times m}$ that will allow the dynamics to evolve on a lower dimensional linear manifold. V may be derived from the principal components (or eigenvectors) of data covariance matrix:

$$C_{xx} = \frac{1}{N \cdot t_f} \mathrm{X}^{\mathrm{T}}\mathrm{X} \tag{8}$$

Where X is a collection of simulated time samples of variables x and η_i for $t = 1,\ldots,t_f$ and N different initial conditions.

2.2.2 Basis of Eigenvectors Derived from Data Covariance Matrix

Let us define an augmented variable vector as:

$$\tilde{x} = \begin{bmatrix} x \\ \eta \end{bmatrix} = \in \mathbb{R}^{(n+n_\eta)\times 1} \tag{9}$$

As previously mentioned we create a data matrix X from simulated data from variables x and η :

$$\mathrm{X}=\begin{bmatrix} x(t_{0,1})^T & \eta(t_{0,1})^T \\ x(t_{1,1})^T & \eta(t_{1,1})^T \\ \vdots & \vdots \\ x(t_{f,1})^T & \eta(t_{f,1})^T \\ \vdots & \vdots \\ \vdots & \vdots \\ x(t_{0,N})^T & \eta(t_{0,N})^T \\ x(t_{1,N})^T & \eta(t_{1,N})^T \\ \vdots & \vdots \\ x(t_{f,N})^T & \eta(t_{f,N})^T \end{bmatrix} \in \mathbb{R}^{N\cdot t_f \times (n+n_\eta)} \tag{10}$$

We perform Eigen-decomposition on the covariance matrix:

$$\begin{aligned} &C_{xx}=\frac{1}{N\cdot t_f}\mathbf{X}^{\mathrm{T}}\mathbf{X}\approx\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}\in\mathbb{R}^{(n+n_\eta)\times(n+n_\eta)} \\ &\mathbf{V}=\begin{bmatrix}\mathbf{V}_x\\ \mathbf{V}_\eta\end{bmatrix}\in\mathbb{R}^{(n+n_\eta)\times m}\Rightarrow\mathbf{V}_x\in\mathbb{R}^{n\times m};\mathbf{V}_\eta\in\mathbb{R}^{n_\eta\times m} \\ &m-\#\ of\ eigenvectors\ after\ truncation \end{aligned} \tag{11}$$

Where V is a matrix consisting of m eigenvectors corresponding to the m largest eigenvalues in eigenvalue matrix $\boldsymbol{\Lambda}$. The number of eigenvectors kept (m) can either be determined through analysis of variance, cross-validation or examination of unexplained output variance[32]. In the case of cell/matrix model being studied we can also determine m based on the magnitude of certain dynamics within the approximated system.

Through orthogonal projection, we may derive the latent space representation of the augmented state variable vector $\tilde{x}$:

$$z(t)=\mathbf{V}^{\mathrm{T}}\underbrace{\begin{bmatrix}x(t)\\ \eta(t)\end{bmatrix}}_{\tilde{x}(t)}=\mathbf{V}_x{}^{\mathrm{T}}x(t)+\mathbf{V}_\eta{}^{\mathrm{T}}\eta(t)\in\mathbb{R}^{m\times 1} \tag{12}$$

We can also approximate the original augmented variable vector (and consequently x and η) using latent variable (z):

$$\underbrace{\begin{bmatrix} x(t) \\ \eta(t) \end{bmatrix}}_{\tilde{x}(t)} \approx \mathbf{V}z(t) \Rightarrow x(t) \approx \mathbf{V}_x z(t); \eta(t) \approx \mathbf{V}_\eta z(t) \tag{13}$$

2.2.3 Calculation of Linear State Equation in Reduced-Order Latent Variable Space

Differentiating(6):

$$\frac{d\mathbf{z}}{dt}(t) = V_x^T \frac{dx}{dt} + V_\eta^T \frac{d\boldsymbol{\eta}}{dt} \tag{14}$$

Substituting equation (5) into equation(14):

$$\frac{d\mathbf{z}}{dt} = V_x^T \left(W_\eta \boldsymbol{\eta}\right) + V_\eta^T \frac{d\boldsymbol{\eta}}{dt} \tag{15}$$

We now substitute the approximate of ηfound using latent variable z, given by equation(12)

$$\frac{d\mathbf{z}}{dt} \approx V_x^T \left(\mathbf{W}_\eta \mathbf{V}_\eta z\right) + V_\eta^T \frac{d\boldsymbol{\eta}}{dt} \tag{16}$$

Since the dynamics of η is nonlinear (as shown in equation(6)) we wish to approximate $\frac{d\eta}{dt}$ using latent variable z .This is in order to linearize the nonlinear dynamics with respect to the latent representation of the state and auxiliary variables. It is important to note that if we attempted to linearize $\frac{d\eta}{dt}$ with respect to $\frac{dx}{dt}$ using a fixed Jacobian, the resulting structure would be collinear and add no new information to the system. Furthermore, if auxiliary variables consisted of linear constitutive laws, the equations would also be collinear and therefore redundant. Consequently, it is necessary to include only nonlinear constitutive laws within the auxiliary variables and that furthermore linearize with respect to the state and auxiliary variables as opposed to linearizing the Jacobian. We therefore apply principal component regression (or PCR). PCR allows for the regression of a specified output on the latent variables $z \in \mathbb{R}^{m\times 1}$. Using ordinary least squares we can obtain a vector of estimated regression coefficients (equal to the number of latent variables). Let the output data matrix be defined as:

$$\mathbf{Y} = \begin{bmatrix} \dot{\eta}(t_{0,1})^T \\ \dot{\eta}(t_{1,1})^T \\ \vdots \\ \dot{\eta}(t_{f,1})^T \\ \vdots \\ \vdots \\ \dot{\eta}(t_{0,N})^T \\ \dot{\eta}(t_{1,N})^T \\ \vdots \\ \dot{\eta}(t_{f,N})^T \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{n_\eta} \end{bmatrix} \in \mathbb{R}^{N \cdot t_f \times n_\eta} \tag{17}$$

$$where\ \dot{\eta} = \frac{d\eta}{dt}$$

And the latent variable matrix be defined as:

$$\mathbf{Z} = \begin{bmatrix} z(t_{0,1})^T \\ z(t_{1,1})^T \\ \vdots \\ z(t_{f,1})^T \\ \vdots \\ \vdots \\ z(t_{0,N})^T \\ z(t_{1,N})^T \\ \vdots \\ z(t_{f,N})^T \end{bmatrix} \in \mathbb{R}^{N \cdot t_f \times m} \tag{18}$$

Here we have estimated $\frac{d\eta}{dt}$ numerically from simulated time samples of η_i using the backward Euler approximation:

$$\dot{\eta}(t) \approx \frac{\eta(t) - \eta(t - \Delta t)}{\Delta t} \tag{19}$$

For the *jth* column in data matrix **Y** we consider each sample individually:

$$\mathbf{y}_j = \begin{bmatrix} y_j(t_{0,1}) \\ y_j(t_{1,1}) \\ \vdots \\ y_j(t_{f,1}) \\ \vdots \\ \vdots \\ y_j(t_{0,N}) \\ y_j(t_{1,N}) \\ \vdots \\ y_j(t_{f,N})^T \end{bmatrix} \Rightarrow \hat{y}_j(t) = K_j z(t) \quad (t = t_{0,1}, t_{1,1}, \ldots, t_{f,1}, t_{1,N}, \ldots t_{f,N}) \quad (20)$$

Where $K_j \in \mathbb{R}^{1\times m}$ is the vector of regression coefficients. The ordinary least squares estimate of K_j is found by:

$$K_j^{\,o} = \arg\min \sum_{t=t_{0,1}}^{t_{f,N}} (y_j(t) - K_j z(t))^2 \quad (21)$$

Finally we write:

$$\begin{aligned} &\mathbf{Y} \approx \mathbf{ZK} \\ &\mathbf{K} = \begin{bmatrix} K_1^{\,T} & K_2^{\,T} & \cdots & K_{n_\eta}^{\,T} \end{bmatrix} \in \mathbb{R}^{m\times n_\eta} \end{aligned} \quad (22)$$

Using these results we approximate $\frac{d\eta}{dt}$ as:

$$\widehat{\frac{d\eta}{dt}} = \hat{\dot{\eta}}(t) = \mathbf{K}\cdot z(t) \quad (23)$$

Substituting equation (23) into (16) we obtain:

$$\dot{z}(t) \approx \underbrace{\left(\mathbf{V}_x{}^{\mathrm{T}}\mathbf{W}_\eta\mathbf{V}_\eta + \mathbf{V}_\eta{}^{\mathrm{T}}\mathbf{K}\right)}_{A} z(t) \tag{24}$$

Examining (24) we have effectively represented the augmented system in(6) in latent variable space.

2.3 Linking Individual Reduced Order Models through the Dynamics of Shared Variables

2.3.1Problem Statement

Let us now consider that the previously described dynamical system is a component of a larger (more complex) nonlinear system where multiple components are connected through a shared field (ϕ). Each individual component influences the dynamics of the field state. In the context of biophysical system being studied, each component represents the dynamics of single-cell/ECM interaction. In the larger more complex system, multiple cells interact through as shared ECM environment. Therefore the field represents the ECM.

We wish to predict the behavior of the larger more complex nonlinear system by linking the linear latent space models of the individual components through the field. The linearity of the dynamics of individual components facilitates the integration to from a more complex system since we may employ linear systems analysis. This will be accomplished by obtaining an estimate the nonlinear dynamics of the shared (multi-component) field based on influence of each individual component.

We have made several assumptions:

1. Each component has the same A matrix (in equation(24)) but starts at different initial conditions. In the context of the biophysical model being studied, this means that each cell (component) expresses the same phenotype. Under this assumption, the kith component may be defined by:

$$\begin{aligned} \dot{z}^k(t) &\approx \underbrace{\left(\mathbf{V}_x{}^{\mathrm{T}}\mathbf{W}_\eta\mathbf{V}_\eta + \mathbf{V}_\eta{}^{\mathrm{T}}\mathbf{K}\right)}_{A} z^k(t) \\ z^k(0) &= z^{k,0} \end{aligned} \tag{25}$$

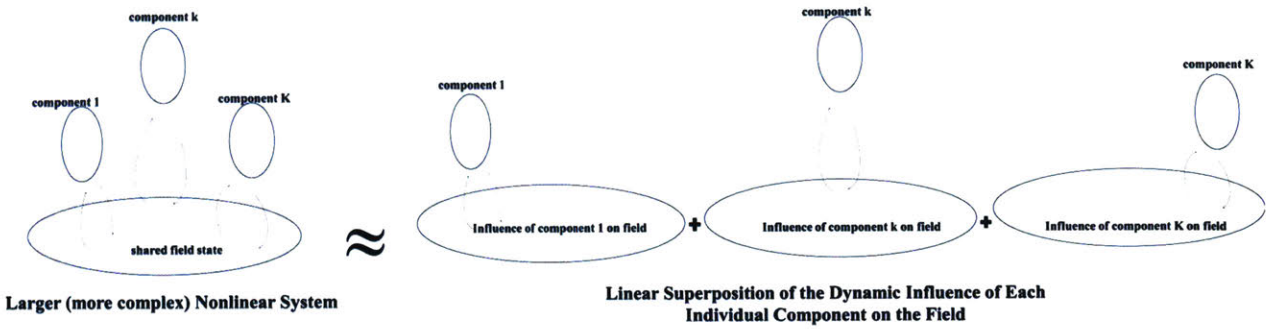


Figure 5: Conceptual diagram showing how the more complex nonlinear system will be estimated using superposition of the influence of multiple linear components through a shared field.

Where we have used the linear latent space model to describe the dynamics of the component.

2. Components are not directly connected, and can only influence each other through the shared field. This means that the field is the only means of energy exchange between components. In the context of the biophysical model being studied we further restrict to purely mechanical energy exchange between components and the shared ECM. Although multiple energy domains exist for the system, examining the mechanical aspect could be sufficient to describe critical behaviors[3].
3. The change in field state by each individual component it sufficiently small between time points. In the context of biophysical system being studied this means that the strain of the cell on the ECM is below 5%. (i.e. large deformations do not occur quickly between time points.)

Under these assumptions, we may find an estimate of the shared (multi-component) field through linear superposition of the dynamic influence of each individual component on the field. Although, the estimation may not be exact, it will be valid if it can reproduce the specified behaviors and interactions characterized in the complex nonlinear system.

Let us rearrange the augmented variables defined in $\tilde{x}$ (equation(26)) into two distinct groups that represent field variables (ϕ) and component variables (α). For the *kth* component:

$$\tilde{x}^k = \begin{bmatrix} x^k \\ \eta^k \end{bmatrix} \xleftrightarrow{mutation} \begin{bmatrix} \alpha^k \\ \phi^k \end{bmatrix} \in \mathbb{R}^{(n+n_\eta)\times 1} \tag{26}$$

Here, $\alpha^k \in \mathbb{R}^{n_\alpha \times 1}$ is the group of state and auxiliary variables corresponding to component k . In the case of the biophysical model being studied, this corresponds to the position and velocities of the cell and the internal elastic forces and damping forces of the cell. $\phi^k \in \mathbb{R}^{n_\phi \times 1}$ is the group of state and auxiliary variables corresponding to the field with respect to an isolated component/field system. In the case of the biophysical model being studied, this corresponds to the deformations, traction forces and viscoelastic forces of the ECM considering interaction with a single, isolated cell.

2.3.2 Isolation of Shared Variables Dynamics

The first step in linking the linear latent space models of the individual components is to isolate the field variables $\left(\phi^k\right)$ in latent space. First we isolate these variables in the original (augmented) space:

$$\tilde{x}^k = \begin{bmatrix} \alpha^k \\ \phi^k \end{bmatrix} = \begin{bmatrix} \alpha^k \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \phi^k \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{n_\alpha \times n_\alpha} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \alpha^k \\ \phi^k \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_{n_\phi \times n_\phi} \end{bmatrix} \begin{bmatrix} \alpha^k \\ \phi^k \end{bmatrix} \tag{27}$$

Using the previously derived orthogonal projection we may write the latent space representation as a sum of the latent representations of the component and field separately:

$$\begin{aligned} z^k &= V^T \begin{bmatrix} \alpha^k \\ \phi^k \end{bmatrix} = V^T \begin{bmatrix} \alpha^k \\ 0 \end{bmatrix} + V^T \begin{bmatrix} 0 \\ \phi^k \end{bmatrix} = V^T \begin{bmatrix} I_{n_\alpha \times n_\alpha} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha^k \\ \phi^k \end{bmatrix} + V^T \begin{bmatrix} 0 & 0 \\ 0 & I_{n_\phi \times n_\phi} \end{bmatrix} \begin{bmatrix} \alpha^k \\ \phi^k \end{bmatrix} \\ &\approx \underbrace{V^T \begin{bmatrix} I_{n_\alpha \times n_\alpha} & 0 \\ 0 & 0 \end{bmatrix} Vz(t)}_{z_\alpha^k} + \underbrace{V^T \begin{bmatrix} 0 & 0 \\ 0 & I_{n_\phi \times n_\phi} \end{bmatrix} Vz(t)}_{z_\phi^k} \end{aligned} \tag{28}$$

Were we have substituted for the approximate the original augmented variable vector defined in equation(13). We define P_ϕ and $\left(I - P_\phi\right)$ as projection matrices to isolate the field components in latent space:

$$
\begin{aligned}
z^k &= z_\phi^k + z_\alpha^k \\
z_\phi^k &= P_\phi z^k \\
z_\alpha^k &= \left(I - P_\phi\right) z^k \\
P_\phi &= V^T \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_\phi \times n_\phi} \end{bmatrix} V; \left(I - P_\phi\right) = V^T \begin{bmatrix} \mathbf{I}_{n_a \times n_a} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} V
\end{aligned} \tag{29}
$$

2.3.3 Superposition of Shared Variables of Shared Variable Dynamics

Using the above definition, we may link the field dynamics of each individual component in the following manner:

$$
\begin{aligned}
\dot{z}^k &= \left(I - P_\phi\right) \dot{z}^k + \dot{z}_\phi^k \\
\dot{z}_\phi^k &= P_\phi \dot{z}_\phi^k
\end{aligned} \tag{30}
$$

$$
\dot{z}^{k,emergent} = \dot{z}^k + \sum_{\ell \neq k} \dot{z}_\phi^\ell = \left(I - P_\phi\right) \dot{z}^k + \underbrace{\dot{z}_\phi^k + \sum_{\ell \neq k} \dot{z}^\ell}_{\dot{z}_\phi^{emergent}} \tag{31}
$$

In the above equations we have modified dynamics of component k by essentially replacing the individual influence of the field $\left(\dot{z}_\phi^k\right)$ by the estimate of the shared field defined by:

$$
\dot{z}_\phi^{emergent} = \sum_{\ell=1}^{K} \dot{z}_\phi^\ell \tag{32}
$$

Note that the model of the individual component (equation(25)) must be sufficiently trained to be robust to the emergent dynamics.

Substituting equation (25) into equation (31) we may write:

$$
\dot{z}^{k,emergant} \approx A z^k + \sum_{\ell \neq k} P_\phi \left(A z_\phi^\ell\right) \tag{33}
$$

In matrix form (for components $k = 1$ -K) we may write:

$$\begin{bmatrix} \dot{z}^{1,emergant} \\ \dot{z}^{2,emergant} \\ \vdots \\ \dot{z}^{K,emergant} \end{bmatrix} \approx \begin{bmatrix} A & P_\phi A & \cdots & P_\phi A \\ P_\phi A & A & \cdots & P_\phi A \\ \vdots & \vdots & \ddots & \vdots \\ P_\phi A & P_\phi A & \cdots & A \end{bmatrix} \begin{bmatrix} z^1 \\ z^2 \\ \vdots \\ z^K \end{bmatrix} \tag{34}$$

By definition:

$$\begin{bmatrix} z^{1,emergant} \\ z^{2,emergant} \\ \vdots \\ z^{K,emergant} \end{bmatrix} \approx \begin{bmatrix} I & P_\phi & \cdots & P_\phi \\ P_\phi & I & \cdots & P_\phi \\ \vdots & \vdots & \ddots & \vdots \\ P_\phi & P_\phi & \cdots & I \end{bmatrix} \begin{bmatrix} z^1 \\ z^2 \\ \vdots \\ z^K \end{bmatrix} \tag{35}$$

Substituting equation (35) into equation(34):

$$\begin{bmatrix} \dot{z}^{1,emergant} \\ \dot{z}^{2,emergant} \\ \vdots \\ \dot{z}^{K,emergant} \end{bmatrix} \approx \begin{bmatrix} A & P_\phi A & \cdots & P_\phi A \\ P_\phi A & A & \cdots & P_\phi A \\ \vdots & \vdots & \ddots & \vdots \\ P_\phi A & P_\phi A & \cdots & A \end{bmatrix} \begin{bmatrix} I & P_\phi & \cdots & P_\phi \\ P_\phi & I & \cdots & P_\phi \\ \vdots & \vdots & \ddots & \vdots \\ P_\phi & P_\phi & \cdots & I \end{bmatrix}^{\#} \begin{bmatrix} z^{1,emergant} \\ z^{2,emergant} \\ \vdots \\ z^{K,emergant} \end{bmatrix} \tag{36}$$

#– pseudo inverse

Equation (36) represents the K coupled dynamic equations that can be used to predict the behaviors of the larger (more complex) nonlinear system.

3. Application of Approach to 1-Dimensional Lumped Parameter Cell-Matrix Mode

3.1 Definition of 1-D Cell Motility Model on a 1-D Elastic Substrate

For proof of concept, we begin with a 1-D simple lumped parameter model of interacting cell/ECM behavior. Although quite simple, it is still possible to reproduce bio-mechanical interactive behaviors. The general structure is modified from the 1-D cell motility model given in [33] . The cell is treated as a viscoelastic deformable body [34]interacting with a 1-D deformable ECM[35]. The cell body consists of two masses (which represent the leading edge and trailing edge of the cell) connected through a spring and damper (which represents the viscoelastic nature of the cell). The ECM is made up of springs connected through multiple nodes.

The cell can attach to the ECM at the leading and trailing edge through the multiple nodes present within the ECM. To move forward, the current cell attachments must be broken and the cell can "grab" and attach to the closest ECM nodes. Depending on the node attachments the ECM stiffness parameters of the sections behind the trailing edge, between the leading and trailing edges, and in front of the leading edge will change. These parameters are nonlinear functions that change depending on the position of the ECM node attachments. Consequently the equilibrium position of the leading and trailing edge are adjusted to compensate for the changing ECM stiffness's. Furthermore there is a constant stiffness (G_1, G_2) associated with the ECM node attachments of the leading and trailing edges.

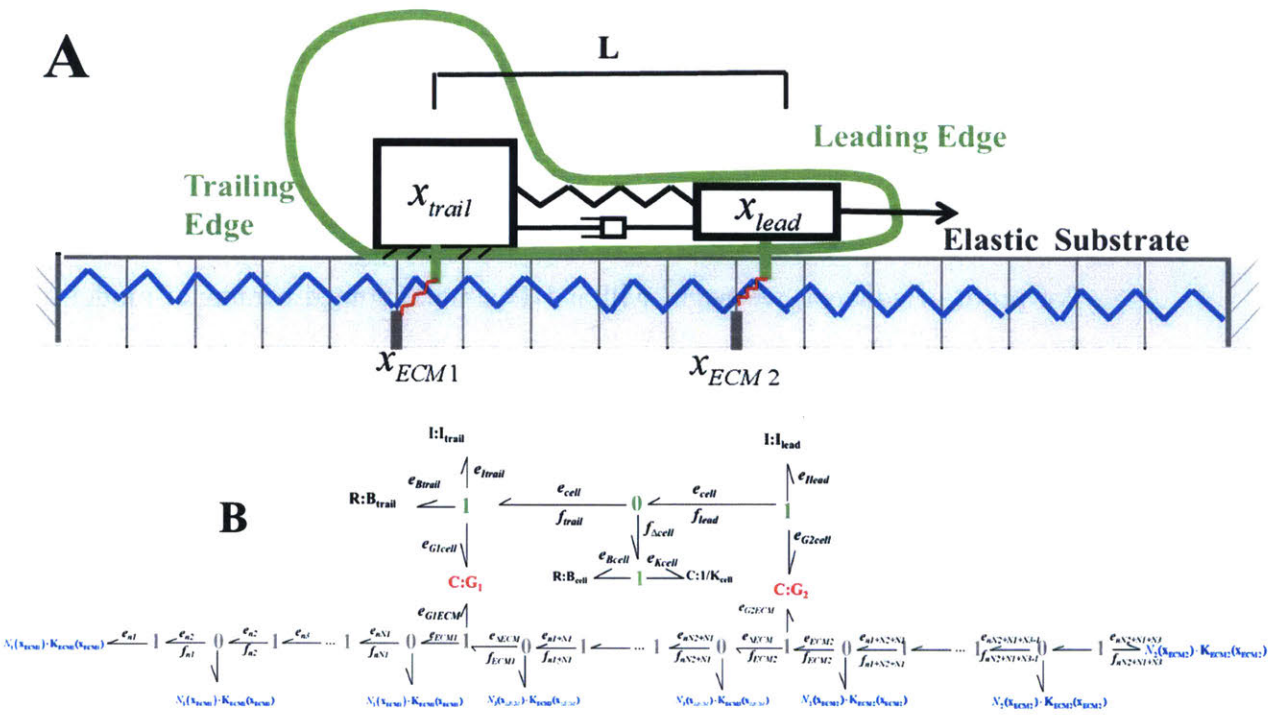


Figure 6: A) overview of the mechanical structure of 1-D lumped parameter cell migration model B) Bond graph corresponding to the 1-D lumped parameter system. Bond graph representing cell dynamics (green junctions) is linked to the bond graph representation

3.2 Bond Graph Representation

Consider the bond graph shown in Fig. 6b which describes the 1-D lumped parameter cell/ECM dynamics. The elements (masses, springs, and dampers) representing the system are connected through lines, termed power bonds. The two variables are associated with each bond, the product of which represents power flowing through the bond. In the mechanical domain, variables represent the force and velocity respectively. The 1's and 0's describe two types of junctions. The former represents the mechanical domain equivalent to Kirchhoff's Voltage Law, while the latter represents the mechanical domain equivalent to Kirchhoff's Current Law.

The bond graph representing cell dynamics (green junctions) is linked to the bond graph representing ECM dynamics (grey junctions) through the elastic elements representing cell to ECM node attachments (red elements). Spring elements connecting adjacent nodes within the ECM (blue elements) are functions of the number of nodes within the sections behind the trailing edge , between the leading and trailing edges , and in front of the leading edge and are also functions of the

corresponding stiffness parameters within each section. The state variables within the system are given by the vector:

$$\begin{aligned}
&\mathbf{x} = \left(x_{lead}, x_{trail}, p_{lead}, p_{trail}, x_{ECM1}, x_{ECM2}, x_{\Delta ECM}, x_{\Delta cell}, x_{n_1}, x_{n_2}, ..., x_{n_{N_{ECM}}} \right)^T \in \mathbb{R}^{n_x \times 1} \\
&x_{\Delta ECM} = x_{ECM1} - x_{ECM2} \\
&x_{\Delta cell} = x_{lead} - x_{trail}
\end{aligned} \tag{37}$$

Where x_{lead}, x_{trail}, x_{ECM1}, x_{ECM2}, $x_{\Delta ECM}$, $x_{\Delta cell}$ are positions corresponding to the leading edge, trailing edge and corresponding ECM node attachments as described in Fig. 1. Variables p_{lead} and p_{trail} represent the momentum of the leading and trailing edge and $x_{n_1}, x_{n_2}, ..., x_{n_{N_{ECM}}}$ (N_{ECM} = total # of nodes within ECM) represent the position of each node within the ECM.

3.3 Deriving State Equations using Bond Graphs

Using the bond graph we may write the dynamic equations of the system as a combination of the state variables $(x's)$ and the effort and flow variables $(e's, f's)$ associated with the (linear and nonlinear) constitutive laws of the elements described within the bond graph.

$$\begin{aligned}
&\frac{dx_{trail}}{dt} = f_{trail} && \frac{dx_{ECM1}}{dt} = f_{ECM1} \\
&\frac{dp_{trail}}{dt} = e_{cell} - e_{\Delta ECM} - e_{ECM1} - e_{Btrail} && \frac{dx_{ECM2}}{dt} = f_{ECM2} \\
&\frac{dx_{lead}}{dt} = f_{lead} && \frac{dx_{\Delta ECM}}{dt} = f_{ECM2} - f_{ECM1} \\
&\frac{dp_{lead}}{dt} - e_{ECM2} + e_{\Delta ECM} - e_{cell} && \frac{dx_{\Delta cell}}{dt} = f_{lead} - f_{trail}
\end{aligned} \tag{38}$$

$$\frac{dx_{n_i}}{dt} = f_{n_i} = \begin{cases} f_{ECM1} \cdot \left(1 - \dfrac{i}{N_1}\right) & for\ i = [1, \ldots, N_1] \\ f_{\Delta ECM} \cdot \left(\dfrac{i - N_1}{N_2}\right) + f_{ECM1} & for\ i = [N_1, \ldots, N_1 + N_2] \\ f_{ECM2} \cdot \left(1 - \dfrac{i - (N_1 + N_2)}{N_3}\right) & for\ i = [N_1 + N_2, \ldots, N_1 + N_2 + N_3] \end{cases}$$

$$N_{ECM} = N_1 + N_2 + N_3$$

Equation (38) describes the state variable dynamics as a linear combination of effort and flow variables. Rewriting the system in matrix form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{W}_{\eta} \cdot \boldsymbol{\eta} \tag{39}$$

Where $\mathbf{W}_{\eta} \in \mathbb{R}^{n_x \times n_\eta}$ is a constant weight matrix and $\boldsymbol{\eta}$ represents the auxiliary variables are defined by:

$$\begin{aligned}\boldsymbol{\eta} = [&f_{trail}, f_{lead}, e_{cell}, e_{Btrail}, f_{ECM1}, f_{ECM2}, \\ &e_{ECM1}, e_{ECM2}, e_{\Delta ECM}, f_{n_1}, f_{n_2}, \ldots, f_{n_{N_{ECM}}}] \in \mathbb{R}^{n_\eta \times 1}\end{aligned} \tag{40}$$

It has been shown that augmenting the state variables with auxiliary variables ($\boldsymbol{\eta}$) derived from the system dynamics is necessary for depicting the nonlinear system dynamics behavior as a whole[26] . The augmented state vector may be defined as:

$$\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\eta} \end{pmatrix} \in \mathbb{R}^{\ell \times 1} \tag{41}$$

The augmented variable vector is termed sufficiently informing in that it in can completely reproduce the nonlinear dynamics of the system. Here we propose that an approximate linear variation can be found within the augmented data.

3.4 Linear Reduced Order Latent Representation using Simulated Data

3.4.1 Modifications from section 2.2

Although the variables described in the augmented vector given by (41) are sufficient to inform the dynamics of isolated single-cell/ECM, additional variables are added for the multi-cell ECM case. This is in order to track the cell's leading and trailing edge location with respect to it's ECM. Furthermore we must ensure that the ECM variables are consistent among the individual models. Consequently, we record the index of the ECM node currently attached to the leading and trailing cell edges (i_{lead}, i_{trail}). Specifying these indices will further create an address for the cell on the ECM. The original ECM state variables x_{ECM1}, x_{ECM2} in (37) (which represent the positions of the attached ECM nodes) are defined as follows:

$$\left.\begin{aligned} x_{ECM1} &= x_{node i_{lead}} \\ f_{ECM1} &= \dot{x}_{node i_{lead}} \\ f_{ECM1} &= F_{node i_{lead}} \end{aligned}\right\} \text{if node } i_{lead} \text{ is attached to leading edge } (i_{lead} \neq i_{trail})$$
$$\left.\begin{aligned} x_{ECM2} &= x_{i_{trail}} \\ f_{ECM2} &= \dot{x}_{i_{trail}} \\ f_{ECM2} &= F_{i_{trail}} \end{aligned}\right\} \text{if node } i_{trail} \text{ is attached to leading edge } (i_{trail} \neq i_{lead}) \tag{42}$$

We also include the forces $e_{n_i}\ (i=1,\ldots,N_{ECM})$ of all the ECM the nodes comprising the ECM. We redefine auxiliary variables to include the additional aforementioned variables.

$$\begin{aligned} \boldsymbol{\eta}_\phi &= \left[f_{n_1}, f_{n_2}, \ldots, f_{n_{N_{ECM}}}, e_{n_1}, e_{n_2}, \ldots, e_{n_{N_{ECM}}} \right]^T \in \mathbb{R}^{n_{\eta_\phi} \times 1} \\ \boldsymbol{\chi}_\phi &= \left[f_{ECM1}, f_{ECM2}, e_{ECM1}, e_{ECM2}, e_{\Delta ECM} \right]^T \in \mathbb{R}^{n_{\chi_\phi} \times 1} \\ \boldsymbol{\eta}_\alpha &= \left[\frac{d(i_{lead})}{dt}, \frac{d(i_{trail})}{dt}, f_{trail}, f_{lead}, e_{cell}, e_{Btrail} \right] \in \mathbb{R}^{n_{\eta_\alpha} \times 1} \end{aligned} \tag{43}$$

Note that $\boldsymbol{\eta}_\alpha$ contains only variables associated with the cell dynamics and $\boldsymbol{\eta}_\phi$ contains only variables associated with the ECM. Following from (42)we may write a relationship between $\boldsymbol{\eta}_\phi$ and $\boldsymbol{\chi}_\phi$:

$$\boldsymbol{\chi}_\phi = \mathbf{M}_{\chi_\phi} \boldsymbol{\eta}_\phi \tag{44}$$

Where $\mathbf{M}_{\chi\phi}$ is a $n_{\chi_\phi} \times n_{\eta_\phi}$ binary membership matrix. If the leading edge of the cell is attached the ECM node the elements at the 1st and 3rd and 5th rows and column are 1. If the trailing edge of the cell is attached the ECM node the elements at the 2nd and 4th, rows and column are 1 and 5ht row are -1. All other elements representing unattached ECM nodes are 0.

Augmenting equation (3) with the additional variables we may write:

$$\underbrace{\frac{d}{dt}\begin{bmatrix} i_{lead} \\ i_{trail} \\ \mathbf{x} \\ e_{n_1} \\ \vdots \\ e_{n_{N_{ECM}}} \end{bmatrix}}_{\frac{d\mathbf{x}^*}{dt}} = \underbrace{\begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{W}_\eta & \mathbf{I}_{n_{N_{ECM}} \times n_{N_{ECM}}} \end{bmatrix}}_{\mathbf{W}_\eta^*} \underbrace{\begin{bmatrix} \boldsymbol{\eta}_\alpha \\ \boldsymbol{\chi}_\phi \\ \boldsymbol{\eta}_\phi \end{bmatrix}}_{\boldsymbol{\eta}^*} \tag{45}$$

The augmented state vector may be re-defined as:

$$\tilde{\mathbf{x}}^{*} = \begin{pmatrix} \mathbf{x}^{*} \\ \boldsymbol{\eta}^{*} \end{pmatrix} \in \mathbb{R}^{\ell^{*}\times 1}$$
$$\ell^{*} = \boldsymbol{n}_{x^{*}} + \boldsymbol{n}_{\eta^{*}} \tag{46}$$
$$\boldsymbol{n}_{x^{*}} = 2 + \boldsymbol{n}_{x} + \boldsymbol{n}_{ECM}; \boldsymbol{n}_{\eta^{*}} = 2 + \boldsymbol{n}_{\eta} + \boldsymbol{n}_{ECM}$$

Since all the variables contained in equation (41) are included in the new set of variables the new set is still sufficient to describe the single-cell/ECM behavior. We conduct simulations of the nonlinear switching model of the single-cell/ECM dynamics described using a diverse set of initial conditions. With these simulations, a large number of aforementioned variables equation (46) are sampled. A data matrix is formed by arranging these samples in the following manner:

$$\mathbf{X}^{\mathrm{T}} = \begin{bmatrix} x^{*}(t_{0,1})^{T} & \eta^{*}(t_{0,1})^{T} \\ x^{*}(t_{1,1})^{T} & \eta^{*}(t_{1,1})^{T} \\ \vdots & \vdots \\ x^{*}(t_{f,1})^{T} & \eta^{*}(t_{f,1})^{T} \\ \vdots & \vdots \\ \vdots & \vdots \\ x^{*}(t_{0,N})^{T} & \eta^{*}(t_{0,N})^{T} \\ x^{*}(t_{1,N})^{T} & \eta^{*}(t_{1,N})^{T} \\ \vdots & \vdots \\ x^{*}(t_{f,N})^{T} & \eta^{*}(t_{f,N})^{T} \end{bmatrix} \in \mathbb{R}^{N\cdot t_{f}\times \ell^{*}} \tag{47}$$

We use the covariance of the data matrix in (47) to transform the variables specified in (46) into latent space as previously outlined in section 2.2.2. and 2.2.3 to obtain equation.

3.4.2 Model Evaluation and Analysis

As can be seen in figure 7, there is good agreement between the trajectories of the real cell compared to the reduced order latent variable model. The total mean squared error $MSE = .02$.

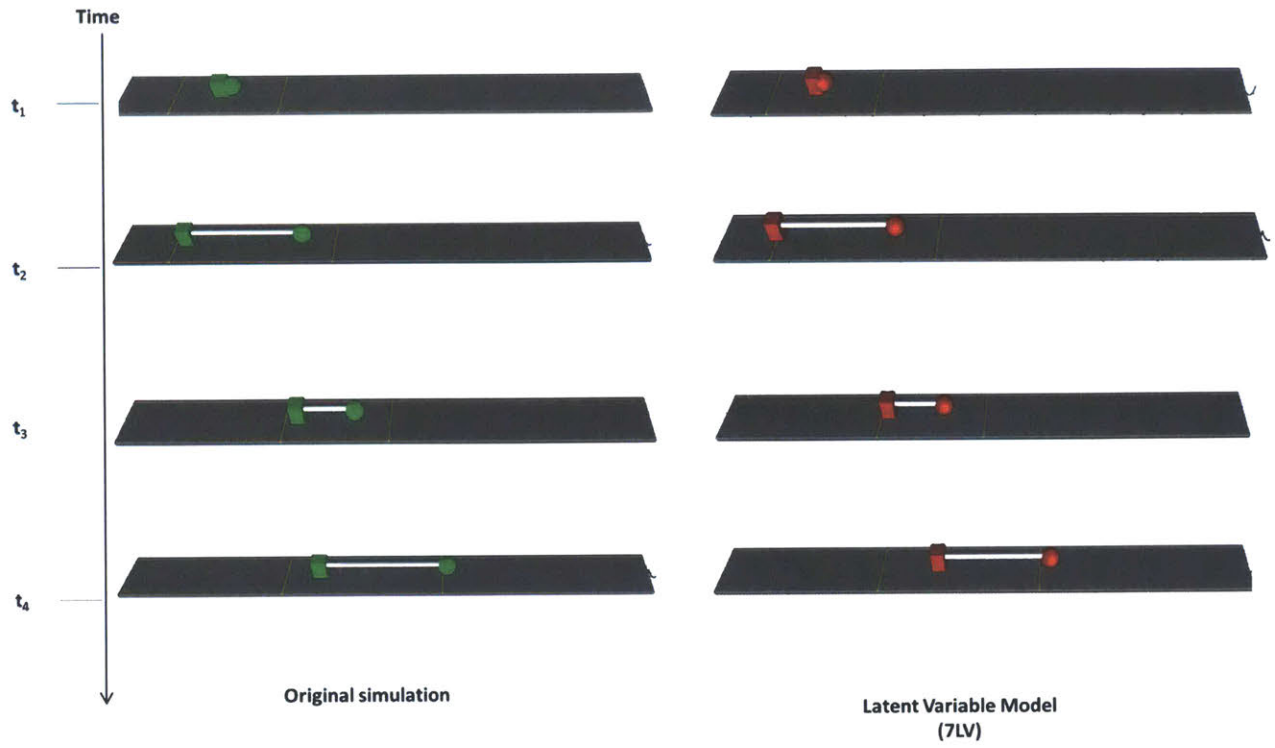


Figure 7: Trajectories of the real cell compared to the reduced order latent variable model

3.5 Linking Individual Reduced Order Models through the Dynamics of Shared Variables

3.5.1Modifications from section 2.3

First, we isolate variables in latent space:

$$\dot{\mathbf{z}}(t)=\mathbf{V}^{\mathrm{T}}\begin{bmatrix}\dot{\mathbf{x}}^{*}\\ \dot{\boldsymbol{\eta}}^{*}\end{bmatrix}=\mathbf{V}^{\mathrm{T}}\begin{bmatrix}\mathbf{W}_{\boldsymbol{\eta}}{}^{*}\boldsymbol{\eta}^{*}\\ \dot{\boldsymbol{\eta}}^{*}\end{bmatrix}=\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\begin{bmatrix}\boldsymbol{\eta}_{\alpha}\\ \boldsymbol{\chi}_{\phi}\\ \boldsymbol{\eta}_{\phi}\\ \dot{\boldsymbol{\eta}}_{\alpha}\\ \dot{\boldsymbol{\chi}}_{\phi}\\ \dot{\boldsymbol{\eta}}_{\phi}\end{bmatrix}$$

$$=\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\begin{bmatrix}\boldsymbol{\eta}_{\alpha}\\ 0\\ 0\\ \dot{\boldsymbol{\eta}}_{\alpha}\\ 0\\ 0\end{bmatrix}+\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\begin{bmatrix}0\\ \boldsymbol{\chi}_{\phi}\\ 0\\ 0\\ \dot{\boldsymbol{\chi}}_{\phi}\\ 0\end{bmatrix}+\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\begin{bmatrix}0\\ 0\\ \boldsymbol{\eta}_{\phi}\\ 0\\ 0\\ \dot{\boldsymbol{\eta}}_{\phi}\end{bmatrix} \tag{48}$$

$$\overline{\overline{\mathbf{W}}}=\begin{bmatrix}\mathbf{W}_{\boldsymbol{\eta}}^{*} & \mathbf{0}_{n_{x^*}\times n_{\eta^*}}\\ \mathbf{0}_{n_{\eta^*}\times n_{\eta^*}} & \mathbf{I}_{n_{\eta^*}\times n_{\eta^*}}\end{bmatrix}\in\mathbb{R}^{\left(n_{x^*}+n_{\eta^*}\right)\times 2n_{\eta^*}}$$

Further simplification gives:

$$\dot{\mathbf{z}}(t)=\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\mathbf{M}_{\alpha}\begin{bmatrix}\boldsymbol{\eta}_{\alpha}\\ \boldsymbol{\chi}_{\phi}\\ \boldsymbol{\eta}_{\phi}\\ \dot{\boldsymbol{\eta}}_{\alpha}\\ \dot{\boldsymbol{\chi}}_{\phi}\\ \dot{\boldsymbol{\eta}}_{\phi}\end{bmatrix}+\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\begin{bmatrix}0\\ \mathbf{M}_{\chi\phi}\boldsymbol{\eta}_{\phi}\\ 0\\ 0\\ \mathbf{M}_{\chi\phi}\dot{\boldsymbol{\eta}}_{\phi}\\ 0\end{bmatrix}+\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\mathbf{C}_{\chi_{\phi}}\begin{bmatrix}0\\ 0\\ \boldsymbol{\eta}_{\phi}\\ 0\\ 0\\ \dot{\boldsymbol{\eta}}_{\phi}\end{bmatrix} \tag{49}$$

Where we substitute equation(44) and $\mathbf{M}_{\alpha}$ is defined as:

$$\mathbf{M}_{\alpha}=\begin{bmatrix}
\mathbf{I}_{n_{\eta\alpha}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\eta\phi}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\eta\phi}} \\
\mathbf{0}_{n_{\chi\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\chi\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\chi\phi}\times n_{\eta\phi}} & \mathbf{0}_{n_{\chi\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\chi\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\chi\phi}\times n_{\eta\phi}} \\
\mathbf{0}_{n_{\eta\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\phi}} \\
\mathbf{0}_{n_{\eta\alpha}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\eta\phi}} & \mathbf{I}_{n_{\eta\alpha}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\eta\phi}} \\
\mathbf{0}_{n_{\chi\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\chi\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\chi\phi}\times n_{\eta\phi}} & \mathbf{0}_{n_{\chi\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\chi\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\chi\phi}\times n_{\eta\phi}} \\
\mathbf{0}_{n_{\eta\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\phi}}
\end{bmatrix}\in\mathbb{R}^{2\boldsymbol{\eta}^{*}\times 2\boldsymbol{\eta}^{*}} \qquad (50)$$

Furthermore $\mathbf{C}\in\mathbb{R}^{\ell^{*}\times 2n_{\eta^{*}}}$ is defined as:

$$\mathbf{C}=\begin{bmatrix}
\mathbf{0}_{n_{\eta\alpha}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\eta\phi}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\eta\phi}} \\
\mathbf{0}_{n_{\chi\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\chi\phi}\times n_{\chi\phi}} & \mathbf{M}_{\chi\phi} & \mathbf{0}_{n_{\chi\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\chi\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\chi\phi}\times n_{\eta\phi}} \\
\mathbf{0}_{n_{\eta\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\phi}} \\
\mathbf{0}_{n_{\eta\alpha}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\eta\phi}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\alpha}\times n_{\eta\phi}} \\
\mathbf{0}_{n_{\chi\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\chi\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\chi\phi}\times n_{\eta\phi}} & \mathbf{0}_{n_{\chi\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\chi\phi}\times n_{\chi\phi}} & \mathbf{M}_{\chi\phi} \\
\mathbf{0}_{n_{\eta\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\alpha}} & \mathbf{0}_{n_{\eta\phi}\times n_{\chi\phi}} & \mathbf{0}_{n_{\eta\phi}\times n_{\eta\phi}}
\end{bmatrix}\in\mathbb{R}^{2\boldsymbol{\eta}^{*}\times 2\boldsymbol{\eta}^{*}} \qquad (51)$$

Let us define the variables related to the ECM, cell and interface as:

$$\dot{\mathbf{z}}_{\phi}=\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\begin{bmatrix}0\\0\\\boldsymbol{\eta}_{\phi}\\0\\0\\\dot{\boldsymbol{\eta}}_{\phi}\end{bmatrix} \qquad (52)$$

$$\dot{\mathbf{z}}_{\alpha} = \mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\begin{bmatrix}\boldsymbol{\eta}_{\alpha}\\0\\0\\\dot{\boldsymbol{\eta}}_{\alpha}\\0\\0\end{bmatrix} = \mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\mathbf{M}_{\alpha}\begin{bmatrix}\boldsymbol{\eta}_{\alpha}\\\chi_{\phi}\\\boldsymbol{\eta}_{\phi}\\\dot{\boldsymbol{\eta}}_{\alpha}\\\dot{\chi}_{\phi}\\\dot{\boldsymbol{\eta}}_{\phi}\end{bmatrix} = \mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\mathbf{M}_{\alpha}\underbrace{\begin{bmatrix}\left(\mathbf{W}_{\eta}^{\ *}\right)^{\#}\mathbf{V}_{\mathbf{x}^*}\\\mathbf{V}_{\boldsymbol{\eta}^*}\end{bmatrix}}_{\mathbf{V}^*}\dot{\mathbf{z}} \tag{53}$$

$$\dot{\mathbf{z}}_{\chi_{\phi}} = \mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\begin{bmatrix}0\\\chi_{\phi}\\0\\0\\\dot{\chi}_{\phi}\\0\end{bmatrix} = \mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\mathbf{C}\begin{bmatrix}0\\0\\\boldsymbol{\eta}_{\phi}\\0\\0\\\dot{\boldsymbol{\eta}}_{\phi}\end{bmatrix} = \mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\mathbf{C}\left(\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\right)^{\#}\dot{\mathbf{z}}_{\phi} \tag{54}$$

Then substituting equation(49):

$$\dot{\mathbf{z}}(t) = \left(\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\mathbf{M}_{\alpha}\mathbf{V}^{*}\right)\dot{\mathbf{z}} + \mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\mathbf{C}\left(\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\right)^{\#}\dot{\mathbf{z}}_{\phi} + \dot{\mathbf{z}}_{\phi} \tag{55}$$

Finally we may write:

$$\dot{\mathbf{z}}(t) = \left(\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\mathbf{M}_{\alpha}\mathbf{V}\right)\mathbf{A}\mathbf{z}(t) + \left(\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\mathbf{C}\left(\mathbf{V}^{\mathrm{T}}\overline{\overline{\mathbf{W}}}\right)^{\#} + \mathbf{I}\right)\dot{\mathbf{z}}_{\phi} \tag{56}$$

Equation (56) provides the state equations where the latent variables associated with ECM dynamics as an additive term.

Using the modified framework in equation we propose to replace the ECM dynamics of the individual isolated cells ($\dot{\mathbf{z}}_{\phi}(t)$) with an emergent state variable ($\dot{\mathbf{z}}_{\phi}^{emergant}(t)$) that represents the multi-cell environment. In order to this, a second key assumption is necessary. Mainly, it must possible to estimate the multi-cell ECM environment and corresponding ECM states by superposing ECM dynamics of multiple isolated single-cell ECM models. Although, the estimation may not be exact, it will be valid if it can reproduce the specified emergent behavior which as previously mentioned is the cells moving forward towards each other.

The ECM variables of individual single agent models are added to create the combined ECM environment influenced by multiple cells. The state equation given by equation specified for each individual (j) agent is:

$$\dot{\mathbf{z}}^{emergant} = \left(\mathbf{V}^{\mathbf{T}}\overline{\overline{\mathbf{W}}}\mathbf{M}_{\alpha}\mathbf{V}^{*}\right)\mathbf{A}\mathbf{z}(t) + \left(\mathbf{V}^{\mathbf{T}}\overline{\overline{\mathbf{W}}}\mathbf{C}\left(\mathbf{V}^{\mathbf{T}}\overline{\overline{\mathbf{W}}}\right)^{\#} + \mathbf{I}\right)\dot{\mathbf{z}}_{\phi}^{emergant} \tag{57}$$

The individual ECM are supposed to create an estimate of the combined ECM environment influenced by multiple cells:

$$\hat{\dot{\mathbf{z}}}_{\phi}^{emergent} = \sum_{j=1}^{2}\dot{\mathbf{z}}_{\phi,j} \tag{58}$$

Where $\dot{\mathbf{z}}_{\phi,j} = \mathbf{P}_{\phi}\cdot\dot{\mathbf{z}}_{j}$

$$\mathbf{P}_{\phi} = \mathbf{V}^{\mathbf{T}}\overline{\overline{\mathbf{W}}}\mathbf{M}_{\phi}\mathbf{V}^{*} \tag{59}$$

3.5.2 Model Evaluation and Analysis

The position trajectories of the original nonlinear simulation (green) and the linear superposition estimation (red) are shown in figure 8. As can be seen, approximation of the ECM using linear superposition leads to the production of the correct emergent behavior (i.e. the two cells move towards each other). These results suggest that it is possible to reproduce the general required motion to indicate emergent behavior.

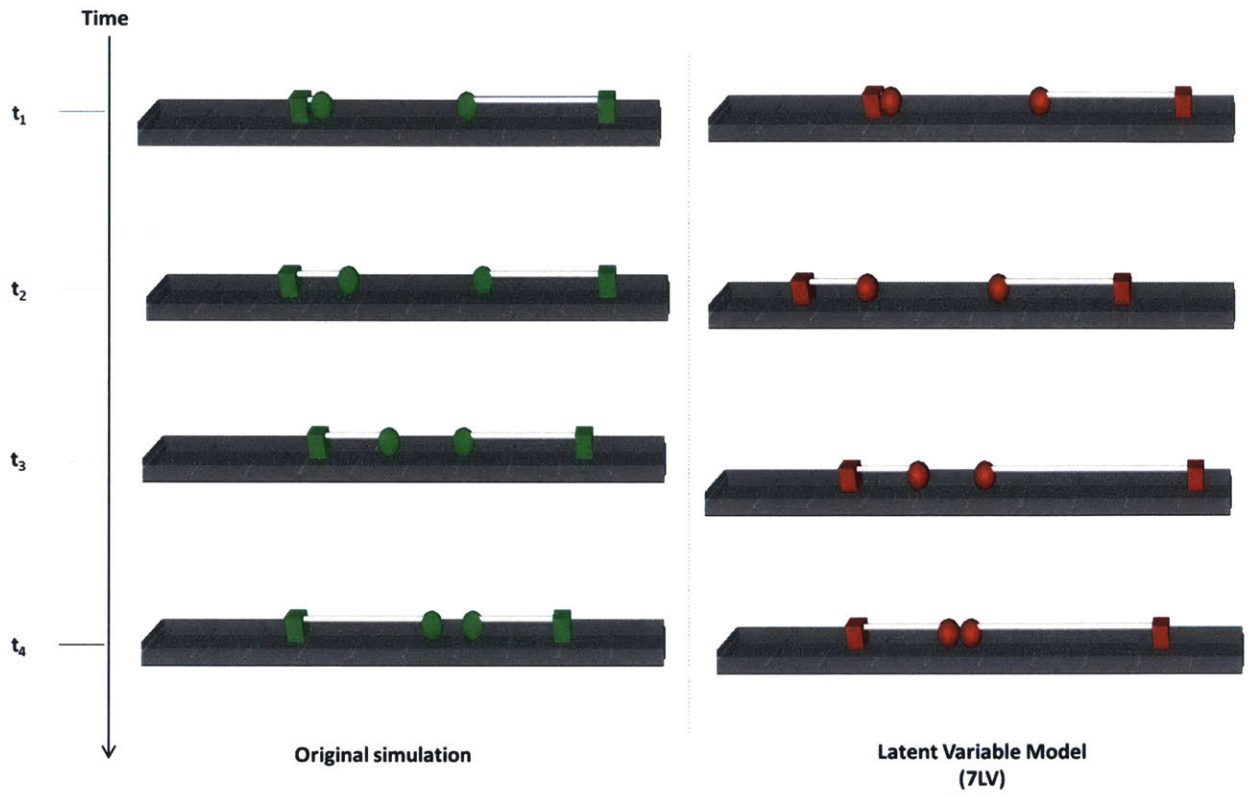


Figure 8: Python simulation of position trajectories of the original nonlinear simulation (green) and the linear state equation where variables are estimated using superposition (red).

4. Application of Approach to 3-Dimensional Distributed Parameter Cell-Matrix Model

4.1 Definition of Nonlinear Distributed Parameter Model of 3-D Cell Spreading on a 2.5D Elastic Substrate

Consider the 3-D distributed parameter cell model shown in figure 9. The full biophysical model simulates cell migration and spreading on an elastic substrate and incorporates focal adhesion dynamics, cytoskeleton and nucleus remodeling, actin motor activity, and lamellipodia protrusion[11], [13]. The inner and outer membrane of the cell is divided into a triangular mesh consisting of 200 nodes. The ECM is a complex fibrous network that is divided into2000 nodes. It measures 2um thick and 60um in diameter. The ECM fibers are modeled after Collagen Type I. The bulk stiffness of the substrate is estimated to be 5kPa.

The cell membrane deforms and gains traction as the nodes distributed over the outer membrane bond to nodes on the ECM surface, and form focal adhesions. The ECM deforms due to the cell and can also influence the cell dynamics because the cell's anterior-posterior polarization

(or cell's polarity) is changed based on the direction of local maximum stiffness in the ECM[36]. Consequently, the cell continuously updates its lamellipodial protrusions in order to reflect the cell's polarity change to local maximum stiffness direction.

The dynamical system to be linearized is a set of nonlinear differential equations that describe the deformation of the cell outer membrane and ECM substrate.

4.2 Bond Graph Representation

In order to create a bond graph describing the deformations of the outer cell membrane we must consider the 3-D position of each node on the outer membrane and each node within the ECM substrate:

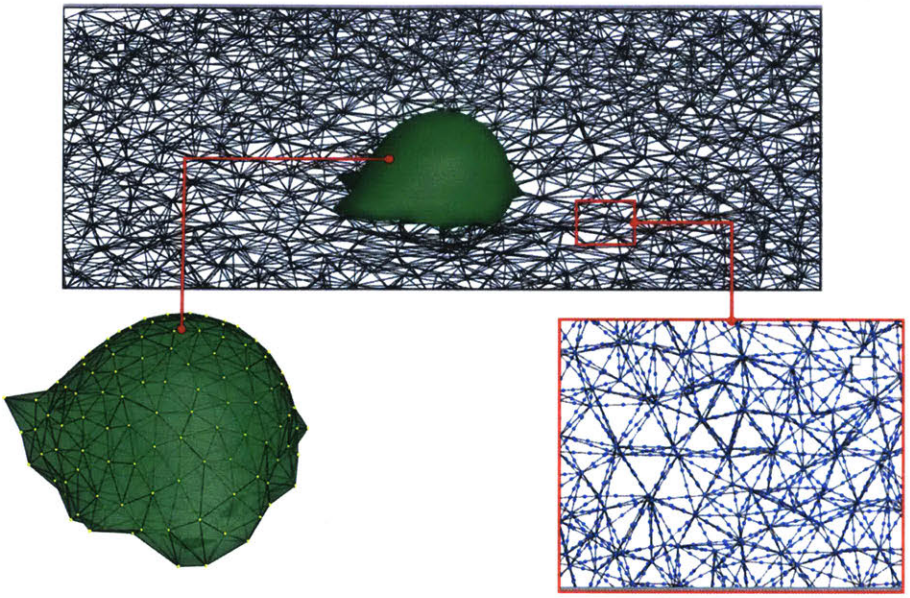


Figure 9: Nonlinear Distributed Parameter Model of 3-D Cell Spreading on a 2.5D Elastic Substrate. The membrane of the cell is divided into a triangular mesh consisting of 200 nodes. The ECM is a complex

$$\begin{aligned} &\left(x^c = \left(x_1^c, \ldots, x_{200}^c\right)^T \in \mathbb{R}^{3\cdot(200)\times 1}\right) \\ &x_i^c = \left(\mathrm{x}_i^e, \mathrm{y}_i^e, \mathrm{z}_i^e\right) \\ &\left(x^e = \left(x_1^e, \ldots, x_{2000}^e\right)^T \in \mathbb{R}^{3\cdot(200)\times 1}\right) \\ &x_i^e = \left[\mathrm{x}_i^e, \mathrm{y}_i^e, \mathrm{z}_i^e\right] \end{aligned} \tag{60}$$

Furthermore, we must identify the modeling elements (e.g. masses, springs, and dampers) representing energy storage and dissipation in the system and their connectivity. Let us examine the forces interacting on a membrane node and ECM node shown in figure 10.

Each membrane node is acted upon by five forces. It should be noted that the inertia of the node is not considered. Therefore the node is a point mass. These forces are summarized in Table 2:

Force	Equation	Modeling Element
Frictional Dissipative Force	$F_D^c = \Phi_{R_1}\left(\frac{dx^c}{dt}\right)$	Linear Damper
Cortical Tension Force	$F_T^{c,i} = \Phi_T\left(x^c\right)$	Nonlinear Spring
Elastic Energy Force	$F_E^c = \Phi_E\left(x^c\right)$	Nonlinear Spring
Focal Adhesion Force	$F_{FA}^c = \Phi_{FAc}\left(x^c, x^e\right)$	Nonlinear Spring
Lamellipodium Force	$F_L^c = \begin{cases} \text{constant} \\ 0 \end{cases}$	Source

Table 2: Summary of forces acting at a single (massless) membrane node

Each ECM is acted upon by 3 forces. The constitutive equations for these forces are summarized in Table 3:

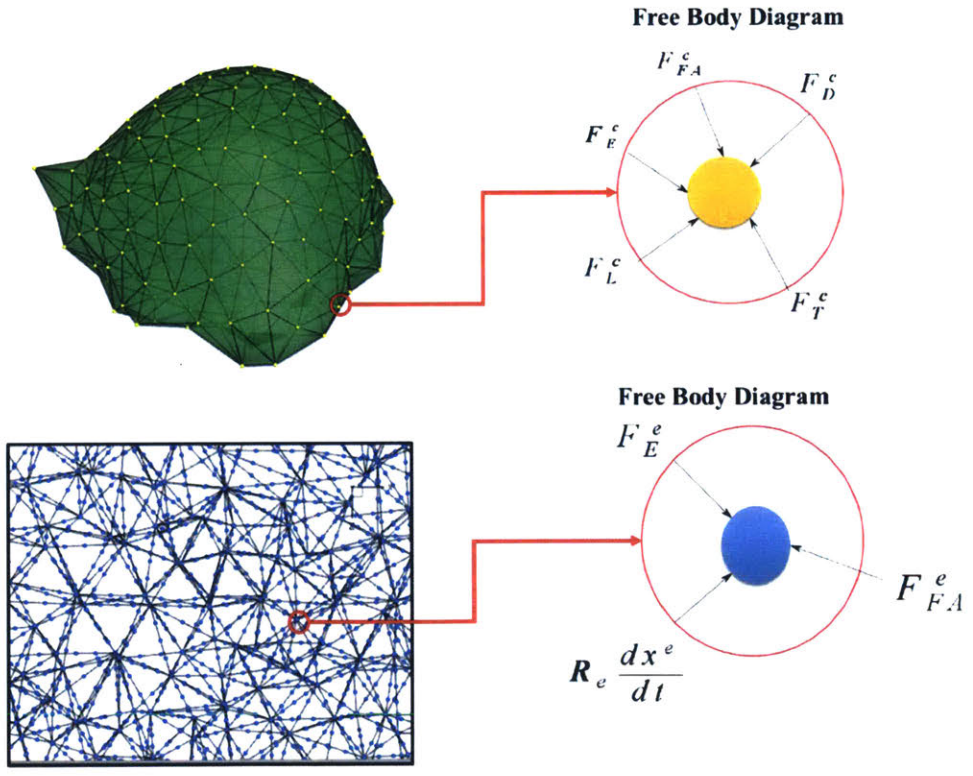


Figure 10: Forces interacting on a single membrane node and ECM node single node

Force	Equation	Modeling Element
Elastic Energy Force	$F_E^e = \Phi_E\left(x^e\right)$	Nonlinear Spring
Focal Adhesion Force	$F_{FA}^e = \Phi_{FAe}\left(\boldsymbol{x}^c, \boldsymbol{x}^e\right)$	Nonlinear Spring
Frictional Dissipative Force	$F_D^e = \Phi_{R_2}\left(\frac{dx^e}{dt}\right)$	Linear Damping

Table 3: Summary of forces acting at a single (massless) ECM node

Membrane nodes can bond to the ECM nodes through focal adhesions. When the focal adhesion forces of the bonded membrane ECM node pair are equal and opposite. Therefore the *ith* membrane node bonded to the *jth* ECM node:

$$F_{FA,i}^c = -F_{FA,j}^e \text{ (when bonded)} \tag{61}$$

However, if the bond between the *ith* membrane node bonded to the *jth* ECM node is ruptured the focal adhesion force automatically switches to zero:

$$\begin{aligned} F_{FA,i}^c &= 0 \\ F_{FA,j}^e &= 0 \end{aligned} \text{ (when ruptured)} \tag{62}$$

This adds another source of nonlinearity to the system.

The forces described in the above tables are 3-Dimensional and involve nonlinear kinematics to represent each spring modeling element. An accurate bond graph describing the node requires the use of and modified bond graph representation used for multiple dimensions[37].

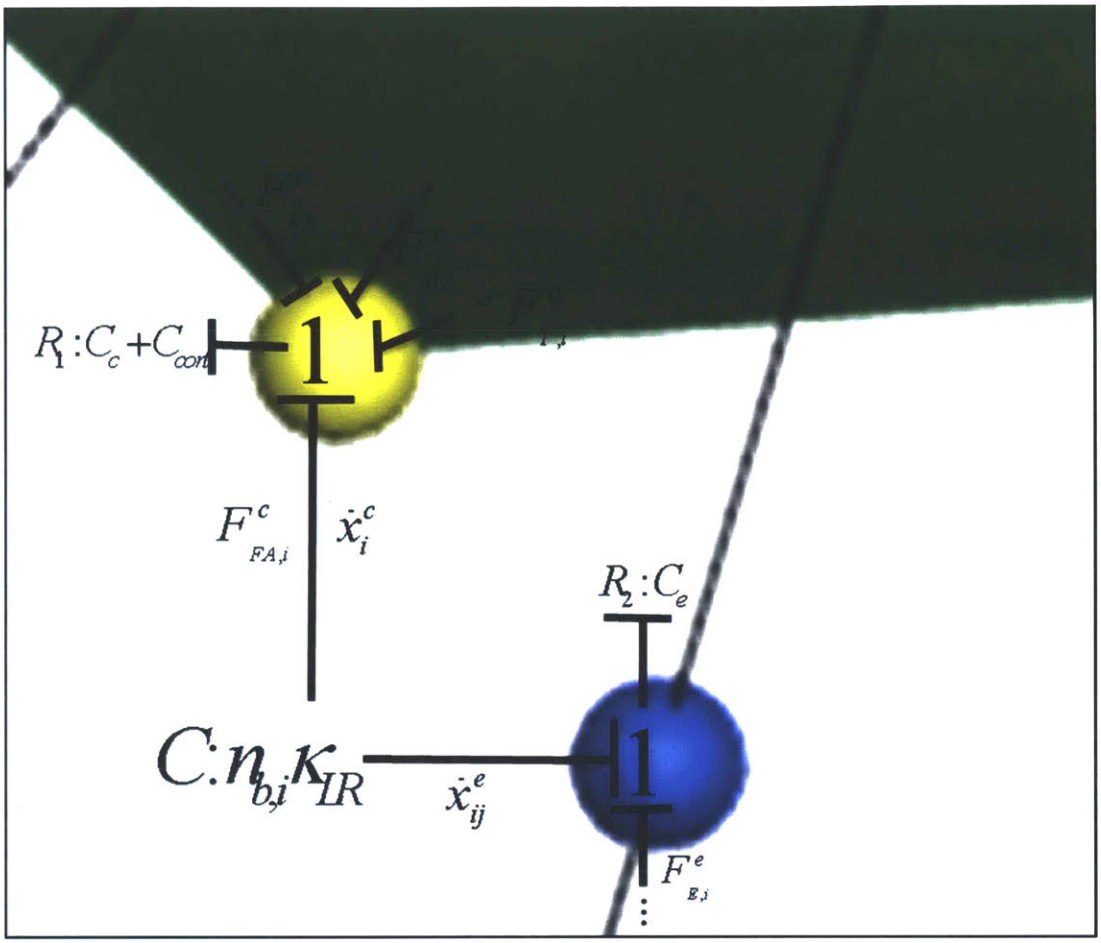


Figure 11: Simplified Conceptual Bond Graph of Cell ECM Interface

However for simplicity let us consider the simplified conceptual Bond Graph structure in figure 11 which contains the bonds and causality necessary to derive the governing equations.

Here, $R_1 = C_c + C_{cort}$ and $R_2 = C_e$ represent the linear damping elements for the membrane node and ECM node respectively. The focal adhesion bond is shown as with a multiport capacitive element $C = n_{b,i} \cdot \kappa_{LR}$ (termed C-Field). C-fields are multi-port generalizations of the basic scalar elements, and can be used in order to model complex multi-dimensional systems[28], [38], [39].

4.3 Deriving Equations from Bond Graphs

From above the bond graph may list the governing equations:

1 *junction equations*

for $i = 1,...,200$

$$(C_c + C_{cort})\frac{\mathrm{d}x_{\mathrm{i}}^{c}}{\mathrm{dt}} + \mathrm{F}_{FA}^{c} + \mathrm{F}_{E,i}^{c} + \mathrm{F}_{L,i}^{c} + \mathrm{F}_{T,i}^{c} = 0$$

for $i = 1,...,2000$

$$C_{\mathrm{e}}\frac{\mathrm{d}x_{\mathrm{i}}^{\mathrm{e}}}{\mathrm{dt}} + \mathrm{F}_{\mathrm{FA,i}}^{\mathrm{e}} + \mathrm{F}_{\mathrm{E,i}}^{\mathrm{e}} = 0$$

constitutive equations from nonlinear elements

$$\begin{aligned}
&\mathrm{F}_{FA}^{c} = \Phi_{FAc}\left(x^{c}, x^{e}\right)\\
&\mathrm{F}_{E}^{c} = \Phi_{E}\left(x^{c}\right)\\
&\mathrm{F}_{L}^{c} = \begin{cases}\textbf{constant}\\ 0\end{cases}\\
&\mathrm{F}_{\mathrm{T}}^{c,i} = \Phi_{T}\left(x^{c}\right)\\
&\mathrm{F}_{FA}^{c} = \Phi_{FAe}\left(x^{c}, x^{e}\right)\\
&\mathrm{F}_{E}^{e} = \Phi_{E}\left(x^{e}\right)\\
&\mathrm{F}_{FA}^{e} = \Phi_{FAe}\left(x^{c}, x^{e}\right)
\end{aligned} \tag{63}$$

From causality, the damping linear damping forces are dependent on the other forces: Therefore we may write for the *ith* membrane node and *ith* ECM node:

$$\begin{aligned}
\frac{\mathrm{d}x_{\mathrm{i}}^{c}}{\mathrm{dt}} &= \frac{1}{(C_c + C_{cort})}\left(\mathrm{F}_{FA}^{c} + \mathrm{F}_{E,i}^{c} + \mathrm{F}_{L,i}^{c} + \mathrm{F}_{T,i}^{c}\right)\\
\frac{\mathrm{d}x_{\mathrm{i}}^{\mathrm{e}}}{\mathrm{dt}} &= \frac{1}{C_{\mathrm{e}}}\left(\mathrm{F}_{\mathrm{FA,i}}^{\mathrm{e}} + \mathrm{F}_{\mathrm{E,i}}^{\mathrm{e}}\right)
\end{aligned} \tag{64}$$

The state equation describing the 3-Dimensional position of the n_{mem} = 200 and the n_{ECM} = 2000 nodes are given by:

$$\underbrace{\begin{bmatrix} \frac{dx_1^c}{dt} \\ \vdots \\ \frac{dx_{n_{mem}}^c}{dt} \\ \frac{dx_1^e}{dt} \\ \vdots \\ \frac{dx_{n_{ECM}}^e}{dt} \end{bmatrix}}_{\frac{dx}{dt}} = \underbrace{\begin{bmatrix} \frac{1}{(C_c+C_{cort})} & & & \frac{1}{(C_c+C_{cort})} & & & \frac{1}{(C_c+C_{cort})} & & & \frac{1}{C_e} & & \frac{1}{C_e} & \\ & \ddots & & & \ddots & & & \ddots & & & \ddots & & \ddots \\ & & \frac{1}{(C_c+C_{cort})} & & & \frac{1}{(C_c+C_{cort})} & & & \frac{1}{(C_c+C_{cort})} & & \frac{1}{C_e} & & \frac{1}{C_e} \end{bmatrix}}_{W_\eta} \underbrace{\begin{bmatrix} F_{FA,1}^c \\ \vdots \\ F_{FA,n_{mem}}^c \\ F_{E,1}^c \\ \vdots \\ F_{E,n_{mem}}^c \\ F_{T,1}^c \\ \vdots \\ F_{T,n_{mem}}^c \\ F_{L,i}^c \\ \vdots \\ F_{L,n_{mem}}^c \\ \vdots \\ F_{FA,n_{ECM}}^e \\ F_{E,1}^e \\ \vdots \\ F_{E,n_{ECM}}^e \end{bmatrix}}_{\eta} \tag{65}$$

Equation (65) reveals a linear representation of the system with respect auxiliary variable vector r_i. Furthermore we augment the system by explicitly writing the dynamics of $r_i$as previously explained in section 2.1.2. For clarity we re-write equation(6):

$$\begin{aligned} \frac{dx}{dt} &= \mathbf{W}_\eta \cdot \boldsymbol{\eta} \in \mathbb{R}^{n\times 1} \quad \left(n = 3\cdot n_{mem} + 3\cdot n_{ECM}\right) \\ \frac{d\boldsymbol{\eta}}{dt} &= \mathrm{H}\left(x, \boldsymbol{\eta}\right) \in \mathbb{R}^{n_\eta \times 1} \end{aligned} \tag{66}$$

4.4 Incorporation of Cell Polarity

As previously discussed in section 4.1, the cell continuously updates its lamellipodial protrusions in order to reflect its polarity change to local maximum stiffness direction. As shown in figure 12, depending on the polarity direction (d_{pol}) the cell membrane is split into a leading and trailing edge. The lamellipodial protrusion force$\left(F_L^c\right)$ of all membrane nodes at the trailing edge is set to zero. Depending the distance of the membrane node from the elastic substrate, the lamellipodial protrusion force at the leading edge can be nonzero.

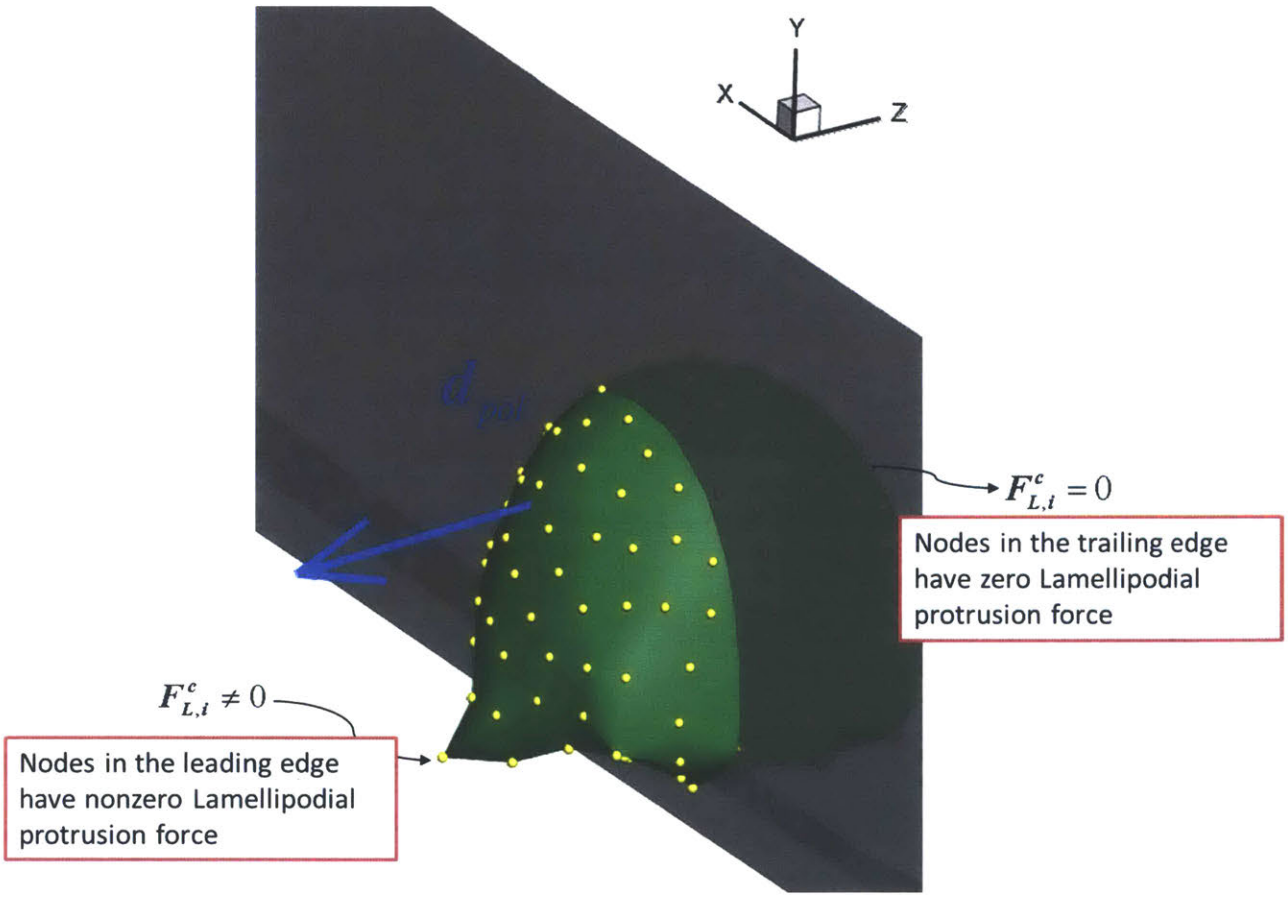


Figure 12: Polarization direction determines the leading edge of the cell and lamellipodial protrusion force

We wish to represent the dependency of lamellipodial protrusion force on the cell polarity direction within the state equations given in equation(65). First we re-write equation (65) to isolate the lamellipodial protrusion forces at each node:

$$\underbrace{\begin{bmatrix} \frac{dx_1^c}{dt} \\ \vdots \\ \frac{dx_{n_{cm}}^c}{dt} \\ \frac{dx_1^e}{dt} \\ \vdots \\ \frac{dx_{n_{ECM}}^e}{dt} \end{bmatrix}}_{\frac{d\mathbf{x}}{dt}} = \underbrace{\begin{bmatrix} \frac{1}{(C_c+C_{cort})} & & & & & & & & & & & \\ & \ddots & & & & & & & & & & \\ & & \frac{1}{(C_c+C_{cort})} & & & & & & & & & \\ & & & \frac{1}{(C_c+C_{cort})} & & & & & & & & \\ & & & & \ddots & & & & & & & \\ & & & & & \frac{1}{(C_c+C_{cort})} & & & & & & \\ & & & & & & \frac{1}{(C_c+C_{cort})} & & & & & \\ & & & & & & & \ddots & & & & \\ & & & & & & & & \frac{1}{C_e} & & & \\ & & & & & & & & & \ddots & & \\ & & & & & & & & & & \frac{1}{C_e} & \\ & & & & & & & & & & & \ddots \\ & & & & & & & & & & & \frac{1}{C_e} \end{bmatrix}}_{W_0} \underbrace{\begin{bmatrix} F_{FA,1}^c \\ \vdots \\ F_{FA,n_{cm}}^c \\ F_{E,1}^c \\ \vdots \\ F_{E,n_{cm}}^c \\ F_{T,1}^c \\ \vdots \\ F_{T,n_{cm}}^c \\ 0 \\ \vdots \\ 0 \\ F_{FA,1}^e \\ \vdots \\ F_{FA,n_{ECM}}^e \\ F_{E,1}^e \\ \vdots \\ F_{E,n_{ECM}}^e \end{bmatrix}}_{\eta} + \underbrace{\frac{1}{(C_c+C_{cort})} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ F_{L,i}^c \\ \vdots \\ F_{L,n_{cm}}^c \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{u} \tag{67}$$

Next, we define $\mathbf{u}$ as:

$$\mathbf{u} = \frac{1}{(C_c + C_{cort})} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathrm{F}_{L,i}^c \\ \vdots \\ \mathrm{F}_{L,n}^c \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{n\times 1} \Rightarrow F_{L,i}^c = \begin{cases} 0 & \textit{trailing edge} \\ F_L & \textit{leading edge} \end{cases} \tag{68}$$

In order to relate the lamellopodial protrusion force at each node to the global polarity direction vector $\left(d_{pol} \in \mathbb{R}^{3\times 1}\right)$, we have created a membership function matrix $\mathbf{M}_{pol} \in \mathbb{R}^{n_\eta \times 1}$.

$$\mathbf{M}_{pol} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ \vdots & \cdots \; \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n_\eta \times 360} \tag{69}$$

The *jith* column of $\mathbf{M}_{pol}$ corresponds the j degree polarity angle $\theta_{pol} = \arctan\left(\frac{\left|d_{pol0} \times d_{pol}\right|}{d_{pol0} \cdot d_{pol}}\right)$ (rounded to the nearest integer degree). Therefore, there are 360 columns in the matrix. Each element in the *jth* column of the $\mathbf{M}_{pol}$ takes a 1 or 0 value depending on whether $F_{L,i}^{c}$ has a zero or nonzero value. We re-write $\mathbf{u}$ as:

$$\mathbf{u}\left(d_{pol}\right) = \frac{1}{\left(C_c + C_{cort}\right)} \mathbf{M}_{pol}\left(:, \arctan\left(\frac{\left|d_{pol0} \times d_{pol}\right|}{d_{pol0} \cdot d_{pol}}\right)\right) \cdot F_L \tag{70}$$

$$F_L = const$$

If the polarity direction is pre-determined we may write:

$$\begin{aligned} \frac{dx}{dt} &= \mathbf{W}_\eta^{*} \cdot \boldsymbol{\eta}^{*} + \mathbf{u}\left(d_{pol}\right) \\ \frac{d\boldsymbol{\eta}^{*}}{dt} &= \mathrm{H}\left(x, \boldsymbol{\eta}^{*}\right) \end{aligned} \tag{71}$$

Equation (71) describes open-loop augmented state equations since the polarity is shown as an independent input to the system. However as previously mentioned a cell's polarity is changed based on the direction of local maximum stiffness in the ECM. In order to mathematically represent this relationship, we use an equation modified from [36]:

$$\frac{d\left(d_{pol}\right)}{dt} = \kappa\left(d_{pol} \times \left(d^{\max}{}_{stiff,ECM} \times d_{pol}\right)\right) \tag{72}$$

Here $d^{\max}{}_{stiff,ECM}$ is a 3 dimensional vector representing the direction of local maximum stiffness in the ECM and a nonlinear function of the auxiliary and state variables of the system:

$$\boldsymbol{d}^{\max}{}_{stiff,ECM} = \Phi_d\left(x,\eta\right) \in \mathbb{R}^{3\times 1} \tag{73}$$

In equation (73) we do not specify the relationship explicitly but only acknowledge that it will be some nonlinear function of the original state and the auxiliary variables. Therefore we may write the closed loop augmented state equations as:

$$\begin{aligned}
\frac{dx}{dt} &= \mathbf{W}_{\eta}^{*} \cdot \boldsymbol{\eta}^{*} + \mathbf{u}\left(d_{pol}\right) \\
\frac{d\boldsymbol{\eta}^{*}}{dt} &= \mathrm{H}\left(x,\boldsymbol{\eta}^{*}\right) \\
\frac{d\left(d_{pol}\right)}{dt} &= \boldsymbol{\kappa}\left(d_{pol} \times \left(\Phi_d\left(x,\boldsymbol{\eta}^{*}\right) \times d_{pol}\right)\right)
\end{aligned} \tag{74}$$

4.5 Linear Reduced Order Latent Representation using Simulated Data

4.5.1 Modifications from section 2.2

For the given system, the augmented state vector modified from (9) is:

$$\tilde{x} = \begin{bmatrix} x \\ \eta^{*} \end{bmatrix} \in \mathbb{R}^{\left(n+n_{\eta}\right)\times 1} \tag{75}$$

Where we have eliminated the lamellipodial protrusion force from the definition of the auxiliary variable vector:

$$\eta^* = \eta - \frac{1}{(C_c + C_{cort})} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ F^c_{L,i} \\ \vdots \\ F^c_{L,n} \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{76}$$

In order to create a reduced order representation of the augmented system in latent variable space, we calculate the transformation V^* as shown in equation (11)in section 2.2.2. Here, V^* is derived using the data covariance matrix of simulated time samples of x and η^* for $t = 1,...,t_f$ and N different initial conditions:

$$\mathrm{X}^* = \begin{bmatrix} x(t_{0,1})^T & \eta^*(t_{0,1})^T \\ x(t_{1,1})^T & \eta^*(t_{1,1})^T \\ \vdots & \vdots \\ x(t_{f,1})^T & \eta^*(t_{f,1})^T \\ \vdots & \vdots \\ \vdots & \vdots \\ x(t_{0,N})^T & \eta^*(t_{0,N})^T \\ x(t_{1,N})^T & \eta^*(t_{1,N})^T \\ \vdots & \vdots \\ x(t_{f,N})^T & \eta^*(t_{f,N})^T \end{bmatrix} \in \mathbb{R}^{N \cdot t_f \times (n+n_\eta)} \tag{77}$$

Using V^* we can find the latent space representation as in(12). Differentiating we again obtain:

$$\frac{d\mathbf{z}}{dt} = V_x^{*T}\frac{dx}{dt} + V_\eta^{*T}\frac{d\boldsymbol{\eta}^*}{dt} \tag{78}$$

Substituting equation 4.7:

$$\frac{d\mathbf{z}}{dt} = V_x^{*T}\left(\mathbf{W}_\eta^* \cdot \boldsymbol{\eta}^* + \mathbf{u}\left(d_{pol}\right)\right) + V_\eta^{*T}\frac{d\boldsymbol{\eta}^*}{dt} \tag{79}$$

As previously discussed in section 2.2.2 we can estimate an approximate of η^* using latent variables and substitute into equation(79):

$$\begin{aligned} \eta^* &= V_\eta^* z(t) \\ \frac{d\mathbf{z}}{dt} &\approx V_x^{*T}\left(\mathbf{W}_\eta^* V_\eta^* z + \mathbf{u}\left(d_{pol}\right)\right) + V_\eta^{*T}\frac{d\boldsymbol{\eta}^*}{dt} \end{aligned} \tag{80}$$

As explained in 2.2.3, we can approximate $\frac{d\eta}{dt}$ using PCR:

$$\widehat{\frac{d\eta^*}{dt}} = \hat{\dot{\eta}}^*(t) = \mathbf{K}^* \cdot z(t) \tag{81}$$

Substituting:

$$\begin{aligned} \frac{d\mathbf{z}}{dt} &\approx V_x^{*T}\left(\mathbf{W}_\eta^* V_\eta^* z + \mathbf{u}\left(d_{pol}\right)\right) + V_\eta^{*T}\mathbf{K}^* \cdot z(t) \\ &= \left(V_x^{*T}\mathbf{W}_\eta^* V_\eta^* + V_\eta^{*T}\mathbf{K}^*\right) z(t) + V_x^{*T}\mathbf{u}\left(d_{pol}\right) \end{aligned} \tag{82}$$

We have left $\mathbf{u}$ explicit in the equation(82), as we will use the exact equations given in (70) and (72) to drive the reduced order latent variable system. However, the nonlinear relationship described in (73) will be estimated using PCR. In order to do this, let us rearrange the augmented

variables defined in $\tilde{x}$ into two distinct groups that represent field variables (ϕ) and component variables (α) as discussed section 2.3.2:

$$\begin{aligned}
\tilde{x} &= \begin{bmatrix} x \\ \eta^* \end{bmatrix} = \begin{bmatrix} \alpha \\ \phi \end{bmatrix} \in \mathbb{R}^{(n+n_\eta)\times 1} \\
\alpha &= \left[x^c, \mathrm{F}_{\mathrm{T}}^{c,i}, \mathrm{F}_E^c, \mathrm{F}_{FA}^c \right]^T \\
\phi &= \left[x^e, \mathrm{F}_E^e, \mathrm{F}_{FA}^e \right]^T
\end{aligned} \tag{83}$$

Since, as previously stated, the direction of maximum ECM stiffness depends the cell location $x_{center} = \frac{1}{n_{mem}} \sum_{i=1}^{n_{mem}} x_i^c$ and the stresses (or Forces) within the field, we will regress only the latent variable representation associated with these variables to onto $d^{\max}{}_{stiff,ECM}$:

$$\begin{aligned}
\hat{\boldsymbol{d}}^{\max}_{stiff,ECM} &= K_d \begin{bmatrix} z_{F\phi} \\ z_{x_c} \end{bmatrix} \\
z_{F\phi} &= P_{F_\phi} z \\
z_{x_c} &= P_{x_{center}} z \\
P_{F_\phi} &= \boldsymbol{V}^{*T} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n_\phi - n_{ECM} \times n_{n_\phi - n_{ECM}}} \end{bmatrix} \boldsymbol{V}^* \\
P_{x_{center}} &= \boldsymbol{V}^{*T} \left(\frac{1}{n_{mem}} \cdot \begin{bmatrix} J_{n_{mem} \times n_{mem}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \right) \boldsymbol{V}^* \\
J_{n_{mem} \times n_{mem}} &= matrix\ of\ ones
\end{aligned} \tag{84}$$

Where coefficient matrix K_d can be may be estimated by the least squares method. We may now write the reduced order latent space equations as:

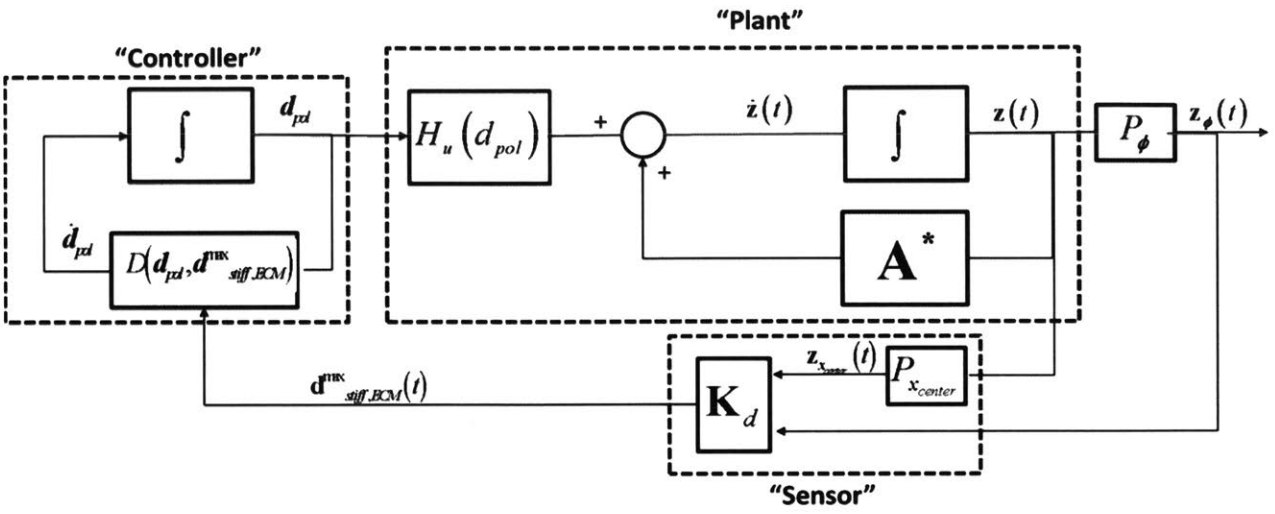


Figure 13: Block Diagram of the reduced order latent space represented in (85)

$$\frac{d\mathbf{z}}{dt} \approx \underbrace{\left(V_x^{*T}\mathbf{W}_\eta^* V_\eta^* + V_\eta^{*T}\mathbf{K}^*\right)}_{A^*} z(t) + \underbrace{V_x^{*T}\frac{1}{(C_c + C_{cort})}\mathbf{M}_{pol}\left(:, \arctan\left(\frac{|d_{pol0} \times d_{pol}|}{d_{pol0} \cdot d_{pol}}\right)\right)\cdot F_L}_{H_u(d_{pol})}$$

$$\frac{d(d_{pol})}{dt} = \underbrace{\kappa\left(d_{pol} \times \left(K_d \begin{bmatrix} z_{F\phi} \\ z_{x_c} \end{bmatrix} \times d_{pol}\right)\right)}_{D(d_{pol}, d^{max}_{stiff,ECM})} = K_d \begin{bmatrix} z_{F\phi} \\ z_{x_c} \end{bmatrix}\left(d_{pol} \cdot d_{pol}\right) - d_{pol}\left(d_{pol} \cdot K_d \begin{bmatrix} z_{F\phi} \\ z_{x_c} \end{bmatrix}\right) \tag{85}$$

Were we have rewritten input polarity dynamics as using triple product expansion. Under the assumptions stated in 2.3.1 we can link the above system. Figure 13 shows the block diagram of the equations in (85). From the basic structure we may identify pseudo plant (A matrix), controller (polarity equation) and sensor (direction of maximum stiffness equation) blocks.

4.5.2 Model Evaluation and Analysis

In order to evaluate the model, calculated the latent space trajectory and compared with the original simulation.

As can be seen in figure 14there is good agreement between the deformations of the real cell compared to the reduced order latent variable model. Figure 15 shows the mean squared error as a function of latent variable dimension (*m*):

$$MSE = \frac{1}{t_f}\sum_{t=0}^{t_f-1}\left(e_t^T e_t\right) \tag{86}$$

Where e_t is the residual error between the linear state transitions and the original nonlinear simulation at time *t*. As can be seen, as the number of latent variables increase the mean squared error decreases significantly. At 100 latent variables, the mean square error is $MSE = 5\times10^{-15}$. However, since the cell deforms around 1×10^{-10} per time step the mean squatted error for 10 latent variables is still sufficiently low.

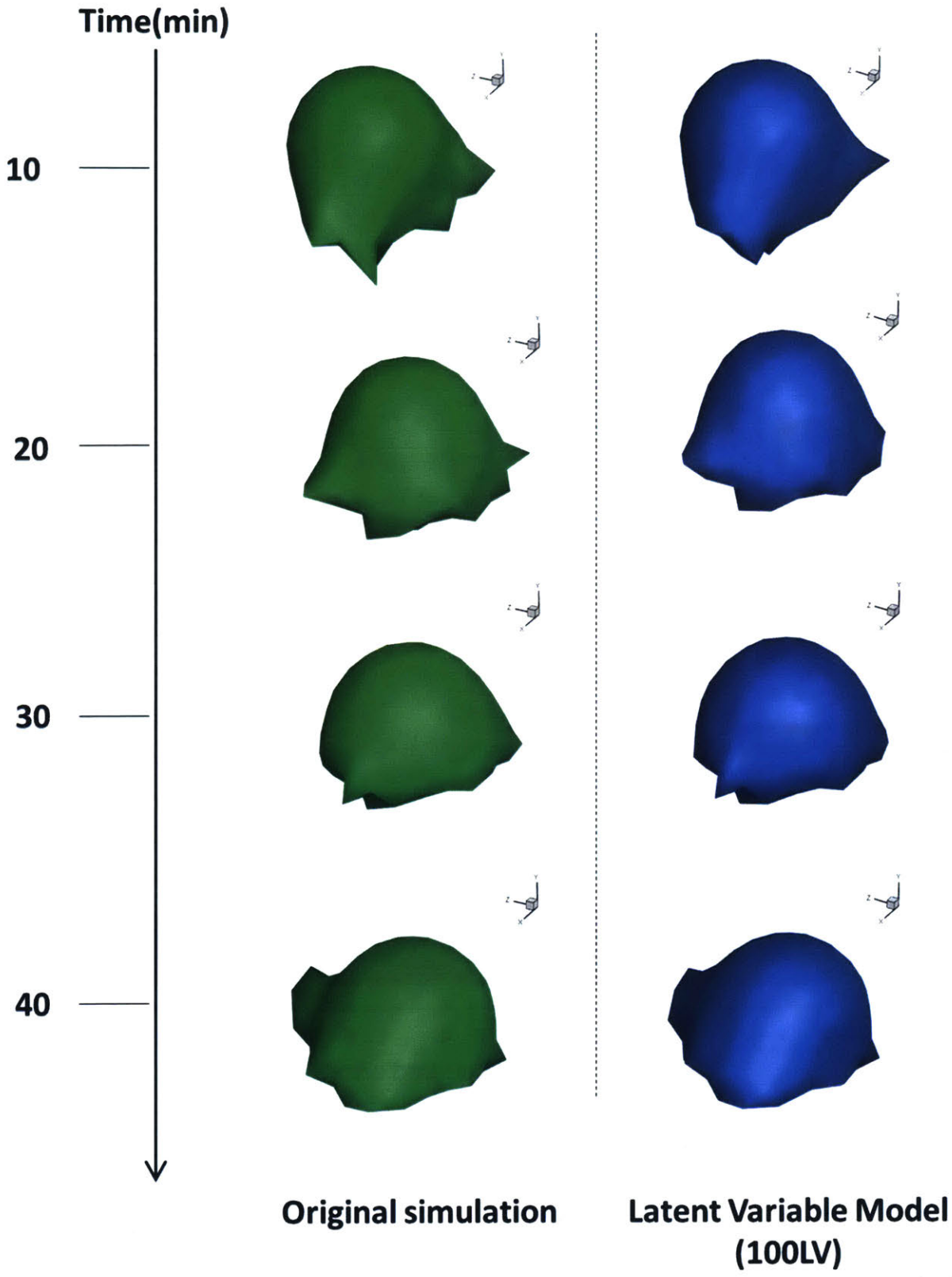


Figure 14: Comparison of cell morphologies over time between the original simulation (green) and the latent variable simulation using 100 latent variables(blue).

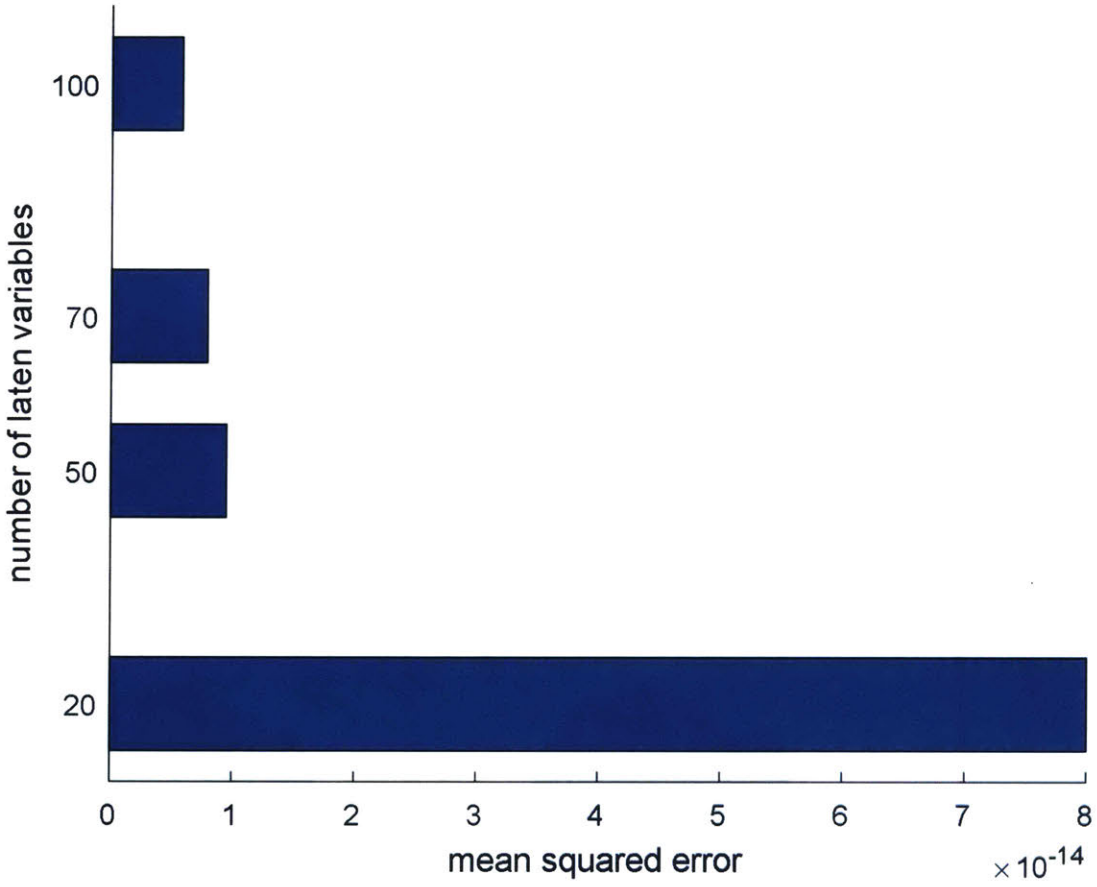


Figure 15: Mean squared error as a function of latent variable dimension (m)

The main distinction between the number of latent variables considered and the resulting model is that as more latent variables are included, higher order dynamics may be described. This means that with 10 latent variables the model can capture the average lower frequency deformations of the cell and ECM, but does not capture fast fluctuations displayed in the original simulation. However if 100 latent variables are included within the model, these fluctuation may be reproduced.

Figures 16 and 17 show the eigenvalues for the A^* matrix (which corresponds to open loop poles of the system) of the 10 latent variable and 100 latent variable model. As can be seen in figure 16 the majority poles for the 10 latent variable model reveal more damped frequencies. However when 100 latent variables are used within the model, the imaginary component of complex pole pairs are larger, indicating higher frequencies within the model.

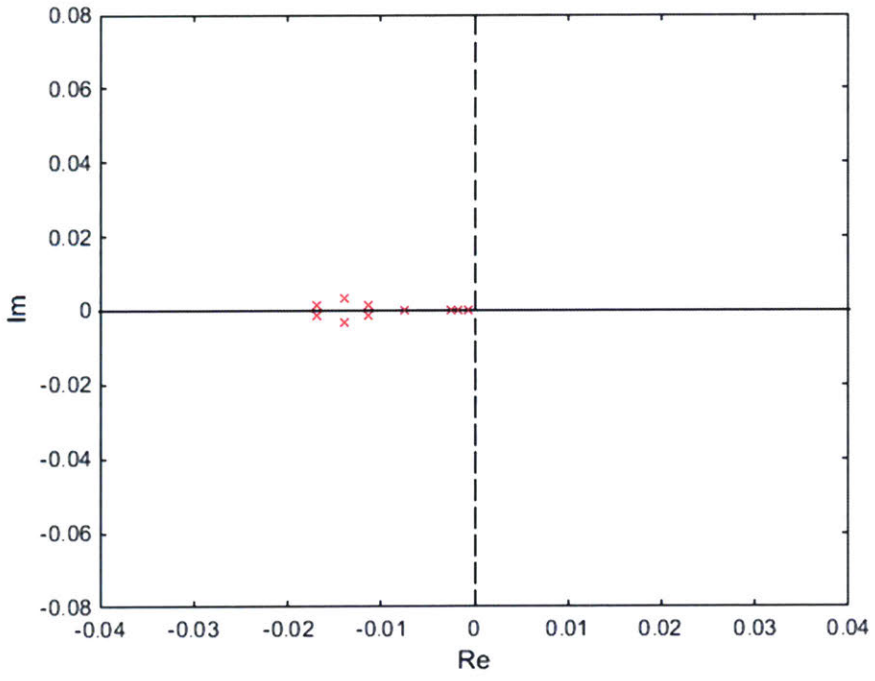


Figure 16: Eigenvalues for the matrix (which corresponds to open loop poles of the system) of the 10 latent variable model. The majority poles are near the real axis indicating lower frequencies.

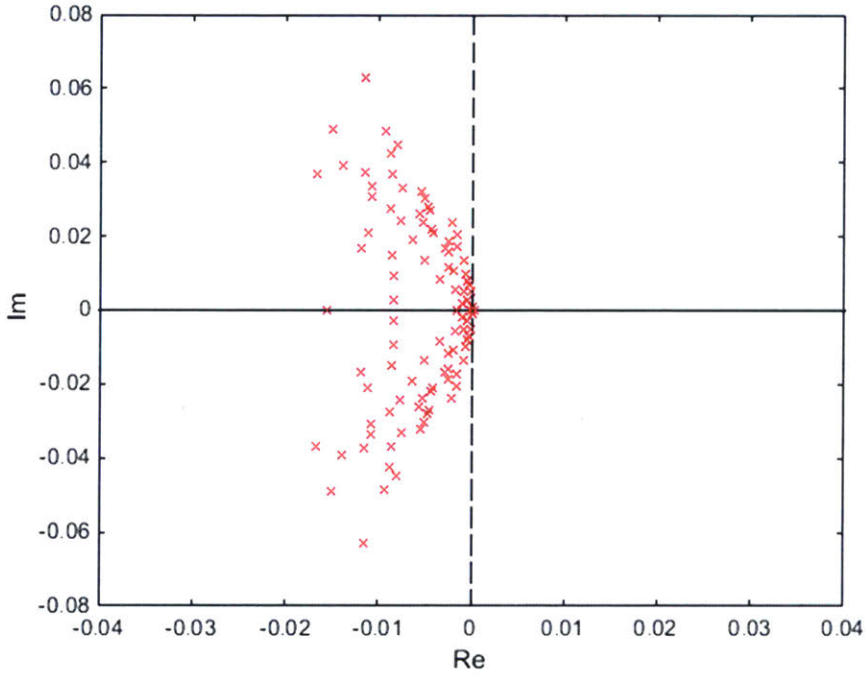


Figure 17: Eigenvalues for the matrix (which corresponds to open loop poles of the system) of the 100 latent variable model. The imaginary component of complex pole pairs are larger, indicating higher frequencies within the model.

Since the majority of the dynamics within the reduced order latent space model have been represented linearly, the computation time from the original simulation is decreased significantly. Figure 18 show the computation time a function of latent variable dimension (m). This is compared to the original simulation which took seven hours to compute the same number of time points.

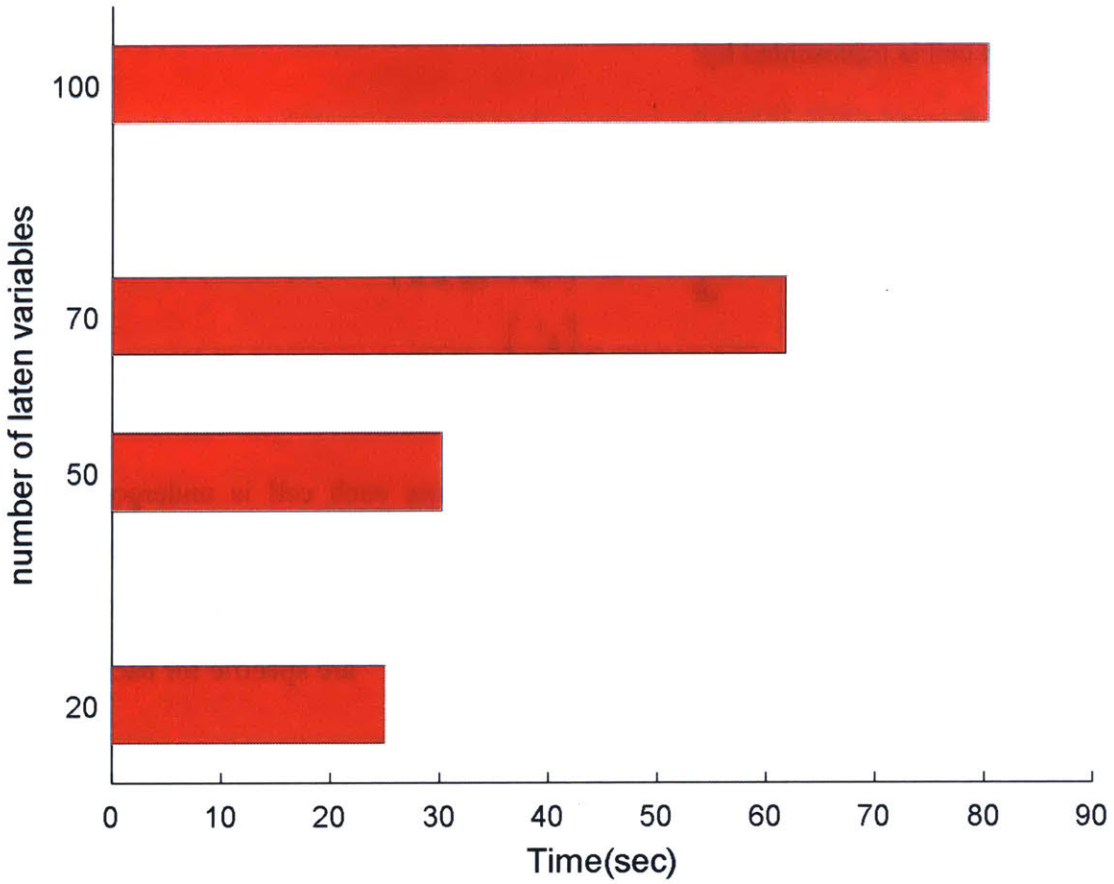


Figure 18: Computation time a function of latent variable dimension (m). This is compared to the original simulation which too seven hours to compute the same number of time points

4.6 Linking Individual Reduced Order Models through the Dynamics of Shared Variables

Now let us consider the case of multiple cells interacting on an elastic substrate. As mentioned previously, experimental findings showed that cells transmit forces through the ECM causing increased deformation and compaction of the ECM between the cells [4], [5]. We would like to estimate the interaction of the multicellular system by linking the latent space models of the individual cells (derived in the previous section) through the ECM.

Consider K reduced order latent space models of a single cell spreading on an elastic substrate. The *kth* cell is represented by:

$$\begin{aligned}
&\frac{dz^k}{dt} \approx A^* \cdot z^k + H_u^k\left(d_{pol}^k\right); \quad z^k(0) = z^{k,0} \\
&\frac{d\left(d_{pol}^k\right)}{dt} = D^k\left(d_{pol}^k, d_{stiff,ECM}^{\max,k}\right) \\
&d_{stiff,ECM}^{\max,k} = K_d^k \begin{bmatrix} z_{F\phi}^k \\ z_{x_c}^k \end{bmatrix}
\end{aligned} \tag{87}$$

Here matrix A^* is the same for each cell because each cell is undergoing the same phenotype (i.e cell spreading interactions on an elastic substrate). However the internal mechanics (i.e. cell polarity) will differ depending on the location and local environment of the cell. Therefore the polarity equation $H_u^k\left(d_{pol}^k\right)$ and initial conditions $z^k(0) = z^{k,0}$ are specific for each cell.

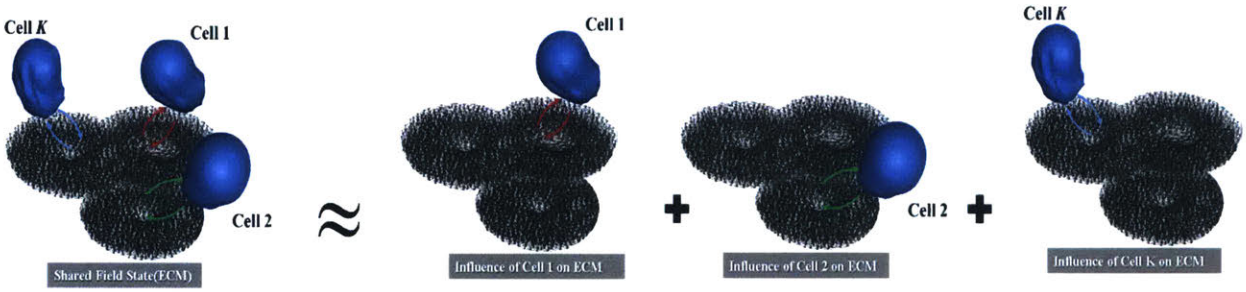


Figure 19: Conceptual diagram showing how multi-cell system will be estimated using superposition of the influence of multiple linear single cell models through a shared ECM.

4.6.1Modifications from section 2.3

First, we isolate the ECM forces of individual cell models by using projection matrix P_{F_ϕ} as described in equation(84):

$$\begin{aligned} z_\phi^k &= P_{F\phi} z^k \\ P_{F\phi} &= \boldsymbol{V}^{*T} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n_\phi - n_{ECM} \times n_{n_\phi - n_{ECM}}} \end{bmatrix} \boldsymbol{V}^* \end{aligned} \tag{88}$$

We link the ECM dynamics of each individual cell in the following manner:

$$\begin{aligned} \dot{z}^k &= \left(I - P_{F\phi} - P_{map} P_{F\phi}\right) \dot{z}^k + \dot{z}_{F\phi}^k + P_{map} \dot{z}_{F\phi}^k \\ \dot{z}_{F\phi}^k &= P_{F\phi} \dot{z}_{F\phi}^k \end{aligned} \tag{89}$$

$$\begin{aligned} \dot{z}^{k,emergent} &= \dot{z}^k + \left(I + P_{map}\right) \cdot \sum_{\ell \neq k} \dot{z}_{F\phi}^\ell \\ &= \left(I - P_{F\phi} - P_{map} P_{F\phi}\right) \dot{z}^k + \dot{z}_{F\phi}^k + P_{map} \dot{z}_{F\phi}^k + \left(I + P_{map}\right) \cdot \sum_{\ell \neq k} \dot{z}_{F\phi}^\ell \\ &= \left(I - P_{F\phi} - P_{map} P_{F\phi}\right) \dot{z}^k + \left(I + P_{map}\right) \cdot \sum_{\ell=1}^{K} \dot{z}_{F\phi}^\ell \end{aligned} \tag{90}$$

Where a mapping function matrix (P_{map}), derived from the simulated training data) is used to constrain the focal adhesion forces of connected membrane and ECM nodes to be equal and opposite.

Substituting equation (87)into equation (90) we may write:

$$\dot{z}^{k,emergant} \approx \left(A^* z^k + H_u^k\left(d_{pol}^k\right)\right) + \left(I + P_{map}\right) P_{F\phi} \cdot \sum_{\ell \neq k} \left(A^* z^\ell + H_u^\ell\left(d_{pol}^\ell\right)\right) \tag{91}$$

In matrix form (for components $k = 1, \ldots, K$) we may write:

$$\begin{bmatrix} \dot{z}^{1,emergant} \\ \dot{z}^{2,emergant} \\ \vdots \\ \dot{z}^{K,emergant} \end{bmatrix} \approx \begin{bmatrix} A^* & \left(I+P_{map}\right)P_{F\phi}A^* & \cdots & \left(I+P_{map}\right)P_{F\phi}A^* \\ \left(I+P_{map}\right)P_{F\phi}A^* & A^* & \cdots & \left(I+P_{map}\right)P_{F\phi}A^* \\ \vdots & \vdots & \ddots & \vdots \\ \left(I+P_{map}\right)P_{F\phi}A^* & \left(I+P_{map}\right)P_{F\phi}A^* & \cdots & A^* \end{bmatrix} \begin{bmatrix} z^1 \\ z^2 \\ \vdots \\ z^K \end{bmatrix}$$
$$+ \begin{bmatrix} I & \left(I+P_{map}\right)P_{F\phi} & \cdots & \left(I+P_{map}\right)P_{F\phi} \\ \left(I+P_{map}\right)P_{F\phi} & I & \cdots & \left(I+P_{map}\right)P_{F\phi} \\ \vdots & \vdots & \ddots & \vdots \\ \left(I+P_{map}\right)P_{F\phi} & \left(I+P_{map}\right)P_{F\phi} & \cdots & I \end{bmatrix} \begin{bmatrix} H_u^1\left(d_{pol}^1\right) \\ H_u^2\left(d_{pol}^2\right) \\ \\ H_u^K\left(d_{pol}^K\right) \end{bmatrix} \tag{92}$$

Substituting equation (35) into equation(92):

$$\begin{bmatrix} \dot{z}^{1,emergant} \\ \dot{z}^{2,emergant} \\ \vdots \\ \dot{z}^{K,emergant} \end{bmatrix} \approx \begin{bmatrix} A^* & \left(I+P_{map}\right)P_{F\phi}A^* & \cdots & \left(I+P_{map}\right)P_{F\phi}A^* \\ \left(I+P_{map}\right)P_{F\phi}A^* & A^* & \cdots & \left(I+P_{map}\right)P_{F\phi}A^* \\ \vdots & \vdots & \ddots & \vdots \\ \left(I+P_{map}\right)P_{F\phi}A^* & \left(I+P_{map}\right)P_{F\phi}A^* & \cdots & A^* \end{bmatrix} \begin{bmatrix} I & P_{F\phi} & \cdots & P_{F\phi} \\ P_{F\phi} & I & \cdots & P_{F\phi} \\ \vdots & \vdots & \ddots & \vdots \\ P_{F\phi} & P_{F\phi} & \cdots & I \end{bmatrix}^{\#} \begin{bmatrix} z^{1,emergant} \\ z^{2,emergant} \\ \vdots \\ z^{K,emergant} \end{bmatrix}$$
$$+ \begin{bmatrix} I & \left(I+P_{map}\right)P_{F\phi} & \cdots & \left(I+P_{map}\right)P_{F\phi} \\ \left(I+P_{map}\right)P_{F\phi} & I & \cdots & \left(I+P_{map}\right)P_{F\phi} \\ \vdots & \vdots & \ddots & \vdots \\ \left(I+P_{map}\right)P_{F\phi} & \left(I+P_{map}\right)P_{F\phi} & \cdots & I \end{bmatrix} \begin{bmatrix} H_u^1\left(d_{pol}^1\right) \\ H_u^2\left(d_{pol}^2\right) \\ \\ H_u^K\left(d_{pol}^K\right) \end{bmatrix} \tag{93}$$

#– pseudo inverse

Equation (93) represents the K coupled dynamic equations that we will use to predict the behaviors of the larger multi-cell system.

4.6.2 Model Evaluation and Analysis for 2 Cells

Since we approximate the (originally nonlinear) multi-cell ECM environment by superposition individual (linear) single-cell/ECM dynamics, we do not expect the predicted trajectories to be exact to that of the original nonlinear simulation (with same initial conditions). However the matrix-mediated mechanical interaction leading to emergent behavior are reproduced may be reproduced.

A key finding in Guo et al. showed that the ECM within the central gap between two interacting cells was significantly deformed compared to areas on the periphery of the central gap, which remained static. This indicated directed matrix-mediated force transmission in between cells [3]. Similarly this phenomena can be reproduced within the multi-cell spreading model presented. Figure 20 shows a contour plot of the of ECM nodes for the latent variable model and original simulation. The larger red arrows indicate velocity has reached a threshold over 1.6nm/s with highest speeds directly underneath the cell (at 10nm/s). This value of 1.6 nm/s is consistent with deformation velocity experiments between 2 cells outlined in Guo et al[3]. As can be seen, ECM node speeds are only significant within the central gap as compared to the surrounding ECM where the velocities are zero. Here we also show that the latent variable model is capable of reproducing this phenomena which is present within the original simulation.

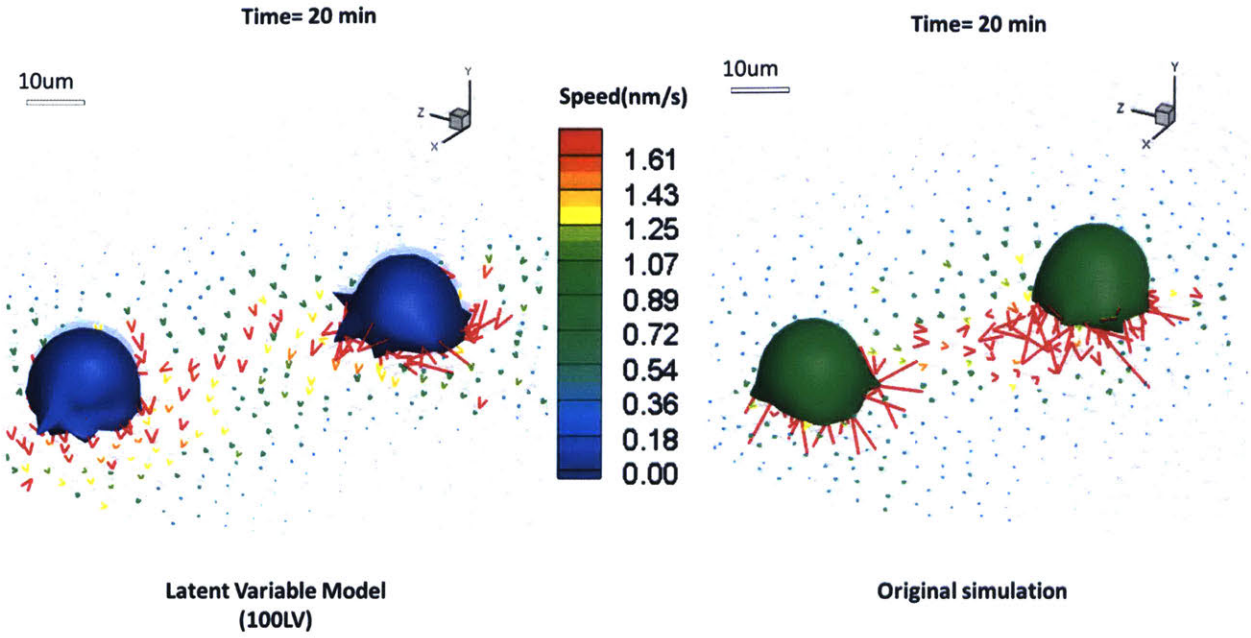


Figure 20: Contour plot of the of ECM nodes for the latent variable model and original simulation. The larger red arrows indicate velocity has reached a threshold over 2nm/s with highest speeds directly underneath the cell (at 10nm/s). The latent variable model is capable of reproducing this phenomena which is present within the original simulation

Using, the latent variable model we may also reveal more pronounced deformations (indicated by higher ECM node velocities) when the cells are spaced closer together. Figure 21 shows a contour plots of cells spaced at 10um and 30 um at 20 minute and 40 minute time points. As can be seen, the ECM node velocity between the cells increase as the as the gap between the cells decrease. Figure 22 summarizes the average velocity of ECM node s in-between cells (red) and areas on the periphery (blue) for 10um spacing and 30um spacing. As expected, the ECM node velocities increase in between the cells for smaller gap sizes indicating increased mechanical interaction as closer distances.

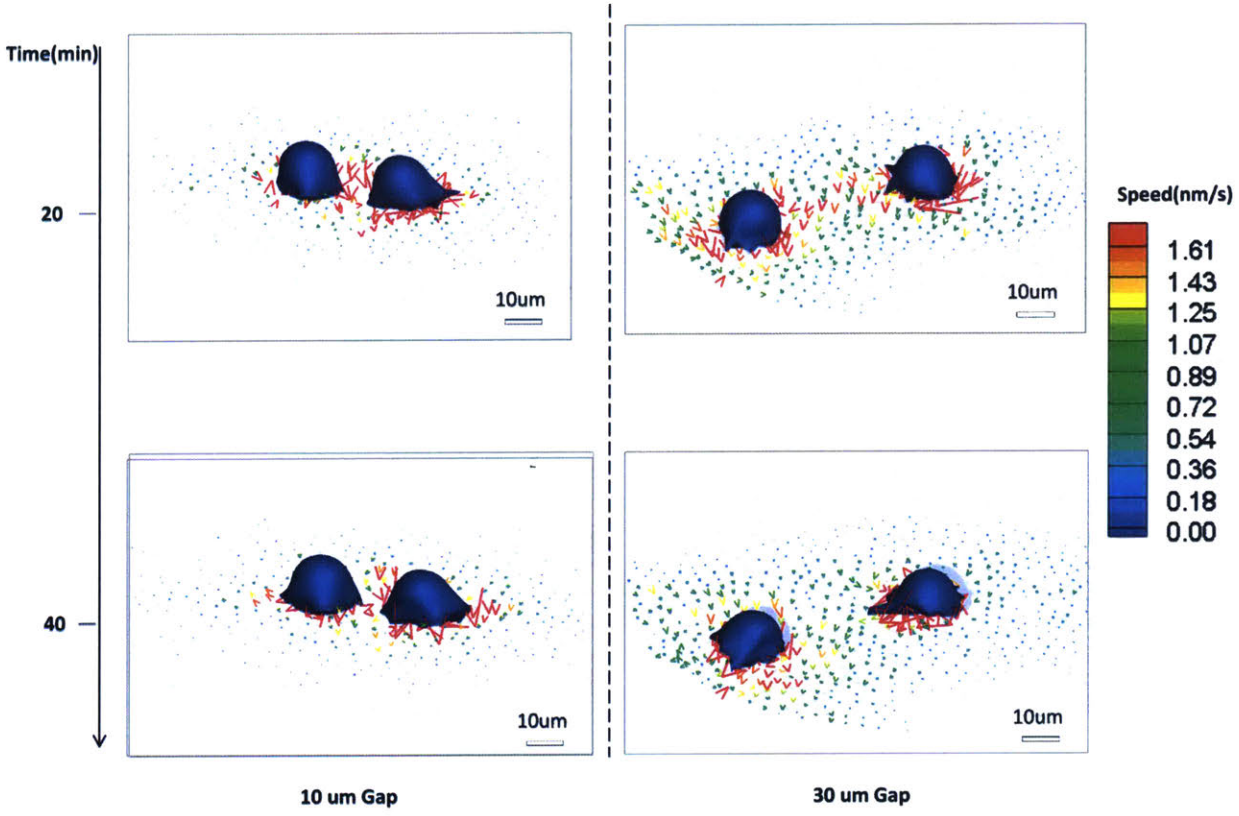


Figure 21: Contour plots of cells spaced at 10um and 30 um at 20 minute and 40 minute time points.

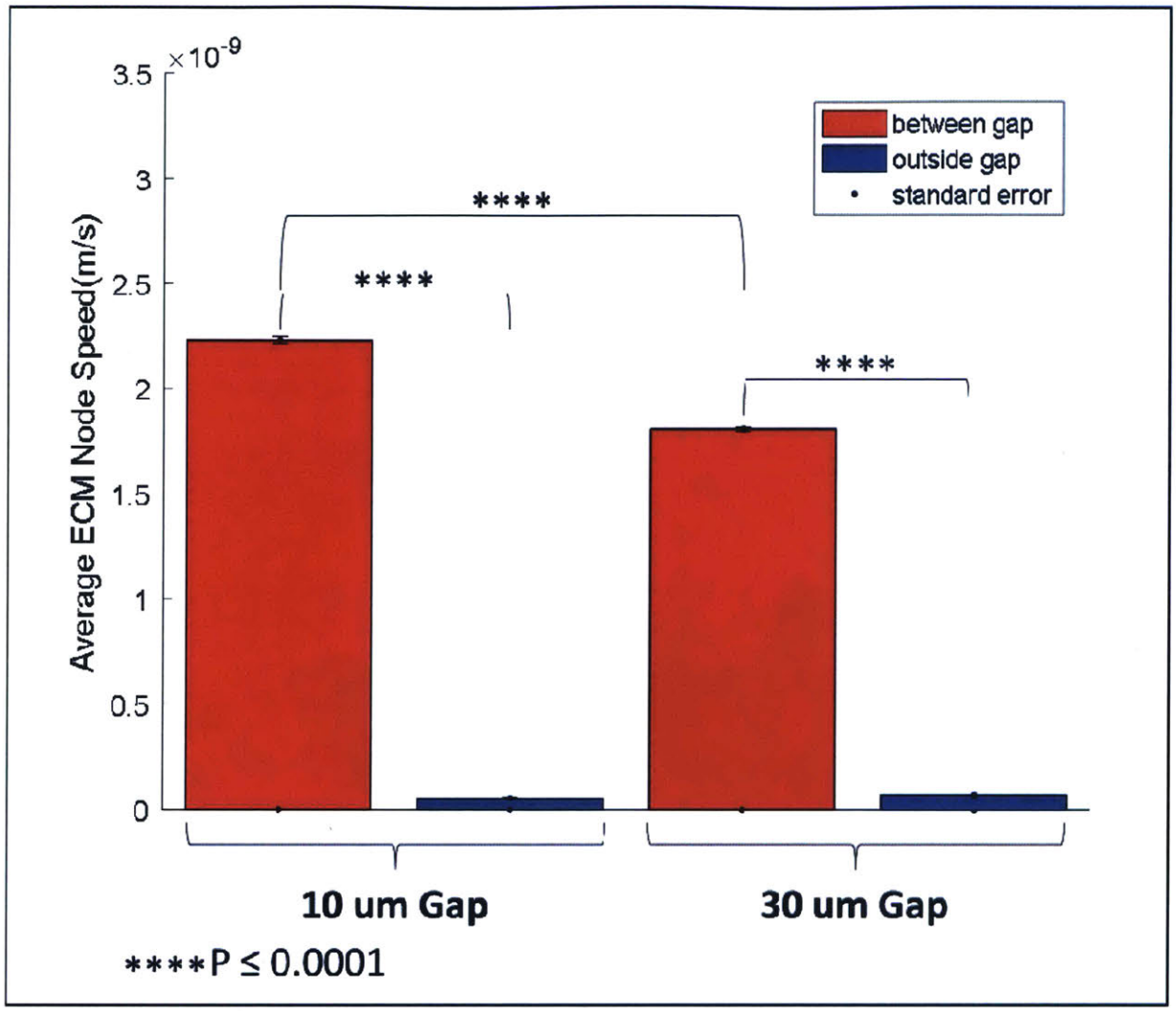


Figure 22: The average velocity of ECM node s in-between cells (red) and areas on the periphery (blue) for 10um spacing and 30um spacing.

We also analyzed cell spreading within the gap by measuring/analyzing the changing gap size (or distance between cell's protrusions within the gap). Figure 23 shows the changing gap size normalized by the original gap size over time. As can be seen the cells spread more extensively towards each other when closer together which is consistent with the increase in non-dispersed force transmission between cells that are closer together.

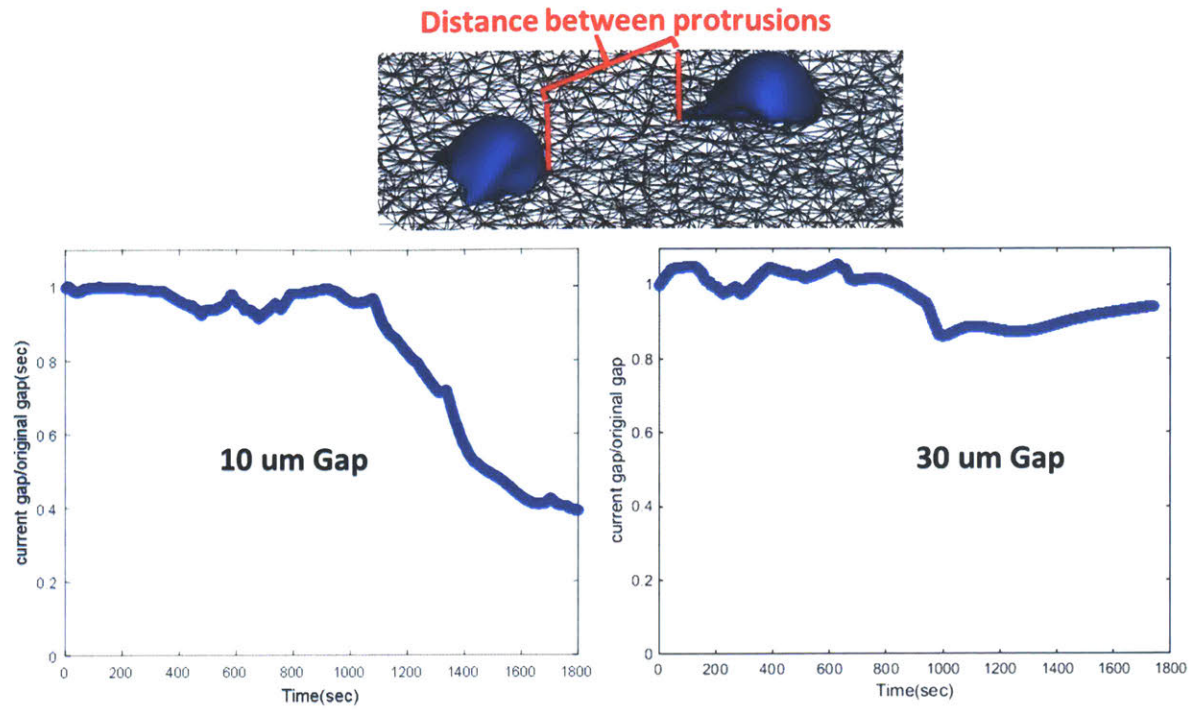


Figure 23: Changing gap size (or distance between cell's protrusions) normalized by the original gap size over time.

Using the latent variable approach, the computation time from the original simulation is decreased significantly. Table 4 summarizes the computation times for 20 latent variable model and 100 latent variable model for 2 cells compared to the original simulation computation time.

	20 LV model	**100LV model**	**Real Simulation**
Computation time	2 minutes	20 minutes	10 hours

Table 4: Computation times for 20 latent variable model and 100 latent variable model for 2 cells compared to the original simulation computation time

4.6.3 Model Evaluation and Analysis for 10 Cells

Extending the aforementioned approach to 10 cells, we see that the variation in ECM node velocity in between cells with respect to gap size is similar to the 2-cell case. As can be seen by the contour plot (of ten cells placed at random initial locations) cells can interact with multiple cells within its vicinity. However, at a distance of about 40um inter-cell interactions are attenuated. Interestingly, this attenuation distance is consistent with recent observations of mechanical interaction between cells crawling on a 2D substrate which showed an interaction length of the order of 30 um[6]. Figure 24 summarizes the average velocity of ECM nodes in-between cells for 5um spacing, 10-20um spacing and 40-50um spacing. Note that the ECM node velocities near cells in the upper right were not included in the analysis since they are affected by the ECM boundary. As within the 2-cell case ECM node velocity decreases in between cells as gap size increases. However the overall ECM node speeds are higher which is expected since each cell is interacting with more than one cell.

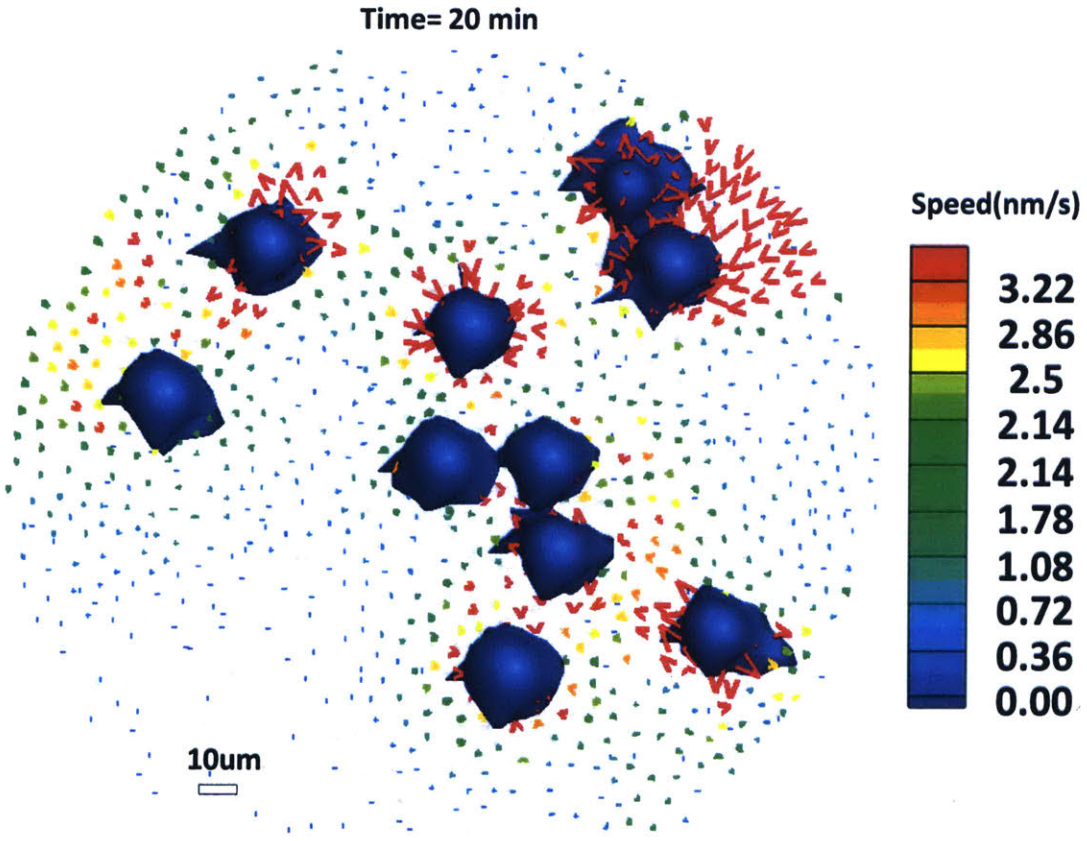


Figure 24: Contour plot of ECM node velocities of ten cells placed at random initial locations on the ECM.

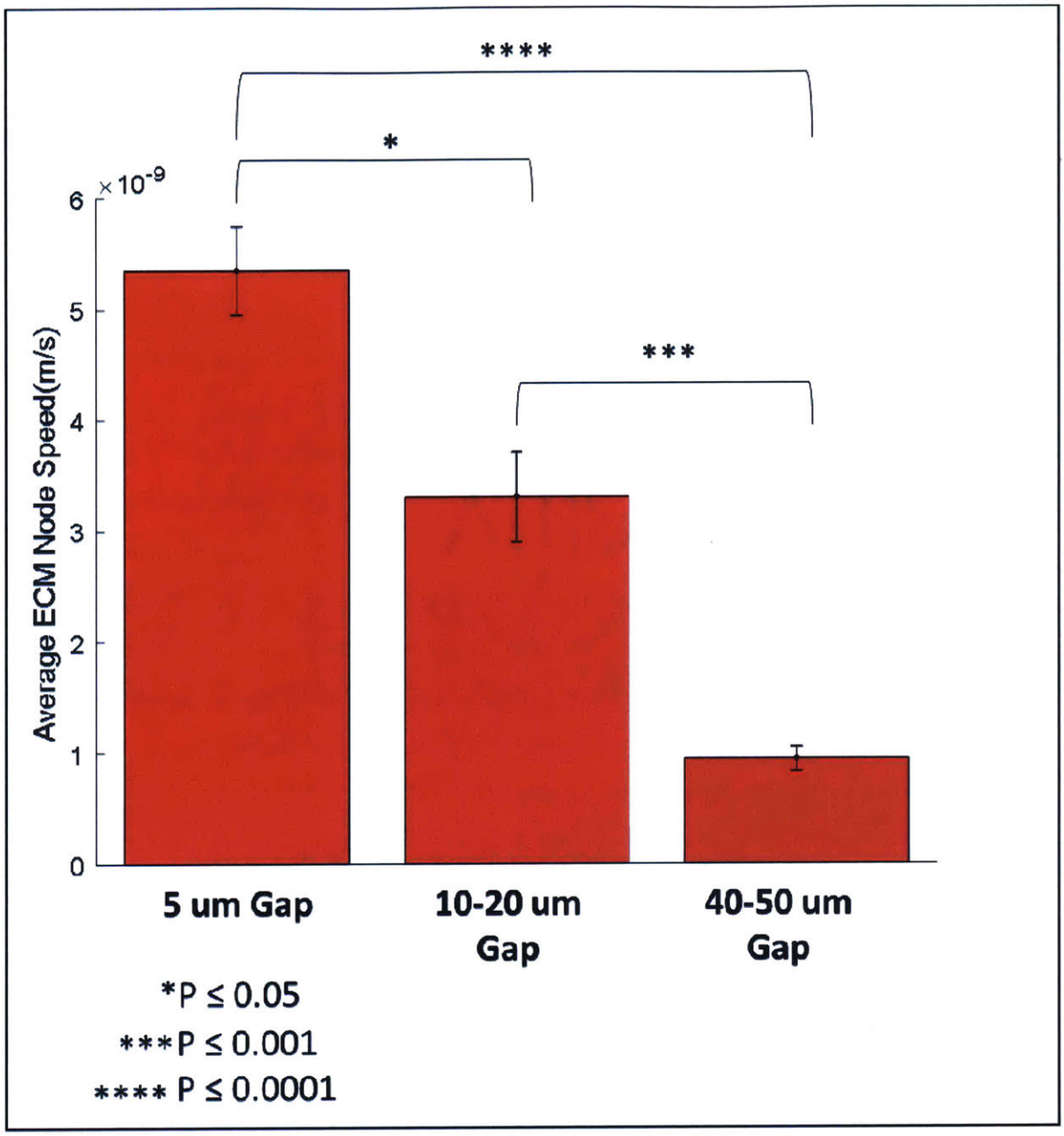


Figure 25: Average velocity of ECM nodes in-between cells for 5um spacing, 10-20um spacing and 40-50um spacing

4.6.4 Extension to 3-D Cells Embedded in a Fibrous matrix

In three dimensions, large shape changes of the cell imply very pronounced deformations of the surrounding matrix. Thus, the phenomena of cells embedded in a 3-D matrix is more pronounced and crucially different from those observed on 2D substrates[10].

A key finding in Fernandez et al. showed that that multiple cells embedded inside a 3-Dimensional collagen gel could spontaneously contract the entire gel volume via collective contractile activity, an important mechanism behind tissue formation. Furthermore, local deformations of a single cell embedded within the gel were insufficient to produce significant contraction of the gel[10].

Similarly this phenomena can be reproduced for cells embedded within a 3-Dimenstional ECM as shown in figure 25. The simulations shown in figure 26compare compaction and densification of the ECM in between two cells for the original simulation(green cells) and the linear superposition approach using 100 latent variables (blue cells). As can be seen the latent variable superposition approach reproduces contraction of the gel boundary which is also present within the original simulation. In addition we have reduced computation time to 20 minutes using the superposition approach (same as for the 2-D case). In Figure 27 we further show that 1 cell produces smaller and more local deformations than 2-cells embedded in a 3-D ECM. This indicates that the mechanism of compaction is dependent on the number and spacing (i.e. density) of cells within the gel. Figure 28 show quantification of the 1 cell vs. 2-cell gel compaction using thickens of the ECM boundary at specific distances along the axial (x) direction as a metric. As can be seen, 2-cell model ECM boundary of the 2-cell model compacts contracts more and also the contraction is more global along the axial direction. The maximum contraction of the 1 cell model is 90% whereas the maxim contraction of the 1-cell model is 67%.

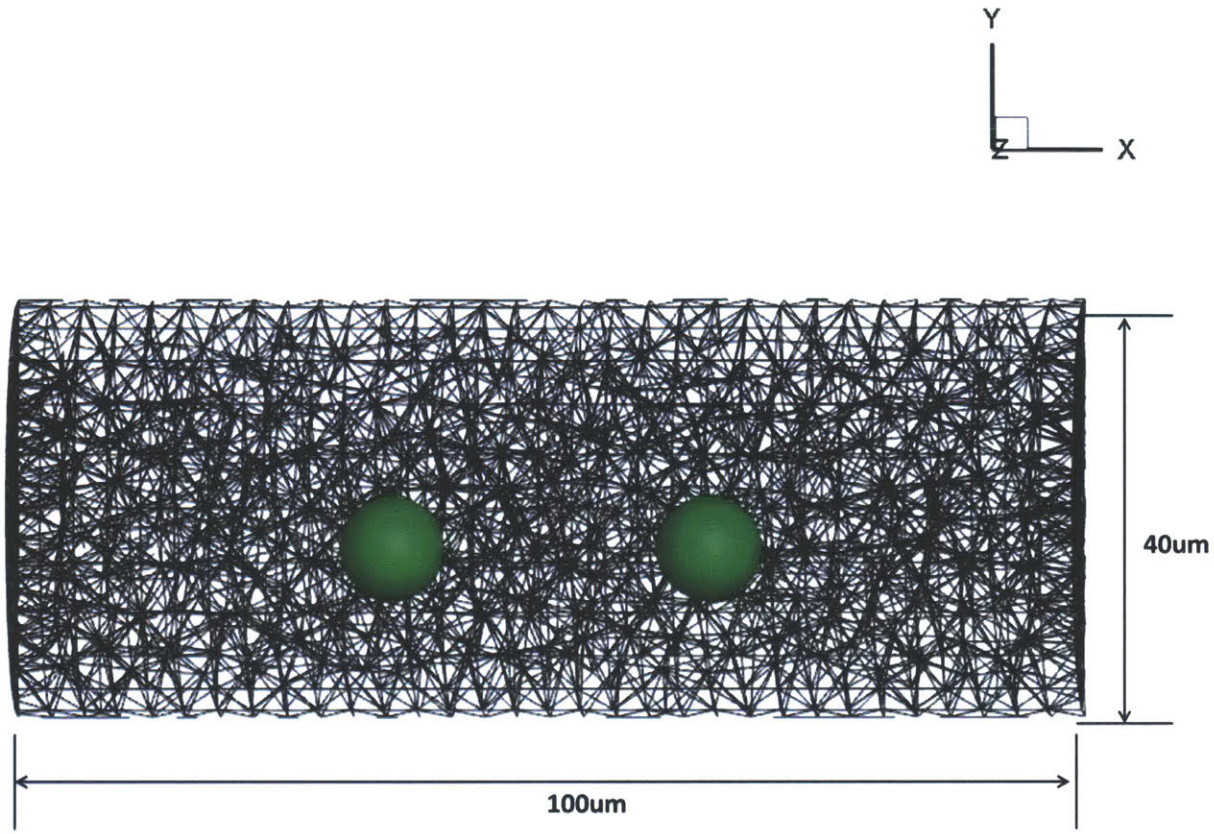


Figure 26: Biophysical model for the study of cells embedded in a 3-D matrix

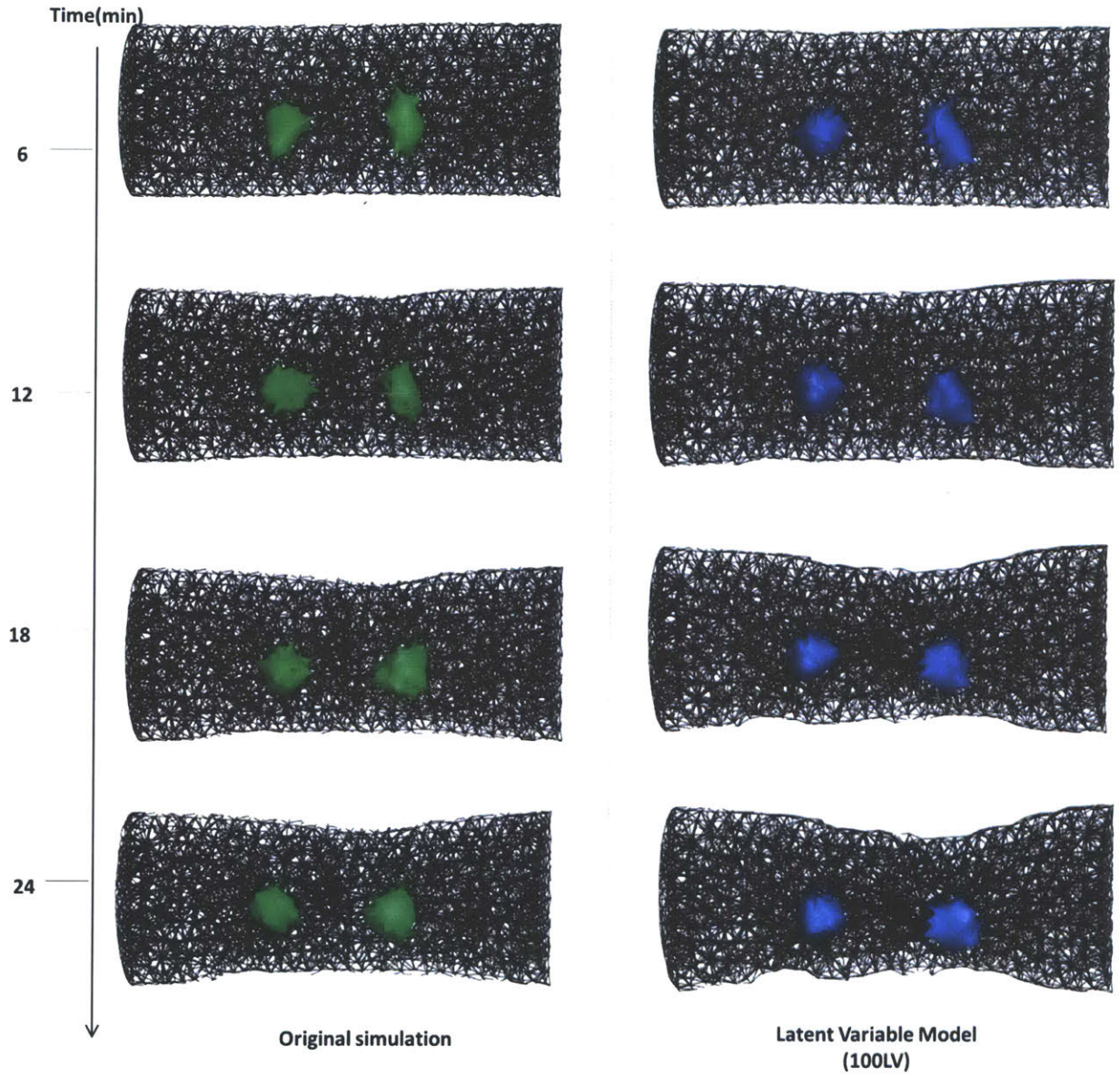


Figure 27: Simulation shows compaction and densification of the ECM in between the two cells for the original simulation (green cells) and the linear superposition approach using 100 latent variables (blue cells).

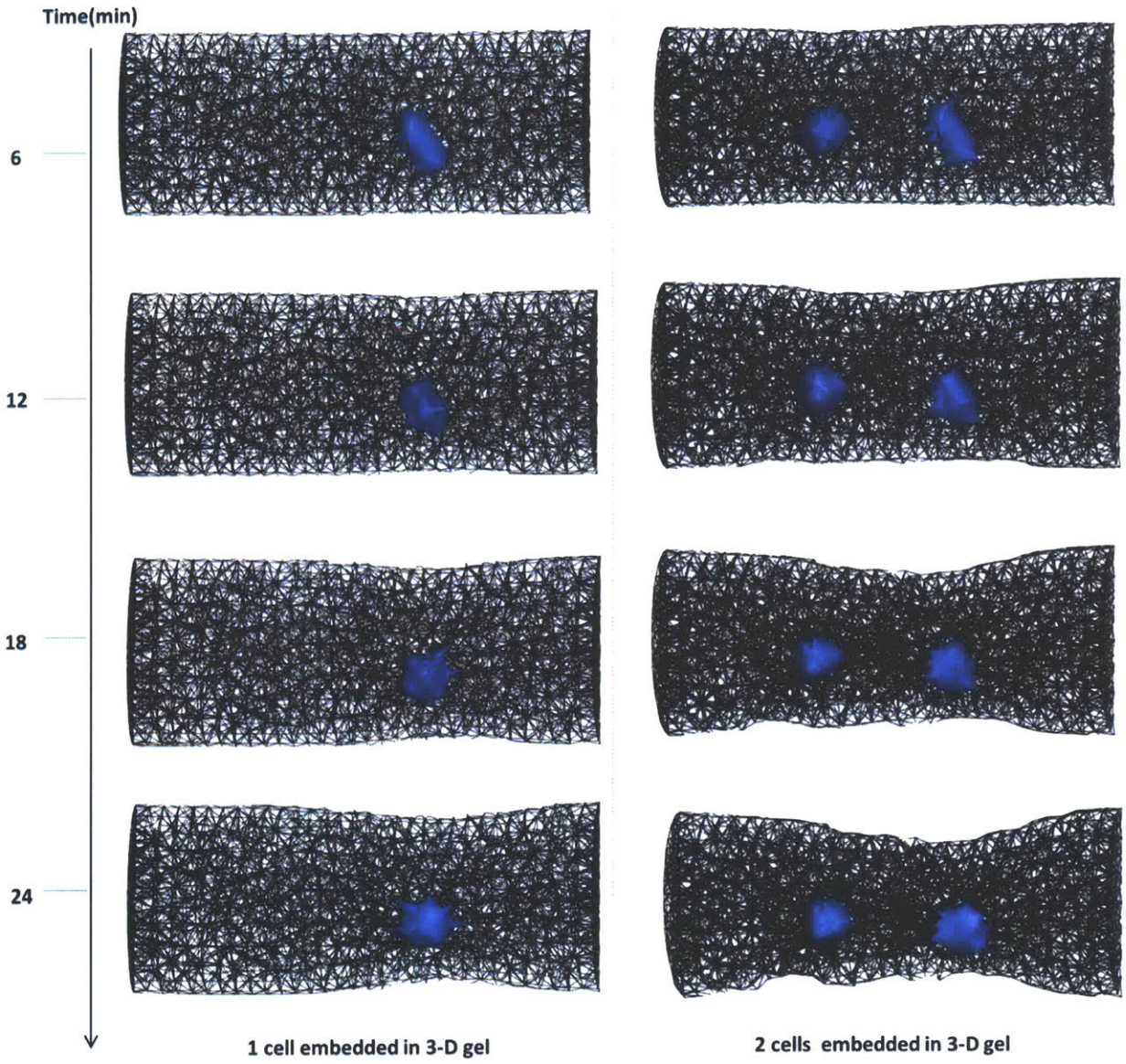


Figure 28: Simulation shows thatthat 1 cell produces smaller and more local deformations than 2-cells embedded in a 3-D ECM.

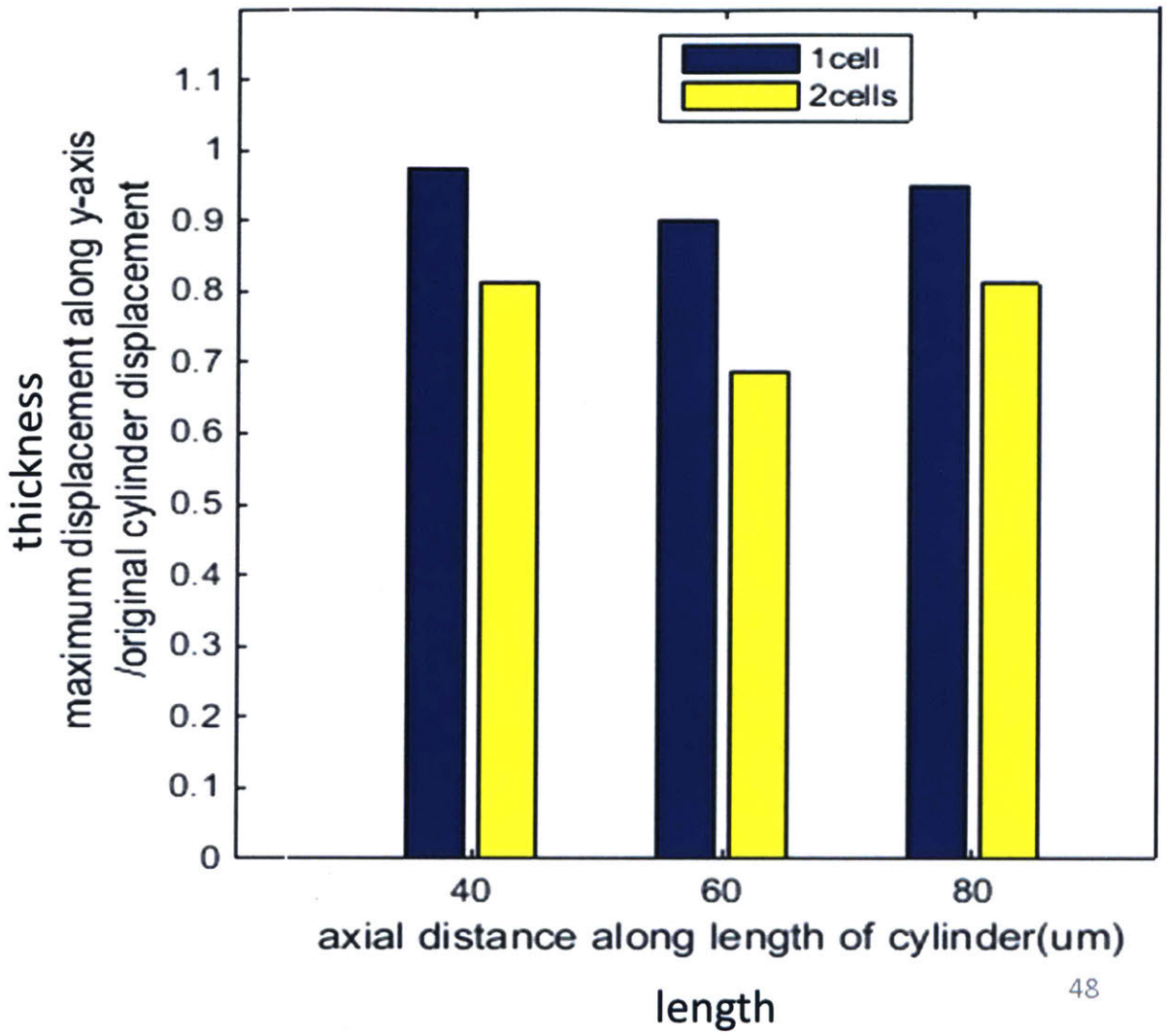


Figure 29: Quantification of the 1 cell (blue) vs. 2-cell (yellow) gel compaction using thickness of the ECM boundary at specific distances along the axial (x) direction as a metric

5. Conclusion

5.1 Contributions of This Work

This thesis has developed a methodology for the linearization and subsequent superposition of single-cell models to explain the emergent behavior among multiple cells. Contributions were made in the 3 major areas: Theoretical, Computational and System Integration, and Biological.

5.1.1 Theoretical Contribution

The nonlinear dynamics describing single-cell/ECM interactions were modeled as linearized subsystems in a more complex framework describing multi-cell interactions. This was achieved by the following:

1. Recasting nonlinear (single-cell) dynamics to higher dimensional space by augmenting the original (single-cell) system with auxiliary variables (that is, output variables on all the nonlinear elements) derived from bond graph representation. This also created a more complete description of the nonlinear dynamics.
2. Transforming the augmented state equations (describing single-cell) to a reduced-order linear representation by projecting the augmented state equations onto a basis of eigenvectors derived from simulated data set. This allowed for the evolution of the states to be described within a lower dimensional linear manifold.

The resulting reduced order latent space model was capable of reproducing nonlinear dynamics (including cell polarity and cell-matrix interactions). Furthermore the linearized structure of individual models facilitated their integration to describe multi-cell behaviors.

5.1.2 Computational and System Integration Contribution

The prediction of a more complex multi-cell mechanical system using the aforementioned linearized subsystems was achieved by the following:

1. Isolating matrix dynamics within the linearized (single-cell) models describing individual cells using linear projection
2. Linking matrix dynamics of individual cell models using linear superposition to approximate a multi-cell environment.

With these combined methodologies, we created a comprehensive framework to facilitate the study systems of interacting nonlinear agents, which would otherwise be prohibitively complex to compute.

5.1.3 Biological Contribution

Using the proposed methodologies, we were able to reproduce intercellular mechanical interactions consistent with published experimental observations for the following cell induced emergent mechanical changes within ECM:

1. Long-range non-dispersed force transmission between cells through measurement of increased deformation velocity between cells. Non-dispersed force transmission is a crucial mechanism for the initiation and maintenance of long-scale multi-cell linear patterns [11].
2. Global contraction of gel volume via collective cell- contractile activity (as opposed to local deformations of single cell embedded within the gel)]. Cell collective contractile activity is an important mechanism behind tissue formation[10].

Furthermore, our analysis proved to be consistent with observations of mechanical interaction between cells on a 2D substrate which an interactions showed attenuated interactions length of the after 30um. We have introduced a new set of computational tools for modeling mechanical interactions between cells and the extracellular matrix. Furthermore, through our study of intermediate emergent behaviors arising within the matrix we can advance our understanding of biomechanical signaling mechanisms during tissue formation and multi-cellular patterning.

5.2 Limitations of Approach

A key limitation within our approach is that in order to reproduce biologically relevant phenomena, one must begin with a biophysical model that accurately represents a subcomponent of the phenomena being studied in order to create relevant simulation dataset. However in the future, simulation data can be supplemented with experimental data in order to create a more accurate dataset. Related to the creation of a dataset, another limitation is the computations of covariance matrices and subsequent eigen-decomposition for the purpose of obtaining an orthogonal basis on which to project augmented variables. For large datasets (either containing too many nodes describing the cell and ECM or too many time samples) these computations may become time consuming or intractable. However, methods of down sampling followed by interpolation of final trajectory could alleviate this issue.

A key limitation within the theoretical framework is the inability to incorporate emergent interactions involving cell to cell mechanical contact in addition to cell/matrix interactions. This is because the causal relationships within the representative bond graph could not be resolved, since both the bidirectional energy exchange through contact with the ECM and the adjacent cell would have to be considered. However, if it was desired to study cell to cell mechanical contact in the absence of ECM, the proposed approach would still be valid. This type of approach may be relevant in the study of close-packed collective systems such as confinement-induced collective migration characterized by cell jamming in pathological situations such as asthma and cancer[40].

In addition, the theoretical framework may be limited the to the representation of extracellular matrices where small strains behave within the linear elastic regime[41]. This it is due to the linear superposition assumption for combining the ECM dynamics for individual cells.

Furthermore, with the current mathematical formulation, we are unable to reproduce degradation of the ECM through secretion of MMP. ECM degradation would be necessary to reproduce sustained movement and migration of the cells particularly in 3-D embedded matrices[42]. Since ECM degradation continuously changes the fiber connectivity through ECM remodeling, we would need to develop a methodology to update the grid structure describing the ECM field within our formulation. However ECM degradation may not be necessary to model gel compaction since whereas isolated cells migrate in a random manner, a higher density cluster of cells remains stationary when contracting the surrounding gel [10].

A large assumption we made within our approach was that the cell's internal adaptive response to mechanical cues could be modeled as the change in cell polarity due to mechanical cues. Cells change their internal state through a complex process of mechanotransduction and intracellular signaling[43]. Future work should incorporate these more complex mechanisms to for more accurate cell represntations.

Future Directions

5.2.1 Inclusion of More than one Phenotype to describe Model of Single cell

In the current model formulation, we assume only one phenotype is expressed through single-cell ECM interactions. This is in order to ensure that the model represents a time invariant system. In reality, the cells exhibits a wide range of dynamic phenotypes including spreading, migration, and differentiation during matrix-mediated emergent behaviors[4]. These dynamics cannot be explained by one reduced order latent variable model. Therefore it would be necessary to create a specific set of latent variables for each dynamic phenotype of an individual cell. In addition, rules would need to be developed to transition between the latent variables describing a specific phenotype. These rules should be based on the underlying subcellular mechanisms (potentially found through experimentation) since it is undesirable to ad-hoc rules.

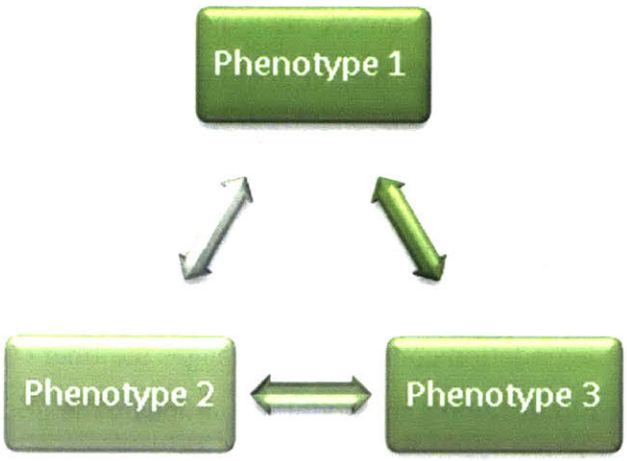


Figure 30: Diagram representing switching between different latent variable models describing multiple phenotypes and individual cell

5.2.2 Modeling Emergence using Pairwise Cell-Cell Mechanical interactions as Building Blocks

The current model assumes that the conditions (i.e. local ECM stiffness) experienced by a single cell in isolation is sufficient to produce the necessary responses for collective behaviors. However, an isolated cell cannot be exposed to the same local stiffness as in the case when more than one cell is present. Therefore, if 2-cell interactions where considered to be the initial subsystem for integration into a larger more complex system, more accurate and global behaviors could be reproduced. However, incorporation of 2 cells would create a larger dataset which may lead to the creation of large matrices that are difficult to handle computationally (as described in the limitations section). Furthermore, more complex nonlinear mechanisms would arise between two interacting cells which would require a larger number of latent variables to accurately represent the more complicated dynamics.

5.2.3 Inclusion of Experimental Data to Inform Model

In preliminary experiments (see appendix B) GFP tagged plasma membrane HUVEC were used to detect cell morphology and observe lamellipodial formation of 2 cells separated by 10-100um. Preliminary analysis measured polarity by calculating the principal axis of elongation along of the cell contour and the cell's movement direction between time frames. With future experiments, polarity metric data could be integrated into the current model in order to tune the polarity direction with respect to ECM properties and cell location. However, future experiments should consider a more relevant experimental set-up where intercellular signaling has been eliminated as a variable within the experiment in order to examine the mechanical interactions between the cells.

A. MATLAB Codes

A.1 Data Preprocessing

```
clear all; close all
tic
%%
Cc = 1e-3;
Ccort = 1e-3;
Ce = Cc;
%% Initialize Variables

%%cell state variables
%%velocities
cort1_node_vx = []; cort1_node_vy = []; cort1_node_vz = [];

cell1_node_vx = []; cell1_node_vy = []; cell1_node_vz = [];

%%positions
cell1_node_x = []; cell1_node_y = []; cell1_node_z = [];

%%forces
FE_node_x = []; FE_node_y = []; FE_node_z = [];
FL_node_x = []; FL_node_y = []; FL_node_z = [];
FT_node_x = []; FT_node_y = []; FT_node_z = [];

Ftract_x_node = [];  Ftract_y_node = []; Ftract_z_node = [];

F_L_x = []; F_L_y = []; F_L_z = [];

F_cort_x = []; F_cort_y = []; F_cort_z = [];

bondn_node = []; %d_AM_node =[];

%%polarity
Polarx = []; Polary = []; Polarz = [];
ds_max_x = []; ds_max_y = []; ds_max_z = [];

%% ECM state variables

ECM_node_x = []; ECM_node_y = []; ECM_node_z = [];

ECM_node_vx = []; ECM_node_vy = []; ECM_node_vz = [];

FEECM_x_node = [];  FEECM_z_node=[]; FEECM_y_node=[];
```

```
FtractECM_x_node = []; FtractECM_z_node=[]; FtractECM_y_node=[];
FD_x_node = []; FD_y_node = []; FD_z_node = [];
%%
T = [];
N_mem = 189;%%number of membrane nodes
%%Cell /ECM attachements
is_cell_attached = [];
grid_index_memnode = {};

cell1_node_x_attached =[];
cell1_node_y_attached =[];
cell1_node_z_attached =[];

x_closestECM = [];
y_closestECM = [];
z_closestECM = [];

for i = 1:N_mem+1
    grid_index_memnode{i} = zeros(1,3000);
end
%%
startRow = 3;

%% Import Files(A1)
TT = 1:3000;
% TT([[10:10:100]]) = [];

for time = TT
    %% Filopodia Coordinates
%     filename1 = strcat('C:\Users\Michaelle\Dropbox
(MIT)\MIT\Graduate\2.THG Research\data for
PLS\old_Data2\Cell_1_',num2str(time),'.000000_s.txt' );
% filename1 = strcat('C:\Users\Michaelle\Dropbox (MIT)\MIT\Graduate\2.THG
Research\Postprocess_new\Output2-
Jan142017\Cell_1_',num2str(time),'.000000_s.sav' );
    filename1 = strcat('C:\Users\Michaelle\Dropbox
(MIT)\MIT\Graduate\2.THG Research\PLS code for collective cell
migration2\Output12April22017cell24\Data\Cell_1_',num2str(time),'.000000_s
.sav' );
    N_mem = 189;%%number of membrane nodes
    [Px,Py,Pz] = importfile1(filename1, 1, 1);
    [dex,dey,dez] = importfile1(filename1, 2, 2);
    [i_mem, node_x, node_y,node_z,node_vx, node_vy,node_vz,FL_x,
FL_y,FL_z, FE_x, FE_y,FE_z, Ftract_x,Ftract_y, Ftract_z,n_b] =
importfile6b(filename1, 3, N_mem+2); %%node #, xyz positions, xyz FE
forces,xyz Ftract forces

%     filename2 = strcat('C:\Users\Michaelle\Dropbox
(MIT)\MIT\Graduate\2.THG Research\Postprocess_new\Output2-
Jan142017\Cortical_1_',num2str(time),'.000000_s.sav' );
    filename2 = strcat('C:\Users\Michaelle\Dropbox
(MIT)\MIT\Graduate\2.THG Research\PLS code for collective cell
migration2\Output12April22017cell24\Data\Cortical_1_',num2str(time),'.0000
00_s.sav' );
```

```
    [i_cort,nodet_vx, nodet_vy,nodet_vz,FT_x, FT_y,FT_z] =
importfile6a(filename2, 1, N_mem);

%    filename3 = strcat('C:\Users\Michaelle\Dropbox
(MIT)\MIT\Graduate\2.THG Research\Postprocess_new\Output2-
Jan142017\ECM_fiber_',num2str(time),'.000000_s.sav' );
   filename3 = strcat('C:\Users\Michaelle\Dropbox (MIT)\MIT\Graduate\2.THG
Research\PLS code for collective cell
migration2\Output12April22017cell24\Data\ECM_fiber_',num2str(time),'.00000
0_s.sav' );

%     N_CL = 6741;%6236; %1646; %%number of CL nodes
    N_ECM = importfile2(filename3,1,1);
    N_CL = 1949;%importfile2(filename3,1,1);
    [i_ECM_UM,i_ECM_M,ECMCL_x, ECMCL_y, ECMCL_z,ECMCL_vx, ECMCL_vy,
ECMCL_vz,FEE_x,FEE_y,FEE_z,FEtract_x,FEtract_y,FEtract_z,FD_x,FD_y,FD_z ]
= importfile6c(filename3, 2, N_CL+1); %% xyz positions, xyz FCL
forces,con. VEGF at each CL, integrity at each CL

 %%velocities
    cort1_node_vxt = [];cort1_node_vyt = [];cort1_node_vzt = [];
    cell1_node_vxt = [];cell1_node_vyt = [];cell1_node_vzt = [];
 %%positions
    cell1_node_xt = [];cell1_node_yt = [];cell1_node_zt = [];
 %%forces
    FL_node_xt = [];FL_node_yt = [];FL_node_zt = [];
    FT_node_xt = [];FT_node_yt = [];FT_node_zt = [];
    FE_node_xt = [];FE_node_yt = [];FE_node_zt = [];
    Ftract_xnodet = []; Ftract_ynodet= [];Ftract_znodet = [];
    bondn_nodet = [];

    %%ECM state variables
    ECM_node_xt = [];ECM_node_yt = [];ECM_node_zt = [];
    ECM_node_vxt = [];ECM_node_vyt = [];ECM_node_vzt = [];
    FEtract_xnodet = []; FEtract_ynodet= []; FEtract_znodet = [];
    FEE_x_nodet = [];  FEE_y_nodet = []; FEE_z_nodet = [];
    FD_xnodet = []; FD_ynodet = []; FD_znodet = [];

    index_unmapped = []; index_mapped = [];
    index_mem_node = [];

        parfor i = 1:N_mem
            cell1_node_xt = [cell1_node_xt,node_x(i)];
            cell1_node_yt = [cell1_node_yt,node_y(i)];
            cell1_node_zt = [cell1_node_zt,node_z(i)];

            cell1_node_vxt = [cell1_node_vxt,node_vx(i)];
            cell1_node_vyt = [cell1_node_vyt,node_vy(i)];
            cell1_node_vzt = [cell1_node_vzt,node_vz(i)];

            cort1_node_vxt = [cort1_node_vxt,nodet_vx(i)];
            cort1_node_vyt = [cort1_node_vyt,nodet_vy(i)];
            cort1_node_vzt = [cort1_node_vzt,nodet_vz(i)];

            FL_node_xt = [FL_node_xt ,FL_x(i)];
```

```
            FL_node_yt = [FL_node_yt,FL_y(i)];
            FL_node_zt = [FL_node_zt,FL_z(i)];

            FT_node_xt = [FT_node_xt ,FT_x(i)];
            FT_node_yt = [FT_node_yt,FT_y(i)];
            FT_node_zt = [FT_node_zt,FT_z(i)];

            FE_node_xt = [FE_node_xt ,FE_x(i)];
            FE_node_yt = [FE_node_yt,FE_y(i)];
            FE_node_zt = [FE_node_zt,FE_z(i)];

            Ftract_xnodet = [Ftract_xnodet,Ftract_x(i)];
            Ftract_ynodet = [Ftract_ynodet,Ftract_y(i)];
            Ftract_znodet = [Ftract_znodet,Ftract_z(i)];

            bondn_nodet = [bondn_nodet,n_b(i)];
            index_mem_node = [index_mem_node,i_mem];

        end
        %%Cell attachements
        i_cell_attach{time} = find(bondn_nodet > 0);
        is_cell_attached_t = zeros(1,length(i_mem));
        is_cell_attached_t(i_cell_attach{time}) = 1;



        parfor i = 1:N_CL
            index_unmapped = [index_unmapped,i_ECM_UM(i)];
            index_mapped = [index_mapped,i_ECM_M(i)];

            ECM_node_xt = [ECM_node_xt,ECMCL_x(i)];
            ECM_node_yt = [ECM_node_yt,ECMCL_y(i)];
            ECM_node_zt = [ECM_node_zt,ECMCL_z(i)];

            ECM_node_vxt = [ECM_node_vxt,ECMCL_vx(i)];
            ECM_node_vyt = [ECM_node_vyt,ECMCL_vy(i)];
            ECM_node_vzt = [ECM_node_vzt,ECMCL_vz(i)];

            FEE_x_nodet = [FEE_x_nodet ,FEE_x(i)];
            FEE_y_nodet = [FEE_y_nodet ,FEE_y(i)];
            FEE_z_nodet = [FEE_z_nodet ,FEE_z(i)];

            FEtract_xnodet = [FEtract_xnodet,FEtract_x(i)];
            FEtract_ynodet = [FEtract_ynodet,FEtract_y(i)];
            FEtract_znodet = [FEtract_znodet,FEtract_z(i)];

            FD_xnodet = [FD_xnodet,FD_x(i)];
            FD_ynodet = [FD_ynodet,FD_y(i)];
            FD_znodet = [FD_znodet,FD_z(i)];
        end
        %%ECM crosslink w/ nonzero traction force
        mag2_Ftract = FEtract_xnodet.*FEtract_xnodet +
FEtract_ynodet.*FEtract_ynodet + FEtract_znodet.*FEtract_znodet;
        j_ECM_nzTractF = find(mag2_Ftract);
```

```
        ECM_node_xt_tract = ECM_node_xt(j_ECM_nzTractF);
        ECM_node_yt_tract = ECM_node_yt(j_ECM_nzTractF);
        ECM_node_zt_tract = ECM_node_zt(j_ECM_nzTractF);

          %%calculate closest distance between attached membrance node and
ECM CL
          %%node
            cell1_node_xt_attached = zeros(1,N_mem) ;
            cell1_node_yt_attached = zeros(1,N_mem) ;
            cell1_node_zt_attached = zeros(1,N_mem) ;

            x_closestECMt = zeros(1,N_mem) ;
            y_closestECMt = zeros(1,N_mem) ;
            z_closestECMt = zeros(1,N_mem);

        for i = 1:length(i_cell_attach{time})
            i_attached = i_cell_attach{time}(i);
            cell1_node_xt_attached(i_attached) =
cell1_node_xt(i_attached);
            cell1_node_yt_attached(i_attached) =
cell1_node_yt(i_attached);
            cell1_node_zt_attached(i_attached) =
cell1_node_zt(i_attached);

            dx =
cell1_node_xt_attached(i_attached).*ones(size(ECM_node_xt_tract)) -
ECM_node_xt_tract;
            dy =
cell1_node_yt_attached(i_attached).*ones(size(ECM_node_yt_tract)) -
ECM_node_yt_tract;
            dz =
cell1_node_zt_attached(i_attached).*ones(size(ECM_node_zt_tract)) -
ECM_node_zt_tract;

            CL_memnode_distance = sqrt(dx.*dx + dy.*dy + dz.*dz);
            [m_d,grid_index_j] = min(CL_memnode_distance);
            %% ith membrane node  address(on ECM grid) = index of closest
ECM CL node
            grid_index_memnode_t = j_ECM_nzTractF(grid_index_j);

            x_closestECMt(i_attached) = ECM_node_xt_tract(grid_index_j);
            y_closestECMt(i_attached) = ECM_node_yt_tract(grid_index_j);
            z_closestECMt(i_attached) = ECM_node_zt_tract(grid_index_j);

            grid_index_memnode{i_attached}(time) = grid_index_memnode_t;
        end

       cell1_node_x_attached
=[cell1_node_x_attached;cell1_node_xt_attached];
       cell1_node_y_attached
=[cell1_node_y_attached;cell1_node_yt_attached];
       cell1_node_z_attached
=[cell1_node_z_attached;cell1_node_zt_attached];

       x_closestECM = [x_closestECM;x_closestECMt];
```

```
y_closestECM = [y_closestECM;y_closestECMt];
z_closestECM = [z_closestECM;z_closestECMt];

 Polarx = [Polarx;Px];
 Polary = [Polary;Py];
 Polarz = [Polarz;Pz];

 ds_max_x = [ds_max_x;dex];
 ds_max_y = [ds_max_y;dey];
 ds_max_z = [ds_max_z;dez];

 cell1_node_x = [cell1_node_x;cell1_node_xt];
 cell1_node_y = [cell1_node_y;cell1_node_yt];
 cell1_node_z = [cell1_node_z;cell1_node_zt];

 cell1_node_vx = [cell1_node_vx;cell1_node_vxt];
 cell1_node_vy = [cell1_node_vy;cell1_node_vyt];
 cell1_node_vz = [cell1_node_vz;cell1_node_vzt];

 cort1_node_vx = [cort1_node_vx;cort1_node_vxt];
 cort1_node_vy = [cort1_node_vy;cort1_node_vyt];
 cort1_node_vz = [cort1_node_vz;cort1_node_vzt];

 FE_node_x = [FE_node_x;FE_node_xt];
 FE_node_y = [FE_node_y;FE_node_yt];
 FE_node_z = [FE_node_z;FE_node_zt];

 FL_node_x = [FL_node_x;FL_node_xt];
 FL_node_y = [FL_node_y;FL_node_yt];
 FL_node_z = [FL_node_z;FL_node_zt];

 FT_node_x = [FT_node_x;FT_node_xt];
 FT_node_y = [FT_node_y;FT_node_yt];
 FT_node_z = [FT_node_z;FT_node_zt];

 bondn_node = [bondn_node;bondn_nodet];

 Ftract_x_node = [Ftract_x_node;Ftract_xnodet];
 Ftract_y_node = [Ftract_y_node;Ftract_ynodet];
 Ftract_z_node = [Ftract_z_node;Ftract_znodet];


 FtractECM_x_node = [FtractECM_x_node;FEtract_xnodet];
 FtractECM_y_node = [FtractECM_y_node;FEtract_ynodet];
 FtractECM_z_node = [FtractECM_z_node;FEtract_znodet];

 FD_x_node = [FD_x_node;FD_xnodet];
 FD_y_node = [FD_y_node;FD_ynodet];
 FD_z_node = [FD_z_node;FD_znodet];

 FEECM_x_node = [FEECM_x_node;FEE_x_nodet];
 FEECM_y_node = [FEECM_y_node;FEE_y_nodet];
 FEECM_z_node = [FEECM_z_node;FEE_z_nodet];
```

```
        ECM_node_x = [ECM_node_x;ECM_node_xt];
        ECM_node_y = [ECM_node_y;ECM_node_yt];
        ECM_node_z = [ECM_node_z;ECM_node_zt];

        ECM_node_vx = [ECM_node_vx;ECM_node_vxt];
        ECM_node_vy = [ECM_node_vy;ECM_node_vyt];
        ECM_node_vz = [ECM_node_vz;ECM_node_vzt];

        T = [T;time];

        %%which membrance nodes are attached
       is_cell_attached = [is_cell_attached;is_cell_attached_t];

end

%%

F_Eerrorx = Ce.* ECM_node_vx - (FtractECM_x_node + FEECM_x_node);

F_Eerrory = Ce.* ECM_node_vy - (FtractECM_y_node + FEECM_y_node);

F_Eerrorz = Ce.* ECM_node_vz - (FtractECM_z_node + FEECM_z_node);

sumFE = FtractECM_z_node + FEECM_z_node + F_Eerrorz;

figure(1)
plot(sumFE, Ce.* ECM_node_vz)
%%
save('Cell_f_dxec24','FtractECM_x_node','FtractECM_y_node','FtractECM_z_no
de',...

'FEECM_x_node','FEECM_y_node','FEECM_z_node','ECM_node_vx','ECM_node_vy',.
..
    'ECM_node_vz','F_Eerrorx','F_Eerrory','F_Eerrorz','-v7.3')
save('Cell_xec24','ECM_node_x','ECM_node_y','ECM_node_z','-v7.3')

%
save('Cell_f_dxe44','FtractECM_x_node','FtractECM_y_node','FtractECM_z_nod
e',...
%
'FEECM_x_node','FEECM_y_node','FEECM_z_node','ECM_node_vx','ECM_node_vy',.
..
%     'ECM_node_vz','F_Eerrorx','F_Eerrory','F_Eerrorz','-v7.3')
% save('Cell_xe44','ECM_node_x','ECM_node_y','ECM_node_z','-v7.3')

%Find cell node attachements
%%
F_errorx = (Ccort + Cc).* cell1_node_vx - (Ftract_x_node +  FE_node_x +
FL_node_x - FT_node_x + Ccort.*cort1_node_vx);

F_errory = (Ccort + Cc).* cell1_node_vy - (Ftract_y_node +  FE_node_y +
FL_node_y - FT_node_y + Ccort.*cort1_node_vy);
```

```
    F_errorz = (Ccort + Cc).* cell1_node_vz - (Ftract_z_node +  FE_node_z +
    FL_node_z - FT_node_z + Ccort.*cort1_node_vz);
    %%
    save('Cell_f_dxc24','Ftract_x_node','Ftract_y_node','Ftract_z_node','FE_no
    de_x','FE_node_y','FE_node_z','FL_node_x','FL_node_y','FL_node_z','FT_node
    _x','FT_node_y','FT_node_z',...

    'cort1_node_vx','cort1_node_vy','cort1_node_vz','cell1_node_vx','cell1_nod
    e_vy','cell1_node_vz','F_errorx','F_errory','F_errorz','-v7.3')
    save('Cell_xc24','cell1_node_x','cell1_node_y','cell1_node_z','-v7.3')

    %
    save('Cell_f_dx44','Ftract_x_node','Ftract_y_node','Ftract_z_node','FE_nod
    e_x','FE_node_y','FE_node_z','FL_node_x','FL_node_y','FL_node_z','FT_node_
    x','FT_node_y','FT_node_z',...
    %
    'cort1_node_vx','cort1_node_vy','cort1_node_vz','cell1_node_vx','cell1_nod
    e_vy','cell1_node_vz','F_errorx','F_errory','F_errorz','-v7.3')
    % save('Cell_x44','cell1_node_x','cell1_node_y','cell1_node_z','-v7.3')
    %%
    save('Cell_ECM_interfacec24','i_cell_attach','grid_index_memnode','-v7.3')

    save('Polarityc24', 'Polarx', 'Polary', 'Polarz','-v7.3')

    save('highest_stiff_c24', 'ds_max_x', 'ds_max_y', 'ds_max_z','-v7.3')

save('index_and_ECMnum12c24', 'index_unmapped','index_mapped','N_ECM','-
v7.3')
```

A.2 Orthogonal Transformations

```
clear all
close all
%%
Cc = 1e-3;
Ccort = 1e-3;
Ce = Cc;

tt = [];
%%
% load Cell_f_dx2
% load Cell_x2

f_dx1 = matfile('Cell_f_dx22');
xx1 = matfile('Cell_x22');

f_dx2 = matfile('Cell_f_dx44');
xx2 = matfile('Cell_x44');

time = 1:3000;
T = length(time);
dt = 1;
%%
```

```
cell1_node =
[[xx1.cell1_node_x;xx2.cell1_node_x],[xx1.cell1_node_y;xx2.cell1_node_y],[
xx1.cell1_node_z;xx2.cell1_node_z]];
cell1_node_v =
[[f_dx1.cell1_node_vx;f_dx2.cell1_node_vx],[f_dx1.cell1_node_vy;f_dx2.cell
1_node_vy],[f_dx1.cell1_node_vz;f_dx2.cell1_node_vz]];
Ftract_node =
[[f_dx1.Ftract_x_node;f_dx2.Ftract_x_node],[f_dx1.Ftract_y_node;f_dx2.Ftra
ct_y_node],[f_dx1.Ftract_z_node;f_dx2.Ftract_z_node]];
FE_node =
[[f_dx1.FE_node_x;f_dx2.FE_node_x],[f_dx1.FE_node_y;f_dx2.FE_node_y],[f_dx
1.FE_node_z;f_dx2.FE_node_z]];
FL_node =
[[f_dx1.FL_node_x;f_dx2.FL_node_x],[f_dx1.FL_node_y;f_dx2.FL_node_y],[f_dx
1.FL_node_z;f_dx2.FL_node_z]];
FT_node =
[[f_dx1.FT_node_x;f_dx2.FT_node_x],[f_dx1.FT_node_y;f_dx2.FT_node_y],[f_dx
1.FT_node_z;f_dx2.FT_node_z]];
F_cort =
[Ccort.*[f_dx1.cort1_node_vx;f_dx2.cort1_node_vx],Ccort.*[f_dx1.cort1_node
_vy;f_dx2.cort1_node_vy],Ccort.*[f_dx1.cort1_node_vz;f_dx2.cort1_node_vz]]
;
F_error =
[[f_dx1.F_errorx;f_dx2.F_errorx],[f_dx1.F_errory;f_dx2.F_errory],[f_dx1.F_
errorz;f_dx2.F_errorz]];

sumFF = Ftract_node+  FE_node + FL_node- FT_node+ F_cort + F_error;
F_error2c1 = (Ccort + Cc).*[cell1_node_v(1,:);diff(cell1_node(1:3000,:))]-
sumFF(1:3000,:);
F_error2c2 = (Ccort +
Cc).*[cell1_node_v(1+3000,:);diff(cell1_node(3000+1:end,:))]-
sumFF(1+3000:6000,:);
F_error2 = [F_error2c1;F_error2c2];
%
% load Cell_f_dxe2
% load Cell_xe2

f_dxe1 = matfile('Cell_f_dxe22');
xxe1 = matfile('Cell_xe22');

f_dxe2 = matfile('Cell_f_dxe44');
xxe2 = matfile('Cell_xe44');

ECM_node =
[[xxe1.ECM_node_x;xxe2.ECM_node_x],[xxe1.ECM_node_y;xxe2.ECM_node_y],[xxe1
.ECM_node_z;xxe2.ECM_node_z]];
ECM_node_v =
[[f_dxe1.ECM_node_vx;f_dxe2.ECM_node_vx],[f_dxe1.ECM_node_vy;f_dxe2.ECM_no
de_vy],[f_dxe1.ECM_node_vz;f_dxe2.ECM_node_vz]];
FtractECM_node=
[[f_dxe1.FtractECM_x_node;f_dxe2.FtractECM_x_node],[f_dxe1.FtractECM_y_nod
e;f_dxe2.FtractECM_y_node],[f_dxe1.FtractECM_z_node;f_dxe2.FtractECM_z_nod
e]];
```

```
FEECM_node =
[[f_dxe1.FEECM_x_node;f_dxe2.FEECM_x_node],[f_dxe1.FEECM_y_node;f_dxe2.FEE
CM_y_node],[f_dxe1.FEECM_z_node;f_dxe2.FEECM_z_node]];
F_Eerror =
[[f_dxe1.F_Eerrorx;f_dxe2.F_Eerrorx],[f_dxe1.F_Eerrory;f_dxe2.F_Eerrory],[
f_dxe1.F_Eerrorz;f_dxe2.F_Eerrorz]];

sumFFe = FtractECM_node+  FEECM_node + F_Eerror;
% FF_error2 = Ce.*[ECM_node_v(1,:);diff(ECM_node)]- sumFFe;

FF_error2c1 = (Ccort + Cc).*[ECM_node_v(1,:);diff(ECM_node(1:3000,:))]-
sumFFe(1:3000,:);
FF_error2c2 = (Ccort +
Cc).*[ECM_node_v(1+3000,:);diff(ECM_node(3000+1:end,:))]-
sumFFe(3000+1:6000,:);
FF_error2 = [FF_error2c1;FF_error2c2];

%%

%% intergrate with Data (xdot = dxdt)
% Y(:,1) =
cell1_node(1,:);%[cell1_node_x(1,:),cell1_node_y(1,:),cell1_node_z(1,:)] ;
% Ye(:,1) = ECM_node(1,:);%
%
% Y(:,1+3000) =
cell1_node(1+3000,:);%[cell1_node_x(1,:),cell1_node_y(1,:),cell1_node_z(1,
:)] ;
% Ye(:,1+3000) = ECM_node(1+3000,:);%
%
% Y1(:,1) = zeros(size(cell1_node(1,:)));
% Y2(:,1) = Y(:,1);
%
% Y1(:,1+3000) = zeros(size(cell1_node(1+3000,:)));
% Y2(:,1+3000) = Y(:,1+3000);
%
% Ye1(:,1) = zeros(size(Ye(:,1)));
% Ye2(:,1) = Ye(:,1);
%
% Ye1(:,1+3000) = zeros(size(Ye(:,1+3000)));
% Ye2(:,1+3000) = Ye(:,1+3000);
%
% for i = 2:T
%   ti = time(i-1);
%   yi_c1 = Y(:,i-1);
%   yi_c2 = Y(:,i-1+3000);
%
%   yi1_c1 = Y1(:,i-1);
%   yi2_c1 = Y2(:,i-1);
%
%   yi1_c2 = Y1(:,i-1+3000);
%   yi2_c2 = Y2(:,i-1+3000);
%
%
%   yei_c1 = Ye(:,i-1);
%   yei_c2 = Ye(:,i-1+3000);
%
```

```
%    yei1_c1 = Ye1(:,i-1);
%    yei2_c1 = Ye2(:,i-1);
%
%    yei1_c2 = Ye1(:,i-1+3000);
%    yei2_c2 = Ye2(:,i-1+3000);
%
%    sumF1_c1 = Ftract_node(i-1,:);
%    sumF1_c2 = Ftract_node(i-1+3000,:);
%
%    sumF2_c1 = FE_node(i-1,:) + FL_node(i-1,:) - FT_node(i-1,:)...
%        + F_cort(i-1,:) + F_error(i-1,:) +F_error2(i-1,:);
% %    sumF_c1 = sumF1_c1+sumF2_c1;
%
%    sumF2_c2 = FE_node(i-1+3000,:) + FL_node(i-1+3000,:) - FT_node(i-
1+3000,:)...
%        + F_cort(i-1+3000,:) + F_error(i-1+3000,:) +F_error2(i-1+3000,:);
% %    sumF_c2 = sumF1_c2+sumF2_c2;
%
%    sumFe1_c1 = FtractECM_node(i-1,:);
%    sumFe1_c2 = FtractECM_node(i-1+3000,:);
%
%
%    sumFe2_c1 =FEECM_node(i-1,:) + F_Eerror(i-1,:) + FF_error2(i-1,:);
%
%    sumFe_c1 = sumFe1_c1+sumFe2_c1;
%
%    sumFe2_c2 =FEECM_node(i-1+3000,:) + F_Eerror(i-1+3000,:) +
FF_error2(i-1+3000,:);
%
%    sumFe_c2 = sumFe1_c2+sumFe2_c2;
%
% %
%    dYdt1_c1(:,1) = (1/(Ccort + Cc)).*sumF1_c1;
%    dYdt2_c1(:,1) = (1/(Ccort + Cc)).*sumF2_c1;
%    dYdt_c1(:,1) = dYdt1_c1(:,1) + dYdt2_c1(:,1);
% %
%    dYdt1_c2(:,1) = (1/(Ccort + Cc)).*sumF1_c2;
%    dYdt2_c2(:,1) = (1/(Ccort + Cc)).*sumF2_c2;
% %    dYdt_c2(:,1) = dYdt1_c2(:,1) + dYdt2_c2(:,1);
%
%    dYedt1_c1(:,1) = (1/( Ce)).*sumFe1_c1;
%    dYedt2_c1(:,1) = (1/( Ce)).*sumFe2_c1;
%    dYedt_c1(:,1) = dYedt1_c1(:,1) + dYedt2_c1(:,1);
%
%    dYedt1_c2(:,1) = (1/( Ce)).*sumFe1_c2;
%    dYedt2_c2(:,1) = (1/( Ce)).*sumFe2_c2;
% %    dYedt_c2(:,1) = dYedt1_c2(:,1) + dYedt2_c2(:,1);
%
%    Y1(:,i) = yi1_c1  + dt*dYdt1_c1(:,1);
%    Y2(:,i) = yi2_c1  + dt*dYdt2_c1(:,1);
%    Y(:,i) = Y1(:,i) + Y2(:,i);
%
%    Y1(:,i+3000) = yi1_c2  + dt*dYdt1_c2(:,1);
%    Y2(:,i+3000) = yi2_c2  + dt*dYdt2_c2(:,1);
%    Y(:,i+3000) = Y1(:,i+3000) + Y2(:,i+3000);
%
%    Ye1(:,i) = yei1_c1  + dt*dYedt1_c1(:,1);
```

```
%    Ye2(:,i) = yei2_c1  + dt*dYedt2_c1(:,1);
%    Ye(:,i) = Ye1(:,i) + Ye2(:,i);
%
%    Ye1(:,i+3000) = yei1_c2  + dt*dYedt1_c2(:,1);
%    Ye2(:,i+3000) = yei2_c2  + dt*dYedt2_c2(:,1);
%    Ye(:,i+3000) = Ye1(:,i+3000) + Ye2(:,i+3000);
%    tt = [tt;ti];
% end
%
% cell1_node_FA = Y1';
% cell1_node_alpha = Y2';
% save('split_x1_13_FLPn','cell1_node_FA','cell1_node_alpha','-v7.3')
%
% ECM_node_FA = Ye1';
% ECM_node_alpha = Ye2';
% save('split_xe1_13_FLPn','ECM_node_FA','ECM_node_alpha','-v7.3')
%%
%
N_mem = size(cell1_node,2)./3;
N_CL = size(ECM_node,2)./3;
% load split_x
% load split_xe
%%
% figure(1)
% plot(Ye1(1:1000:end,1)+Ye2(1:1000:end,1),'o')
% hold on
% plot(ECM_node(1,1:1000:end))
%
% figure(2)
%
plot(Y1(100,3001:6000)'+Y2(100,3001:6000)',cell1_node(3001:6000,100),'o')
% hold on
% plot(cell1_node(3001:6000,100),cell1_node(3001:6000,100),'r')
% %
% figure(3)
% plot(Y(1:N_mem,2555)',cell1_node(2555,1:N_mem),'o')
% hold on
% plot(cell1_node(2555,1:N_mem),cell1_node(2555,1:N_mem),'r')
% %%
% figure(4)
% plot(Ye(1:N_CL,2555)',ECM_node(2555,1:N_CL),'o')
% hold on
% plot(ECM_node(2555,1:N_CL),ECM_node(2555,1:N_CL),'r')
% %
%% Form A matrix
% N_mem = size(xx.cell1_node_x,2);
% N_CL = size(xxe.ECM_node_x,2);
% %%
% %%node ix;
% % A_eta = zeros(3*N_mem, (3*N_mem)*6);
% % for i = 1:N_mem
% %     for j = 0:5
% %         A_eta(i,i + j*(3*N_mem)) = 1;
% %     end
% % end
% %
% % %%node iy;
```

```
% % for i = 1:N_mem
% %     for j = 0:5
% %         A_eta((i + N_mem), (i + N_mem) + j*(3*N_mem)) = 1;
% %     end
% % end
% %
% % %%node iz;
% % for i = 1:N_mem
% %     for j = 0:5
% %         A_eta((i + 2*N_mem), (i + 2*N_mem) + j*(3*N_mem)) = 1;
% %     end
% % end
% % %%
 A_eta = sparse([eye(3*N_mem,3*N_mem), eye(3*N_mem,3*N_mem)]) ;
 A_eta1 = sparse([eye(3*N_mem,3*N_mem)]) ;
 A_eta2 = sparse([eye(3*N_mem,3*N_mem)]) ;
% spy(A_eta)
%  A_etae = sparse([eye(3*N_CL,3*N_CL), eye(3*N_CL,3*N_CL)]) ;
 A_etae1 = sparse([eye(3*N_CL,3*N_CL)]) ;
 A_etae2 = sparse([eye(3*N_CL,3*N_CL)]) ;
% spy(A_etae)
A_etatot = sparse([A_eta1,zeros(3*N_mem,3*N_mem +
6*N_CL);zeros(3*N_mem,3*N_mem),A_eta2,zeros(3*N_mem,6*N_CL);zeros(3*N_CL,6
*N_mem),A_etae1,zeros(3*N_CL,3*N_CL);zeros(3*N_CL,6*N_mem+3*N_CL),A_etae2]
);%sparse([eye(3*N_mem+3*N_CL,3*N_mem),
eye(3*N_mem+3*N_CL,3*N_mem),eye(3*N_CL+3*N_mem,3*N_CL),
eye(3*N_CL+3*N_mem,3*N_CL)]);
%% form eta
% eta = (1/(Ccort + Cc)).*[Ftract_node,FE_node+FL_node-
FT_node+F_cort+F_error+F_error2];
eta1 = (1/(Ccort + Cc)).*[Ftract_node];
eta2 = (1/(Ccort + Cc)).*[FE_node-
FT_node+F_cort+F_error+F_error2];%+(1/(Ccort + Cc)).*FL_node;

% eta = eta./10e-10;
eta1 = eta1./10e-10;
eta2 = eta2./10e-10;

% etae = (1/Ce).*[FtractECM_node,FEECM_node+F_Eerror+FF_error2];
etae1 = (1/Ce).*[FtractECM_node];
etae2 = (1/Ce).*[FEECM_node+F_Eerror+FF_error2];

% etae = etae./10e-10;
etae1 = etae1./10e-10;
etae2 = etae2./10e-10;

% n_eta = size(eta,2);
n_eta1 = size(eta1,2);
n_eta2 = size(eta2,2);

% n_etae = size(etae,2);
n_etae1 = size(etae1,2);
n_etae2 = size(etae2,2);
n_etatot =size([eta1,eta2,etae1,etae2],2); %size([eta,etae],2);
%
```

```
% clear Ftract_node FE_node  FT_node F_cort F_error FtractECM_node
FEECM_node F_Eerror
%% intergrate with Data using A (xdot  = A*eta)
% % Y(:,1) = cell1_node(1,:)./10e-
10;%[cell1_node_x(1,:),cell1_node_y(1,:),cell1_node_z(1,:)] ;
% % Ye(:,1) = ECM_node(1,:)./10e-
10;%[ECM_node_x(1,:),ECM_node_y(1,:),ECM_node_z(1,:) ] ;
% %
% % Y1(:,1) =cell1_node_FA(1,:)./10e-10; %zeros(size(cell1_node(1,:)));
% % Y2(:,1) = cell1_node_alpha(1,:)./10e-10;
% %
% % Ye1(:,1) = ECM_node_FA(1,:)./10e-10;
% % Ye2(:,1) = ECM_node_alpha(1,:)./10e-10;
% %
% % Ytot(:,1) =[Y1(:,1);Y2(:,1);Ye1(:,1);Ye2(:,1)]; %[Y(:,1);Ye(:,1)];
% %
% % for i = 2:T
% %   ti = time(i-1);
% % %   yi = abs(Y(:,i-1));
% % %   yei = Ye(:,i-1);
% %   yyi =Ytot(:,i-1); %yi;yei];
% %
% % %   sumF = A_eta*eta(i-1,:)'; %Ftract_node(i-1,:)+  FE_node(i-1,:) +
FL_node(i-1,:) - FT_node(i-1,:) + F_cort(i-1,:) + F_error(i-1,:);
% % %   sumFe = A_etae*etae(i-1,:)';
% %
% %   etatot(i-1,:) = [eta1(i-1,:),eta2(i-1,:),etae1(i-1,:),etae2(i-
1,:)]';
% %   dYdttot(:,1) = A_etatot*etatot(i-1,:)';
% %
% % %   dYdt(:,1) = (1/(Ccort + Cc)).*sumF;
% % %   dYedt(:,1) = sumFe;
% %
% % %   Y(:,i) = abs(yi)  + dt*dYdt(:,1);
% % %   Ye(:,i) = yei  + dt*dYedt(:,1);
% %   Ytot(:,i) =yyi + dt*dYdttot(:,1);
% %
% %   tt = [tt;ti];
% % end
% % %
% % % plot(Ye(1:N_CL,2555).*10e-10, ECM_node(2555,1:N_CL))
% % % plot(Y2(3*N_mem+N_CL+1:3*N_mem+2*N_CL,2555).*10e-10,
ECM_node(2555,N_CL+1:2*N_CL))
% % %%
% % % figure(1)
% % % plot(Ye(1:1000:end,:),'o')
% % % hold on
% % % plot(ECM_node(1:end-1,1:1000:end))
% % %
% % figure(2)
% % plot(Ytot(1:3*N_mem,1:100:2000)'.*10e-
10,cell1_node_FA(1:100:2000,:),'o')
% % hold on
% % plot(cell1_node_FA,cell1_node_FA,'r')
%
%
% % figure(2)
```

```
% % plot(Ye',ECM_node,'o')
% % hold on
% % plot(ECM_node,ECM_node,'r')
%% PCA Analysis
xxprime = matfile('split_x1_13_FLPn');
xxeprime = matfile('split_xe1_13_FLPn');
% % %
x = [xxprime.cell1_node_FA(1:end-1,:),xxprime.cell1_node_alpha(1:end-1,:),
xxeprime.ECM_node_FA(1:end-1,:), xxeprime.ECM_node_alpha(1:end-1,:)]./10e-
10;%cell1_node(1:end-1,:);
% % %
n_x_FA = size(xxprime.cell1_node_FA(1:end-1,:),2);
n_x_alpha = size(xxprime.cell1_node_alpha(1:end-1,:),2);
n_xe_FA = size(xxeprime.ECM_node_FA(1:end-1,:),2);
n_xe_alpha = size(xxeprime.ECM_node_alpha(1:end-1,:),2);
%
n_xphi = size(ECM_node(1:end-1,:),2);
% %
clear cell1_node ECM_node
n_x = size(x,2);
% %
% % x_dot = [cell1_node_v(1:end-1,:),ECM_node_v(1:end-1,:)]./10e-10;
% %
clear cell1_node_v ECM_node_v
% %
% % eta_dot = diff(eta,1,1)./dt;
eta_dot1 = diff(eta1,1,1)./dt;
eta_dot2 = diff(eta2,1,1)./dt;
%
% % etae_dot = diff(etae,1,1)./dt;
etae_dot1 = diff(etae1,1,1)./dt;
etae_dot2 = diff(etae1,1,1)./dt;
% % [eta1(i-1,:),eta2(i-1,:),etae1(i-1,:),etae2(i-1,:)]
x_star = sparse([x,eta1(1:end-1,:),eta2(1:end-1,:),etae1(1:end-
1,:),etae2(1:end-1,:)]);
% %
% clear eta1 etae1 eta2 etae2 x
% % %
% [N, ncol_PCA] = size(x_star);
% [x_star0, mu, sigma] = zscore(x_star);
% x_star0 = x_star0.*repmat(sigma', [1, size(x_star0,1)])';
% % % % Cxx =(1/N)* x_star'*x_star;
% % % % save('covarXX2_3000','Cxx','-v7.3')
% tic
% Cxx0 =(1/N)* (x_star0')*x_star0;
% save('covarXX100_3000_split1_13FLP','Cxx0','-v7.3')
% comptime1 = toc;
% % % % %
% % % % % %%
%%
% m_covX = matfile('covarXX100_3000_split1_13FLP.mat');
% % % % % load covarXX20_3000_split12
% % % % %%
% % % % % %
% tic
% [Vpca,Dtot] = eigs(m_covX.Cxx0,100) ;
% comptime2 = toc;
```

```
% %%
% save('EVD_Cxxe_scale100_split13_FLP','Vpca','Dtot','-v7.3');
% % % % % % %%
% % % % % % % load EVD_Cxx0
% % % % % % % load EVD_Cxxe2
% % % % % % load EVD_Cxxe_scale12
% % load EVD_Cxxe_scale20_split12_FLP
%%
% mPCA = matfile('EVD_Cxxe_scale100_split13_FLP.mat');
% [lambdatot,sorti] = sort(diag(abs(mPCA.Dtot)),'descend');
% Dtot = diag(lambdatot);
% Vpca = mPCA.Vpca;
% Vpca = Vpca(:,sorti);
% % m = rank(Dtot);
% % figure(3)
% % bar(log(lambdatot))
% % title('Eigenvalues of Covariance Matrix');
% % ylabel('log \lambda_i')
% % % % % % % % %
% % % % % % % %%
% % % k = sum(lambdatot)./trace(m_covX.Cxx0);%sum(lambdatot)./trace(Cxx0);
% % % save('k20_split13_FLP','k')
% % % % % % % %%
% tic
% parfor i = 1:size(x_star0,1)
%     Zpca(i,:) = Vpca'*x_star0(i,:)';
% end
% % % % % % % %%
% eedot = [eta_dot1,eta_dot2,etae_dot1,etae_dot2];
% % % % % % %
% parfor i = 1:size(eedot,2)
%     K(:,i) = regress(eedot(:,i), Zpca);
% end
% % % % % % % % %
% % clear eta_dot etae_dot
% % % % % % % % % % %%
% % save('regress_coeffesplit-100LV_scale013_FLP','K');
% % comptime3 =toc;
% % % % % % % % % %
% load regress_coeffesplit-100LV_scale013_FLP
% % % % % % % % % % %%
% % % eta_dotpca =Zpca*K;
% % %%
% % % figure(4)
% % % plot(eta_dotpca(1:500:end),eedot(1:500:end),'o')
% % %%
% % % clear eedot eta_dotpca
% % % % %%
% V_x = Vpca(1:n_x,:);
% V_eta = Vpca(n_x+1:n_x+n_etatot,:);
% A = (V_x'*A_etatot*V_eta+V_eta'*K');
% % %
% B = V_x'*A_etatot*mu(n_x+1:n_x+n_etatot)';
% lambda_A = eig(A);
% % % % % %
% % % figure(5)
% % % c = -pi:.01:pi;
```

```
% % % cx = cos(c);
% % % cy = -sin(c);
% % %
% % % line(cx, cy, 'color', 'k');
% % % hold on
% % % plot(cx,zeros(size(cy)),'k--')
% % % hold on
% % % plot(zeros(size(cx)),cy,'k--')
% % % hold on
% % % title('Eigenvalues State of State Transition Matrix A')
% % %
plot(complex(lambda_A),'o','color','r','markersize',5,'linewidth',1);
% % % % % % %
% % % % % clear V_x V_eta A_etatot
% % % % % % %%
% I_x_FA = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_x_FA(1:n_x_FA,1:n_x_FA) = eye(n_x_FA,n_x_FA);
%
% I_x_alpha = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_x_alpha (n_x_FA+1:n_x_FA+n_x_alpha,n_x_FA+1:n_x_FA+n_x_alpha) =
eye(n_x_alpha,n_x_alpha);
%
% I_xe_FA  = sparse((zeros(n_x+n_etatot,n_x+n_etatot)));
% I_xe_FA  (n_x_FA+n_x_alpha+1:n_x_FA+n_x_alpha+n_xe_FA
,n_x_FA+n_x_alpha+1:n_x_FA+n_x_alpha+n_xe_FA ) = eye(n_xe_FA,n_xe_FA);
%
% I_xe_alpha  = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_xe_alpha
(n_x_FA+n_x_alpha+n_xe_FA+1:n_x_FA+n_x_alpha+n_xe_FA+n_xe_alpha,...
%     n_x_FA+n_x_alpha+n_xe_FA+1:n_x_FA+n_x_alpha+n_xe_FA+n_xe_alpha ) =
eye(n_xe_alpha,n_xe_alpha);
%
% I_eta_FA = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_eta_FA(n_x+1:n_x+n_eta1,n_x+1:n_x+n_eta1) = eye(n_eta1,n_eta1);
%
% I_eta_alpha = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_eta_alpha
(n_x+n_eta1+1:n_x+n_eta1+n_eta2,n_x+n_eta1+1:n_x+n_eta1+n_eta2) =
eye(n_eta2,n_eta2);
%
% I_etae_FA  = sparse((zeros(n_x+n_etatot,n_x+n_etatot)));
% I_etae_FA
(n_x+n_eta1+n_eta2+1:n_x+n_eta1+n_eta2+n_etae1,n_x+n_eta1+n_eta2+1:n_x+n_e
ta1+n_eta2+n_etae1 ) = eye(n_etae1,n_etae1);
%
% I_etae_alpha  = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_etae_alpha
(n_x+n_eta1+n_eta2+n_etae1+1:n_x+n_eta1+n_eta2+n_etae1+n_etae2,...
%     n_x+n_eta1+n_eta2+n_etae1+1:n_x+n_eta1+n_eta2+n_etae1+n_etae2 ) =
eye(n_etae2,n_etae2);
% % % % %
% % % % % % II = I_x_FA + I_x_alpha + I_xe_FA + I_xe_alpha + I_eta_FA +
I_eta_alpha + I_etae_FA + I_etae_alpha;
% % % % % % % spy(II)
% % % % % % I_xphi = zeros(n_x+n_etatot,n_x+n_etatot);
% % % % % % I_xphi(n_xa+1:n_x,n_xa+1:n_x) = eye(n_xphi,n_xphi);
% % % % % %
```

```
% % % % % % I_etaalpha = zeros(n_x+n_etatot,n_x+n_etatot);
% % % % % % I_etaalpha(n_x+1:n_x+n_eta,n_x+1:n_x+n_eta) =
eye(n_eta,n_eta);
% % % % % %
% % % % % % I_etaphi = zeros(n_x+n_etatot,n_x+n_etatot);
% % % % % %
I_etaphi(n_x+n_eta+1:n_x+n_eta+n_etae,n_x+n_eta+1:n_x+n_eta+n_etae) =
eye(n_etae,n_etae);
% % % % % %
% % % % % % I_alpha = I_xalpha + I_etaalpha;
% % % % % % I_phi = I_xphi + I_etaphi;
% % % % % %
% P_x_FA= Vpca'*I_x_FA*Vpca;
% P_x_alpha = Vpca'*I_x_alpha*Vpca;
% P_xe_FA = Vpca'*I_xe_FA*Vpca;
% P_xe_alpha = Vpca'*I_xe_alpha*Vpca;
% P_eta_FA = Vpca'*I_eta_FA*Vpca;
% P_eta_alpha = Vpca'*I_eta_alpha*Vpca;
% P_etae_FA = Vpca'*I_etae_FA*Vpca;
% P_etae_alpha = Vpca'*I_etae_alpha*Vpca;
% % % % % % % %
% % % %%
% tic
% Zhat(:,1) = Zpca(1+3000,:)';
% T = 3000;
% for i = 2:T-1
%     ti = time(i-1);
%     zi = Zhat(:,i-1);
% %     etahat(:,i-1) = V_eta*zi;
% %     eta_dothat(:,i-1) = K'*zi;
% %     z_FAi = P_x_FA*zi;
% %     z_alphai = P_x_alpha*zi;
% %     ze_FA = P_xe_FA*zi;
% %     ze_alpha = P_xe_alpha*zi;
% %     zeta_FA = P_eta_FA*zi;
% %     zeta_alpha = P_eta_alpha*zi;
% %     zetae_FA = P_etae_FA*zi ;
% %     zetae_alpha = P_etae_alpha*zi;
%
%     dZdt(:,1) = A*zi + B + V_x'*((1/(Ccort +
Cc)).*[zeros(n_x_FA,1);FL_node(i-
1+3000,:)';zeros(n_xe_FA+n_xe_alpha,1)]./10e-10);%V_x'*(A_eta*etahat(:,i-
1)) + V_eta'*eta_dothat(:,i-1);%
%     Zhat(:,i) = zi  + dt*dZdt(:,1);
%      tt = [tt;ti];
% end
% comptme3 = toc;
% %%
% figure(6)
% plot(Zhat(:,1:10:end),Zpca(1+3000:10:T-1+3000,:)','o')
% hold on
% plot(Zpca(1:T-1,:)',Zpca(1:T-1,:)')
%%
% xhat0 = (Vpca*Zhat);
% xhat = (Vpca*Zhat + repmat(mu', [1, size(x_star0(1:T-1,:),1)])).*10e-10;
x_star1 = x_star.*10e-10;
x_star1 = x_star1';
```

```
% % %%
% % figure(7)
% % % plot(xhat(1,1:2999))
% % % hold on
% % % plot(x_star1(1:2999,1),'o')
% % %
% % % plot(xhat(1:3*N_mem,2555))
% % % hold on
% % % plot(x_star1(2555,1:3*N_mem),'o')
% % % plot(xhat(1:3*N_mem,1:10:2999),x_star1(1:10:2999,1:3*N_mem)','o')
% % %
plot(xhat(3*N_mem+2*N_CL+1:3*N_mem+3*N_CL,1:10:2999),x_star1(1:10:2999,3*N
_mem+2*N_CL+1:3*N_mem+3*N_CL)','o')
% %
% % plot(xhat(:,1:100:2999),x_star1(3000+1:100:3000+2999,:)','o')
% % %
% % xhat = xhat';
% % x_star1 = x_star1';
% % x1_cell = xhat(1:n_x_FA,:) + xhat(n_x_FA+1:n_x_FA+n_x_alpha,:);
x1_cell =full( x_star1(1:n_x_FA,:) +
x_star1(n_x_FA+1:n_x_FA+n_x_alpha,:));
x1_cell = x1_cell';
x1_ECM = full(x_star1(n_x_FA+n_x_alpha+1:n_x_FA+n_x_alpha+n_xe_FA,:) +
x_star1(n_x_FA+n_x_alpha+n_xe_FA+1:n_x_FA+n_x_alpha+n_xe_FA+n_xe_alpha,:))
;%xhat(:,n_x_FA+n_x_alpha+1:n_x_FA+n_x_alpha+n_xe_FA) +
xhat(:,n_x_FA+n_x_alpha+n_xe_FA+1:n_x_FA+n_x_alpha+n_xe_FA+n_xe_alpha) ;
x1_ECM = x1_ECM';
v1_ECM = diff(x1_ECM);
```

A.3 Polarity Control

```
clear all
close all

%%
m_D = matfile('highest_stiff_22');
m_P = matfile('Polarity22');
kk = (1/60); %%s^-1
Pvhat = [];
Pvhat(1,:) = [ m_P.Polarx(1,1) ;m_P.Polary(1,1);m_P.Polarz(1,1)]';
P_v0 = [ m_P.Polarx(1,1) ;m_P.Polary(1,1);m_P.Polarz(1,1)]';
%%
for t = 1:3000-1

    P_x = m_P.Polarx(t,1) ;
    P_y = m_P.Polary(t,1) ;
    P_z = m_P.Polarz(t,1) ;

    P_v(t,:) = [P_x;P_y;P_z]';

    P_vr(t,:) = round(P_v(t,:),1,'significant');

    d_x = m_D.ds_max_x(t,1) ;
```

```
    d_y = m_D.ds_max_y(t,1) ;
    d_z = m_D.ds_max_z(t,1) ;

    d_v(t,:) = [d_x;d_y;d_z]';

    d_vn(t,:) = d_v(t,:)./norm(d_v(t,:));

%     d_vnr(t,:) = round(d_vn(t,:),3,'significant');
%     Pvhatr(t,:) = round(Pvhat(t,:),3,'significant');

%     ang_dv(t) = atan2d(norm(cross(P_v0,d_vn(t,:))),dot(P_v0,d_vn(t,:)));
%     ang_dv(t)=wrapTo360(ang_dv(t));
%     ang_dvr(t)=round(ang_dv(t));

    dPv(t,:)=  (Pvhat(t,:)*(Pvhat(t,:)'*d_vn(t,:)) -
d_vn(t,:)*(Pvhat(t,:)'*Pvhat(t,:))) ;
    Pvhat(t+1,:) = Pvhat(t,:) + kk.*dPv(t,:);
    Pvhat(t+1,:) = Pvhat(t+1,:)./norm(Pvhat(t+1,:));

end
figure(1)
plot(P_v,Pvhat(1:end-1,:))
%
% % save('Controller_realdat', 'd_v','P_v','d_vn','ang_dv','ang_dvr')
% % load Controller_realdat
%% Regress d_maxstiff
m_M = matfile('Modelled_100_1and24_FLP');
% m_dim = matfile('dimensions_modelled20_13_FLP');

% load Modelled_10_FLP2_1
load dimensions_modelled20_13_FLP

% N_mem = m_dim.N_mem;
Vpca = m_M.Vpca;
%%
mm_P = matfile('IandP_1and24_100');
%%
P_phi = (mm_P.P_etae_FA + mm_P.P_etae_alpha);%P_phi = Vpca'*I_phi*Vpca;
P_alpha= (eye(size(P_phi)) - P_phi);%P_alpha = Vpca'*I_alpha*Vpca;
%%
T = 2999;
x_star0 = m_M.x_star0;

for i = 1:T
    Zpca(i,:) = Vpca'*x_star0(i,:)';
end

Zphi = P_phi*Zpca';

sumx =zeros(3, size(x_star0,2));
sumx(1,1:N_mem) =(1/N_mem).*ones(1,N_mem);
sumx(2,N_mem+1:2*N_mem) = (1/N_mem).*ones(1,N_mem);
sumx(3,2*N_mem+1:3*N_mem) = (1/N_mem).*ones(1,N_mem);
```

```
V_c = sumx*Vpca;

Zc = (V_c'*V_c)*Zpca';

% d_vn = d_vn(1:end-1,:);

[d_v0, mud, sigmad] = zscore(d_vn);

[Zpca0, muZ, sigmaZ] = zscore(Zpca);

% ang_dvr = ang_dvr(1:end-1);
% %
% [ang_dvr0, mu, sigma] = zscore(ang_dvr);

%%

% [Kd,Kdint,r,rint,stats] = regress(ang_dvr0',
[ones(size(Zphi,2),1),Zpca0] ); %regress(d_v(:,i),
[ones(size(Zphi,2),1),Zphi',Zc'] ); %regress(d_v(:,i), [Zphi',Zc'] );


for i = 1:size(d_vn,2)

   [Kd(:,i),Kdint,r,rint,stats] =regress(d_vn(:,i),
[ones(size(Zphi,2),1),Zphi',Zc'] ); %regress(d_v(:,i),
[ones(size(Zphi,2),1),Zphi',Zc'] ); %regress(d_v(:,i), [Zphi',Zc'] );
   stats1(:,i) =  stats';
end

% YY = (d_v')*[Zphi',Zc'];
% XX= [Zphi',Zc']'*[Zphi',Zc'];
% Kd = YY*inv(XX);
%%

% ang_dvrpca = Kd'*[ones(size(Zphi,2),1),Zpca0]';
% ang_dvrpca = ang_dvrpca.*repmat(sigma', [1, size(ang_dvrpca,2)]) +
repmat(mu', [1, size(ang_dvrpca,2)]);
%

d_vpca = Kd'*[ones(size(Zphi,2),1),Zphi',Zc']';

% d_vpca = d_vpca.*repmat(sigmad', [1, size(d_vpca,2)]) + repmat(mud', [1,
size(d_vpca,2)]);
d_vpca = d_vpca';

%%
figure(2)

% plot(ang_dvr',ang_dvrpca,'o')
% hold on
% plot(ang_dvr',ang_dvr','--')
```

```
plot(d_vpca,d_vn,'o')
hold on
plot(d_vn,d_vn,'--')
%%
kk = (1/60); %%s^-1

PvhatZ(1,:) = P_v(1,:);
for t = 1:3000-1

%     P_x = m_P.Polarx(t,1) ;
%     P_y = m_P.Polary(t,1) ;
%     P_z = m_P.Polarz(t,1) ;

%     P_v(t,:) = [P_x;P_y;P_z]';

%     d_x = m_D.ds_max_x(t,1) ;
%     d_y = m_D.ds_max_y(t,1) ;
%     d_z = m_D.ds_max_z(t,1) ;

%     d_v(t,:) = [d_x;d_y;d_z]';

    d_vpca(t,:) = d_vpca(t,:)./norm(d_vpca(t,:));

    dPv(t,:)=  (PvhatZ(t,:)*(PvhatZ(t,:)'*d_vpca(t,:)) -
d_vpca(t,:)*(PvhatZ(t,:)'*PvhatZ(t,:))) ;
    PvhatZ(t+1,:) = PvhatZ(t,:) + kk.*dPv(t,:);
    PvhatZ(t+1,:) = PvhatZ(t+1,:)./norm(PvhatZ(t+1,:));
end
PvhatZr = round(PvhatZ(:,:),1,'significant');

figure(3)
plot(P_v,PvhatZ(1:end-1,:),'o')
%%

% m_angle =matfile('Potrude33new');
m_angle = matfile('Potrude22new'); %matfile('Potrude33');

P_v0 =P_v(1,:);

for t = 1:3000-1

    anghat(t) =
atan2d(norm(cross(P_v0,PvhatZr(t,:))),dot(P_v0,PvhatZr(t,:)));
    anghat(t)=wrapTo360(anghat(t));
    anghatd(t)=round(anghat(t));
    anghatdnew(t)=round(anghatd(t),1,'significant');

end

k_max = find(anghatd > max(m_angle.angd));
anghatd(k_max) = max(m_angle.angd);
%
[b,bint,r,rint,stats] = regress(anghatd',
[ones(size(anghatd')),m_angle.angd(:,1:end-1)'] );
```

```
figure(4)
 plot(anghat,'o')
 hold on
 plot(anghatd,'r-')
 hold on
 plot(m_angle.angd(:,1:end-1),'go')

 figure(5)
 plot(m_angle.angd(:,1:end-1),anghatd,'o')
 angdest = anghatd;
%%

save('Potrude1and24est_1_100LV2','angdest','Kd','P_v0','PvhatZr','P_v','P_
vr','-v7.3');
```

A.4 2-Cell Superposition Approach

```
clear all
close all
%%
Cc = 1e-3;
Ccort = 1e-3;
Ce = Cc;
% N_mem = 189;
%%
%%
load dimensions_model1ed20_13_FLP
load Cell_ECM_interfacec24
%%
T = 2000;
f_dx1 = matfile('Cell_f_dx22');
xx1 = matfile('Cell_x22');

f_dx2 = matfile('Cell_f_dxc24');
xx2 = matfile('Cell_xc24');

%%
% cell1_node =
[[xx1.cell1_node_x;xx2.cell1_node_x;xx3.cell1_node_x],[xx1.cell1_node_y;xx
2.cell1_node_y;xx3.cell1_node_y],[xx1.cell1_node_z;xx2.cell1_node_z;xx3.ce
ll1_node_z]];
% cell1_node_v =
[[f_dx1.cell1_node_vx;f_dx2.cell1_node_vx;f_dx3.cell1_node_vx],[f_dx1.cell
1_node_vy;f_dx2.cell1_node_vy;f_dx3.cell1_node_vy],[f_dx1.cell1_node_vz;f_
dx2.cell1_node_vz;f_dx3.cell1_node_vz]];
% Ftract_node =
[[f_dx1.Ftract_x_node;f_dx2.Ftract_x_node;f_dx3.Ftract_x_node],[f_dx1.Ftra
ct_y_node;f_dx2.Ftract_y_node;f_dx3.Ftract_y_node],[f_dx1.Ftract_z_node;f_
dx2.Ftract_z_node;f_dx3.Ftract_z_node]];
% FE_node =
[[f_dx1.FE_node_x;f_dx2.FE_node_x;f_dx3.FE_node_x],[f_dx1.FE_node_y;f_dx2.
FE_node_y;f_dx3.FE_node_y],[f_dx1.FE_node_z;f_dx2.FE_node_z;f_dx3.FE_node_
z]];
```

```
FL_node =
[[f_dx1.FL_node_x;f_dx2.FL_node_x],[f_dx1.FL_node_y;f_dx2.FL_node_y],[f_dx
1.FL_node_z;f_dx2.FL_node_z]];
% FT_node =
[[f_dx1.FT_node_x;f_dx2.FT_node_x;f_dx3.FT_node_x],[f_dx1.FT_node_y;f_dx2.
FT_node_y;f_dx3.FT_node_y],[f_dx1.FT_node_z;f_dx2.FT_node_z;f_dx3.FT_node_
z]];
% F_cort =
[Ccort.*[f_dx1.cort1_node_vx;f_dx2.cort1_node_vx;f_dx3.cort1_node_vx],Ccor
t.*[f_dx1.cort1_node_vy;f_dx2.cort1_node_vy;f_dx3.cort1_node_vy],Ccort.*[f
_dx1.cort1_node_vz;f_dx2.cort1_node_vz;f_dx3.cort1_node_vz]];
% F_error =
[[f_dx1.F_errorx;f_dx2.F_errorx;f_dx3.F_errorx],[f_dx1.F_errory;f_dx2.F_er
rory;f_dx3.F_errory],[f_dx1.F_errorz;f_dx2.F_errorz;f_dx3.F_errorz]];
%
% sumFF = Ftract_node+  FE_node + FL_node- FT_node+ F_cort + F_error;
% F_error2c1 = (Ccort +
Cc).*[cell1_node_v(1,:);diff(cell1_node(1:3000,:))]- sumFF(1:3000,:);
% F_error2c2 = (Ccort +
Cc).*[cell1_node_v(1+3000,:);diff(cell1_node(3000+1:6000,:))]-
sumFF(1+3000:6000,:);
% F_error2c3 = (Ccort +
Cc).*[cell1_node_v(1+2*3000,:);diff(cell1_node(2*3000+1:end,:))]-
sumFF(1+2*3000:6000+3000,:);
% F_error2 = [F_error2c1;F_error2c2;F_error2c3];

% f_dxe1 = matfile('Cell_f_dxe22');
% xxe1 = matfile('Cell_xe22');
%
% f_dxe2 = matfile('Cell_f_dxc24');
% xxe2 = matfile('Cell_xc24');
%
%
% ECM_node =
[[xxe1.ECM_node_x;xxe2.ECM_node_x;xxe3.ECM_node_x],[xxe1.ECM_node_y;xxe2.E
CM_node_y;xxe3.ECM_node_y],[xxe1.ECM_node_z;xxe2.ECM_node_z;xxe3.ECM_node_
z]];
% ECM_node_v =
[[f_dxe1.ECM_node_vx;f_dxe2.ECM_node_vx;f_dxe3.ECM_node_vx],[f_dxe1.ECM_no
de_vy;f_dxe2.ECM_node_vy;f_dxe3.ECM_node_vy],[f_dxe1.ECM_node_vz;f_dxe2.EC
M_node_vz;f_dxe3.ECM_node_vz]];
% FtractECM_node=
[[f_dxe1.FtractECM_x_node;f_dxe2.FtractECM_x_node;f_dxe3.FtractECM_x_node]
,[f_dxe1.FtractECM_y_node;f_dxe2.FtractECM_y_node;f_dxe3.FtractECM_y_node]
,[f_dxe1.FtractECM_z_node;f_dxe2.FtractECM_z_node;f_dxe3.FtractECM_z_node]
];
% FEECM_node =
[[f_dxe1.FEECM_x_node;f_dxe2.FEECM_x_node;f_dxe3.FEECM_x_node],[f_dxe1.FEE
CM_y_node;f_dxe2.FEECM_y_node;f_dxe3.FEECM_y_node],[f_dxe1.FEECM_z_node;f_
dxe2.FEECM_z_node;f_dxe3.FEECM_z_node]];
% F_Eerror =
[[f_dxe1.F_Eerrorx;f_dxe2.F_Eerrorx;f_dxe3.F_Eerrorx],[f_dxe1.F_Eerrory;f_
dxe2.F_Eerrory;f_dxe3.F_Eerrory],[f_dxe1.F_Eerrorz;f_dxe2.F_Eerrorz;f_dxe3
.F_Eerrorz]];
%
% sumFFe = FtractECM_node+  FEECM_node + F_Eerror;
% % FF_error2 = Ce.*[ECM_node_v(1,:);diff(ECM_node)]- sumFFe;
```

```
%
% FF_error2c1 = (Ccort + Cc).*[ECM_node_v(1,:);diff(ECM_node(1:3000,:))]-
sumFFe(1:3000,:);
% FF_error2c2 = (Ccort +
Cc).*[ECM_node_v(1+3000,:);diff(ECM_node(3000+1:6000,:))]-
sumFFe(3000+1:6000,:);
% FF_error2c3 = (Ccort +
Cc).*[ECM_node_v(1+2*3000,:);diff(ECM_node(2*3000+1:end,:))]-
sumFFe(2*3000+1:6000+3000,:);
% FF_error2 = [FF_error2c1;FF_error2c2;FF_error2c3];
%
%%
m_Model = matfile('Modelled_100_1and24_FLP.mat');
% x_star = m_Model.x_star1;
% [x,eta(1:end-1,:),etae(1:end-1,:)]
% etja = [Ftract_node,FE_node+FL_node-FT_node+F_cort+F_error+F_error2];
% etae = [FtractECM_node,FEECM_node+F_Eerror+FF_error2];
% tic
% parfor time = 1:T-1
%     I_phitochi1 = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
%     I_phitochi2 = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% %     I_chitochi = zeros(n_x+n_etatot,n_x+n_etatot);
%     for i = 1:length(i_cell_attach{time})    %%name/index of attached mem
node
%         i_attached = i_cell_attach{time}(i);
%         ii = grid_index_memnode{i_attached}(time);
%         I_phitochi1(i_attached,n_x_FA+n_x_alpha + ii) = 1;
%         I_phitochi2(n_x+i_attached,n_x+n_eta1+n_eta2 + ii) = 1;
% %         I_chitochi(n_x+i_attached,n_x+i_attached) = 1;
%     end
%    II_chi_phi1{time} = sparse(I_phitochi1);
%    II_chi_phi2{time} = sparse(I_phitochi2);
% %    II_chi_chi{time} = sparse(I_chitochi);
% %    Chi_phi1(time,:) =II_chi_phi1{time}*x_star(time,:)';
% %    Chi_phi2(time,:) =II_chi_phi2{time}*x_star(time,:)';
% %    Chi_chi(time,:) =II_chi_chi{time}*x_star(time,:)';
% end
% comptime = toc;
% save('Projectionsc24','II_chi_phi1','II_chi_phi2','-v7.3')

% m_Profj = matfile('Projectionsc24.mat');
m_Profj = matfile('Projections24L.mat');

%%
% I_x_FA = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_x_FA(1:n_x_FA,1:n_x_FA) = eye(n_x_FA,n_x_FA);

% I_x_alpha = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_x_alpha (n_x_FA+1:n_x_FA+n_x_alpha,n_x_FA+1:n_x_FA+n_x_alpha) =
eye(n_x_alpha,n_x_alpha);
%
% I_xe_FA  = sparse((zeros(n_x+n_etatot,n_x+n_etatot)));
% I_xe_FA  (n_x_FA+n_x_alpha+1:n_x_FA+n_x_alpha+n_xe_FA
,n_x_FA+n_x_alpha+1:n_x_FA+n_x_alpha+n_xe_FA ) = eye(n_xe_FA,n_xe_FA);

% I_xe_alpha  = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
```

```
% I_xe_alpha
(n_x_FA+n_x_alpha+n_xe_FA+1:n_x_FA+n_x_alpha+n_xe_FA+n_xe_alpha,...
%     n_x_FA+n_x_alpha+n_xe_FA+1:n_x_FA+n_x_alpha+n_xe_FA+n_xe_alpha ) =
eye(n_xe_alpha,n_xe_alpha);

% I_eta_FA = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_eta_FA(n_x+1:n_x+n_eta1,n_x+1:n_x+n_eta1) = eye(n_eta1,n_eta1);
%
% I_eta_alpha = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_eta_alpha
(n_x+n_eta1+1:n_x+n_eta1+n_eta2,n_x+n_eta1+1:n_x+n_eta1+n_eta2) =
eye(n_eta2,n_eta2);
%
% I_etae_FA  = sparse((zeros(n_x+n_etatot,n_x+n_etatot)));
% I_etae_FA
(n_x+n_eta1+n_eta2+1:n_x+n_eta1+n_eta2+n_etae1,n_x+n_eta1+n_eta2+1:n_x+n_e
ta1+n_eta2+n_etae1 ) = eye(n_etae1,n_etae1);
%
% I_etae_alpha  = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_etae_alpha
(n_x+n_eta1+n_eta2+n_etae1+1:n_x+n_eta1+n_eta2+n_etae1+n_etae2,...
%     n_x+n_eta1+n_eta2+n_etae1+1:n_x+n_eta1+n_eta2+n_etae1+n_etae2 ) =
eye(n_etae2,n_etae2);

% x_FA = x_star*I_x_FA;
%%
% plot(-(Ce/(Cc + Ccort)).*Chi_phi1(:,1:n_x),x_FA(1:999,1:n_x),'o')
% hold on
% plot(x_FA(1:999,1:n_x),x_FA(1:999,1:n_x))
%%
% plot(-(Ce/(Cc +
Ccort)).*Chi_phi1(50,i_cell_attach{50}),x_FA(50,i_cell_attach{50}),'o')
% hold on
% plot(x_FA(50,i_cell_attach{50}),x_FA(50,i_cell_attach{50}))
%%

%
% % plot(-(Ce/(Cc + Ccort)).*Chi_phi2(500,n_x+i_cell_attach{500}),
eta_FA(500,n_x + i_cell_attach{500}),'o')
% % hold on
% % plot(eta_FA(500,n_x+i_cell_attach{500}),eta_FA(500,n_x +
i_cell_attach{500}))
% % load Modelled_10
Vpca = m_Model.Vpca;
V_x = Vpca(1:n_x,:);

clear Vpca
mm_P = matfile('IandP_1and24_100.mat');
tt = [];
time = 1:2000;
T = 2000;
dt = 1;
Zhat(:,1) = m_Model.Zpca(1,:)';
II_chi_phi2 = m_Profj.II_chi_phi2;
II_chi_phi1 = m_Profj.II_chi_phi1;
```

```
Pphi = (mm_P.P_xe_FA + mm_P.P_etae_FA + mm_P.P_xe_alpha+
mm_P.P_etae_alpha);
Palpha = (eye(size(Pphi)) - Pphi);
tic
for i = 2:T-1
    ti = time(i-1);
    zi = Zhat(:,i-1);

%     z_alphai = mm_P.P_x_alpha*zi;
%     ze_FA = mm_P.P_xe_FA*zi;
%     ze_alpha = mm_P.P_xe_alpha*zi;
%     zeta_alpha = mm_P.P_eta_alpha*zi;

    z_alpha = Palpha*zi;
%     zetae_FA = mm_P.P_etae_FA*zi ;
%     zetae_alpha = mm_P.P_etae_alpha*zi;
    zphi = Pphi*zi;%zetae_FA + ze_FA + ze_alpha+ zetae_alpha;
    H1 = ((m_Model.Vpca)'*(-(Ce/(Cc +
Ccort)).*II_chi_phi1{ti})*m_Model.Vpca);
    H2 = ((m_Model.Vpca)'*(-(Ce/(Cc +
Ccort)).*II_chi_phi2{ti})*m_Model.Vpca);
    z_FAi = H1*zphi;%mm_P.P_x_FA*zi;
    zeta_FA = H2*zphi;%%mm_P.P_eta_FA*zi;



%     etahat(:,i-1) = V_eta*zi;
%     eta_dothat(:,i-1) = K'*zi;
    dZdt(:,1) = m_Model.A*Palpha*zi +m_Model.A*(H2+eye(100,100))*zphi ...
        + m_Model.B + V_x'*((1/(Ccort + Cc)).*[zeros(n_x_FA,1);FL_node(i-
1,:)';zeros(n_xe_FA+n_xe_alpha,1)]./10e-10);%A*zi;%V_x'*(A_eta*etahat(:,i-
1)) + V_eta'*eta_dothat(:,i-1);
    Zhat(:,i) = zi  + dt*dZdt(:,1);
     tt = [tt;ti];

     HH1(:,:,i) = H1;
     HH2(:,:,i) = H2;
end
save('Hmap_cell1_1and24_100','HH1','HH2','-v7.3')
comptime2 = toc;
```

A.5 10-Cell Superposition Approach

```
clear all
close all
%%
Cc = 1e-3;
Ccort = 1e-3;
Ce = Cc;
% N_mem = 189;
%%
%%
load dimensions_model1ed20_13_FLP
```

```
% load Cell_ECM_interfacec24
%%
T = 2000;
% f_dx1 = matfile('Cell_f_dx22');
% xx1 = matfile('Cell_x22');

% f_dx2 = matfile('Cell_f_dxc24');
% xx2 = matfile('Cell_xc24');

%%
% cell1_node =
[[xx1.cell1_node_x;xx2.cell1_node_x;xx3.cell1_node_x],[xx1.cell1_node_y;xx
2.cell1_node_y;xx3.cell1_node_y],[xx1.cell1_node_z;xx2.cell1_node_z;xx3.ce
ll1_node_z]];
% cell1_node_v =
[[f_dx1.cell1_node_vx;f_dx2.cell1_node_vx;f_dx3.cell1_node_vx],[f_dx1.cell
1_node_vy;f_dx2.cell1_node_vy;f_dx3.cell1_node_vy],[f_dx1.cell1_node_vz;f_
dx2.cell1_node_vz;f_dx3.cell1_node_vz]];
% Ftract_node =
[[f_dx1.Ftract_x_node;f_dx2.Ftract_x_node;f_dx3.Ftract_x_node],[f_dx1.Ftra
ct_y_node;f_dx2.Ftract_y_node;f_dx3.Ftract_y_node],[f_dx1.Ftract_z_node;f_
dx2.Ftract_z_node;f_dx3.Ftract_z_node]];
% FE_node =
[[f_dx1.FE_node_x;f_dx2.FE_node_x;f_dx3.FE_node_x],[f_dx1.FE_node_y;f_dx2.
FE_node_y;f_dx3.FE_node_y],[f_dx1.FE_node_z;f_dx2.FE_node_z;f_dx3.FE_node_
z]];
% FL_node =
[[f_dx1.FL_node_x;f_dx2.FL_node_x],[f_dx1.FL_node_y;f_dx2.FL_node_y],[f_dx
1.FL_node_z;f_dx2.FL_node_z]];
% FT_node =
[[f_dx1.FT_node_x;f_dx2.FT_node_x;f_dx3.FT_node_x],[f_dx1.FT_node_y;f_dx2.
FT_node_y;f_dx3.FT_node_y],[f_dx1.FT_node_z;f_dx2.FT_node_z;f_dx3.FT_node_
z]];
% F_cort =
[Ccort.*[f_dx1.cort1_node_vx;f_dx2.cort1_node_vx;f_dx3.cort1_node_vx],Ccor
t.*[f_dx1.cort1_node_vy;f_dx2.cort1_node_vy;f_dx3.cort1_node_vy],Ccort.*[f
_dx1.cort1_node_vz;f_dx2.cort1_node_vz;f_dx3.cort1_node_vz]];
% F_error =
[[f_dx1.F_errorx;f_dx2.F_errorx;f_dx3.F_errorx],[f_dx1.F_errory;f_dx2.F_er
rory;f_dx3.F_errory],[f_dx1.F_errorz;f_dx2.F_errorz;f_dx3.F_errorz]];
%
% sumFF = Ftract_node+  FE_node + FL_node- FT_node+ F_cort + F_error;
% F_error2c1 = (Ccort +
Cc).*[cell1_node_v(1,:);diff(cell1_node(1:3000,:))]- sumFF(1:3000,:);
% F_error2c2 = (Ccort +
Cc).*[cell1_node_v(1+3000,:);diff(cell1_node(3000+1:6000,:))]-
sumFF(1+3000:6000,:);
% F_error2c3 = (Ccort +
Cc).*[cell1_node_v(1+2*3000,:);diff(cell1_node(2*3000+1:end,:))]-
sumFF(1+2*3000:6000+3000,:);
% F_error2 = [F_error2c1;F_error2c2;F_error2c3];

% f_dxe1 = matfile('Cell_f_dxe22');
% xxe1 = matfile('Cell_xe22');
%
% f_dxe2 = matfile('Cell_f_dxc24');
```

```
% xxe2 = matfile('Cell_xc24');
%
%
% ECM_node =
[[xxe1.ECM_node_x;xxe2.ECM_node_x;xxe3.ECM_node_x],[xxe1.ECM_node_y;xxe2.E
CM_node_y;xxe3.ECM_node_y],[xxe1.ECM_node_z;xxe2.ECM_node_z;xxe3.ECM_node_
z]];
% ECM_node_v =
[[f_dxe1.ECM_node_vx;f_dxe2.ECM_node_vx;f_dxe3.ECM_node_vx],[f_dxe1.ECM_no
de_vy;f_dxe2.ECM_node_vy;f_dxe3.ECM_node_vy],[f_dxe1.ECM_node_vz;f_dxe2.EC
M_node_vz;f_dxe3.ECM_node_vz]];
% FtractECM_node=
[[f_dxe1.FtractECM_x_node;f_dxe2.FtractECM_x_node;f_dxe3.FtractECM_x_node]
,[f_dxe1.FtractECM_y_node;f_dxe2.FtractECM_y_node;f_dxe3.FtractECM_y_node]
,[f_dxe1.FtractECM_z_node;f_dxe2.FtractECM_z_node;f_dxe3.FtractECM_z_node]
];
% FEECM_node =
[[f_dxe1.FEECM_x_node;f_dxe2.FEECM_x_node;f_dxe3.FEECM_x_node],[f_dxe1.FEE
CM_y_node;f_dxe2.FEECM_y_node;f_dxe3.FEECM_y_node],[f_dxe1.FEECM_z_node;f_
dxe2.FEECM_z_node;f_dxe3.FEECM_z_node]];
% F_Eerror =
[[f_dxe1.F_Eerrorx;f_dxe2.F_Eerrorx;f_dxe3.F_Eerrorx],[f_dxe1.F_Eerrory;f_
dxe2.F_Eerrory;f_dxe3.F_Eerrory],[f_dxe1.F_Eerrorz;f_dxe2.F_Eerrorz;f_dxe3
.F_Eerrorz]];
%
% sumFFe = FtractECM_node+  FEECM_node + F_Eerror;
% % FF_error2 = Ce.*[ECM_node_v(1,:);diff(ECM_node)]- sumFFe;
%
% FF_error2c1 = (Ccort + Cc).*[ECM_node_v(1,:);diff(ECM_node(1:3000,:))]-
sumFFe(1:3000,:);
% FF_error2c2 = (Ccort +
Cc).*[ECM_node_v(1+3000,:);diff(ECM_node(3000+1:6000,:))]-
sumFFe(3000+1:6000,:);
% FF_error2c3 = (Ccort +
Cc).*[ECM_node_v(1+2*3000,:);diff(ECM_node(2*3000+1:end,:))]-
sumFFe(2*3000+1:6000+3000,:);
% FF_error2 = [FF_error2c1;FF_error2c2;FF_error2c3];
%
%%
load FLnode_10cells
m_Model = matfile('Modelled_100_10cells_FLP.mat');
% x_star = m_Model.x_star1;
% [x,eta(1:end-1,:),etae(1:end-1,:)]
% etja = [Ftract_node,FE_node+FL_node-FT_node+F_cort+F_error+F_error2];
% etae = [FtractECM_node,FEECM_node+F_Eerror+FF_error2];
% tic
% parfor time = 1:T-1
%     I_phitochi1 = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
%     I_phitochi2 = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% %     I_chitochi = zeros(n_x+n_etatot,n_x+n_etatot);
%     for i = 1:length(i_cell_attach{time})    %%name/index of attached mem
node
%         i_attached = i_cell_attach{time}(i);
%         ii = grid_index_memnode{i_attached}(time);
%         I_phitochi1(i_attached,n_x_FA+n_x_alpha + ii) = 1;
%         I_phitochi2(n_x+i_attached,n_x+n_eta1+n_eta2 + ii) = 1;
% %         I_chitochi(n_x+i_attached,n_x+i_attached) = 1;
```

```
%     end
%    II_chi_phi1{time} = sparse(I_phitochi1);
%    II_chi_phi2{time} = sparse(I_phitochi2);
% %    II_chi_chi{time} = sparse(I_chitochi);
% %    Chi_phi1(time,:) =II_chi_phi1{time}*x_star(time,:)';
% %    Chi_phi2(time,:) =II_chi_phi2{time}*x_star(time,:)';
% %    Chi_chi(time,:) =II_chi_chi{time}*x_star(time,:)';
% end
% comptime = toc;
% save('Projectionsc24','II_chi_phi1','II_chi_phi2','-v7.3')

% m_Profj = matfile('Projectionsc24.mat');
m_Profj = matfile('Projections24L.mat');

%%
% I_x_FA = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_x_FA(1:n_x_FA,1:n_x_FA) = eye(n_x_FA,n_x_FA);

% I_x_alpha = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_x_alpha (n_x_FA+1:n_x_FA+n_x_alpha,n_x_FA+1:n_x_FA+n_x_alpha) =
eye(n_x_alpha,n_x_alpha);
%
% I_xe_FA  = sparse((zeros(n_x+n_etatot,n_x+n_etatot)));
% I_xe_FA  (n_x_FA+n_x_alpha+1:n_x_FA+n_x_alpha+n_xe_FA
,n_x_FA+n_x_alpha+1:n_x_FA+n_x_alpha+n_xe_FA ) = eye(n_xe_FA,n_xe_FA);

% I_xe_alpha  = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_xe_alpha
(n_x_FA+n_x_alpha+n_xe_FA+1:n_x_FA+n_x_alpha+n_xe_FA+n_xe_alpha,...
%     n_x_FA+n_x_alpha+n_xe_FA+1:n_x_FA+n_x_alpha+n_xe_FA+n_xe_alpha ) =
eye(n_xe_alpha,n_xe_alpha);

% I_eta_FA = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_eta_FA(n_x+1:n_x+n_eta1,n_x+1:n_x+n_eta1) = eye(n_eta1,n_eta1);
%
% I_eta_alpha = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_eta_alpha
(n_x+n_eta1+1:n_x+n_eta1+n_eta2,n_x+n_eta1+1:n_x+n_eta1+n_eta2) =
eye(n_eta2,n_eta2);
%
% I_etae_FA  = sparse((zeros(n_x+n_etatot,n_x+n_etatot)));
% I_etae_FA
(n_x+n_eta1+n_eta2+1:n_x+n_eta1+n_eta2+n_etae1,n_x+n_eta1+n_eta2+1:n_x+n_e
ta1+n_eta2+n_etae1 ) = eye(n_etae1,n_etae1);
%
% I_etae_alpha  = sparse(zeros(n_x+n_etatot,n_x+n_etatot));
% I_etae_alpha
(n_x+n_eta1+n_eta2+n_etae1+1:n_x+n_eta1+n_eta2+n_etae1+n_etae2,...
%     n_x+n_eta1+n_eta2+n_etae1+1:n_x+n_eta1+n_eta2+n_etae1+n_etae2 ) =
eye(n_etae2,n_etae2);

% x_FA = x_star*I_x_FA;
%%
% plot(-(Ce/(Cc + Ccort)).*Chi_phi1(:,1:n_x),x_FA(1:999,1:n_x),'o')
% hold on
% plot(x_FA(1:999,1:n_x),x_FA(1:999,1:n_x))
```

```
%%
% plot(-(Ce/(Cc +
Ccort)).*Chi_phi1(50,i_cell_attach{50}),x_FA(50,i_cell_attach{50}),'o')
% hold on
% plot(x_FA(50,i_cell_attach{50}),x_FA(50,i_cell_attach{50}))
%%

%
% % plot(-(Ce/(Cc + Ccort)).*Chi_phi2(500,n_x+i_cell_attach{500}),
eta_FA(500,n_x + i_cell_attach{500}),'o')
% % hold on
% % plot(eta_FA(500,n_x+i_cell_attach{500}),eta_FA(500,n_x +
i_cell_attach{500}))
% % load Modelled_10
Vpca = m_Model.Vpca;
V_x = Vpca(1:n_x,:);

clear Vpca
mm_P = matfile('IandP_1and24_100.mat');
tt = [];
time = 1:2000;
T = 2000;
dt = 1;
Zhat(:,1) = m_Model.Zpca(1,:)';
II_chi_phi2 = m_Profj.II_chi_phi2;
II_chi_phi1 = m_Profj.II_chi_phi1;
Pphi = (mm_P.P_xe_FA + mm_P.P_etae_FA + mm_P.P_xe_alpha+
mm_P.P_etae_alpha);
Palpha = (eye(size(Pphi)) - Pphi);
tic
for i = 2:T-1
    ti = time(i-1);
    zi = Zhat(:,i-1);

%     z_alphai = mm_P.P_x_alpha*zi;
%     ze_FA = mm_P.P_xe_FA*zi;
%     ze_alpha = mm_P.P_xe_alpha*zi;
%     zeta_alpha = mm_P.P_eta_alpha*zi;

    z_alpha = Palpha*zi;
%     zetae_FA = mm_P.P_etae_FA*zi ;
%     zetae_alpha = mm_P.P_etae_alpha*zi;
    zphi = Pphi*zi;%zetae_FA + ze_FA + ze_alpha+ zetae_alpha;
    H1 = ((m_Model.Vpca)'*(-(Ce/(Cc +
Ccort)).*II_chi_phi1{ti})*m_Model.Vpca);
    H2 = ((m_Model.Vpca)'*(-(Ce/(Cc +
Ccort)).*II_chi_phi2{ti})*m_Model.Vpca);
    z_FAi = H1*zphi;%mm_P.P_x_FA*zi;
    zeta_FA = H2*zphi;%%mm_P.P_eta_FA*zi;



%     etahat(:,i-1) = V_eta*zi;
%     eta_dothat(:,i-1) = K'*zi;
    dZdt(:,1) = m_Model.A*Palpha*zi +m_Model.A*(H2+eye(100,100))*zphi ...
```

```
        + m_Model.B + V_x'*((1/(Ccort + Cc)).*[zeros(n_x_FA,1);FL_node(i-
1,:)';zeros(n_xe_FA+n_xe_alpha,1)]./10e-10);%A*zi;%V_x'*(A_eta*etahat(:,i-
1)) + V_eta'*eta_dothat(:,i-1);
    Zhat(:,i) = zi  + dt*dZdt(:,1);
     tt = [tt;ti];

     HH1(:,:,i) = H1;
     HH2(:,:,i) = H2;
end
save('Hmap_cell1_10cells_100','HH1','HH2','-v7.3
```

B. Experiments in the Context of the Biophysical Model being Studied

B.1 Set up and Protocol

GFP tagged plasma membrane HUVEC was used to detect in cell morphology and observe lamellipodial formation of 2 cells separated by 10-100um. Perti dishes were coated with 20:1 PDMS crosslinker ratio to obtain stiffness of ~800kPa stiffness. 10ug/ml of fibronectin was coated on top and left overnight. The next day, GFP tagged plasma membrane HUVEC were seeded at 5000 cells in a 52mm diameter dish (~3 cells/mm^2). Imaging was done at 16X magnitude for an hour and a half.

B.2 Experimental Evaluation

The spreading (in between the cell gap) and migration towards each other analyzed by examining the minimum distance between the two cells and the centroid distance between the cells. As can be seen in figure 17, the analysis verified that the cells migrate and extend towards each other.

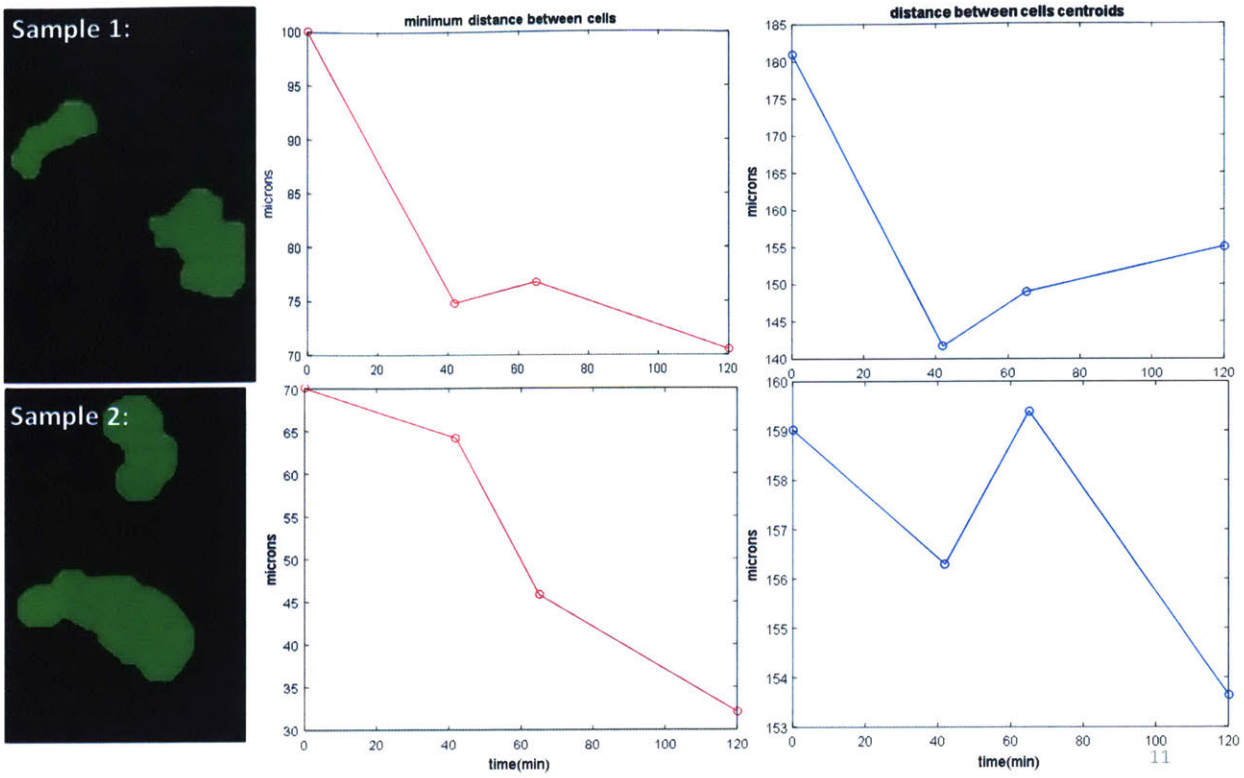


Figure 31: Minimum distance between the two cells and the centroid distance between the cells.

As seen in figure 28, cell directionality and polarity was analyzed by examining the principal axis of elongation along the cell contour (blue arrow) and the cell's movement direction between time frames (yellow). The average(green) of the two arrow was used as an indication of cell directionality. Although for some samples the cells seemed be directed towards each other, results were somewhat inconsistent and more data further experiments are necessary.

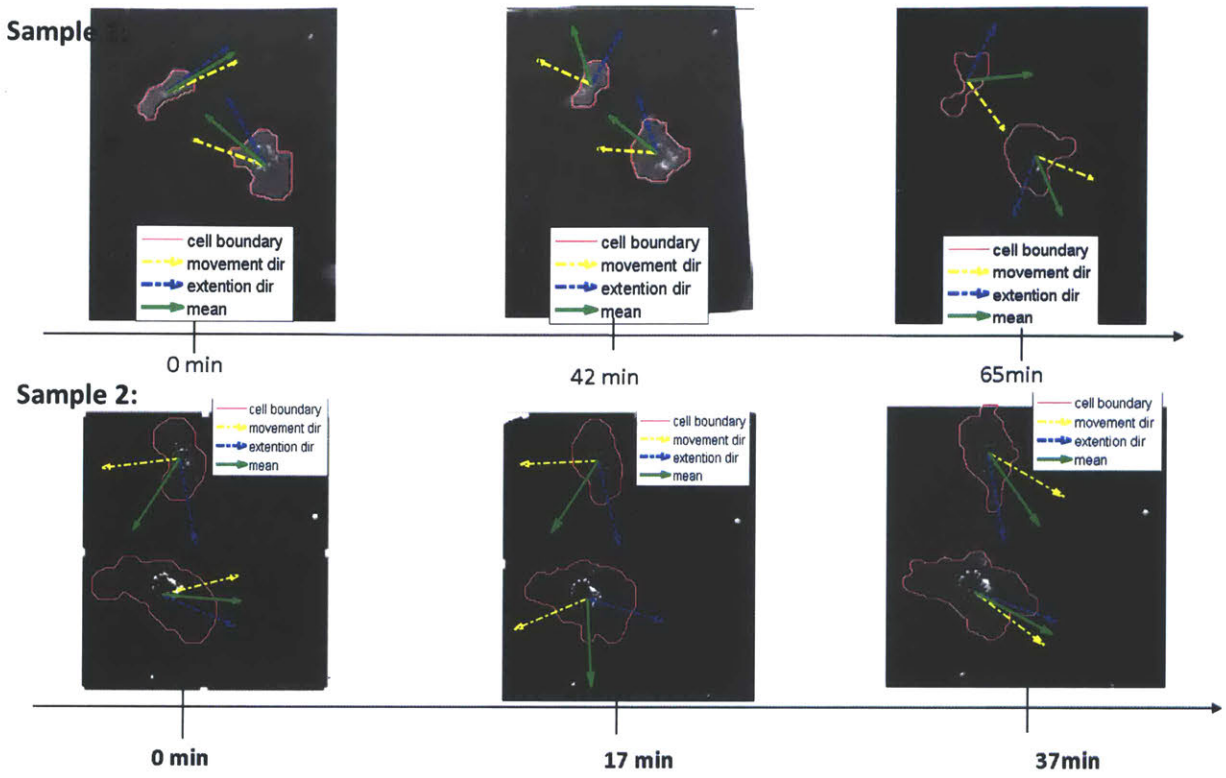


Figure 32: Cell directionality and polarity analysis. Principal axis of elongation along the cell contour (blue arrow) and the cell's movement direction between time frames (yellow). The average (green).

Bibliography

[1] H. Wang, A. S. Abhilash, C. S. Chen, R. G. Wells, and V. B. Shenoy, "Long-Range Force Transmission in Fibrous Matrices Enabled by Tension-Driven Alignment of Fibers," *Biophys. J.*, vol. 107, no. 11, pp. 2592–2603, Dec. 2014.

[2] C. M. Nelson *et al.*, "Emergent patterns of growth controlled by multicellular form and mechanics," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 102, no. 33, pp. 11594–11599, Aug. 2005.

[3] C.-L. Guo, M. Ouyang, J.-Y. Yu, J. Maslov, A. Price, and C.-Y. Shen, "Long-range mechanical force enables self-assembly of epithelial tubular patterns," *Proc. Natl. Acad. Sci.*, vol. 109, no. 15, pp. 5576–5582, Apr. 2012.

[4] S. E. M. Boas, M. M. Palm, P. Koolwijk, and R. M. H. Merks, "Computational Modeling of Angiogenesis: Towards a Multi-Scale Understanding of Cell–Cell and Cell–Matrix Interactions," in *Mechanical and Chemical Signaling in Angiogenesis*, vol. 12, C. A. Reinhart-King, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 161–183.

[5] R. F. M. van Oers, E. G. Rens, D. J. LaValley, C. A. Reinhart-King, and R. M. H. Merks, "Mechanical Cell-Matrix Feedback Explains Pairwise and Collective Endothelial Cell Behavior In Vitro," *PLOS Comput. Biol.*, vol. 10, no. 8, p. e1003774, Aug. 2014.

[6] C. A. Reinhart-King, M. Dembo, and D. A. Hammer, "Cell-Cell Mechanical Communication through Compliant Substrates," *Biophys. J.*, vol. 95, no. 12, pp. 6044–6051, Dec. 2008.

[7] A. W. Johnson and B. Harley, *Mechanobiology of Cell-Cell and Cell-Matrix Interactions*. Springer Science & Business Media, 2011.

[8] J. P. Winer, S. Oake, and P. A. Janmey, "Non-Linear Elasticity of Extracellular Matrices Enables Contractile Cells to Communicate Local Position and Orientation," *PLOS ONE*, vol. 4, no. 7, p. e6382, Jul. 2009.

[9] X. Ma *et al.*, "Fibers in the Extracellular Matrix Enable Long-Range Stress Transmission between Cells," *Biophys. J.*, vol. 104, no. 7, pp. 1410–1418, Apr. 2013.

[10] P. Fernandez and A. R. Bausch, "The compaction of gels by cells: a case of collective mechanical activity," *Integr. Biol.*, vol. 1, no. 3, pp. 252–259, Mar. 2009.

[11] M.-C. Kim, D. M. Neal, R. D. Kamm, and H. H. Asada, "Dynamic Modeling of Cell Migration and Spreading Behaviors on Fibronectin Coated Planar Substrates and Micropatterned Geometries," *PLoS Comput Biol*, vol. 9, no. 2, p. e1002926, Feb. 2013.

[12] M.-C. Kim, J. Whisler, Y. R. Silberberg, R. D. Kamm, and H. H. Asada, "Cell Invasion Dynamics into Three Dimensional Extracellular Matrix Fiber Network(Accepted 9/2015)," *PLoS Comput. Biol.*, 2015.

[13] M.-C. Kim, M. N. Mayalu, and H. H. Asada, "Dynamic modeling of collective cell migration on an elastic substrate of extracellular matrix fiber network," 2016, pp. 6911–6916.

[14] D. Vader, A. Kabla, D. Weitz, and L. Mahadevan, "Strain-Induced Alignment in Collagen Gels," *PLOS ONE*, vol. 4, no. 6, p. e5902, Jun. 2009.

[15] C.-M. Lo, H.-B. Wang, M. Dembo, and Y. Wang, "Cell Movement Is Guided by the Rigidity of the Substrate," *Biophys. J.*, vol. 79, no. 1, pp. 144–152, Jul. 2000.

[16] A. Zemel and S. A. Safran, "Active self-polarization of contractile cells in asymmetrically shaped domains," *Phys. Rev. E*, vol. 76, no. 2, p. 021905, Aug. 2007.

[17] K. Takakuda and H. Miyairi, "Tensile behaviour of fibroblasts cultured in collagen gel," *Biomaterials*, vol. 17, no. 14, pp. 1393–1397, Jul. 1996.

[18] B. C. Thorne, A. M. Bailey, and S. M. Peirce, "Combining experiments with multi-cell agent-based modeling to study biological tissue patterning," *Brief. Bioinform.*, vol. 8, no. 4, pp. 245–257, Mar. 2007.

[19] A. B. Tepole and E. Kuhl, "Systems-based approaches toward wound healing," *Pediatr. Res.*, vol. 73, no. 4–2, pp. 553–563, 2013.

[20] G. Kerschen, J. Golinval, A. F. Vakakis, and L. A. Bergman, "The Method of Proper Orthogonal Decomposition for Dynamical Characterization and Order Reduction of Mechanical Systems: An Overview," *Nonlinear Dyn.*, vol. 41, no. 1–3, pp. 147–169, Aug. 2005.

[21] S. Ahuja and C. W. Rowley, "Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators," *J. Fluid Mech.*, vol. 645, pp. 447–478, 2010.

[22] A. C. Antoulas, "An overview of approximation methods for large-scale dynamical systems," *Annu. Rev. Control*, vol. 29, no. 2, pp. 181–190, 2005.

[23] P. Kerfriden, P. Gosselet, S. Adhikari, and S. P. A. Bordas, "Bridging proper orthogonal decomposition methods and augmented Newton–Krylov algorithms: An adaptive model order reduction for highly nonlinear mechanical problems," *Comput. Methods Appl. Mech. Eng.*, vol. 200, no. 5–8, pp. 850–866, Jan. 2011.

[24] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Compressive sampling and dynamic mode decomposition," Dec. 2013.

[25] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic Mode Decomposition with Control," *SIAM J. Appl. Dyn. Syst.*, vol. 15, no. 1, pp. 142–161, Jan. 2016.

[26] H. Harry Asada, Faye Wu, Alexandre Girard, and Michaelle N. Mayalu, "A Data-Driven Approach to Precise Linearization of Nonlinear Dynamical Systems in Augmented Latent Space," in *American Control Conference Proceedings(Preprint)*, Boston, MA, 2015.

[27] D. L. Margolis, "A survey of bond graph modelling for interacting lumped and distributed systems," *J. Frankl. Inst.*, vol. 319, no. 1, pp. 125–135, Jan. 1985.

[28] W. Borutzky, *Bond Graph Methodology: Development and Analysis of Multidisciplinary Dynamic System Models*. Springer Science & Business Media, 2009.

[29] A. M. Bos and M. J. L. Tiernego, "Formula manipulation in the bond graph modelling and simulation of large mechanical systems," *J. Frankl. Inst.*, vol. 319, no. 1–2, pp. 51–65, Jan. 1985.

[30] P. J. Gawthrop, J. Cursons, and E. J. Crampin, "Hierarchical bond graph modelling of biochemical networks," *Proc R Soc A*, vol. 471, no. 2184, p. 20150642, Dec. 2015.

[31] P. J. Gawthrop and E. J. Crampin, "Energy-based Analysis of Biochemical Cycles using Bond Graphs," *Proc. R. Soc. Math. Phys. Eng. Sci.*, vol. 470, no. 2171, pp. 20140459–20140459, Sep. 2014.

[32] U. Kruger and L. Xie, *Advances in statistical monitoring of complex multivariate processes: with applications in industrial process control*. John Wiley & Sons, 2012.

[33] E. L. Barnhart, G. M. Allen, F. Jülicher, and J. A. Theriot, "Bipedal Locomotion in Crawling Cells," *Biophys. J.*, vol. 98, no. 6, pp. 933–942, Mar. 2010.

[34] L. Yang, J. C. Effler, B. L. Kutscher, S. E. Sullivan, D. N. Robinson, and P. A. Iglesias, "Modeling cellular deformations using the level set formalism," *BMC Syst. Biol.*, vol. 2, no. 1, p. 68, Jul. 2008.

[35] G. De Santis, A. B. Lennon, F. Boschetti, B. Verhegghe, P. Verdonck, and P. J. Prendergast, "How can cells sense the elasticity of a substrate? An analysis using a cell tensegrity model," *Eur. Cell. Mater.*, vol. 22, pp. 202–213, 2011.

[36] C. Borau, R. D. Kamm, and J. M. García-Aznar, "Mechano-sensing and cell migration: a 3D model approach," *Phys. Biol.*, vol. 8, no. 6, p. 066008, Dec. 2011.

[37] P. C. Breedveld, "Multibond graph elements in physical systems theory," *J. Frankl. Inst.*, vol. 319, no. 1, pp. 1–36, Jan. 1985.

[38] J. H. Ginsberg, *Advanced Engineering Dynamics*. Cambridge University Press, 1998.

[39] "Wiley: System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems, 5th Edition - Dean C. Karnopp, Donald L. Margolis, Ronald C. Rosenberg." [Online]. Available: http://www.wiley.com/WileyCDA/WileyTitle/productCd-047088908X.html. [Accessed: 28-Apr-2017].

[40] J.-A. Park *et al.*, "Unjamming and cell shape in the asthmatic airway epithelium," *Nat. Mater.*, vol. 14, no. 10, pp. 1040–1048, Oct. 2015.

[41] M. S. Hall *et al.*, "Fibrous nonlinear elasticity enables positive mechanical feedback between cells and ECMs," *Proc. Natl. Acad. Sci.*, vol. 113, no. 49, pp. 14043–14048, Dec. 2016.

[42] P. Lu, K. Takai, V. M. Weaver, and Z. Werb, "Extracellular Matrix Degradation and Remodeling in Development and Disease," *Cold Spring Harb. Perspect. Biol.*, vol. 3, no. 12, Dec. 2011.

[43] R. Mayor and S. Etienne-Manneville, "The front and rear of collective cell migration," *Nat. Rev. Mol. Cell Biol.*, vol. 17, no. 2, pp. 97–109, Feb. 2016.

[44] Q. Zhao *et al.*, "Multilinear Subspace Regression: An Orthogonal Tensor Decomposition Approach.," in *NIPS*, 2011, vol. 2011, pp. 1269–1277.

[45] G. An, Q. Mi, J. Dutta-Moscato, and Y. Vodovotz, "Agent-based models in translational systems biology," *Wiley Interdiscip. Rev. Syst. Biol. Med.*, vol. 1, no. 2, pp. 159–171, 2009.

[46] J. F. MacGregor and T. Kourti, "Statistical process control of multivariate processes," *Control Eng. Pract.*, vol. 3, no. 3, pp. 403–414, 1995.

[47] T. Katayama, *Subspace methods for system identification*. Springer Science & Business Media, 2006.

[48] M. N. Mayalu, M.-C. Kim, and H. H. Asada, "A Linear Systems Approach to Prediction of Emergent Behaviors of Interacting Nonlinear Agents describing Collective Cell Migration on an Elastic Substrate," *PLoS Comput. Biol. Prep.*

[49] I. B. Bischofs and U. S. Schwarz, "Cell organization in soft media due to active mechanosensing," *Proc. Natl. Acad. Sci.*, vol. 100, no. 16, pp. 9274–9279, Aug. 2003.

[50] D. T. Tambe *et al.*, "Collective cell guidance by cooperative intercellular forces," *Nat. Mater.*, vol. 10, no. 6, pp. 469–475, 2011.

[51] G. F. Oster, J. D. Murray, and A. K. Harris, "Mechanical aspects of mesenchymal morphogenesis," *Development*, vol. 78, no. 1, pp. 83–125, 1983.

[52] J. T. Daub and R. M. H. Merks, "A Cell-Based Model of Extracellular-Matrix-Guided Endothelial Cell Migration During Angiogenesis," *Bull. Math. Biol.*, vol. 75, no. 8, pp. 1377–1399, Aug. 2013.

[53] S. Checa, M. K. Rausch, A. Petersen, E. Kuhl, and G. N. Duda, "The emergence of extracellular matrix mechanics and cell traction forces as important regulators of cellular self-organization," *Biomech. Model. Mechanobiol.*, vol. 14, no. 1, pp. 1–13, Jan. 2015.

[54] R. O. Hynes, "Extracellular matrix: not just pretty fibrils," *Science*, vol. 326, no. 5957, pp. 1216–1219, Nov. 2009.