Dynamic Cellular Actuator Arrays and Expanded Fingerprint Method for Dynamic Modeling

D. MacNair[a,*], J. Ueda[a]

[a] *Bio-Robotics & Human Modeling Lab, School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332*

Abstract

A key step to understanding and producing natural motion is creating a physical, well understood actuator with a dynamic model resembling biological muscle. This actuator can then serve as the basis for building viable, full-strength, and safe muscles for disabled patients, rehabilitation, human force amplification, telerobotics, and humanoid robotic systems. This paper presents a cell-based flexible actuator modeling methodology and the General Fingerprint Method for systematically and efficiently calculating the actuators' respective dynamic equations of motion. The cellular actuator arrays combine many flexible 'cells' in complex and varied topologies for combined large-scale motion. The cells can have varied internal dynamic models and common actuators such as piezoelectric, SMA, linear motor, and pneumatic technologies can fit the model by adding a flexible element in series with the actuator. The topology of the cellular actuator array lends it many of its properties allowing the final muscle to be catered to particular applications. The General Fingerprint Method allows for fast recalculation for different and/or changing structures and internal dynamics, and provides an intuitive base for future controls work. This paper also presents two physical SMA based cellular actuator arrays which validate the presented theory and give a basis for future development.

Keywords: Modular, Flexible, Cellular, Actuator Array, Muscle, Bio-Inspired, Actuation, Fingerprint, Topology, Cell

INTRODUCTION

For decades researchers in physiology have been trying to model and generate natural motions, the movements created by biological systems, in order to both gain a greater understanding of biological muscle and to produce motions similar to muscle. This knowledge, in turn, benefits therapy and rehabilitation for patients who suffered muscle damage or degradation, allowing patients to function more easily in society and, in the best case, make the loss a non-issue in ordinary life. The advances also aid in the development of naturally moving prosthetic devices for those who have lost limbs and

[*] Corresponding author

facilitates the development of human force amplification exoskeletons. Finally muscle-like actuators, and knowledge of the control processes used to generate natural motion, enables humanoid robots to move in more natural ways. This yields more life-like and capable humanoid robotic systems, and is important as robots continue to integrate into human society.

Biological Inspiration

Although many humanoid robots can be said to be biologically inspired in terms of morphology, they are not biologically inspired in terms of how they are actuated; most use some sort of traditional servomotor. Resultant movements are far from those characterizing humans. Multi-celled organisms have specialized cells (muscle cells) that move their body parts by elastic contraction in response to signals from the central nervous system. Biological muscles are non-continuous and non-uniform; muscles consist of several types of muscle tissues with different levels of contractile speed and fatigability [1]. A motor unit, a bundle of muscle fibers with a specific force capability connected to a single motor neuron, is stimulated by nerve impulses. Hence, a single muscle consisting of motor units is structurally quantized in terms of force generation.

Should robots also possess these properties, great strides could be made in terms of cycle time, capability, the number of environments in which robots can be deployed, and cost. The primary way to endow robots with biological abilities is to equip them with biologically inspired actuation. This paper uses the term "biologically-inspired" in regard to actuation in robotics, namely, that the actuators themselves have structural and operational characteristics in common with muscles. Specifically, the actuator systems of interest will 1) have a modular structure: the actuator selectively activates distinct units (recruitment), and 2) possess elasticity allowing impulse signals to produce a smooth contraction. The modular architecture may be non-uniform, or hybrid, consisting of different materials, such as piezoelectric actuators (with high-speed, but low-force) and shape-memory alloy (SMA) actuators (with slow-speed, but high-force), or different sizes, such as using differing diamater SMA actuators, allowing a wider working range. Due to the elasticity and viscosity of the muscle tissue, twitch and tetanic contraction create a relatively damped force profile [2, 3, 4].

Actuator Array Prior Work

A key step along the way to understanding the natural motion [5, 6, 7] is producing a physical, well understood test platform with a dynamic model closely resembling biological muscle. This test bed can then serve as the basis for experiments to better understand the interrelated nature of the nervous system and the muscles, for kinematics/dynamics experiments to understand balance and synergies, and for building viable, full-strength, and safe muscles for prosthetics, human force amplification, and humanoid robotic systems.

[8] presents biologically-inspired cell array actuators consisting of many small cells interconnected in various layouts, or topologies, to achieve muscle-like motion. In this work all cells were identical and the topology of the cell array actuator, represented compactly using a two row set of matricies or "fingerprint", differentiated static properties such as displacement, force capacity, force descritization, and robustness. The

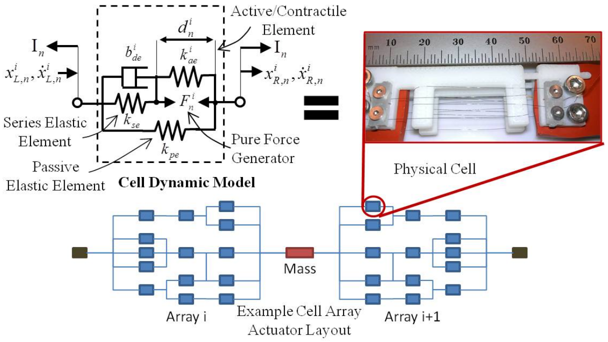

Figure 1: HILL-TYPE MODEL, PHYSICAL CELL, AND EXAMPLE CELL ARRAY ACTUATOR LAYOUT.

cells were operated using a bi-stable stochastic all-on all-off broadcast control method to reduce wiring complexity, control signals, and hysteresis error [9]. [10] expanded upon the results of [8] looking more specifically into force descritization and presented methods for generating all possible cell array topologies given a limited number of cells. This later result is critical to designers wanting to explore the possibilities of the cell arrays and select a topology specific to their criteria.

[10] was a static modeling of the form $\mathbf{A}\cdot\mathbf{x} = \mathbf{c}+\mathbf{u}$ where $\mathbf{A}$ represented the topology, $\mathbf{x}$ the state vector, $\mathbf{c}$ constants for the system such as endpoint locations, and $\mathbf{u}$ the control input. This was then solved for the state vector using $\mathbf{x} = \mathbf{A}^{-1}\cdot(\mathbf{c}+\mathbf{u})$. While this allowed for easily exploring stochastic properties and guided topology selection, it did not generate the dynamic equations of motion.

In [11], an expanded version of the fingerprint method presented in [10] allows for the dynamic analysis cell array actuators built from cells based on Miga NanoMuscle 704 SMA actuators connected in series with coil springs. The cell model for these actuator arrays was a modified Hill-Type model shown in Fig. 1.

Graph Theoritic Modeling

Methods for graphically representing complex multibody systems and obtaining governing equations include bond graphs and graphic theoretic modeling (GTM), or linear graphs [12]. McPhee has presented a series of publications on the applications of linear graph theory to flexible multibody systems [13, 12, 14, 15]. The key concept is to introduce a matrix, named incidence matrix, to represent a complex topology of a multibody system. This graph-theoretic approach enables automatic generation of dynamic equations [16, 17]. While these methods do generate the equations of motion for the dynamic systems of interest to the current work, they are highly general and generate redundant equations requiring careful manual selection of state variables, or cut-sets, to have physical meaning to the reduced escholon form required to systematically generate the equations of motion. This manual selection is not conducive to the automatic generation required to analyze many different actuator topologies and differing internal cell structures. Additionally, the incidence matrix in the graphical techniques consists of a list of connections between all elements in the system while the expanded fingerprint method presented below uses the self-contained cell structures to separate the internal cell dynamics from the topology dynamics, thus greatly simplifying and shrinking the incidence matrix (represented as G and H matricies in the theory below). [18] presents a similar simplifying method called Newton-Raphson Mixed Nodal Tableau by separating internal dynamics of photovoltaic cells and then "stamping" these repeated dynamics into a larger system as a single element and using GTM to generate the final equations of motion. This allowed for a simplier process for generating the dynamic equations of motion and allowed for the non-linear dynamics of the photovoltaic system while treating the rest of the system as linear. [18] focused on electrical systems, specifically photovoltaic power systems, resulting in an application specific algorithm with fixed equations that cannot be easily applied to mechanical systems.

Current Contribution

The current contribution generalizes the expanded fingerprint method presented in [11] to allow for general linear actuation technologies. Actuator technologies such as piezoelectrics, ultrasonic motors, linear stepper motors, hydraulics, pneumatics, and shape memory alloy (SMA) can all be represented. The method provides a direct process to generate the standard state-space form $\dot{\mathbf{X}} = \mathbf{A} \cdot \mathbf{X} + \mathbf{B} \cdot \mathbf{u}$ from any array layout, or topology, and varied cell internal dynamics. The resultant state-space form allows for using standard controllability, stability, etc. analysis techniques for the actuator arrays and for simulating their responses as a part of larger dynamic systems.

The approach is built to aid automation and simulation of the cell array actuators, allows for fast recalculation for different cell array topologies, and provides an intuitive base for future controls work on cell array actuators. The dynamics representing a given cell array actuator could be generated using Dymola, SimScape, GTM [16, 17], or other computational methods based on base principles. The presented expanded fingerprint method allows the dynamics to be calculated with less human effort, less computational effort, and with greater speed, especially when comparing different topologies and cell dynamics. While the current work builds on the idea of stamping presented in [18], that work used fixed equations for photovoltaic systems while the presented method represents the general interactions between mechanical systems.

Two different physical SMA based cell array actuator cell designs, and thus design methodologies, are presented and used to validate the presented theory. The first cell design, from [11], consists of two Miga NanoMuscle 704 SMA actuators mounted to an ABS rapid prototyped shell, and the second is a more practical silicone-based SMA actuator array design inspired by the M-lines and Z-lines of sarcomeres in biological muscle. SMA actuators have been used in the past as muscle-like actuators [19] and a major drawback has been nonlinear dynamics due to hysteresis. The current work utilizes a flexible structure as well as forward-loop non-linear input force to separate these non-linearities from the linear system dynamics elements, allowing for the use of a wide range of linear actuators without compromising the bio-inspired intent of the muscles.

MODELING

Cell Definition

Cell array actuators are collections of cells connected in various arrangements, or topologies, to provide the large-scale motion required of a muscle system. The topology of the array is critical to determining the array's properties. A cell consists of a linear actuator connected in series with a flexible element which allows the linear actuator to achieve its full travel, even when externally fully blocked, and mitigates the differences in current length between the different cells. The actuator can be based on any linear actuation technology so long as 1) the actuator can be represented as a spring, rigid or flexible, with a pure force preloading the spring; 2) the actuator's pure force follows a known force versus time function, f(t), when activated; 3) that function is minimally dependent on external cell load and therefore the interaction is negligible. The first criteria is generally true of most actuators, whether they are mostly rigid like a motor or flexible like shape memory alloy, but it does limit the use of actuators with large amounts of creep. Modifications to the presented theory can allow for high-creep actuators, however these are not considered in the current work. The second criteria is generally easy to achieve through measurement. The criteria implies that the external effects on the cell, whether displacement or force, should have little effect on the actuator's ability to reach the commanded position. This can be met through design, for example by choosing a low stiffness flexible element to combine in series with an actuator to ensure the actuator can move through its full displacement. Actuator technologies such as piezoelectrics, ultrasonic motors, linear stepper motors, hydraulics, pneumatics, and shape memory alloy (SMA) can all meet the three criteria.

Cell Modeling

The model of a cell comes through inspiration from studies in physiology. In physiology, the linear hill-type model [20] shown in Fig. 1 is a widely accepted and utilized dynamic model for muscle dynamics. The model consists of a passive elastic element, a series elastic element, and a contractile element. In most accepted literature, the passive elastic and series elastic elements are modeled as springs while the contractile element does not have consistently defined dynamics and in many cases is simply treated as a function of force versus time. In physiology this is done due to lack of knowledge of the dynamics of the contractile element, an open area of research, however the current work uses it as a method of model simplification. If the above three criteria are met, the internal dynamics of the actuator can be decoupled from the dynamics of the actuator array and treated instead as a feed-forward effect on the actuator array dynamics. The actuator, or contractile element in the hill-type model, is treated as a spring acted on by a pure force preload. The force, F_n, is given by equation (1) where k_{ae} is the spring constant of the actuator and s_n is the displacement of the actuator at a given point in time. The result is a shortening of the neutral length of the actuator's 'spring' by the amount of the displacement of the actuator.

$$F_n = k_{ae} \cdot (s_n) \tag{1}$$

The series elastic element represents the flexible connective material between actuators and can be anything from metal springs to rubber to even cloth. In general this element should have a lower spring constant than the actuator itself to meet the three criteria. Here the hill-type model is modified slightly to add in an additional damper in parallel with the series elastic element to represent the damping effects of many material choices, such as silicone rubber, and to allow for additional design options. In the current work the connecting structure should be linear to allow for representation by a linear model, though the work can be extended to include non-linear materials where creep and hysteresis play larger roles.

The parallel elastic element represents spring-like forces carried across the cell but not seen by the series elastic and contractile elements. These are generally kept small to avoid compressive forces internal to the cell but are maintained through the theory to allow for greater design flexibility.

Topology

Having more cells in series tends to give an array more displacement, more in parallel gives more force and higher robustness, and having a non-uniform structure can give higher force discretization and more fine-tuned control [10]. A physical cell has two significant mass elements connected by the SMA actuator and spring. In the array model, the cells are each treated as mass-less elements and the mass of the cells is combined with the mass of the connecting structures holding the cells together (referred to as masses). All cells connect directly to masses and no mass is connected directly to any other mass. Furthermore, the design restrictions in [10] are upheld, namely that an array consists of discrete equal relaxed length layers and all cells are identical. This last restriction will be relaxed in future work but is maintained here to simplify explination of the theory. The array topology can be represented using the fingerprint method from [10], a layer based set of three row matrices encoding the connection information of the array. The fingerprint transcription consists of segmenting the array into layers just after the cells and before the incoming connecting structures. The layer can then be represented as a front structure and a back structure sharing mid-layer nodes. For each mid-layer node, a hexadecimal number represents a binary encoding of the outgoing nodes from the previous layer that the mid-layer node connects to. For example, in Fig. 2 the incoming structure layer two would be represented by [&1, &E, &10] showing the first mid-layer node connects to the first outgoing node, 00001; the second mid-layer node connects to the next three outgoing nodes, 01110; and the third mid-layer node connects to the last outgoing node, 10000. The back structure for each mid-layer node is represented by the number of cells connected to that mid-layer node. "-1" is used to signify the end of the array as there are no additional cells for the final layer of the array. Fig. 2 shows the complete transcription of a fourteen cell array with special focus given to the second layer for explanation purposes.

This fingerprint is used as a part of the expanded fingerprint method to quickly generate the equations of motion for the array, which in turn aids in exploring the array's properties. For additional details regarding the fingerprint, autogeneration of fingerprints, etc. readers are referred to [10].

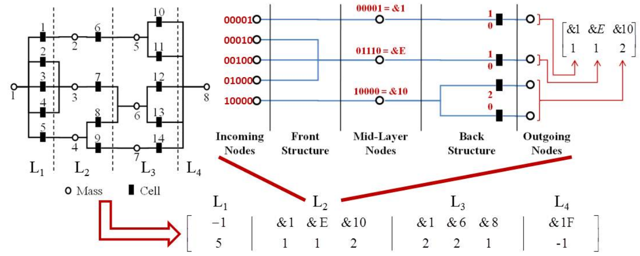

Figure 2: EXAMPLE OF BUILDING A FINGERPRINT FROM A ACTUATOR ARRAY TOPOLOGY.

Array Modeling

Any actuator array, i, built from hill-type model cells (Fig. 1) can be represented in state-space form by choosing the states to be the position (x_m^i) and velocity ($\dot{x}_m^i$) of each mass, m, and the length of the contractile element ($d_{1,n}^i$) of each cell, n. The subscript for $d_{1,n}^i$ refers to the internal state number for the cell; for which there is only one, though the subscript is maintained for generalization. Equation (2) represents the tension force carried across cell n, the cumulative force carried by the passive elastic element, the active elastic elements, and exerted by the input actuator. This is the force exerted on the mass elements that cell n connects to. Since the mass of the connection between the series elastic and contractile elements is negligible, the force carried by each end of the cell is equal. For each elastic element, the force carried across the element is the spring constant, k, of the elastic element multiplied by the change in length of the element.

$$I_n^i = k_{pe} \cdot \left(x_{R,n}^i - x_{L,n}^i\right) + k_{ae} \cdot \left(d_{1,n}^i\right) + F_n^i \tag{2}$$

Equation (3) shows the force balance of the upper portion of the Hill-type model for cell n. Here the cumulative force carried by the series elastic element and the damper element is the same as cumulative force carried by the active elastic element and the input force F_n.

$$k_{se} \cdot \left(x_{R,n}^i - x_{L,n}^i - d_{1,n}^i\right) + b_{de} \cdot \left(\dot{x}_{R,n}^i - \dot{x}_{L,n}^i - \dot{d}_{1,n}^i\right) = k_{ae} \cdot \left(d_{1,n}^i\right) + F_n^i \tag{3}$$

In both equations, $x_{L,n}^i$ is the position of the mass connected to the left of cell n and $x_{R,n}^i$ is the position of the mass connected to the right of cell n. $d_{1,n}^i$ is the length of cell n's contractile element, and $\dot{d}_{1,n}^i$ is the time derivative of $d_{1,n}^i$. k_{se}, k_{pe}, and k_{ae} are the spring constants of the series elastic, parallel elastic, and actuator elastic elements respectively. b_{de} is the damping coefficient of the damper and F_n^i is the activation pure force acting across the damper. F_n^i is zero when the cell is inactive and F when active. The electrical dynamics act significantly faster than the physical dynamics and are thus not modeled. Fig. 1 shows the elements and associated variables and can be used as a reference to varify equations (2) and (3).

Equation (3) can be rearranged to solve for $\dot{d}^i_{1,n}$ as a function of the state variables, as shown in (4). $\dot{d}^i_{1,n}$ shows the change in the contractile element length, an essential part of the final dynamics.

$$\begin{aligned}\dot{d}^i_{1,n} = \frac{k_{se}}{b_{de}} \cdot \left(x^i_{R,n} - x^i_{L,n}\right) - \frac{k_{se} + k_{ae}}{b_{de}} \cdot \left(d^i_{1,n}\right) - \frac{1}{b_{de}} \cdot \left(F^i_n\right) \\ + \left(\dot{x}^i_{R,n} - \dot{x}^i_{L,n}\right)\end{aligned} \tag{4}$$

Since masses are only connected to cells, a force balance for each mass contains only forces from cells on either side of the mass. Cells on the left side pull the mass in the negative direction while those on the right pull in the positive direction. Equation (5) shows the resultant acceleration for mass m with list of cells L^i_m connected to the left and list of cells R^i_m connected to the right. The mass of mass m is $mass^i_m$.

$$\begin{aligned}\ddot{x}^i_m = &-\frac{k_{pe}}{mass^i_m} \cdot \left[\sum_{n=L^i_m} 1 + \sum_{n=R^i_m} 1\right] \cdot x^i_m \\ &+\frac{k_{pe}}{mass^i_m} \cdot \left[\sum_{n=L^i_m} x^i_{L,n} + \sum_{n=R^i_m} x^i_{R,n}\right] \\ &-\frac{k_{ae}}{mass^i_m} \cdot \left[\sum_{n=L^i_m} d^i_{1,n} - \sum_{n=R^i_m} d^i_{1,n}\right] \\ &-\frac{1}{mass^i_m} \cdot \left[\sum_{n=L^i_m} F^i_n - \sum_{n=R^i_m} F^i_n\right]\end{aligned} \tag{5}$$

Using mass 6 in Fig. 2 as an example, the summations in (5) would be (6):

$$\begin{aligned} L^i_m &= [7, 8] & R^i_m &= [12, 13] \\ \sum_{n=L^i_m} 1 &= 2 & \sum_{n=R^i_m} 1 &= 2 \\ \sum_{n=L^i_m} x^i_{L,n} &= x^i_{L,7} + x^i_{L,8} = x^i_3 + x^i_4 & \sum_{n=R^i_m} x^i_{R,n} &= x^i_{R,12} + x^i_{R,13} = 2 \cdot x^i_8 \\ \sum_{n=L^i_m} d^i_{1,n} &= d^i_{1,7} + d^i_{1,8} & \sum_{n=R^i_m} d^i_{1,n} &= d^i_{1,12} + d^i_{1,13}\end{aligned} \tag{6}$$

The trivial equation (7) completes the needed state space equations.

$$\dot{x}^i_m = \frac{\mathrm{d}}{\mathrm{d}t} x^i_m \tag{7}$$

The end masses, x_1^i and x_M^i for M^i masses in an array i are considered the external connections. If the connection point is rigid, the position value is a constant. If the point is a free mass, m_m^i is the mass of this point. If outside dynamics act on the point, for instance when an antagonist actuator array also connects to the mass, the force of the outside dynamics is added to (5) and $mass_m^i$ is adjusted to reflect the combined mass at this point.

EXPANDED FINGERPRINT METHOD

The standard linear state-space form $\dot{\mathbf{X}} = \mathbf{A} \cdot \mathbf{X} + \mathbf{B} \cdot \mathbf{u}$ allows for using standard controllability, stability, etc. analysis techniques. The Expanded Fingerprint Method provides a direct method to generate the standard form from any fingerprint and linear cell internal dynamics, not just the hill-type model based dynamics, given that 1) cells connect solely to masses (thus the only external states needed for cell dynamics are the mass positions and velocities) and 2) the forces at each side of the cell are identical (cell has negligible mass).

Any linear internal cell dynamics which follows this criteria as well as the criteria in the cell and topology subsections above can be represented in the generalized form given by (8) and (9). Function $cod^i\left(y,q\right)$ represents the coefficients in front of the given variable y for internal variable equation $\dot{d}_{q,n}^i$ and $com^i\left(y\right)$ represents the coefficients in front of the given variable y for mass equation $\ddot{x}_m^i$.

$$\begin{aligned}\dot{d}_{q,n}^i =& cod^i\left(x_{L,n}^i, q\right)\cdot\left[x_{L,n}^i\right] + cod^i\left(x_{R,n}^i, q\right)\cdot\left[x_{R,n}^i\right]\\ &+cod^i\left(\dot{x}_{L,n}^i, q\right)\cdot\left[\dot{x}_{L,n}^i\right] + cod^i\left(\dot{x}_{R,n}^i, q\right)\cdot\left[\dot{x}_{R,n}^i\right]\\ &+cod^i\left(F_n^i, q\right)\cdot\left[F_n^i\right] + \sum_{q=1}^{Q}\left(cod^i\left(d_{q,n}^i, q\right)\cdot\left[d_{q,n}^i\right]\right)\end{aligned} \tag{8}$$

$$
\begin{aligned}
\ddot{x}^i_m = &-com^i\left(x^i_{R,n}\right)\cdot\left[\sum_{n=L^i_m} x^i_m\right] - com^i\left(x^i_{L,n}\right)\cdot\left[\sum_{n=R^i_m} x^i_m\right] \\
&+com^i\left(x^i_{L,n}\right)\cdot\left[\sum_{n=L^i_m} x^i_{L,n}\right] + com^i\left(x^i_{R,n}\right)\cdot\left[\sum_{n=R^i_m} x^i_{R,n}\right] \\
&-com^i\left(\dot{x}^i_{R,n}\right)\cdot\left[\sum_{n=L^i_m} \dot{x}^i_m\right] - com^i\left(\dot{x}^i_{L,n}\right)\cdot\left[\sum_{n=R^i_m} \dot{x}^i_m\right] \\
&+com^i\left(\dot{x}^i_{L,n}\right)\cdot\left[\sum_{n=L^i_m} \dot{x}^i_{L,n}\right] + com^i\left(\dot{x}^i_{R,n}\right)\cdot\left[\sum_{n=R^i_m} \dot{x}^i_{R,n}\right] \\
&-com^i\left(F^i_n\right)\cdot\left[\sum_{n=L^i_m} F^i_n\right] + com^i\left(F^i_n\right)\cdot\left[\sum_{n=R^i_m} F^i_n\right] \\
&+\sum_{q=1}^{Q}\left(com^i\left(d^i_{q,n}\right)\cdot\left[\sum_{n=L^i_m} d^i_{q,n}\right] - com^i\left(d^i_{q,n}\right)\cdot\left[\sum_{n=R^i_m} d^i_{q,n}\right]\right) \qquad (9)
\end{aligned}
$$

Equation (5) can be represented in the form of (9), for example, by making the substitutions in (10).

$$
\begin{aligned}
com^i\left(x^i_{R,n}\right) &= com^i\left(x^i_{L,n}\right) = \frac{k_{pe}}{mass^i_m} \\
com^i\left(x^i_{L,n}\right) &= com^i\left(x^i_{R,n}\right) = \frac{k_{pe}}{mass^i_m} \\
com^i\left(\dot{x}^i_{R,n}\right) &= com^i\left(\dot{x}^i_{L,n}\right) = 0 \\
com^i\left(\dot{x}^i_{L,n}\right) &= com^i\left(\dot{x}^i_{R,n}\right) = 0 \\
com^i\left(d^i_{q,n}\right) &= com^i\left(d^i_{q,n}\right) = \frac{k_{ae}}{mass^i_m} \\
com^i\left(F^i_n\right) &= \frac{1}{mass^i_m} \qquad (10)
\end{aligned}
$$

If the state vector is chosen to have a specific form, where states are separated according to muscle and type, a simplified method of generating the dynamic equations of motions exists which is derived from linear algebra.

State vector $\mathbf{X}$ can be defined as (11) to seperate internal array dynamics from connection point and outside dynamics. $\mathbf{d}^i_q$ in (11) represents internal state variable q for all n cells in array i. This is used to group state variables internal to the cells when complex cell dynamics require more than one internal variable. Though there is only one internal variable in the Hill based model described above, the notation is kept for generality.

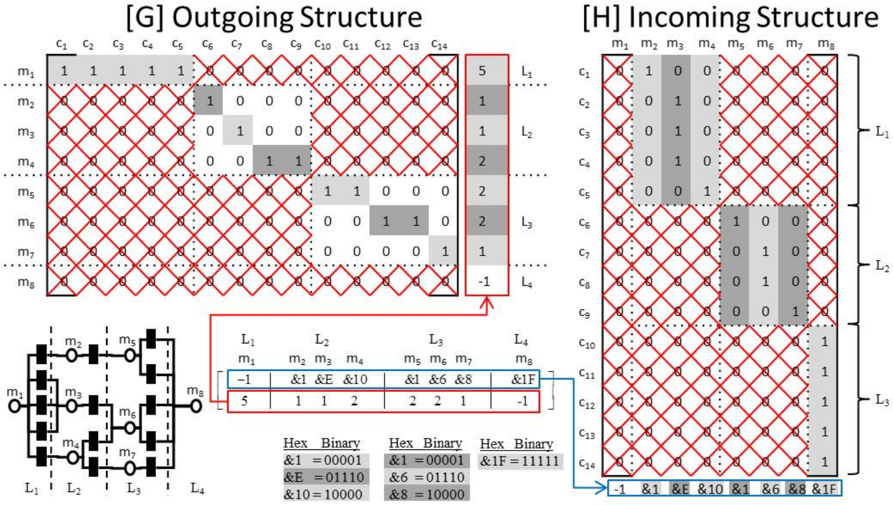

Figure 3: BUILDING $\mathbf{G}^i$ AND $\mathbf{H}^i$, THE OUTGOING AND INCOMING STRUCTURES.

$$\mathbf{X} = \begin{bmatrix} \vdots \\ \mathbf{X}^i \\ \dot{\mathbf{X}}^i \\ \mathbf{D}^i \\ \vdots \\ x_l^c \\ \dot{x}_l^c \\ \vdots \end{bmatrix}, \begin{bmatrix} \mathbf{X}^i \\ \dot{\mathbf{X}}^i \end{bmatrix} = \begin{bmatrix} x_2^i \\ \vdots \\ x_{M-1}^i \\ \dot{x}_2^i \\ \vdots \\ \dot{x}_{M-1}^i \end{bmatrix}, \mathbf{D}^i = \begin{bmatrix} \mathbf{d}_1^i \\ \vdots \\ \mathbf{d}_q^i \\ \vdots \\ \mathbf{d}_Q^i \end{bmatrix}, \mathbf{d}_q^i = \begin{bmatrix} \mathbf{d}_{q,1}^i \\ \vdots \\ d_{q,n}^i \\ \vdots \\ d_{q,N}^i \end{bmatrix} \tag{11}$$

This state vector is built by first defining $\mathbf{m^i} = [m_1^i ... m_M^i]$ as the set of masses belonging to actuator array i with M^i total masses. Also define $\mathbf{n^i} = [n_1^i ... n_N^i]$ as the set of cells belonging to actuator array i with N^i total cells. Actuator array i has states $x_1^i ... x_M^i$, $\dot{x}_1^i ... \dot{x}_M^i$, and $\mathbf{D}^i$. The incoming connection points for actuator array i are m_1^i and m_M^i respectively, with states x_1^i, x_M^i, $\dot{x}_1^i$, and $\dot{x}_M^i$. If a connection point is shared across multiple actuators, for instance when antagonistic pairs are used, the duplicate states must be removed. Define $\mathbf{x^c} = [x_1^c ... x_l^i ... x_L^c]$ and $\dot{\mathbf{x}}^\mathbf{c} = [\dot{x}_1^i ... \dot{x}_l^i ... \dot{x}_L^i]$ as the states of all of the connection points, or endpoints, for each array and remove x_1^i, x_M^i, $\dot{x}_1^i$, and $\dot{x}_M^i$ for all i, thus removing all duplicates. For example if m_M^i is also m_1^{i+1} then $x_M^i = x_1^{i+1} = x_l^c$ and $\dot{x}_M^i = \dot{x}_M^{i+1} = \dot{x}_l^c$ thus reducing four states to two with no duplicates. The remaining states $\mathbf{x^i} = [x_2^i ... x_{M-1}^i]$, $\dot{\mathbf{x}}^\mathbf{i} = [\dot{x}_2^i ... \dot{x}_{M-1}^i]$, and $\mathbf{D}^i$ from the internal states for the array are placed in the final state space form separate from the $\mathbf{x^c}$ and $\dot{\mathbf{x}}^\mathbf{c}$], the connection point dynamics, as shown in (11).

The expanded fingerprint method begins by defining an incoming connections matrix, $\mathbf{H}^i$, and an outgoing connections matrix, $\mathbf{G}^i$, for each array i. $\mathbf{H}^i$ is an N^i-by-M^i matrix and $\mathbf{G}^i$ is an M^i-by-N^i matrix where N^i is the number of cells in array i and M^i is the number of masses. The elements of both $\mathbf{H}^i$ and $\mathbf{G}^i$ are either 1, representing a connection between the associated cell (row in $\mathbf{G}^i$, column in $\mathbf{H}^i$) and mass (column in $\mathbf{G}^i$, row in $\mathbf{H}^i$), or 0, representing no connection. This is shown in equation (12).

$$
\begin{aligned}
\mathbf{G}^i_{m,n} &= \begin{cases} 0 & \text{if mass } m \text{ not connected to cell } n \\ 1 & \text{if mass } m \text{ structure connected to cell } n \end{cases} \\
\mathbf{H}^i_{n,m} &= \begin{cases} 0 & \text{if mass } m \text{ not connected to cell } n \\ 1 & \text{if mass } m \text{ structure connected to cell } n \end{cases}
\end{aligned} \tag{12}
$$

$\mathbf{H}^i$ can be populated automatically from the first row of the fingerprint. Column one is skipped as there are no left-connecting cells to the leftmost mass. Columns two through M^i are populated as shown in Fig. 3. Each subsequent layer begins one row down from the previous layer's lowest entry. $\mathbf{G}^i$ can be populated automatically from the second row of the fingerprint. The last row is skipped as there are no right-connecting cells to the rightmost mass. Rows one through $N^i - 1$ are populated as shown in Fig. 3. Each subsequent layer begins one column right of the previous layer's rightmost entry and contains a number of 1's corresponding to the number of outgoing cells for each mass of the layer, or the elements of the second row of the fingerprint.

Vector $\mathbf{p}^i$ can be defined as a column vector of length N^i with 1's in every element as shown in (13).

$$
p^i_n = 1 \text{ for } n = 1...N^i \tag{13}
$$

Given (12) and (13), $\left(\mathbf{H}^i\right)^\top \cdot \mathbf{p}^i$ gives a vector who's m'th row is the sum of the incoming cells connected to x^i_m, or $\left(\sum_{n=L^i_m} 1\right)$. Likewise, $\mathbf{G}^i \cdot \mathbf{p}^i$ gives a vector who's m'th row is the sum of outgoing cells connected to x^i_m, or $\left(\sum_{n=R^i_m} 1\right)$. With this the simplifications in (14) can be made, where subscript m implies row m and function diag$(\mathbf{y})$ creates a diagonal matrix from the elements of column vector $\mathbf{y}$.

$$
\begin{aligned}
\left[\sum_{n=L^i_m} x^i_m\right] &= \left[\left(\mathbf{H}^i\right)^\top \cdot \mathbf{p}^i\right]_m \cdot x^i_m & \left[\sum_{n=R^i_m} x^i_m\right] &= \left[\mathbf{G}^i \cdot \mathbf{p}^i\right]_m \cdot x^i_m \\
&= \left[\text{diag}\left(\left(\mathbf{H}^i\right)^\top \cdot \mathbf{p}^i\right)\right]_m \mathbf{X}^i & &= \left[\text{diag}\left(\mathbf{G}^i \cdot \mathbf{p}^i\right)\right]_m \mathbf{X}^i \\
\left[\sum_{n=L^i_m} \dot{x}^i_m\right] &= \left[\left(\mathbf{H}^i\right)^\top \cdot \mathbf{p}^i\right]_m \cdot \dot{x}^i_m & \left[\sum_{n=R^i_m} \dot{x}^i_m\right] &= \left[\mathbf{G}^i \cdot \mathbf{p}^i\right]_m \cdot \dot{x}^i_m \\
&= \left[\text{diag}\left(\left(\mathbf{H}^i\right)^\top \cdot \mathbf{p}^i\right)\right]_m \dot{\mathbf{X}}^i & &= \left[\text{diag}\left(\mathbf{G}^i \cdot \mathbf{p}^i\right)\right]_m \dot{\mathbf{X}}^i
\end{aligned} \tag{14}
$$

Row m of $\mathbf{G}^i \cdot \mathbf{H}^i$ contains the connections between masses in the forward (outgoing) direction while the transpose contains the same information in the reverse (incoming) direction relative to mass m. This allows the simplifications in (15).

$$
\begin{aligned}
\left[\sum_{n=L^i_m} x^i_{L,n}\right] &= \left[\left(\mathbf{G}^i \cdot \mathbf{H}^i\right)^\top\right]_m \cdot \mathbf{X}^i & \left[\sum_{n=R^i_m} x^i_{R,n}\right] &= \left[\mathbf{G}^i \cdot \mathbf{H}^i\right]_m \cdot \mathbf{X}^i \\
\left[\sum_{n=L^i_m} \dot{x}^i_{L,n}\right] &= \left[\left(\mathbf{G}^i \cdot \mathbf{H}^i\right)^\top\right]_m \cdot \dot{\mathbf{X}}^i & \left[\sum_{n=R^i_m} \dot{x}^i_{R,n}\right] &= \left[\mathbf{G}^i \cdot \mathbf{H}^i\right]_m \cdot \dot{\mathbf{X}}^i
\end{aligned} \tag{15}
$$

Since in (12), $\mathbf{G}^i$ and $\mathbf{H}^i$ contain the connections between cells and masses for outgoing and incoming cells respectively, the internal variable and force effects on the masses can be simplified according (16). These are the final simplifications needed to remove the summations from (9) and yield an equation for $\ddot{x}^i_m$ with solely linear algebra construction.

$$
\begin{aligned}
\left[\sum_{n=L^i_m} d^i_{q,n}\right] &= \left[\left(\mathbf{H}^i\right)^\top\right]_m \cdot \mathbf{d}^i_q & \left[\sum_{n=R^i_m} d^i_{q,n}\right] &= \left[\mathbf{G}^i\right]_m \cdot \mathbf{d}^i_q \\
\left[\sum_{n=L^i_m} F^i_n\right] &= \left[\left(\mathbf{H}^i\right)^\top\right]_m \cdot \mathbf{F}^i & \left[\sum_{n=R^i_m} F^i_n\right] &= \left[\mathbf{G}^i\right]_m \cdot \mathbf{F}^i
\end{aligned} \tag{16}
$$

By the reverse logic of (16), the n'th row of $\left(\mathbf{G}^i\right)^\top \cdot \mathbf{X}^i$ and $\mathbf{H}^i \cdot \mathbf{X}^i$ yield $x^i_{L,n}$ and $x^i_{R,n}$ respectively. Likewise, the n'th row of $\mathbf{H}^i \cdot \dot{\mathbf{X}}^i$ and $\left(\mathbf{G}^i\right)^\top \cdot \dot{\mathbf{X}}^i$ yield $\dot{x}^i_{R,n}$ and $\dot{x}^i_{L,n}$ respectively. $d^i_{q,n}$ and F^i_n can be replaced by $\mathbf{I}_n \cdot \mathbf{d}^{\mathbf{i}}_{\mathbf{q}}$ and $\mathbf{I}_n \cdot \mathbf{F}^i$ respectively, meaning the n'th row of the identity matrix multiplied by vector $\mathbf{d}^{\mathbf{i}}_{\mathbf{q}}$ or $\mathbf{F}^{\mathbf{i}}$. These substitutions are shown in (17) and when used in (8) yeld an equation for $d^i_{q,n}$ with solely linear algebra construction.

$$
\begin{aligned}
\left[x^i_{L,n}\right] &= \left[\left(\mathbf{G}^i\right)^\top\right]_n \cdot \mathbf{X}^i & \left[x^i_{R,n}\right] &= \left[\mathbf{H}^i\right]_n \cdot \mathbf{X}^i \\
\left[\dot{x}^i_{L,n}\right] &= \left[\left(\mathbf{G}^i\right)^\top\right]_n \cdot \dot{\mathbf{X}}^i & \left[\dot{x}^i_{R,n}\right] &= \left[\mathbf{H}^i\right]_n \cdot \dot{\mathbf{X}}^i \\
\left[d^i_{q,n}\right] &= \mathbf{I}_n \cdot \mathbf{d}^{\mathbf{i}}_{\mathbf{q}} & \left[F^i_n\right] &= \mathbf{I}_n \cdot \mathbf{F}^i
\end{aligned} \tag{17}
$$

In order to combine terms with like domains, eg. $\mathbf{X}^i_m$ or $\dot{\mathbf{X}}^i_m$, and order the equations according to the desired state-space form, define $\boldsymbol{\alpha}^i$ as (18), $\boldsymbol{\beta}^i$ as (19), $\boldsymbol{\gamma}^i$ as (20), $\boldsymbol{\delta}^i$ as (21), $\boldsymbol{\zeta}^i$ as (22), $\boldsymbol{\eta}^i$ as (23), $\boldsymbol{\iota}^i$ as (24), and $\boldsymbol{\kappa}^i$ as (25) for each actuator array i.

$$\boldsymbol{\alpha}^i \begin{bmatrix} \vdots \\ \mathbf{x}_m^i \\ \vdots \end{bmatrix} = \begin{bmatrix} -\text{diag}\Big(com^i\left(x_{R,n}^i\right) \cdot \left(\mathbf{H}^i\right)^\top \cdot \mathbf{p}^i + com^i\left(x_{L,n}^i\right) \cdot \mathbf{G}^i \cdot \mathbf{p}^i \Big) \\ +com^i\left(x_{L,n}^i\right) \cdot \left(\mathbf{G}^i \cdot \mathbf{H}^i\right)^\top + com^i\left(x_{R,n}^i\right) \cdot \mathbf{G}^i \cdot \mathbf{H}^i \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{x}_m^i \\ \vdots \end{bmatrix} \tag{18}$$

$$\boldsymbol{\beta}^i \begin{bmatrix} \vdots \\ \dot{\mathbf{x}}_m^i \\ \vdots \end{bmatrix} = \begin{bmatrix} -\text{diag}\Big(com^i\left(\dot{x}_{R,n}^i\right) \cdot \left(\mathbf{H}^i\right)^\top \cdot \mathbf{p}^i + com^i\left(\dot{x}_{L,n}^i\right) \cdot \mathbf{G}^i \cdot \mathbf{p}^i \Big) \\ +com^i\left(\dot{x}_{L,n}^i\right) \cdot \left(\mathbf{G}^i \cdot \mathbf{H}^i\right)^\top + com^i\left(\dot{x}_{R,n}^i\right) \cdot \mathbf{G}^i \cdot \mathbf{H}^i \end{bmatrix} \begin{bmatrix} \vdots \\ \dot{\mathbf{x}}_m^i \\ \vdots \end{bmatrix} \tag{19}$$

$$\boldsymbol{\gamma}_q^i \cdot \mathbf{d}_q^i = \Big[-com^i\left(d_{q,n}^i\right) \cdot \left(\mathbf{H}^i\right)^\top + com^i\left(d_{q,n}^i\right) \cdot \mathbf{G}^i \Big] \cdot \mathbf{d}_q^i \ \ , \forall q \tag{20}$$

$$\boldsymbol{\delta}^i \cdot \mathbf{F}^i = \Big[com^i\left(F_n^i\right) \cdot \left(\mathbf{H}^i\right)^\top - com^i\left(F_n^i\right) \cdot \mathbf{G}^i \Big] \cdot \mathbf{F}^i \tag{21}$$

$$\boldsymbol{\zeta}_q^i \begin{bmatrix} \vdots \\ \mathbf{x}_m^i \\ \vdots \end{bmatrix} = \Big[cod^i\left(x_{L,n}^i, q\right) \cdot \left[\mathbf{G}^i\right]^\top + cod^i\left(x_{R,n}^i, q\right) \cdot \mathbf{H}^i \Big] \begin{bmatrix} \vdots \\ \mathbf{x}_m^i \\ \vdots \end{bmatrix} \ \ , \forall q \tag{22}$$

$$\boldsymbol{\eta}_q^i \begin{bmatrix} \vdots \\ \dot{\mathbf{x}}_m^i \\ \vdots \end{bmatrix} = \Big[cod^i\Big(\dot{X}_{L,n}^i, q\Big) \cdot \left[\mathbf{G}^i\right]^\top + cod^i\Big(\dot{X}_{R,n}^i, q\Big) \cdot \mathbf{H}^i \Big] \begin{bmatrix} \vdots \\ \dot{\mathbf{x}}_m^i \\ \vdots \end{bmatrix} \ \ , \forall q \tag{23}$$

$$\boldsymbol{\iota}_q^i \cdot \mathbf{d}_q^i = \left[cod^i\left(d_{q,n}^i, q\right) \cdot \mathbf{I} \right] \cdot \mathbf{d}_q^i \ \ , \forall q \tag{24}$$

$$\boldsymbol{\kappa}_q^i \cdot \mathbf{F}^i = \left[cod^i\left(F_n^i, q\right) \cdot \mathbf{I} \right] \cdot \mathbf{F}^i \ \ , \forall q \tag{25}$$

In order to seperate connection point states according to the state vector (11), $\boldsymbol{\alpha}^i$ and $\boldsymbol{\beta}^i$ must be split according to (26). α_{ab}^i, α_{bc}^i, α_{cd}^i, α_{ad}^i, β_{ab}^i, β_{bc}^i, β_{cd}^i, and β_{ad}^i are scaler values; $\boldsymbol{\alpha}_a^i$, $\boldsymbol{\alpha}_b^i$, $\boldsymbol{\alpha}_c^i$, $\boldsymbol{\alpha}_d^i$, $\boldsymbol{\beta}_a^i$, $\boldsymbol{\beta}_b^i$, $\boldsymbol{\beta}_c^i$, and $\boldsymbol{\beta}_d^i$ are one dimensional column/row vectors; and $\boldsymbol{\alpha}_S^i$ and $\boldsymbol{\beta}_S^i$ contain the remaining values of $\boldsymbol{\alpha}^i$ and $\boldsymbol{\beta}^i$ respectively.

$$\boldsymbol{\alpha}^i = \begin{bmatrix} \alpha^i_{ab} & \boldsymbol{\alpha}^i_b & \alpha^i_{bd} \\ \boldsymbol{\alpha}^i_a & \boldsymbol{\alpha}^i_S & \boldsymbol{\alpha}^i_d \\ \alpha^i_{ac} & \boldsymbol{\alpha}^i_c & \alpha^i_{cd} \end{bmatrix} \qquad \boldsymbol{\beta}^i = \begin{bmatrix} \beta^i_{ab} & \boldsymbol{\beta}^i_b & \beta^i_{bd} \\ \boldsymbol{\beta}^i_a & \boldsymbol{\beta}^i_S & \boldsymbol{\beta}^i_d \\ \beta^i_{ac} & \boldsymbol{\beta}^i_c & \beta^i_{cd} \end{bmatrix} \tag{26}$$

Similarly, $\boldsymbol{\gamma}^i_q$ and $\boldsymbol{\delta}^i$ need to be split according to (27) where $\boldsymbol{\gamma}^i_{q,a}$, $\boldsymbol{\gamma}^i_{q,b}$, $\boldsymbol{\delta}^i_a$, and $\boldsymbol{\delta}^i_b$ are one dimensional row vectors and $\boldsymbol{\gamma}^i_{q,S}$ and $\boldsymbol{\delta}^i_S$ contain the remaining values of $\boldsymbol{\gamma}^i_q$ and $\boldsymbol{\delta}^i$ respectively.

$$\boldsymbol{\gamma}^i_q = \begin{bmatrix} \boldsymbol{\gamma}^i_{q,a} \\ \boldsymbol{\gamma}^i_{q,S} \\ \boldsymbol{\gamma}^i_{q,b} \end{bmatrix}, \ \forall q \qquad \boldsymbol{\delta}^i = \begin{bmatrix} \boldsymbol{\delta}^i_a \\ \boldsymbol{\delta}^i_S \\ \boldsymbol{\delta}^i_b \end{bmatrix} \tag{27}$$

Finally, $\boldsymbol{\zeta}^i_q$ and $\boldsymbol{\eta}^i$ need to be split according to (28) where $\boldsymbol{\zeta}^i_{q,a}$, $\boldsymbol{\zeta}^i_{q,b}$, $\boldsymbol{\eta}^i_a$, and $\boldsymbol{\eta}^i_b$ are one dimensional column vectors and $\boldsymbol{\zeta}^i_{q,S}$ and $\boldsymbol{\eta}^i_S$ contain the remaining values of $\boldsymbol{\zeta}^i_q$ and $\boldsymbol{\eta}^i$ respectively.

$$\begin{aligned} \boldsymbol{\zeta}^i_q &= \begin{bmatrix} \boldsymbol{\zeta}^i_{q,a} & \boldsymbol{\zeta}^i_S & \boldsymbol{\zeta}^i_{q,b} \end{bmatrix}, \ \forall q \\ \boldsymbol{\eta}^i_q &= \begin{bmatrix} \boldsymbol{\eta}^i_{q,a} & \boldsymbol{\eta}^i_S & \boldsymbol{\eta}^i_{q,b} \end{bmatrix}, \ \forall q \end{aligned} \tag{28}$$

The internal dynamics of array i can be written as square matrix (29).

$$\mathbf{Array}^i = \begin{array}{c} \begin{pmatrix} \mathbf{X}^i & \dot{\mathbf{X}}^i & \dots & \mathbf{d}^i_q & \dots \end{pmatrix} \\ \begin{bmatrix} \mathbf{0} & \mathbf{I} & \dots & \mathbf{0} & \dots \\ \boldsymbol{\alpha^i_s} & \boldsymbol{\beta^i_s} & \dots & \boldsymbol{\gamma^i_{q,s}} & \dots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ \boldsymbol{\zeta^i_{q,s}} & \boldsymbol{\eta^i_{q,s}} & \dots & \boldsymbol{\iota^i_q} & \dots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{bmatrix} \end{array} \tag{29}$$

State space form of the system can then be written as $\dot{\mathbf{X}} = \mathbf{A} \cdot \mathbf{X} + \mathbf{B} \cdot \mathbf{u}$ where $\mathbf{A}$ and $\mathbf{B}$ are given by (30) and (31) respectively.

$$\mathbf{A} = \begin{bmatrix} \mathbf{Array}^i & \mathbf{0} & \dots & \mathbf{conA}^i_l & \mathbf{conA}^i_{l+1} \\ \mathbf{0} & \mathbf{Array}^{i+1} & \dots & \mathbf{conA}^{i+1}_l & \mathbf{conA}^{i+1}_{l+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{conB}^i_l & \mathbf{conB}^{i+1}_l & \dots & \mathbf{conC}_l & \mathbf{conD}_{l,l+1} \\ \mathbf{conB}^i_{l+1} & \mathbf{conB}^{i+1}_{l+1} & \dots & \mathbf{conD}_{l+1,l} & \mathbf{conC}_{l+1} \end{bmatrix} \tag{30}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}^i & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{B}^{i+1} & \dots \\ \vdots & \vdots & \ddots \\ \mathbf{conF}_l^i & \mathbf{conF}_l^{i+1} & \dots \\ \mathbf{conF}_{l+1}^i & \mathbf{conF}_{l+1}^{i+1} & \dots \end{bmatrix} \tag{31}$$

The effects of x_l^c on array i are given by $\mathbf{conA}_l^i$ and the effects of array i on x_l^c are given by $\mathbf{conB}_l^i$, both of which are defined in (32).

$$\mathbf{conA}_l^i = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{\alpha}_\lambda^i & \boldsymbol{\beta}_\lambda^i \\ \vdots & \vdots \\ \boldsymbol{\zeta}_{q,\lambda}^i & \boldsymbol{\eta}_{q,\lambda}^i \\ \vdots & \vdots \end{bmatrix} \qquad \mathbf{conB}_l^i = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \dots \\ \boldsymbol{\alpha}_\mu^i & \boldsymbol{\beta}_\mu^i & \dots & \boldsymbol{\gamma}_\mu^i & \dots \end{bmatrix} \tag{32}$$

$$\boldsymbol{\alpha}_\lambda^i = \begin{cases} \boldsymbol{\alpha}_b^i & \text{if } x_l^c = x_1^i \\ \boldsymbol{\alpha}_c^i & \text{if } x_l^c = x_M^i \\ \mathbf{0} & \text{otherwise} \end{cases} \qquad \boldsymbol{\alpha}_\mu^i = \begin{cases} \boldsymbol{\alpha}_a^i & \text{if } x_l^c = x_1^i \\ \boldsymbol{\alpha}_d^i & \text{if } x_l^c = x_M^i \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\boldsymbol{\beta}_\lambda^i = \begin{cases} \boldsymbol{\beta}_b^i & \text{if } x_l^c = x_1^i \\ \boldsymbol{\beta}_c^i & \text{if } x_l^c = x_M^i \\ \mathbf{0} & \text{otherwise} \end{cases} \qquad \boldsymbol{\beta}_\mu^i = \begin{cases} \boldsymbol{\beta}_a^i & \text{if } x_l^c = x_1^i \\ \boldsymbol{\beta}_d^i & \text{if } x_l^c = x_M^i \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\boldsymbol{\zeta}_{q,\lambda}^i = \begin{cases} \boldsymbol{\zeta}_{q,a}^i & \text{if } x_l^c = x_1^i \\ \boldsymbol{\zeta}_{q,b}^i & \text{if } x_l^c = x_M^i \\ \mathbf{0} & \text{otherwise} \end{cases}, \forall q \qquad \boldsymbol{\gamma}_{q,\mu}^i = \begin{cases} \boldsymbol{\gamma}_{q,a}^i & \text{if } x_l^c = x_1^i \\ \boldsymbol{\gamma}_{q,b}^i & \text{if } x_l^c = x_M^i \\ \mathbf{0} & \text{otherwise} \end{cases}, \forall q$$

$$\boldsymbol{\eta}_{q,\lambda}^i = \begin{cases} \boldsymbol{\eta}_{q,a}^i & \text{if } x_l^c = x_1^i \\ \boldsymbol{\eta}_{q,b}^i & \text{if } x_l^c = x_M^i \\ \mathbf{0} & \text{otherwise} \end{cases}, \forall q \tag{33}$$

x_l^c's effect on its own acceleration can be written as (34) and the effect of x_g^c on x_l^c's acceleration can be written as (36). Note that $\mathbf{conD}_{g,l} = 0$ unless an actuator array has only one layer.

$$\mathbf{conC}_l = \begin{bmatrix} 0 & 1 \\ \left(\sum_i \alpha_l^i, \forall i\right) & 0 \end{bmatrix} \tag{34}$$

$$\alpha_l^i = \begin{cases} \alpha_{ab}^i & \text{if } x_l^c = x_1^i \\ \alpha_{cd}^i & \text{if } x_l^c = x_M^i \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

$$\mathbf{conD}_{g,l} = \begin{bmatrix} 0 & 0 \\ \left(\sum_i \alpha^i_{g,l}, \forall i\right) & \left(\sum_i \beta^i_{g,l}, \forall i\right) \end{bmatrix} \tag{36}$$

$$\alpha^i_{g,l} = \begin{cases} \alpha^i_{bd} & \text{if } x^c_l = x^i_1 \text{ and } x^c_g = x^i_M \\ \alpha^i_{ac} & \text{if } x^c_l = x^i_M \text{ and } x^c_g = x^i_1 \\ 0 & \text{otherwise} \end{cases} \quad \beta^i_{g,l} = \begin{cases} \beta^i_{bd} & \text{if } x^c_l = x^i_1 \text{ and } x^c_g = x^i_M \\ \beta^i_{ac} & \text{if } x^c_l = x^i_M \text{ and } x^c_g = x^i_1 \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

$\mathbf{B}^i$ represents the effects of the control (actuator) forces on array i and is constructed according to (38). $\mathbf{conF}_l$ represents the effects of these forces on x^c_l and is constructed according to (39).

$$\mathbf{B}^i = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\delta}^i_S \\ \boldsymbol{\kappa}^i_q \end{bmatrix} \tag{38}$$

$$\mathbf{conF}_l = \begin{bmatrix} 0 \\ \boldsymbol{\delta}^i_\mu \end{bmatrix} \tag{39}$$

$$\boldsymbol{\delta}^i_\mu = \begin{cases} \delta^i_a & \text{if } x^c_l = x^i_1 \\ \delta^i_b & \text{if } x^c_l = x^i_M \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

Substituting (29), (32), (34), and (36) into (30) and (38) and (39) into (31) yields the final state space matricies for the dynamics of the combined system.

EXPERIMENTAL VALIDATION

Physical Actuator Array Design

Two different SMA based actuator cell designs were used to validate the presented theory. The first consisted of two Miga NanoMuscle 704 SMA actuators mounted to a ABS rapid prototyped shell. A compression spring (series elastic element) inside the shell connected to a rod which goes through the spring and out to form the outgoing connection point. The arms of the SMA actuators form the incoming connection points, and are activated together when the cell is activated to cancel any moment that would otherwise be generated. The SMA actuators are significantly stiffer than the compression spring. They also take time to heat when activated and time to cool when deactivated. For these two reasons, they are approximated by a pure force acting across a stiff spring (actuator elastic element) and a damper. Magnet wires were chosen to power the SMA actuators in order to reduce the effect of wire stiffness on the

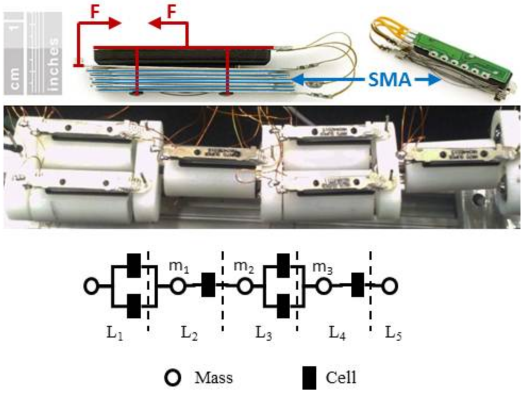

Figure 4: PHYSICAL 6 CELL ARRAY ACTUATOR USED FOR EXPERIMENTAL VALIDATION.

results, however some stiffness remains and was accounted for as the parallel elastic element. The mass, stiffnesses, damping (measured from response time), and force output for each cell were measured using a calibrated Futek 9 newton force transducer powered by an Omega signal conditioner and the results were viewed on a LeCroy 600Mhz scope. The spring constants for the parallel elastic, series elastic, and SMA actuator were found to be 20 N/m, 386.52 N/m, and 1000 N/m respectively by stretching each between two 0.02mm resolution micro-positioners and measuring the output force. The damping coefficient was found to be 500 N/(m/s), and the mass of each mass was 15.83 grams. 6 cells were set up in an isometric contraction arrangement as shown in Fig. 4. The damper element connected in parallel with the series elastic element in the theory was negligible, allowing the dynamics to play a greater role in validating the theory. Additionally, the simple Miga NanoMuscle SMA actuators can be represented accurately for the all-on all-off case as a spring and damper in parallel with a constant pure-force acting on it as an input, as shown in Fig. 5. This allows for the contractile element to be less of a black box, again allowing for further validation of the theory. The setup, however, was extremely fragile and while it produced useful dynamic validation data it is not suggested for use in practical applications. Over the course of the experiment the Miga NanoMuscles tore themselves apart repeatedly. As such the experimental results from this setup were somewhat limited in scope as the authors did not wish to spend a large amount of time and resources to continually rebuild the array.

The significantly more practical silicone-based SMA actuator array design was inspired by the M-lines and Z-lines of sarcomeres in biological muscle and the interaction between the myosin and actin fibers which generate displacement and subsequent contractile force. Fig. 6 shows the molecular representation and actual structure, as seen under a light microscope, of a sarcomere. The actuation of the muscle is provided by myosin interacting with actin in the presense of ATP to create contractile displacement between the Z-line and M-line on either side of the sarcomere. Additionally, the strands of actin and the connecting titin fibers are flexible allowing the contractile displacement

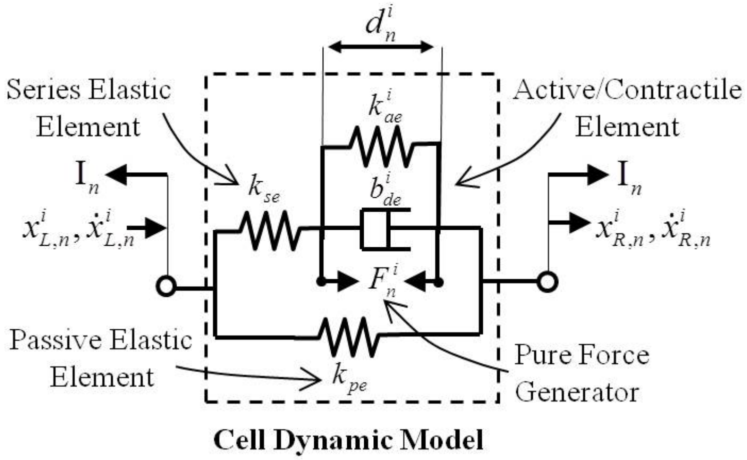

Figure 5: MIGA NANOMUSCLE 704 SMA ACTUATOR CELL MODEL.

of the sarcomere to be translated into contractile force when under the influence of an external load or blocking force. Individual sarcomeres have a consistent contraction length and additional overall displacement, or similarly force, is built by recruiting or activating more sarcomeres in the muscle, not by further displacing a given sarcomere.

The silicone rubber based SMA actuator array, shown in Fig. 7, like its Miga NanoMuscle counterpart capitalizes on this idea of recruitment to negate the effect of hysterisis, an inherent drawback with SMA actuators along with many other linear actuation technologies. The SMA wires behave as a hybrid between actin and myosin fibers. When heated via electrical current, the SMA shifts from its martensite phase to its austensite phase which is roughly 3-4% of its length in the martensite phase. The silicone connecting structure takes the place of the flexible actin and titin fibers and thus provides the translation between displacement and force based on external loading conditions and the length of other cells. As more force or displacement is desired, additional cells are activated causing additional pre-loading of the silicone 'springs' and thus additional force or displacement to the external environment. In the original design a rigid element existed between two sets of SMA wires for each cell to better emulate the M-line of biological muscle, however this was dropped for construction simplicity as a single set of myosin/actin fibers acting against two Z-line elements has roughly the same dynamics as two sets with a rigid M-line between them.

Flexinol 100 m HT SMA wire produced by Dynalloy Inc. was selected as the actuation material primarily for its balance between force and cooling time. Larger diameter SMA wires can generally produce higher forces, but take significantly longer to cool while smaller diameter wires produce significantly less force but also cool more rapidly. Future research will look into combining larger and smaller diameter wire cells in the same actuator array similarly to how biological muscle has slow and fast twitch sarcomeres, however this was not considered in the current work. The spring constant of the Flexinol 100 m HT SMA wire was determined experimentally to be 2.54N/mm

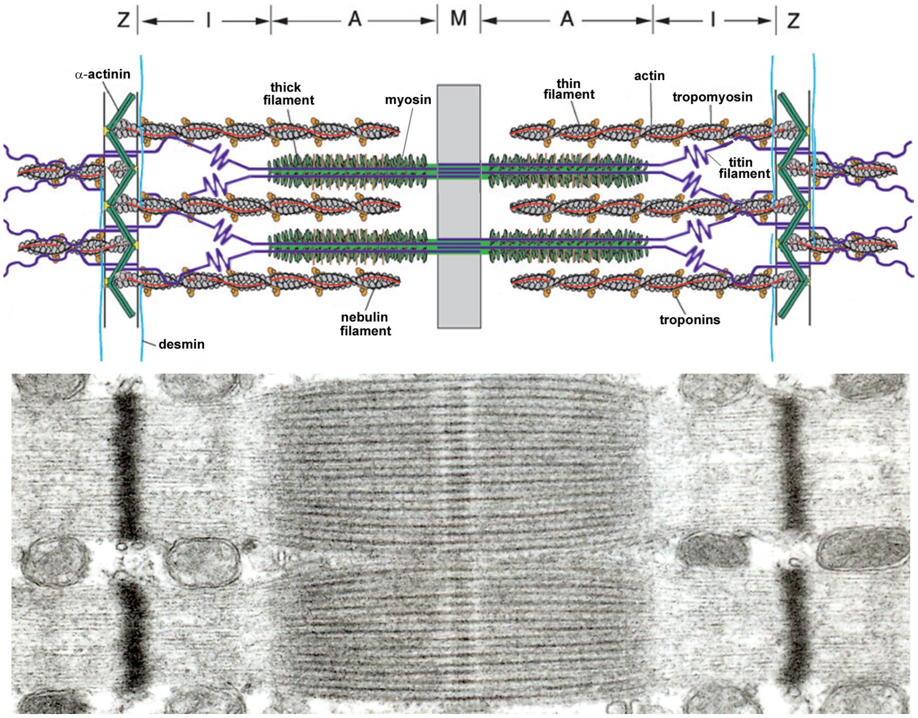

Figure 6: MOLECULAR REPRESENTATION AND STRUCTURE OF A SARCOMERE. IMAGE TAKEN FROM [21] AND USED WITH PERMISSION UNDER THE CREATIVE COMMONS LICENSE.

for the 75mm length of wire used in the final cell array. This was determined using a Futek LBS200 five-pound force sensor and two .02mm resolution micro-positioners with the SMA wire in its relaxed Martensite phase. While the spring constant of the SMA wire does vary greatly between the martensite (relaxed) and austenite (active) phase, the relaxed martensite phase is lower than the active austenite phase and both values are an order of magnitude above that of the silicone connecting structure. For this reason, the value of the martensite phase was taken as the actuator's stiffness. Additionally, during the testing phase the wire was subjected to approximately 8.45 N and remained in its elastic phase. This is roughly five times the force an individual SMA wire experiences in the final actuator array which suggests that, coupled with the silicone connective structure, the SMA wires remain linear and resistant to breaking. In order to increase the force output of a single cell, four wires were used in parallel. Likewise to increase the displacement, the wires were wrapped around a Z-line bracket as shown in Fig. 8. The silicone chosen needed to have a stiffness less than that of the additive SMA wires force for a given displacement so that the SMA wires can always achieve their full displacement. The actual stiffness of the silicone connecting structure is dependent both on the stress-strain properties of the silicone and the geometry of the connecting structure. Several silicone rubber sheets were tested experimentally using the same sensor and micro-positioners as for the SMA testing, and it was determined

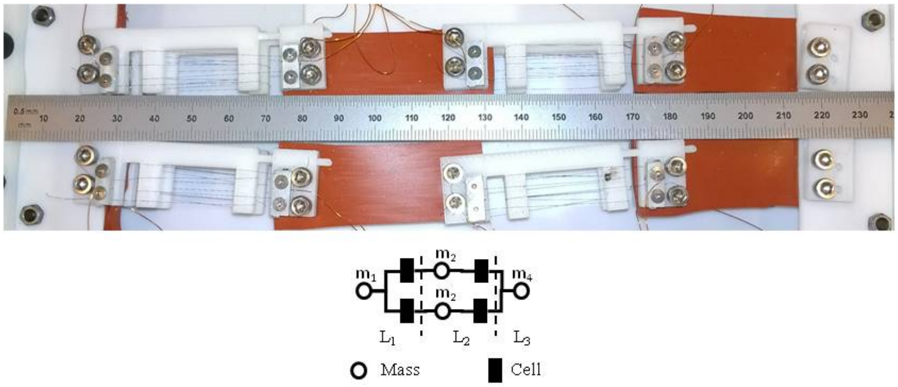

Figure 7: SILICONE RUBBER BASED ACTUATOR ARRAY.

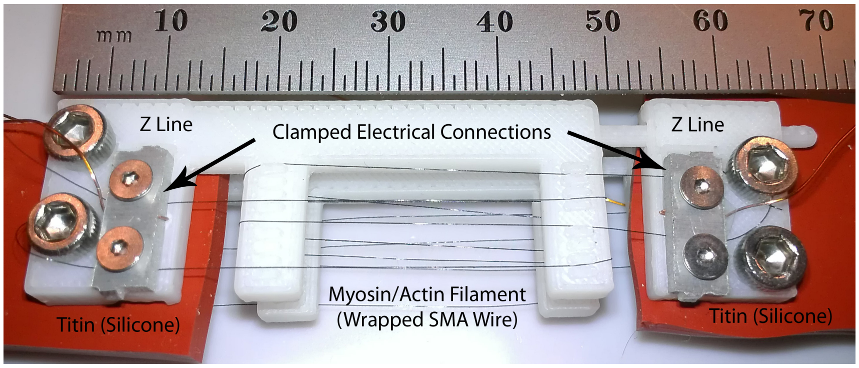

Figure 8: SILICONE RUBBER BASED ACTUATOR ARRAY CELL.

that shore A 20 durometer silicone rubber provided favorable properties. A 30mm x 62mm x 6mm section of silicone rubber, the size used for the final cells, was tested producing a final spring constant of 0.611 N/mm. A stiffer silicone sheet could have been used and would have increased the strength of the resultant muscle significantly, however this was not done for experimental setup reliability reasons. The Z-line brackets were printed with a 3D rapid prototyper using ABS plastic for ease of construction and plastic's inherent electrical resistivity.

While the four parallel SMA wires in each cell for this experimental setup have a theoretical force capability of 18 N and a displacement capability of 5 mm, manufacturing errors caused the actual displacement to be 0.7 mm and thus the effective control force of the cell was $(0.7mm) \cdot (2.54N/mm) = 1.7N$. The effective applied force of the cell was therefore $(0.7mm) \cdot (0.61N/mm) = 0.41N$. While this is significantly lower than what is needed for implimentation in a full scale muscle, a different choice of silicone sheet and an improved design and manufacturing process will dramatically increase the properties. The current SMA wires also take roughly a second to fully heat and to fully cool, however introducing additional cooling mechanisms such as forced

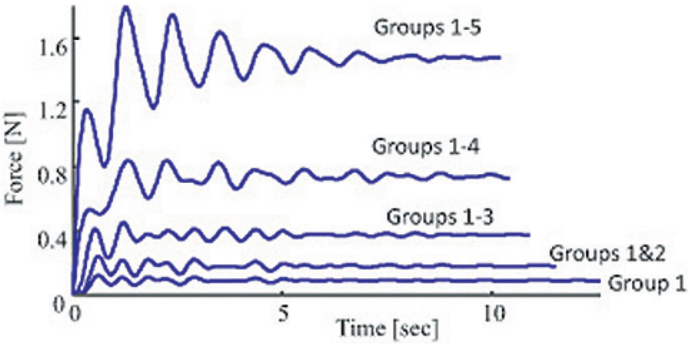

Figure 9: FLOATING-POINT QUANTIZED ACTUATION OF AN NON-UNIFORM ACTUATOR ARRAY.

air or submersion in an oil bath can significantly improve this as well. As the focus of the current research was to validate the presented theory and show viability of the experimental setup, these changes were left to future work.

The properties of a given actuator array are also highly dependent on the topology and choice of internal cell actuation properties (such as SMA wire size). The presented design is highly modular and, depending on the chosen topology, highly robust. Cells are connected together through the silicone structure thus reconfiguring the muscle is as easy as cutting/punching out a different silicone shape and connecting the Z-line elements to it. Likewise, if a cell fails it can simply be disconnected from the surrounding cells and replaced. Parallel structures add robustness meaning if a cell fails either due to an electrical failure (the cell goes dead but still remains intact) or mechanical failure (cell physically breaks leaving zero stiffness) the remaining cells are still able to carry force. Thus an arm using the muscle may have a reduced force capacity but will still be able to function. Take as an illustrative example a robot arm hammering in a dangerous environment. If a critical number of cells fail, the robot may not be able to continue hammering but would still have the force capacity to secure the area and move itself to safety in order to be repaired. Current systems fall limp after a single motor failure requiring an additional robot or human to 'rescue' the robot, putting additional equipment, and possibly lives, at risk.

Simulation

The presented theory was programmed into Mathworks MatLab and the resulting equations of motion were simulated using Matlab's ODE45 numerical solver. Fig. 2, Fig. 4, and Fig. 7 show the topologies for the two physical systems used for theoretical validation plus one additional system with a greater number of cells and a more complex structure as an illustrative example.

For the system in Fig. 2, Fig. 9 shows step responses when the cells in the array are non-uniformly grouped as Group 1={Cell 2}, Group 2={3}, Group 3={4, 8}, Group 4={9, 10, 11}, and Group 5={6, 12, 13, 14}. Cells 1, 5, 7 were not used in this case. The progressive activation inspired by the size principle [1] achieves a fine resolution for a small motor command and a more coarse resolution for a larger command. This is known in physiology as signal dependent noise, a commonly observed yet otherwise unexplained phenomena in biological muscle[7]. Additionally, the function of the signal dependant noise can be controlled by choosing appropriate actuator array topologies and/or different cell force levels yielding linear, exponential, or any number of other monotonically increasing profiles. Note that there is a number of ways in connecting actuator units, mechanically, electrically, or a mixture of these, to realize such actuator groupings. The responses in Fig. 9 show fluctuations from convergence values, mainly due to underdamped modes in the system. Muscle forces during contractions also show fluctuations [7]. Although it has not been fully investigated yet, the identification of such fluctuations would provide an interesting insight into the neuromotor variability. These ideas are explored more fully in [22].

The equations of motion for the three systems, the two physical systems and the one complex example, were also determined by hand from base Newtonian principles and compared with the simulated results providing an additional layer of validation. The hand derivation was timed and compared with the computer's derivation using the presented theory to show the utility of the presented theory, especially with comparing different topologies in a design process or determining robustness properties when cells mechanically fail.

These equations of motion and simulations are critical to developing optimal control strategies for the actuator arrays. Development of the optimal control strategies, however, is highly dependent on the desired application and is left to future work allowing the current work to focus on the dynamics, model identification, and theory validation.

Results

Experimental validation of the above theory was carried out in two stages. The first utilized the highly damped silicone rubber based actuator array shown in Fig. 7 and Fig. 8 to extend the results of [10] to the time domain. In [10] the topology of actuator arrays was used to determine the final actuator array properties, such as force level, but all experimental justification used steady state values. In the current work, the four cells were activated from rest at four different levels: 1 cell on, 2 cells on, 3 cells on, and all cells. For each, the force activation profile was the Sigmoid function $\left(F \cdot \left(\frac{1}{1+e^{-5 \cdot t+6}}\right)\right)$ followed by a similar inverse Sigmoid function $\left(F \cdot \left(1 - \frac{1}{1+e^{-5 \cdot (t-t_d)+6}}\right)\right)$ where t_d is the deactivation time and F is the force of the cell. This function was determined experimentally by measuring the SMA wire contraction profile. The results for each trial are shown in Fig. 10 with the simulation results overlaid. Due to the system's high natural frequency and high damping, the results for the silicone based SMA actuator do not highlight dynamic effects but they do show that predicted force levels of the theory and simulation match those of the experimental results.

Fig. 11 shows force results for the second experimental validation step. All cells were activated for 3 seconds and then deactivated and, as can be seen in the graph, the results track the simulated values very closely including rises and falls in the graph due to individual cell inertia. The track is not perfect, partly because all mass elements were assumed to be uniform in the simulated results whereas in the physical system

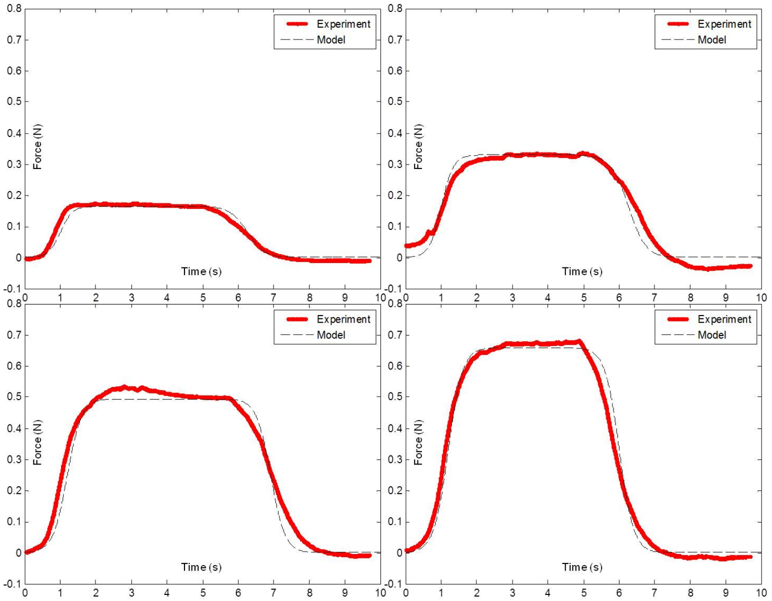

Figure 10: COMPARISON OF 6 CELL ARRAY ACTUATOR PHYSICAL SYSTEM AND SIMULATED RESULTS. ALL CELLS WERE ACTIVATED FOR 3 SECONDS AND THEN DEACTIVATED.

they differed between connection points with two outgoing cells and those with only one outgoing cell. While the experimental setup could be upgraded to normalize the masses, usable actuator arrays will likely have non-uniform masses. This is currently a limitation but is currently being addressed and will be presented in future contributions. It should be noted that no scaling of the physical system nor the simulated results was done and all cell properties were calculated prior to conducting any trials to remove any unintended bias. The results validate the dynamic aspects of the presented theory as the system was not damped. Additionally, since the control input was simply a step acting on a damper it can be seen that no input shaping was done to introduce a bias which further validates the theory.

The Simulink results also matched the theory and physical systems, however programming the system into Simulink from base principles took roughly 15 minutes for the 6-cell model versus 1 minute to form the fingerprint and a negligible computation time with the presented theory. Results were similar for the silicone based actuator array. More complex topologies took significantly longer to program from base principles while topoligical complexity did not affect the provided theory. Finally, any changes to the model, either topological or to cell equations of motion, required a complete reprogram of the Simulink model versus changing only a few constants using the above theory.

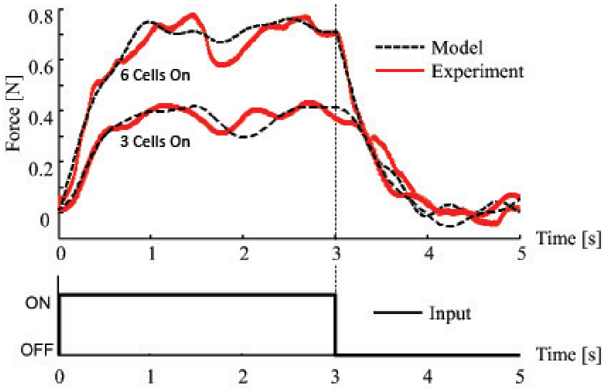

Figure 11: COMPARISON OF 6 CELL ARRAY ACTUATOR PHYSICAL SYSTEM AND SIMULATED RESULTS. ALL CELLS WERE ACTIVATED FOR 3 SECONDS AND THEN DEACTIVATED.

CONCLUSION

This paper presented a generalized expanded fingerprint method which systematically calculates the equations of motion for any linear actuation technology so long as 1) the actuator follows a known displacement versus time function when activated, 2) that function is minimally dependent on external force, and 3) the actuator can be represented as a spring, rigid or flexible, with the displacement being modeled as a pure force preloading the spring. Additionally, the generalized method allows for any internal cell dynamics so long as 1) cells connect solely to masses (thus the only external states needed for cell dynamics are the mass positions and velocities) and 2) the forces at each side of the cell are identical (cell has negligible mass). The presented method aids automation of the discovery of the equations of motion, allows for fast recalculation for different cell array topologies, and provides an intuitive base for future controls work on cell array actuators. While the dynamics representing a given cell array actuator could be generated using other means the presented expanded fingerprint method allows the dynamics to be calculated with less human effort, less computational effort, and with greater speed, especially when comparing different topologies and internal cell dynamics. Finally, the state variables in the presented method have physical significance which greatly aids in intuitively understanding the resultant dynamics.

This paper also provided two different physical cell array actuators. The first, a Miga NanoMuscle 704 SMA based actuator was used primary to validate the dynamic response of the presented theory but was not reliable enough for wider scale implimentation. The second actuator, a SMA system with silicone rubber connecting structures is meant as a guide for building future muscle-like actuators and to provide a biological muscle like base for which to run controls experiments. Both experimental systems matched simulations based on the presented theory. Additionally, simulation results for a more complex 14 cell actuator were presented and shown to exibit properties of biological muscles such as signal dependent noise.

Future Work

Future work will continue to improve the design of the cell array actuators, will begin research into effective control strategies for the arrays, and will apply the actuator arrays and theory to multi-degree of freedom systems. While the full detail given by the presented method may not be needed for multi-degree of freedom systems, the results can be simplified for these systems by simplifying the internal cell dynamics or applying a Henkel Normilization or similar process to the final result. Comparing the presented method with a reduced version, ex. in simulation, will show the validity of the simplification while allowing the simplified model to be used for real-time control. A system identification method will also be developed to model complex actuator array topology and cell internal dynamics. The cumulative research will aid in both learning the mechanisms for generating natural movements and in building robotic systems for use in therapy and rehabilitation, prosthetic devices, human force amplification exoskeletons, and humanoid robotic systems.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. ECCS-0932208.

References

[1] Enoka, R. M., 2008. *Neuromechanics of Human Movement*. Human Kinetics Publishers.

[2] Yamaguchi, G. T., 2005. *Dynamic Modeling of Musculoskeletal Motion: A Vectorized Approach for Biomechanical Analysis in Three Dimensions*.

[3] Van Zandwijk, J. P., Bobbert, M. F., Baan, G. C., and Huijing, P. A., 1996. "From twitch to tetanus: performance of excitation dynamics optimized for a twitch in predicting tetanic muscle forces.". *Biological cybernetics,* **75**(5), Nov., pp. 409–417.

[4] Van Zandwijk, J. P., Bobbert, M. F., Harlaar, J., and Hof, A. L., 1998. "From twitch to tetanus for human muscle: experimental data and model predictions for m. triceps surae.". *Biological cybernetics,* **79**(2), Aug., pp. 121–130.

[5] Johansson, G., 1973. "Visual perception of biological motion and a model for its analysis". *Perception & Psychophysics,* **14**(2), pp. 201–211.

[6] Troje, N. F., 2002. "Decomposing biological motion: a framework for analysis and synthesis of human gait patterns.". *Journal of Vision,* **2**(5), Jan., pp. 371–87.

[7] Jones, K. E., Hamilton, A. F. D. C., Wolpert, D. M., Dimitriou, M., and Franklin, D. W., 2002. "Sources of signal-dependent noise during isometric force production". *Journal of Neurophysiology,* **88**(3), pp. 1533–1544.

[8] MacNair, D. L., and Ueda, J., 2009. "Modeling & characterizing stochastic actuator arrays". In International Conference on Intelligent Robots and Systems, pp. 3232–3237.

[9] Ueda, J., Odhner, L., and Asada, H. H., 2007. "Broadcast Feedback of Stochastic Cellular Actuators Inspired by Biological Muscle Control". *The International Journal of Robotics Research,* **26**(11-12), Nov., pp. 1251–1265.

[10] MacNair, D., and Ueda, J., 2011. "A fingerprint method for variability and robustness analysis of stochastically controlled cellular actuator arrays". *The International Journal of Robotics,* **30**(5), pp. 536–555.

[11] MacNair, D. L., and Ueda, J., 2012. "Expanded Fingerprint Method for Analysis of Dynamic Cellular Actuator Arrays". In ASME Dynamic Systems and Controls Conference.

[12] Sass, L., McPhee, J. M. C., Schmitke, C., Fisette, P., and Grenier, D., 2004. "A Comparison of Different Methods for Modelling Electromechanical Multibody Systems". *Multibody System Dynamics,* **12**(3), pp. 209–250.

[13] McPhee, J. J., 1996. "On the use of linear graph theory in multibody system dynamics". *Nonlinear Dynamics,* **9**(1-2), Feb., pp. 73–90.

[14] Scherrer, M., and Mcphee, J., 2003. "Dynamic Modelling of Electromechanical Multibody Systems". *Multibody System Dynamics,* **9**(1), pp. 87–115.

[15] Shi, P., and McPhee, J., 2002. "Symbolic Programming of a Graph-Theoretic Approach to Flexible Multibody Dynamics". *Mechanics of Structures and Machines,* **30**(1), pp. 123–154.

[16] Diaz-Calderon, A., Paredis, C. J. J., and Khosla, P. K., 2000. "Automatic generation of system-level dynamic equations for mechatronic systems". *Computer-Aided Design,* **32**(5-6), May, pp. 339–354.

[17] Gao, W., and Wang, H., 2012. "An Automatic Dynamics Generation Method for Reconfigurable Modular Robot". *Advances in Reconfigurable Mechanisms and Robots I,* **5**, pp. 551–560.

[18] Chandrashekar, M., Roe, P. H., and Savage, G. J., 1993. "A Unified Approach to Modelling Photovoltaic Power Systems". *Modeling and Simulation,* **23**(4), pp. 313–313.

[19] Madill, D. R., and Wang, D., 1998. "Modeling and L2-Stability of a Shape Memory Alloy Position Control System". *IEEE Transactions on Control Systems Technology,* **6**(4), pp. 473–481.

[20] Hill, A. V., 1938. "The Heat of Shortening and the Dynamic Constants of Muscle". In Proceedings of the Royal Society of London. Series B, Biological Sciences, Vol. 126, pp. 136–195.

[21] Ottenheijm, C. A. C., Heunks, L. M. A., and Dekhuijzen, R. P. N., 2008. "Diaphragm adaptations in patients with COPD.". *Respiratory Research,* **9**(12), Jan.

[22] Ueda, J., MacNair, D., and Brown, E., 2012. "Quantized Control of Compliant Cellular Actuator Arrays for Biological Movement Generation". In ASME Dynamic Systems and Controls Conference.