



UNIVERSITY OF LEEDS

This is a repository copy of *Accurate multivariable arbitrary piecewise model regression of McKibben and Peano muscle static and damping force behavior*.

White Rose Research Online URL for this paper:
<http://eprints.whiterose.ac.uk/137524/>

Version: Accepted Version

Article:

Veale, AJ, Xie, SQ and Anderson, IA (2018) Accurate multivariable arbitrary piecewise model regression of McKibben and Peano muscle static and damping force behavior. *Smart Materials and Structures*, 27 (10). 105048. ISSN 0964-1726

<https://doi.org/10.1088/1361-665X/aadf41>

© 2018 IOP Publishing Ltd. This is an author produced version of a paper published in *Smart Materials and Structures*. Uploaded in accordance with the publisher's self-archiving policy.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Accurate multivariable arbitrary piecewise model regression of McKibben and Peano muscle static and damping force behavior

Allan Joshua Veale¹, Sheng Quan Xie² and Iain Alexander Anderson³

¹ Biomechanical Engineering, Faculty of Engineering Technology, University of Twente, The Netherlands

² School of Electronic and Electrical Engineering, Faculty of Engineering, University of Leeds, England

³ Biomimetics Laboratory, Auckland Bioengineering Institute, The University of Auckland, New Zealand

E-mail: a.j.veale@utwente.nl, s.q.xie@leeds.ac.uk and i.anderson@auckland.ac.nz

August, 2018

Abstract. Machines that efficiently and safely interact with the uncertainty of the natural world need actuators with the properties of living creatures' muscles. However, the inherent nonlinearity of the static and damping properties that the most promising of these muscle-like actuators have makes them difficult to control. Our ability to accurately control these actuators requires accurate models of their behavior. One muscle-like actuator for which no accurate models have been specifically developed is the Peano muscle. This paper presents and validates a model generation algorithm, Multivariable Arbitrary Piecewise MOdel REgression (MAPMORE), that produces accurate models for predicting the static and damping force behavior of Peano muscles, as well as of the popular McKibben muscle. MAPMORE builds a training data processing, muscle-specific model term dictionary, and piecewise function fusion framework around Billings *et al's* Forward Regression Orthogonal Least Squares (FROLS) estimator algorithm. We demonstrate that MAPMORE's static and damping force models have a Normalized Root Mean Square Error (NRMSE) of 48% to 88% of the NRMSE of the most accurate of Peano and McKibben muscles' existing models. The improved accuracy of MAPMORE's models for these artificial muscles potentially aids the muscles' ability to be accurately controlled and hence is a step towards enabling machines that interact with the real world. Further steps could be made by improving MAPMORE's accuracy through the addition of hysteresis operator and lagged terms in the damping force dictionary.

Keywords: fluidic muscle, McKibben muscle, Peano muscle, MAPMORE, soft actuator, static and damping behavior, modeling

Submitted to: *Smart Mater. Struct.*

1. Introduction

Robotics can benefit, particularly in environments with an unpredictable stiffness, from drive systems (actuators) with the inherent physical softness, or compliance of biological muscles. Adding compliance to actuators can improve their ability to safely absorb impacts and interact with objects [1]. Robots have traditionally used actuators such as servo hydraulics or electric motors, which suited robots' original purpose of helping mankind by automating manual labor tasks such as manufacturing and assembly. These tasks were confined to structured, predictable environments. Now, the current challenge is to extend robots' abilities to helping humans in less constrained environments. Example applications include wearable robots that help with the restoration or rehabilitation of people with disabilities [2–4]; automated vehicles that find and rescue people from disaster zones [5]; and devices that operate in hazardous areas [6]. These applications all put robots in unpredictable environments where an unexpected interaction between the robot and an object could damage either of them. This is particularly due to the rigid, high stiffness properties of traditional robots' actuators and structures. A fail-safe solution to this problem is found in the wholly soft and elastic design of biological muscles. This allows living creatures to safely and repeatedly absorb unexpected impacts as well as effectively interact with hard and soft environments [1].

Inspired by the success of biological muscles a large number of manmade artificial muscles have been developed with the aim of mimicking muscles' properties [7–15]. The most commonly used artificial muscles are fluidic muscles, chosen for their soft form [16], high forces [17], ease of manufacturing [18], and reliance on readily available materials and power sources [19, 20].

A fluidic muscle consists of a variable volume pressure chamber. Pressurizing the chamber with the fluid alters its volume, and the chamber's structure couples its volume change to an attached load so the muscle can perform mechanical work on that load. Depending on the materials and geometry of the pressure chamber various movements are possible: contraction [12–15], extension [21], torsion [22], bending [23], and combined three-dimensional (3D) [24, 25].

The most popular fluidic muscle is the McKibben muscle [26, 27], chosen for its simple construction, high force, and inherent compliance [28]. Figure 1a shows how it consists of a rubber tube wrapped in a stiff fiber braid. Pressurizing the tube causes it to increase in diameter, expanding and shortening the braid so the muscle contracts.

The Peano muscle [22], referred to elsewhere as the muscle-motor [14], pouch motor [29], and flat Pneumatic Artificial Muscle (PAM) [30], also contracts when pressurized and can produce high forces. These forces are proportional to its planform area, whereas the McKibben muscle's force scales with its cross-sectional area. Proposed by Mettam in 1959 [14], it is inherently compliant, yet, compared to a McKibben muscle, its pressure chamber has no volume when deflated, potentially resulting in a more compact actuator [31]. It is made of a row of side-by-side inelastic yet flexible tubes. When pressurized,

the tubes' flat deflated form transforms to a cylinder, shortening the length of the tube structure (figure 1b).

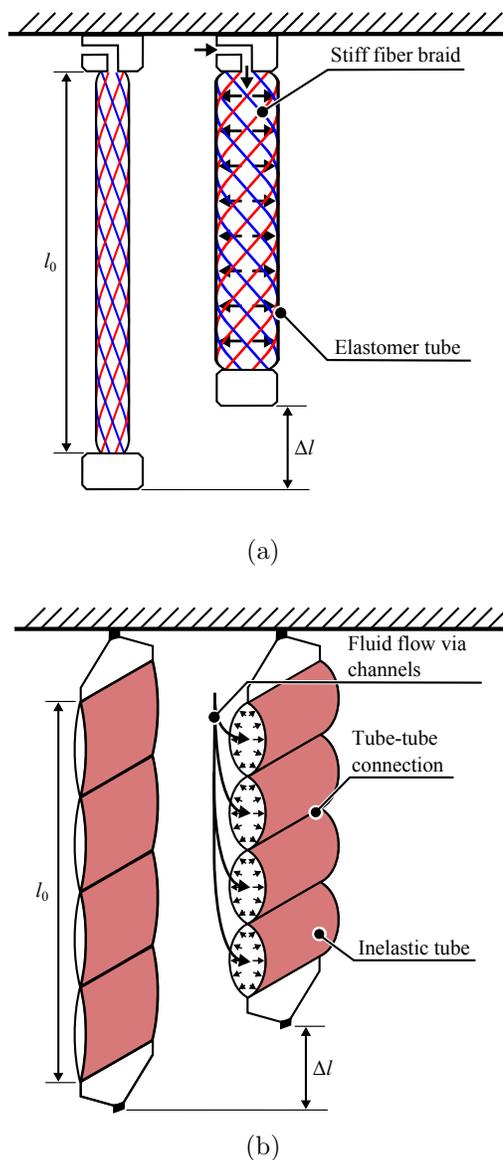


Figure 1: Operating principle of the McKibben (a) and Peano (b) muscles. Pressurizing both muscles shortens them from an initial length l_0 by a displacement Δl . In the McKibben muscle this is achieved by the use of a stiff fiber braid that couples radial expansion of an internal elastomer tube to a contraction of its length. In the Peano muscle, undeflated side-by-side tubes with a flat cross-section tend towards a circular cross-section when pressurized, causing the tube assembly to contract.

The muscles used for testing the models in this work were the commercial McKibben PAM shown in figure 2a (DMSP-20-88N-RM-RM, Festo AG & Co. KG, Esslingen am Neckar, Germany) and the Peano muscle shown in figure 2b (four tubes 0.046 m long and 0.017 25 m wide as in [31]).



Figure 2: The McKibben (a) and Peano (b) muscles used to validate the MAPMORE model

The disadvantage to using artificial muscles such as McKibben and Peano muscles is that they are harder to control than hydraulic and electric actuators. Compared to the rigid materials of traditional actuators, their behavior is nonlinear and challenging to predict [32, 33]. This arises from the complex physics and stochastic properties of their flexible structure and soft materials [33, 34]. Improved control can be achieved by modeling the muscles' behavior, as muscle models can be used to predict how the muscle should behave given a desired muscle behavior. However, the muscles' nonlinear behavior is also challenging to model accurately [33].

The primary contribution of this paper is aimed at producing more accurate models of McKibben and Peano muscles. The hope is that these models will improve the muscles' controllability and hence make them more readily applicable in uncontrolled environments. Specifically this paper presents the validation of Multivariable Arbitrary Piecewise Model REgression (MAPMORE) generated static and damping force models as more accurate than existing accurate models for Peano and McKibben muscles. It also builds on our past work's application of MAPMORE to static behavior [35] by presenting the validated application of MAPMORE to generating muscle damping force models.

The paper is organized as follows: section 2 overviews accurate fluidic muscle static and damping force models; section 3 explains the MAPMORE algorithm and its application to generating damping force models; section 4 presents the validation methods; sections 5 and 6 describe the results and discussion; and section 7 summarizes the conclusions of this work.

2. Fluidic muscle modeling

What kind of model is suitable for accurately predicting the behavior of a fluidic muscle, and particularly for its control? In addition to a model with a low prediction error, it is also desirable to have a versatile and transparent model. Here a versatile model is taken to be applicable to different types of fluidic muscles (for example McKibben and Peano) and different types of behavior. Behaviors considered in this work are static, nonhysteretic force generation for slow movements, and damping force generation for fast, dynamic movements.

Static force, F_s , is the force a muscle produces for a given constant (steadystate) length and pressure. Damping force, F_d , is the difference between the measured force of a muscle, F , and the force it is predicted to produce based on a model of its static force. The relationship between these quantities is simply $F = F_s + F_d$. Hence, if the muscle is only producing static force, $F_d = 0$ and $F = F_s$. Otherwise $F_d \neq 0$.

Hysteresis is also an important muscle behavior [36, 37]. Whenever a muscle moves there will be an energy loss resulting in a hysteresis loop, which varies in size and shape with muscle type. Depending on the application of the muscle, this loop, if unmodeled, may cause model inaccuracy that affects muscle control. Hysteresis modeling's complexity requires it to be treated separately and so it is assumed outside the scope of this work. Potential models that could be considered for muscle hysteresis modeling include those of Vo-Minh *et al* [36], Lin *et al* [37], and Van Damme *et al* [38].

A transparent model is desirable as it contains easily understood terms. These are mathematically simple terms that allow the nature of the muscle's behavior to be interpreted by visual inspection. Additionally, they often have a low computation time for fast execution on a real-time control system. Given these model characteristics, we next survey approaches documented in the literature for the accurate static and damping force modeling of McKibben and Peano muscles.

2.1. Static force models

Physics-based [26, 39, 40], empirical [41, 42], finite element [43], and system identification [44] approaches have all been used to model the static behavior of the McKibben muscle with varying degrees of accuracy. The most accurate of these is Martens and Boblan's [40] physics-based model with a maximum full-scale prediction error of 2.35%. Note that full-scale error is the unsigned ratio of prediction error to the largest measured value in the validation data. The next most accurate was Hošovský *et al*'s [44] adaptive neuro-fuzzy inference system, a system identification approach modeling static force with a full-scale prediction error under 3%. The empirical approach of Sárosi was similarly accurate with a maximum full-scale error of 3.59% [40]. Other approaches tend to have full-scale errors of 5% and higher [26, 39, 43].

Comparing these models' transparency and suitability for control highlights that Hošovský *et al*'s [44] modeling approach is nontransparent. Martens and Boblan's [40] model is specific to the physics of the McKibben muscle and therefore not directly

applicable to the Peano muscle. Hošovský *et al's* approach is considered nontransparent as system identification constructs models by fitting general model terms and coefficients to data. The resulting model can be very accurate, but it is difficult to use it to gain insight into the physical system behind the data.

The Peano muscle's static force has also been modeled using the physics-based virtual work [14, 29], empirical [35, 41], and lumped parameter [31] methods. Maximum full-scale errors with these methods were 8.5 [29], 7.5 [35], and 19.5% [45] respectively. The most accurate of these is Sárosi's model, which has been also demonstrated to be relevant to the McKibben muscle [40]. The second most accurate, Niiyama *et al's* model, is based on the physics of the Peano muscle, and hence is not relevant to the McKibben muscle.

2.2. Damping force models

Damping force models in the literature for the McKibben muscle are limited to combinations of physics-based and empirical terms with empirically fitted coefficients. These are often combined with a static model to form a lumped parameter force model [46]. As there are relatively few damping force models they are separately described. This is also important because, as will be explained later, they have relevance to the application of MAPMORE to damping. These damping force models are summarized in table 1 and rely on linear and nonlinear functions of strain ϵ (where $\epsilon = \Delta l/l_0$ with reference to figure 1), strain rate $\dot{\epsilon}$, and pressure P .

Table 1: Terms used to calculate the damping force component in dynamic McKibben PAM models from the literature. Note that these models use strain rate rather than the muscle velocity used in the original source terms. This is for consistency with the use of strain in static modeling of muscles in this work.

Name	Terms
Linear viscous [47]	$B_0\dot{\epsilon}$
Pressure dependent viscous [48]	$B_1P\dot{\epsilon}$
First order pressure dependent viscous [46, 49]	$(B_0 + B_1P)\dot{\epsilon}$
Hysteresis loss viscous [50]	$(2\zeta(P)\sqrt{ k_t(P, \epsilon) m})\dot{\epsilon}$
Kinetic friction [47]	$\text{sgn}(\dot{\epsilon})\left((B_0 - B_1)e^{\frac{-\dot{\epsilon}}{B_2}} + B_1\right)$
Generalized kinetic friction [51]	$\text{sgn}(\dot{\epsilon})\left(\sum_{k=0}^2 B_k\dot{\epsilon}^k + B_3e^{\frac{- \dot{\epsilon} }{B_4}}\right)$

As shown in table 1, the simplest damping model in the literature is the linear viscous term used by Tondu [47]. This is easily extended to Kerscher *et al's* [48] pressure

dependent viscous term, and then the two combined in Reynolds *et al*'s [46] first order pressure dependent viscous model. Unlike the previous models, Reynolds *et al* fitted separate damping term coefficients for the contraction and extension step responses of a McKibben muscle. They achieved a mean full-scale error of 15% for their combined static and damping force model. They observed an increase in damping with pressure on contraction and a decrease in damping on extension. Similar observations were made by Cao *et al* [49] who used the same first order pressure dependent viscous term fitted separately to contraction and extension step response data. Cao *et al* further distinguished a low and high pressure region. They fitted the pressure coefficients for pressures 200 kPa and below, and above 200 kPa. Thus they used a two segment piecewise version of Reynolds *et al*'s term and fitted it separately to contraction and extension data.

In contrast to the above mentioned empirical approaches, Sárosi *et al* [50] took a physics-based approach with a hysteresis loss viscous term. They based their model on static force models fitted to the upper and lower force-strain curves of a quasistatic muscle load cycle experiment (quasistatic muscle unloading and loading at a constant pressure). They multiplied the strain rate by the critical damping coefficient $2\sqrt{|k_t| m}$ and Lehr's damping coefficient $\zeta(P)$. Where m is the muscle inertial mass, and k_t its stiffness or the derivative of its static force model. Lehr's damping coefficient is the proportion of energy lost due to the rate-independent hysteresis during a load cycle experiment. It occurs over the full stroke of the muscle at a given pressure. Veale *et al* [45] have applied this McKibben muscle damping force model to a combined static and damping force model of the Peano muscle. They obtained a maximum error under 15% strain and a Root Mean Square Error (RMSE) of 2.6% strain.

In addition to Sárosi *et al*'s damping model are Tondu's [47] kinetic friction and Peternel *et al*'s [51] generalized kinetic friction viscous terms. They also have physically meaningful coefficients. Specifically, the coefficient B_0 represents static friction, and B_1 , a limit kinetic friction. These terms are inspired by a combination of textile physics and static-kinetic friction models.

Given this overview of McKibben static and damping force models, several were selected as benchmarks for validating the accuracy of the static and damping models generated by MAPMORE. As mentioned earlier, the focus is foremost on models for control that are accurate, relevant to McKibben and Peano muscles, and have transparent model terms. Thus, the static benchmark model from the literature is that of Sárosi [41]. All the damping models discussed (as listed in table 1) will be benchmarked due to the lack of available literature on their accuracy in modeling the damping force component (accuracies are typically stated for the combined damping and static models of a muscle).

3. The MAPMORE algorithm

MAPMORE is an algorithm that uses experimental data to build multivariable piecewise

functions. It was introduced by Veale *et al* [35] for the accurate modeling of McKibben and Peano muscle quasistatic force behavior. MAPMORE selects the mathematical modeling terms that constitute its piecewise functions from a dictionary of user specified terms and fits the terms to the data. The method that selects these terms is a slightly modified version of Billings *et al*'s [52, 53] Forward Regression Orthogonal Least Squares (FROLS) estimator algorithm. Alternative term selection and fitting algorithms exist [54–57], but FROLS was chosen for use with MAPMORE because it is computationally efficient and simple to implement.

The novelty of MAPMORE lies in the framework it builds around the modified FROLS algorithm so it can accurately model the static and damping force components of McKibben and Peano muscles. This involves extraction, segmentation, and scaling of training data; choice of suitable model terms; and multivariable piecewise function fusion and extrapolation. In this work, the versatility of MAPMORE in accurate model generation is extended beyond the quasistatic force model generation of Veale *et al* [35] to dynamic situations where the damping force component of fluidic muscles is also modeled.

Figure 3 and the explanation that follows outline how MAPMORE works when applied to generating models of a fluidic muscle's damping force component. Note that the symbols in the MAPMORE algorithm referred to in this explanation are summarized at the end of this section in table 2. Also note that the process used to generate the static force component models in this paper is similar, and is described in detail in [35]. In the first step in MAPMORE, in (a) of figure 3, data is collected. This involves the measurement of muscle force F , pressure P , and contraction strain ϵ as it performs contraction and then extension against a gravitational load in response to a pressure step. A total of N_p experiments are performed, each at a different pressure, that cover the range of muscle operating pressures. Next, the data from each experiment is segmented (b). Based on the work of Cao *et al* [49] and Reynolds *et al* [46], the data is segmented by its motion direction; the rise and fall segments when the muscle is contracting and extending respectively. From the vector of measured force data \mathbf{F} the training data damping force component \mathbf{F}_{dt} is calculated (c). Using the force component predicted by the static force model \mathbf{F}_s (see [35] for details), an uncorrected training component \mathbf{F}'_{dt} is calculated as $\mathbf{F}'_{dt} = \mathbf{F} - \mathbf{F}_s$. Then, steadystate static force modeling errors are accounted for by subtracting the steadystate value of \mathbf{F}'_{dt} between the rise and fall segments of each experiment, $F_{dt,ssv}$, from \mathbf{F}'_{dt} :

$$\mathbf{F}_{dt} = \mathbf{F}'_{dt} - F_{dt,ssv} \quad (1)$$

In this example there are $2N_p$ segments of damping force component data, each representing a different combination of pressure and muscle motion direction. The next step (d) uses Billings *et al*'s [52, 53] FROLS algorithm in combination with a dictionary of user specified potential model terms to generate a function f_i of the independent variable \mathbf{x} for each segment i .

Although figure 3 only shows f_i as a function of $\dot{\epsilon}$, \mathbf{x} may contain multiple elements,

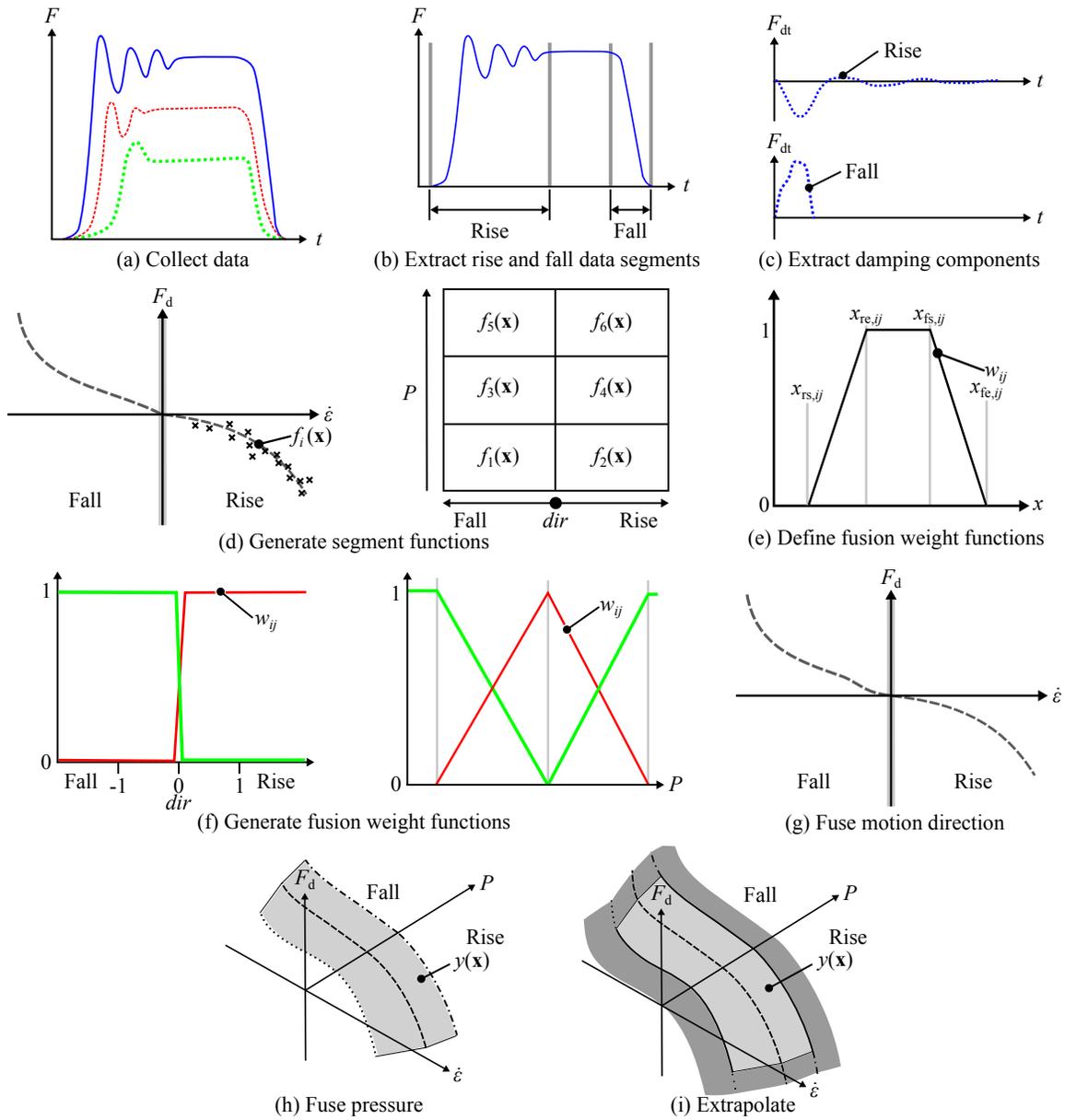


Figure 3: The MAPMORE algorithm used to model the damping force component of a fluidic muscle. First, force data is collected from step response experiments conducted at different pressures (a), the rise and fall transient damping components extracted (b, c), and segment functions $f_i(\mathbf{x})$ for each pressure-motion direction combination built by MAPMORE (d). MAPMORE then fuses these segments together with weight functions (e) for switching between segment functions for rise and fall motion directions, and the different training experiments' pressures (f). The result is a set of smoothed functions that predict the damping component in the different motion directions (g). These are then fused with the pressure weight functions to predict damping force as a function of pressure and strain rate (h). This is the final damping force model that can be extrapolated into and beyond the dark grey zone (i).

depending on the data available for generating $f_i(\mathbf{x})$. In this work, \mathbf{x} includes pressure and strain, and their first and second derivatives. These terms reflect that damping terms in the literature benchmark terms (table 1) include P , ϵ , and $\dot{\epsilon}$; that it is worth investigating whether higher derivatives contribute to model accuracy; and that $\dot{\epsilon}$ is possibly required to model flow restriction in the muscle. Whether all of the values in \mathbf{x} are used by the generated model depends on the terms in the dictionary $D(\mathbf{x})$ that are selected. In this work $D(\mathbf{x})$ consists of a second order polynomial combinations of the elements of \mathbf{x} , based on the second order combination of some of these variables found in the literature (table 1). From $D(\mathbf{x})$ FROLS selects the terms whose linear combination best fits a segment's training data and returns those terms' indices, \mathbf{s} , and their corresponding coefficients $\boldsymbol{\theta}$.

Fusion of the segment functions is achieved with fusion weight functions w_{ij} . These functions smooth the transition between the segments' piecewise functions and are defined for every independent variable with index j of \mathbf{x} and for each segment with index i . As shown in (e), each weight function acts as a switch that turns on its function f_i over the region between the rise start $x_{rs,ij}$ and rise end $x_{re,ij}$, and off between the fall start $x_{fs,ij}$ and fall end $x_{fe,ij}$. Here, weights are used for the variables muscle motion direction, dir , and pressure (f). The weight function for dir is an on/off toggle, reflecting that the muscle can only be moving in two possible directions. The weight function for pressure is an interpolation between the discrete pressures of the experiments for a given motion direction. The weight functions for the other variables in \mathbf{x} are set to zero as they were not used to segment the data. When the weight functions are multiplied by their corresponding piecewise functions they smooth them together (g, h), giving the final model $y(\mathbf{x})$, which predicts F_d . It is feasible that $y(\mathbf{x})$ is used outside the range of data used to train it. Provision is made for this by an extrapolation function E , which determines the behavior of $y(\mathbf{x})$ when $y(\mathbf{x})$ is evaluated outside of the range of the training data (i).

MAPMORE has the potential to generate accurate models for different types of fluidic muscles and for static and dynamic behavior. Its versatility comes from the user defined FROLS dictionary, which may include any input-output function the user believes has value in describing fluidic muscle behavior. MAPMORE's ability to combine these functions in a piecewise manner improves overall model accuracy by acknowledging that a muscle's behavior can significantly change over its operating regime. The validity of this approach has been verified for static behavior [35]. Also validating its applicability to damping force component models in section 4 confirms its applicability to dynamic behavior. In this work a dynamic model is simply the sum of the static and damping force models' predictions. The subsections to follow elaborate on the generation and fusion steps of the MAPMORE algorithm.

3.1. Piecewise Function Generation

The MAPMORE algorithm generates each segment function f_i according to the following steps and accompanying flow chart in figure 4 (note that parenthesized alphabetic labels refer to those in figure 4):

- (i) The segment's independent and dependent variable training data \mathbf{X}_t and \mathbf{y}_t are linearly scaled to lie in the domain $[-1, 0]$ or $[0, 1]$ so the data fits in a unit square and its trend passes through the origin (a). The training data consists of a matrix of rows of \mathbf{x}_t or a vector of rows of y_t . Each row corresponds to a new sample of independent and dependent variable data. The independent variables' scaling slopes and intercepts are $\mathbf{m}_{x,i}$ and $\mathbf{c}_{x,i}$. The slope and intercept to unscale the dependent variable from this unit length domain to the original data units are $m_{y,i}$ and $c_{y,i}$. This scaling is the only modification to Billings *et al*'s FROLS algorithm. Scaling the variables enables the selection and fitting of origin centered dictionary terms to data offset from the origin, like McKibben and Peano muscles' force-strain curves [35].
- (ii) The scaled independent variable training data \mathbf{X}_{ts} are evaluated for each of the N_D terms in the dictionary. The result is N_D term vectors \mathbf{p}_a (b).
- (iii) FROLS iteratively selects terms from $D(\mathbf{x})$ that best match the trend in the scaled dependent variable data \mathbf{y}_{ts} and calculates the terms' corresponding coefficients. In each iteration b , the unselected dictionary terms \mathbf{p}_a are orthogonalized against the previously selected dictionary terms \mathbf{o}_c . The result is the orthogonalized unselected dictionary term \mathbf{q}_a (c). For the first iteration ($b = 1$) there are no previously selected terms to orthogonalize \mathbf{p}_a against, so $\mathbf{q}_a = \mathbf{p}_a$. In successive iterations ($b > 1$), the Gram-Schmidt method in (2) is used to orthogonalize \mathbf{p}_a .

$$\mathbf{q}_a = \mathbf{p}_a - \sum_{c=1}^{b-1} \frac{\mathbf{p}_a^\top \mathbf{o}_c}{\mathbf{o}_c^\top \mathbf{o}_c} \mathbf{o}_c \quad (2)$$

- (iv) Then (3) is used to calculate each term's potential coefficient g_a from \mathbf{q}_a and the scaled dependent variable training data \mathbf{y}_{ts} (d)

$$g_a = \frac{\mathbf{y}_{ts}^\top \mathbf{q}_a}{\mathbf{q}_a^\top \mathbf{q}_a} \quad (3)$$

- (v) Next the extent that the orthogonalized term vector explains the variation in the dependent variable, as measured by the error reduction ratio (ERR), is calculated (e),

$$err_a = g_a^2 \frac{\mathbf{q}_a^\top \mathbf{q}_a}{\mathbf{y}_{ts}^\top \mathbf{y}_{ts}} \quad (4)$$

- (vi) Steps (iii)–(v) are repeated for all the unselected dictionary terms

- (vii) Then the index s_b of the term with the highest ERR is found (f),

$$s_b = \mathit{argmax}(\mathbf{err}) \quad (5)$$

- (viii) The orthogonalized dictionary term vector, coefficient, and ERR are saved to \mathbf{o}_b , θ_b , and ERR_b (g)
- (ix) If the selected model terms' combined ERR is within a specified tolerance ρ (h), FROLS stops, otherwise it continues with steps (iii)–(viii) selecting additional terms. The FROLS also stops if the number of selected dictionary terms b meets the term limit b_1 (i). These termination conditions are described by equations (6) and (7).

$$1 - \sum_{a=1}^b ERR_a \leq \rho \quad (6)$$

$$(b - 1) \geq b_1 \quad (7)$$

The segment function produced by FROLS is then a linear sum of the selected dictionary terms and their coefficients:

$$f_i(\mathbf{x}) = m_{y,i} \left(\sum_{k=1}^{N_{D,i}} \theta_k D_{s_k}(\mathbf{m}_{\mathbf{x},i}^\top \mathbf{x} + \mathbf{c}_{\mathbf{x},i}) \right) + c_{y,i}. \quad (8)$$

Where $N_{D,i}$ is the number of selected terms in this segment function.

3.2. Piecewise Function Fusion

The segment functions are fused with the fusion weight functions w_{ij} , and extrapolation function E to generate the overall model:

$$y(\mathbf{x}) = \sum_{i=1}^{N_s} \left(f_i(E(\mathbf{x}, \mathbf{x}_{ll}, \mathbf{x}_{ul}, \mathbf{x}_{lv}, \mathbf{x}_{uv})) \prod_{j=1}^{N_x} w_{ij}(x_j, x_{rs,ij}, x_{re,ij}, x_{fs,ij}, x_{fe,ij}) \right). \quad (9)$$

Where N_x is the number of independent variables and N_s is the total number of segments. The extrapolation function, which is explained shortly, has the lower and upper limits for \mathbf{x} of \mathbf{x}_{ll} and \mathbf{x}_{ul} . When \mathbf{x} goes beyond the lower and upper limits, the extrapolation function saturates \mathbf{x} to the values \mathbf{x}_{lv} and \mathbf{x}_{uv} . Also in (9) is the weight function, which switches f_i on in the rise transition zone $x_{rs,ij} < x_j \leq x_{re,ij}$, and off in the fall transition zone $x_{fs,ij} < x_j \leq x_{fe,ij}$. Weight functions for segments at the limits of the data domain only need one transition as there are no further adjacent segments to transition to. Some simple ways of modeling this transition zone are the linear, quadratic, and cubic transitions shown in figure 5.

The transition zones bridge adjacent segments. The width of these user selected zones is set by the transition zone size z_s , or the proportion of the segment that is transitioning from its function to the function of the adjacent segment. Varying z_s affects the smoothness of the transition between segment functions (figure 5), but may compromise model accuracy. The transition zones' start locations x_{rs} and x_{fs} , or x_s in general, are calculated according to (10). In this equation, seg_i is the segment location,

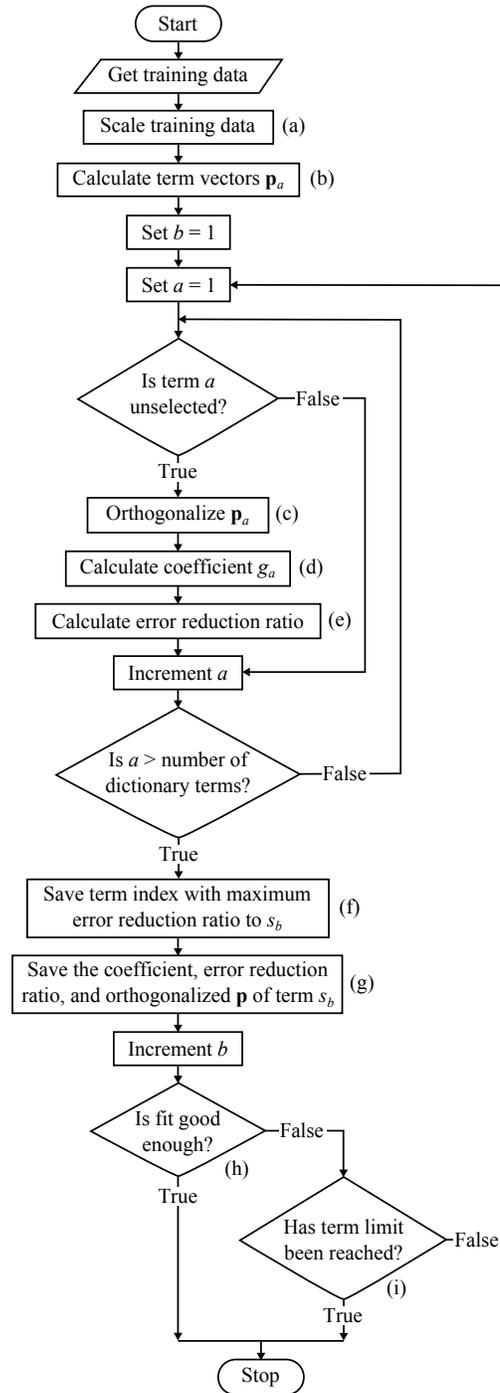


Figure 4: Operation of MAPMORE's FROLS term selection and fitting algorithm. The algorithm begins by evaluating (b) each of the dictionary terms with the scaled training data (a). It then goes through a number of iterations selecting (f, g) the previously unselected dictionary terms that best fit the training data. Within each iteration it orthogonalizes (c) previously unselected dictionary term vectors, calculates their potential coefficient (d), and calculates their fit (e), as measured by the error reduction ratio. When the selected terms cumulatively fit the data well enough (h) or their number reaches the term limit (i), the algorithm stops.

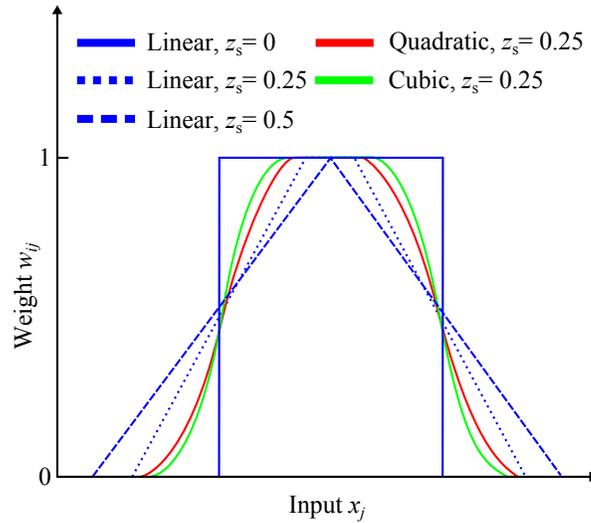


Figure 5: MAPMORE currently supports linear, quadratic, and cubic transitions. The type of transition affects the shape of a weight's switch on and switch off transition zones, where the quadratic and cubic types are smoother alternatives to the linear transition. The width of piecewise function weights is defined by the transition zone size. The transition zone size, or proportion of adjacent segments that the weight acts over is z_s , and can vary from $0 \leq z_s \leq 0.5$. This varies the sharpness of the transition from a vertical step at a segment location to a gradual increase in weight beginning at halfway through the first segment and ending halfway through the second.

and seg_{i-1} and seg_{i+1} are adjacent segment locations, as shown in figure 6. Similarly, the end locations x_{re} and x_{fe} , or x_e in general, are calculated according to (11).

$$x_s = seg_i - z_s(seg_i - seg_{i-1}) \quad (10)$$

$$x_e = seg_i + z_s(seg_{i+1} - seg_i) \quad (11)$$

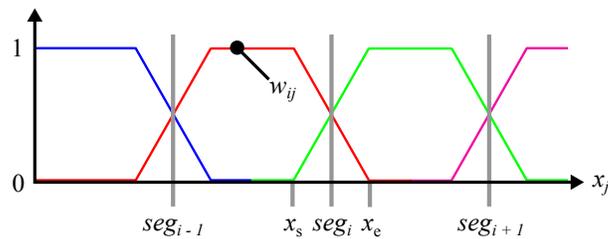


Figure 6: The start and end locations, x_s and x_e , of weight transitions such as w_{ij} are placed either side of a segment location seg_i . They are calculated as a proportion of the width of adjacent segments. Here these segment widths are defined by $seg_{i+1} - seg_i$ and $seg_i - seg_{i-1}$.

Completing the MAPMORE generated model is the extrapolation function E . It allows the user to specify how the model processes inputs beyond the domain of its

training data. Equation (12) shows how this works. When an input x exceeds an upper limit x_{ul} , it will take the value x_{uv} . If the input drops below the lower limit x_{ll} it will take the value x_{lv} .

$$E(\mathbf{x}, \mathbf{x}_{ll}, \mathbf{x}_{ul}, \mathbf{x}_{lv}, \mathbf{x}_{uv}) = \forall j \in \{1, 2, \dots, N_x\} \begin{cases} x_{lv,j} & x_j < x_{ll,j} \\ x_j & x_{ll,j} \leq x_j \leq x_{ul,j} \\ x_{uv,j} & x_{ul,j} < x_j \end{cases} \quad (12)$$

The limits are automatically determined from the domain of the training data, but the limit values are user defined. Their definition determines the extrapolation behavior of the model. For instance, if the model is to continue to evaluate normally outside of the limit values, as in figure 7a, then \mathbf{x}_{lv} and \mathbf{x}_{uv} are set to \mathbf{x} . Alternatively, if the model is to saturate its output at the limit values (figure 7b), then \mathbf{x}_{lv} and \mathbf{x}_{uv} are set to \mathbf{x}_{ll} and \mathbf{x}_{ul} . A third possibility is the model outputs zero when its inputs are beyond the limits of the training data (figure 7c). This behavior is possible by setting the relevant weight functions' outermost transitions to zero at the limits of the training data domain (figure 7d).

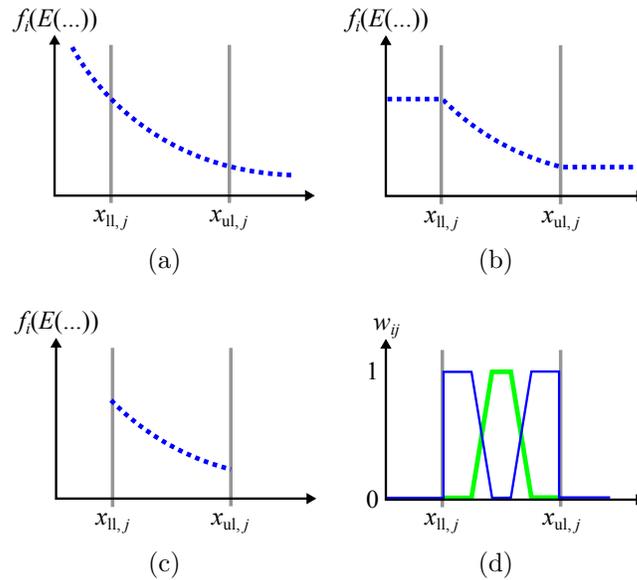


Figure 7: Three possible extrapolation behaviors for a MAPMORE generated model. In (a), the model is evaluated normally beyond the lower and upper limits of its input training data domain $x_{ll,j}$ and $x_{ul,j}$. The second option is to saturate the output at the boundary values of the input domain, as in (b). Last, the model's output can be forced to zero beyond its training domain (c) by setting its outermost weight functions to zero beyond the training data domain limits (d).

4. Model validation methods

The aim of the remainder of this work was to use experimental data to validate the

Table 2: Definitions of the symbols used in the MAPMORE algorithm

Parameter	Definition
a	general index
b	FROLS iteration
b_1	FROLS term selection limit
c	selected term vector index
\mathbf{c}_x	segment independent variables' scaling intercepts
c_y	segment dependent variable unscaling intercept
D	dictionary of candidate function terms
E	extrapolation function
err	orthogonalized term error reduction ratio
ERR	selected term error reduction ratio
f	segment function
g	orthogonalized term potential coefficient
i	segment index
j	independent variable index
k	selected dictionary term index
\mathbf{m}_x	segment independent variables' scaling slopes
m_y	segment dependent variable unscaling slope
N_D	number of dictionary terms
N_s	number of segments
N_x	number of independent variables
\mathbf{o}	selected orthogonalized term vector
\mathbf{p}	term vector
\mathbf{q}	orthogonalized term vector
\mathbf{s}	indices of selected dictionary terms
seg	segment location
w	fusion weight function
x	independent variable(s)
x_e	transition end location
x_{fe}, x_{fs}	transition zone fall end and start
\mathbf{x}_{ll}	independent variables' lower extrapolation limits
\mathbf{x}_{lv}	independent variables' lower extrapolation values
x_{re}, x_{rs}	transition zone rise end and start
x_s	transition start location
\mathbf{X}_t	independent variable segment training data
\mathbf{X}_{ts}	scaled independent variable segment training data
\mathbf{x}_{ul}	independent variables' upper extrapolation limits
\mathbf{x}_{uv}	independent variables' upper extrapolation values
y	MAPMORE generated model
\mathbf{y}_t	dependent variable segment training data
\mathbf{y}_{ts}	scaled dependent variable segment training data
z_s	transition zone size
$\boldsymbol{\theta}$	coefficients of selected dictionary terms
ρ	FROLS ERR termination tolerance

accuracy of models generated by the MAPMORE algorithm for capturing the static and damping behavior of Peano and McKibben muscles. The remainder of this section will list the benchmark models used in validation, give details on the experimental methods used, outline the training methods and equations used for the static and damping force benchmark models, and summarize the MAPMORE configuration used for validation.

MAPMORE's models, if an improvement on existing models, could be useful for the accurate and versatile simulation and hence model-based control of Peano and McKibben muscles' nonlinear behavior. The types of models validated in this work were static force, damping force, and combined static and damping force models (a dynamic model). These models' Normalized RMSEs (NRMSEs) at predicting experimentally measured muscle forces were each compared against the accuracy of accurate and relevant benchmark models.

The static force benchmark models were Sárosi's model and a second order polynomial of muscle pressure and strain. This latter model was chosen as a general reference model that captures the approximately quadratic dependence of static muscle force on strain and linear dependence of static muscle force on pressure [12, 14]. Three damping force models were used as benchmarks. First, Reynolds *et al*'s [46] motion direction dependent first order pressure dependent damping term (which also includes the simpler linear and pressure dependent viscous terms). Second, Sárosi *et al*'s [50] hysteresis loss viscous term. Third, Peternel *et al*'s [51] generalized kinetic friction element (which is an extension of the kinetic friction term).

4.1. Experimental methods

Training and validation of the MAPMORE generated and benchmark static and damping force models were performed with the McKibben PAM and Peano muscle shown in figure 2. The McKibben PAM had a diameter of 0.02 m and active length of 0.088 m. The Peano muscle had four tubes 0.046 m long and 0.017 25 m wide. Load cycle and step response experiments were carried out on these muscles using the conditions summarized in table 3 and the test rig shown in figure 8. The rig used pressurized air to pressurize water in a reservoir via an on/off solenoid valve and a flow restricting valve. The water was then used to hydraulically actuate the muscle under test while measuring its force, pressure, flow rate, and length. Quasistatic load cycle tests were conducted by coupling the electrohydraulic actuator to the muscle. Decoupling the electrohydraulic actuator and pressurizing the muscle enabled step response experiments to be carried out. More details on the test rig are presented in table 4 and [58].

Data from the load cycle experiments was used for training and validation of the static force component models, and step response experiment data was used for training and validation of the damping force component models. As shown in table 3, for each muscle and type of experiment, experiments with different combinations of pressure and inertial mass were conducted. One experiment was done for each combination of test conditions. All this experimental data was used to train the models except for that from

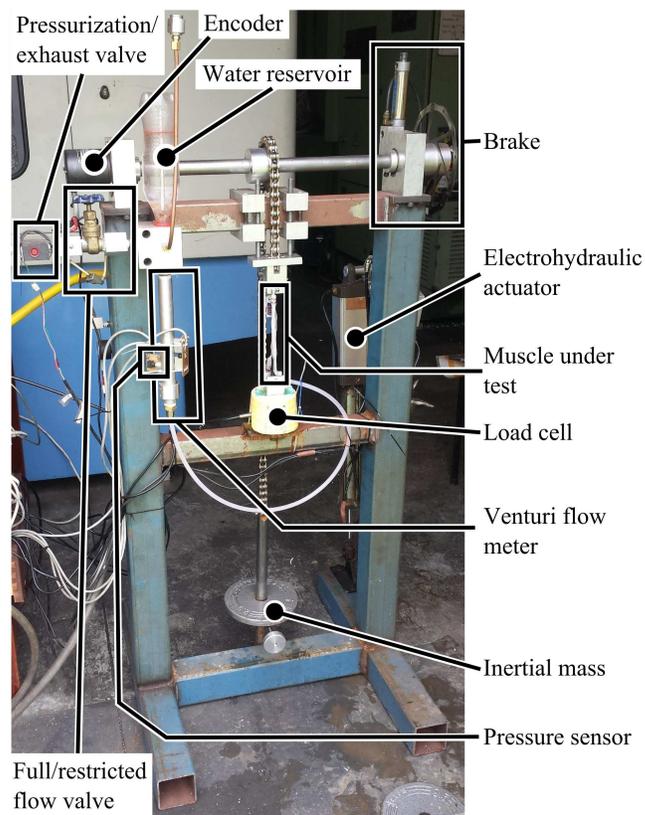


Figure 8: The test rig used to conduct load cycle and step response experiments on the McKibben and Peano muscles for validation of the MAPMORE models

Table 3: McKibben and Peano muscle test conditions

Conditions	McKibben	Peano
Training pressures (kPa), step response inertial mass (N)	80, 42	80, 42
	150, 66	200, 66
	350, 213	500, 213
Validation pressures (kPa), step response inertial mass (N)	250, 115	350, 115

Table 4: A summary of the key components used in the test rig in figure 8

Part	Note
Brake	Custom pneumatically actuated bicycle disc brake
Electro-hydraulic actuator	1.5 kN ORO 230 V, BFT S.p.a, Schio, Italy
Encoder	A860-0300-T001, Fujitsu Fanuc Ltd., Tokyo, Japan
Full/restricted flow valve	Manually operated, Generic gate household water supply type
Load cell	0.7%, Generic S-beam 20 kg, China
Pressure sensor	26PCGNM6G, Honeywell, Morristown, NJ
Pressurization/exhaust valve	3/2 solenoid valve, Humphrey 320 12VDC, Humphrey Products Corp., Kalamazoo, MI
Venturi flow meter	Custom, with flow rate measured by 26PCDFA6D (Honeywell, Morristown, NJ) differential pressure sensor
Water reservoir	0.5 L, Polyethylene terephthalate (PET)

one of the experiments, which was kept aside for the models' validations. Accuracy of the validation data was quantified by the full-scale NRMSE. This is the RMS of the error between the forces predicted by a model and the forces in the experimental data divided by the maximum of the absolute value of the validation data forces F :

$$\text{NRMSE} = \frac{\text{RMS}}{\max(|F|)}. \quad (13)$$

Load cycle experiments, as described in section 4.3, were used to obtain sets of upper and lower force-strain curves for various muscle pressures. As in [35], the average of these upper and lower curves' forces for a given muscle strain were used to train and validate the models.

Step response experiments characterized the dynamic contraction and extension of a muscle against an inertial mass at various pressures. In these experiments the vertical muscle was held lightly tensioned (0.5 N to 1 N) against a stop with a weight (inertial mass) hanging on it. The muscle was then suddenly pressurized while its force, strain, and pressure were measured. After the muscle reached a steady state length the muscle

was exhausted, allowing the inertial mass to re-extend it. This yielded force-time data curves similar to those in figure 3a. The resulting data was used to build and validate MAPMORE or benchmark damping models.

4.2. Static force benchmark models

Sárosi's model [41] accurately predicts static muscle force as a function of pressure and strain according to (14). Coefficients $a_1 - a_6$ fit the model to experimental data.

$$F_s(P, \epsilon) = (a_1 P + a_2) e^{a_3 \epsilon} + a_4 \epsilon P + a_5 P + a_6 \quad (14)$$

The polynomial static force benchmark model is as follows:

$$F_s(P, \epsilon) = a_{2,0} \epsilon^2 + a_{1,1} \epsilon P + a_{1,0} \epsilon + a_{0,1} P + a_{0,0}. \quad (15)$$

It uses its coefficients $a_{2,0}$, $a_{1,1}$, $a_{1,0}$, $a_{0,1}$, and $a_{0,0}$ to fit the model to data. In this work, these models' coefficients were fitted with Matlab's Levenberg-Marquardt numerical solver.

4.3. Damping force benchmark models

The first benchmark model, Reynolds *et al*'s [46] first order pressure dependent damping element predicts the damping force F_d according to (16). In this model, B_{r0} , B_{r1} , B_{f0} , and B_{f1} were fitted by comparing the predicted F_d with F_d measured in step response experiments conducted at various pressures. As the model is motion direction dependent, B_{r0} and B_{r1} were fitted to the rise portions of the experiments, and B_{f0} and B_{f1} to the fall portions. Note that when this model was evaluated, the motion direction dir , was calculated as a rise or fall value based on whether the pressure in the muscle was increasing or decreasing.

$$F_d(\dot{\epsilon}, dir) = \begin{cases} (B_{r0} + B_{r1} P) \dot{\epsilon} & dir = 1 \quad (\text{rise}) \\ (B_{f0} + B_{f1} P) \dot{\epsilon} & dir = -1 \quad (\text{fall}) \end{cases} \quad (16)$$

In contrast to the other benchmark models, Sárosi *et al*'s [50] hysteresis loss damping element was fitted to quasistatic load cycle data. This data characterizes how much force the muscle produces for a given strain and pressure. First, the muscle under test is locked at an initial starting length and pressurized to a target pressure. Second, it is allowed to slowly contract until its force reduces to zero. Last, it is slowly extended to its starting length, and then depressurized. The result is a hysteresis loop consisting of an upper and a lower curve made of force-strain data points.

Before the damping element could be calculated, Sárosi's [41] static model was fitted separately to upper and lower force-strain curves of training data collected at various pressures. The experimental details are described in section 4.1. Thus, equations (17) and (18) predict the static force according to the upper (F_{su}) and lower (F_{sl}) training curves with coefficients $a_{u,1} - a_{u,6}$ and $a_{l,1} - a_{l,6}$.

$$F_{su}(P, \epsilon) = (a_{u,1} P + a_{u,2}) e^{a_{u,3} \epsilon} + a_{u,4} \epsilon P + a_{u,5} P + a_{u,6} \quad (17)$$

$$F_{sl}(P, \epsilon) = (a_{1,1}P + a_{1,2})e^{a_{1,3}\epsilon} + a_{1,4}\epsilon P + a_{1,5}P + a_{1,6} \quad (18)$$

Then the damping force component is

$$F_d(P, \epsilon, \dot{\epsilon}) = - \left(2\zeta(P) \sqrt{|k_t(P, \epsilon)| m} \right) \dot{\epsilon}, \quad (19)$$

where m is the step response experiment inertial mass (listed in section 4.1), k_t is the muscle tensile stiffness, and ζ is the pressure dependent Lehr's damping coefficient. The muscle tensile stiffness is given in (20), and is based on the gradient of F_{su} and the deflated muscle length $n_t w_t$.

$$k_t(P, \epsilon) = (a_{u,3}(a_{u,1}P + a_{u,2})e^{a_{u,3}\epsilon} + a_{u,4}P) / (n_t w_t) \quad (20)$$

Lehr's damping coefficient is

$$\zeta(P) = \frac{U_u - U_l}{U_u}, \quad (21)$$

where U_u and U_l are the areas under the upper and lower static force-strain curves. They were calculated between the minimum and maximum modeled strains ϵ_{\min} and ϵ_{\max} of the muscle at a given pressure P :

$$U_l = \int_{\epsilon_{\min}}^{\epsilon_{\max}} F_{sl}(P, \epsilon) d\epsilon, \quad (22)$$

$$U_u = \int_{\epsilon_{\min}}^{\epsilon_{\max}} F_{su}(P, \epsilon) d\epsilon. \quad (23)$$

The minimum strain was the same as $\epsilon_{t\min}$, the minimum strain in the load cycle training data (based on the muscle's initial starting length in that experiment), which should be the minimum strain for all the training load cycle experiments. The maximum strain was the predicted free-strain and was calculated by solving $F_{su}(P, \epsilon) = 0$ for ϵ at the given pressure.

The last damping force component benchmark model, Peternel *et al*'s [51] generalized kinetic friction damping element, predicts damping force as

$$F_d(\dot{\epsilon}) = \text{sgn}(\dot{\epsilon}) \left(\sum_{k=0}^2 B_k \dot{\epsilon}^k + B_3 e^{\frac{-|\dot{\epsilon}|}{B_4}} \right). \quad (24)$$

Coefficients $B_0 - B_4$ were fitted to the damping force component training data using Matlab's Levenberg Marquardt algorithm, which was also used to fit the other benchmark models' coefficients.

4.4. MAPMORE default configuration

The variety of function generation, fusion, and extrapolation parameters of the MAPMORE algorithm means that there are many possible options for configuring it to generate models. Here we started with a configuration based on simplicity, mid-range values, and inspection of the nonlinearity in the data to be modeled. This configuration was not optimized. The static model was generated with four segments, a fourth order polynomial dictionary of strain ($D(\mathbf{x}) = \mathbf{1}, \mathbf{x}, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4$), one model term, simple linear transition zones of size $z_s = 0.25$, and a simple continue extrapolation behavior (as in figure 7a). The damping model was generated with a second order polynomial dictionary of ϵ , $\dot{\epsilon}$, P , and \dot{P} (excluding the constant term); one model term; a linear transition zone with $z_s = 0.25$; and a continue extrapolation behavior.

5. Results

The following sections present the results of comparing the MAPMORE generated models' accuracies with those of the benchmarks models with highlights of notable results. First, the fit and accuracy of static force component models are compared, then, the accuracy of different damping force component models is explored. Last, the static and damping models are combined to predict the overall dynamic behavior of McKibben and Peano muscles.

5.1. Static force component models

Figures 9 and 10 show that the fit of the MAPMORE generated and Sárosi's static force models are both good for validation data from the McKibben and Peano muscles. The MAPMORE model is more accurate, with an NRMSE of 2.2% (compared to Sárosi's model's 2.5%) for the McKibben muscle and an NRMSE of 1.1% (compared to Sárosi's model's 1.6%) for the Peano muscle. The simple polynomial model had a higher, but still fair NRMSE of 7% for the McKibben muscle and 2.8% for the Peano muscle. Its accuracy tended to decrease at strains above 5% in the McKibben muscle (figure 9), where it increasingly underestimated force with increasing strain.

5.2. MAPMORE damping force component model motion and pressure segmentation

Before proceeding with the comparison of the MAPMORE generated and benchmark damping force models, an investigation was made into the effect of segmentation of the different independent variables. That is, the accuracy of damping force models with and without pressure dependency, and with and without motion direction dependency. In a pressure dependent model, pressure is an independent variable of damping force, and the model is built from a selection of step response experiments conducted at different pressures, as shown in figure 3a. In a pressure independent model, damping force is assumed to be independent of pressure and the model is trained from the step response

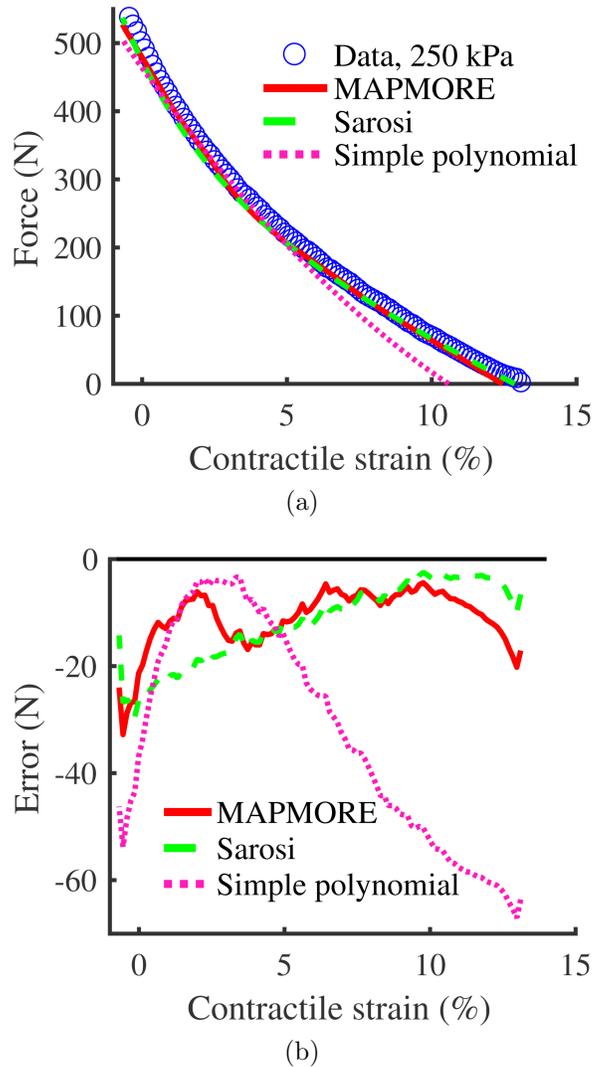


Figure 9: Validation fit (a) and error (b) curves for static models of the McKibben muscle. Shown are MAPMORE's, Sárosi's, and the simple polynomial models.

experiment with the highest pressure (if more than one training data experiment is available). A motion direction dependent model has segment functions for the rise and fall directions. It is trained from step response experimental data separated into these rise and fall transient segments. If the model is motion independent, these rise and fall data segments are combined and collectively used by MAPMORE to build a model that does not have separate segment functions for each motion direction.

The results of this comparison are shown for the McKibben and Peano muscles in figure 11. These results show that both muscles are most accurately modeled by pressure independent, motion direction dependent MAPMORE generated models. Removing pressure dependency from a model with motion direction dependency decreased NRSME. The importance of motion direction dependency confirms previous literature that separately modeled the motion direction of McKibben muscles [46, 49]. In

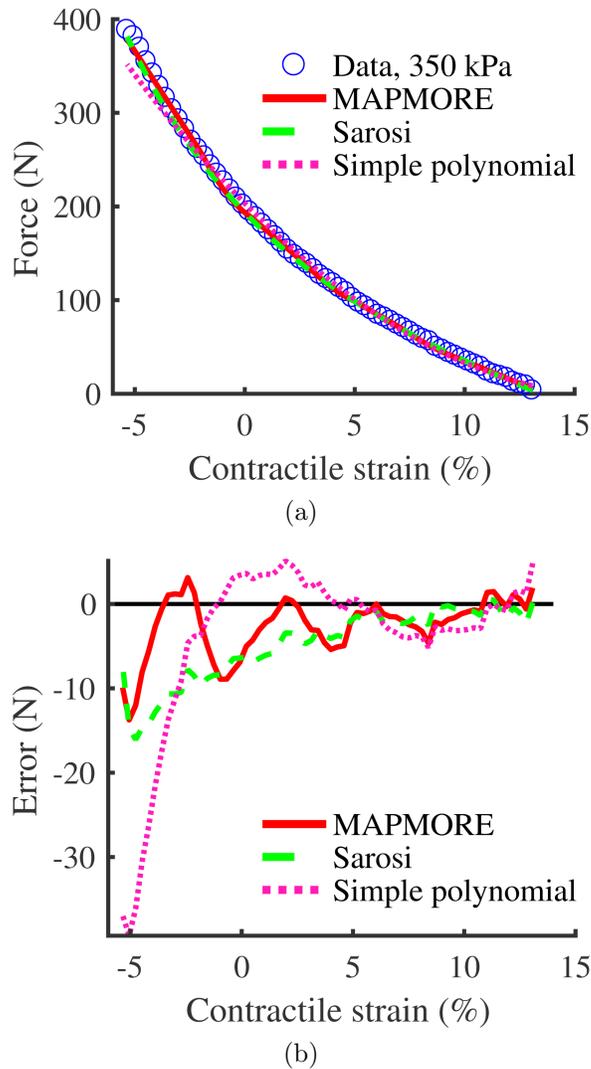


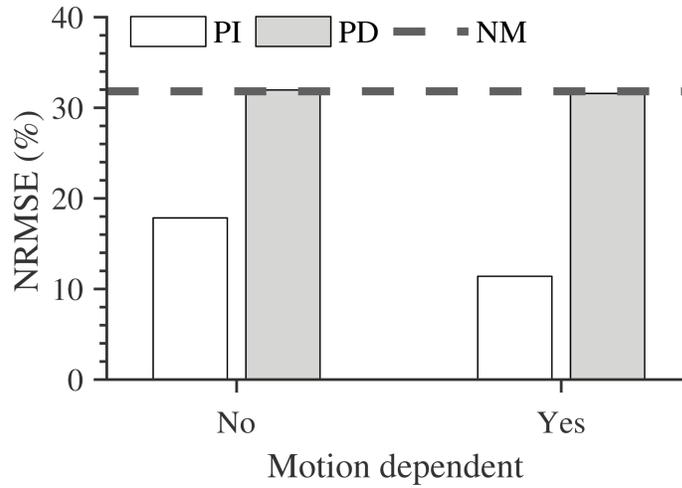
Figure 10: Validation fit (a) and error (b) curves for static models of the Peano muscle. Shown are MAPMORE's, Sárosi's, and the simple polynomial models.

general, figure 11 shows that adding pressure dependency to a model increased NRMSE to about the same value of a static force model (that is, a force model assuming no damping model, or $F_d = 0$). The exception was the Peano muscle with no motion dependency (figure 11b).

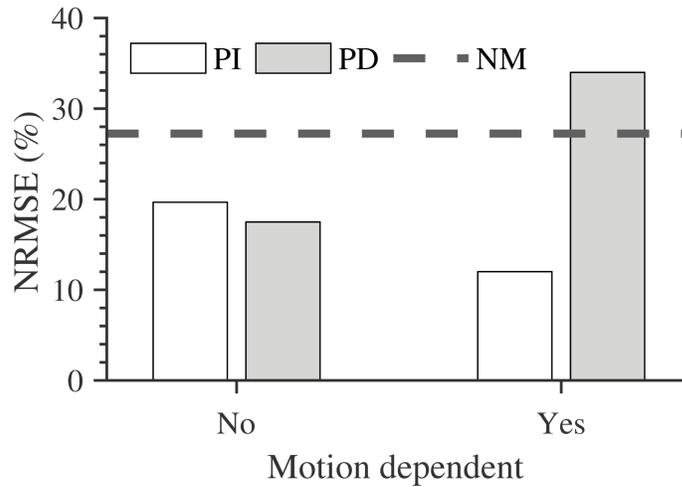
The most accurate models for McKibben and Peano muscles were those with motion direction segmentation and with no pressure segmentation. They and the MAPMORE default configuration described in section 4.4 were used to generate the remainder of the damping force models in this work. These damping models are summarized in table 5.

5.3. Damping force component models

Comparing the results of MAPMORE's and the benchmarks' damping models' fits and NRMSEs in figures 12 and 13 shows that MAPMORE's models had the lowest



(a)



(b)

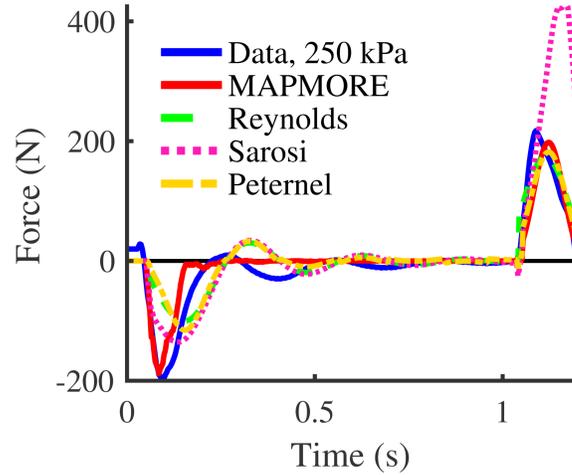
Figure 11: Comparison of validation NRMSEs for McKibben (a) and Peano (b) MAPMORE damping force models. Models are those with and without motion dependent segmentation, and with pressure independent (PI) or pressure dependent (PD) segmentation. These NRMSEs are referenced against the NRMSE assuming a damping force of $F_d = 0$ (no model - NM).

Table 5: The damping force model terms generated by MAPMORE

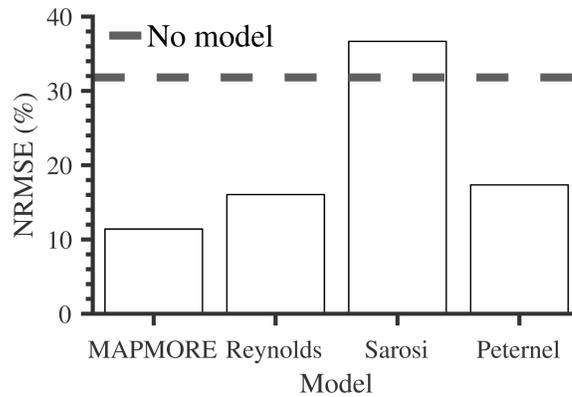
Direction	McKibben muscle	Peano muscle
Rise	$F_d = -15.6 \dot{P}\dot{\epsilon}$	$F_d = 0.202 \dot{P}\dot{\epsilon}$
Fall	$F_d = 34.9 \times 10^6 \dot{\epsilon}^2$	$F_d = -16.6 \times 10^3 \dot{\epsilon}$

NRMSEs (11.4% for the McKibben muscle and 12.0% for the Peano muscle), followed

by Reynolds' models (16.0 % for the McKibben muscle and 25.1 % for the Peano muscle), and Peternel's models (17.3 % for the McKibben muscle and 26.0 % for the Peano muscle). Sárosi's damping models had the highest NRMSEs (36.7 % for the McKibben muscle and 41.4 % for the Peano muscle), higher than no damping model (31.8 % for the McKibben muscle and 27.3 % for the Peano muscle with $F_d = 0$).



(a)



(b)

Figure 12: Validation fit curve (a) and NRMSE comparison (b) for damping models of the McKibben muscle. Shown are MAPMORE's, Reynolds', Sárosi's, and Peternel's damping models.

Referring to the fit of the McKibben muscle damping force models in figure 12a, all the models but Sárosi's fitted the fall region well, slightly underestimating and lagging the positive damping force fall peak in the data. MAPMORE also modeled the negative damping force rise peak accurately in magnitude and timing, whereas the benchmark models' rise peaks lagged the data and underestimated the peak's magnitude. None of the models captured the oscillation in the steadystate portion of the step response data. MAPMORE's model predicted no oscillation, and the benchmark models predicted a 3.4 Hz oscillation with a similar magnitude to that shown in the data damping force, but with a 90° phase lag.

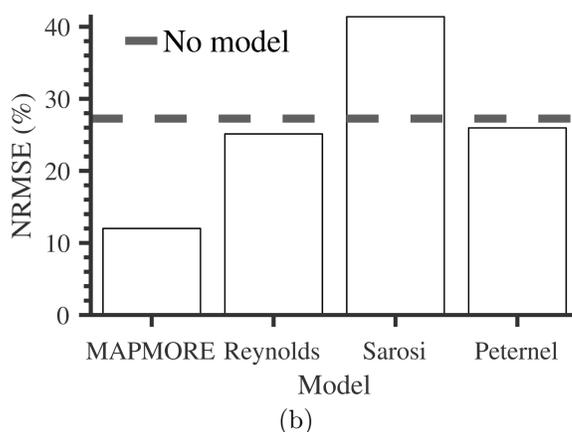
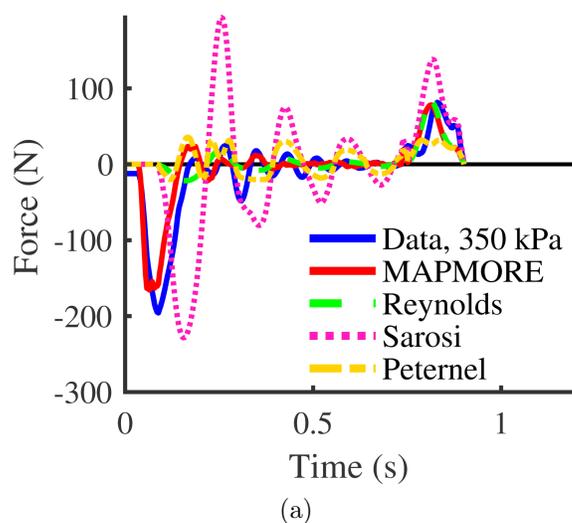


Figure 13: Validation fit curve (a) and NRMSE comparison (b) for damping models of the Peano muscle. Shown are MAPMORE's, Reynolds', Sárosi's, and Peternel's damping models.

In figure 13a the Peano muscle's damping force fall region is approximately modeled in magnitude and shape by MAPMORE's and Reynolds' models, but the models' fall peaks lag that of the data by 0.02s. Only MAPMORE's model predicted the timing of the negative rise peak, but underestimated its magnitude and duration. None of the models in the validation predicted the oscillations seen in the data between its rise and fall peaks.

5.4. Dynamic force models

The final validation tests compared the accuracy of combined static and damping models with the total muscle force measured during step response experiments. The benchmark dynamic model was formed from the most accurate of the static and damping

benchmark models (Sárosi's and Reynolds' models - see sections 4.2 and 4.3). The step response experiments used for validation of the dynamic models were the same as those used for validation of the damping models as described in section 4.1. Accuracy of the dynamic models was calculated using equation (13), where the numerator was the RMS of the errors between the measured force and the sum of the static and damping force component predictions.

Comparing the dynamic models' accuracies in figure 14 shows that both in the McKibben and Peano muscles the addition of damping force models improved accuracy compared to only a static force model. Figure 14 also shows that the combined static and damping force model (dynamic model) of MAPMORE was more accurate than the dynamic benchmark model for both muscle types.

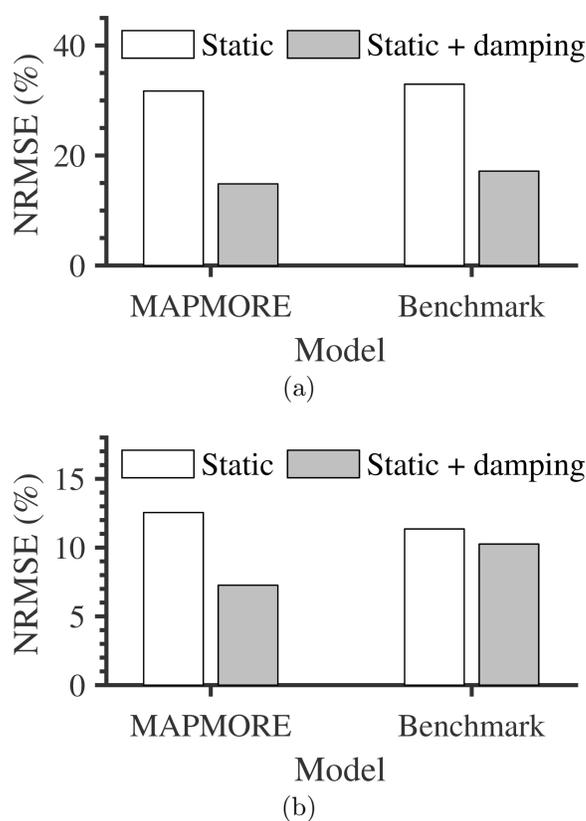
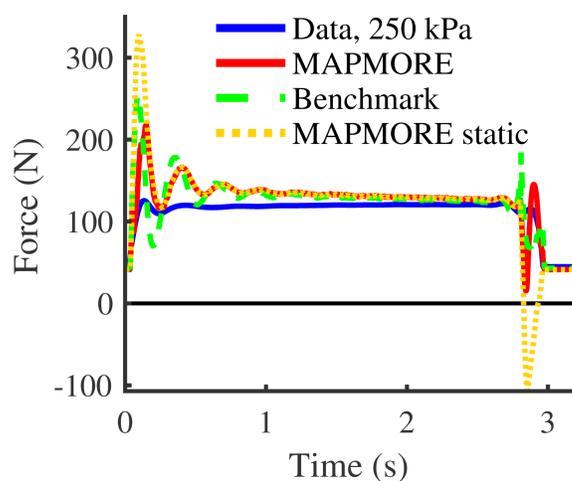


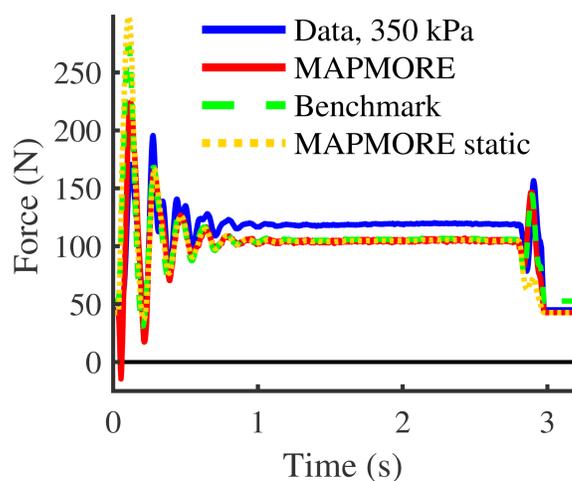
Figure 14: Validation NRMSE comparison for static, and combined static and damping models of the McKibben (a) and Peano (b) muscles. The benchmark models are Sárosi's static and Reynolds' damping force models.

The validation fits (figure 15) show how the addition of a damping force model to MAPMORE's static model increased accuracy compared to a static only model. First, the damping model reduced the static model's overestimation of muscle force during the rise region. Second, the damping model reduced the underestimation of muscle force during the fall region compared to a static force only model.

In the McKibben muscle (figure 14a), the MAPMORE dynamic model had an



(a)



(b)

Figure 15: Validation fits for MAPMORE static, and MAPMORE and benchmark dynamic models of the McKibben (a) and Peano (b) muscles. The benchmark model consists of Sárosi’s static and Reynolds’ damping force models.

NRMSE of 14.8% compared to the benchmark model’s 17.2%. Both these models did not follow the overdamped trend of the rise and fall regions seen in this muscle’s step response data. They did, however, converge towards the correct value of the steadystate muscle force over the region between the rise and fall regions.

In the Peano muscle (figure 15b), the steadystate force error during the time between the rise and fall regions remained constant, and could be due to muscle hysteresis, which is outside the scope of modeling in this work. As with the McKibben muscle, the MAPMORE dynamic model was more accurate, with an NRMSE of 7.3% compared to that of 10.3% for the benchmark model (figure 14b). Both models followed the trend and timing of the Peano muscle force in the rise and fall regions reasonably.

6. Discussion

Validation of MAPMORE’s models in the previous section showed it models static and dynamic McKibben and Peano muscle behavior with good accuracy compared to accurate benchmark models. However, the validation results also raised a number of questions, which are discussed here.

6.1. Static force component models

The results in figures 9 and 10 demonstrate the accuracy of MAPMORE’s models for capturing the static behavior of McKibben and Peano muscles [35]. The second order nature of the simplified polynomial static model has a limited complexity to its curvature in the three dimensions it maps to the force-pressure-strain surface of the muscles’ static behaviors. Higher order polynomial models can also be tried [42], but the risk is that overfitting can occur. Overfitting can result in large prediction errors for independent variable values significantly outside of the values used in the training data set [35]. Sárosi’s empirical model is accurate, but has the disadvantage compared to MAPMORE of needing a initial values for its coefficients to be selected to ensure a good fit. This process can be aided with methods such as a genetic algorithms, but at the expense of an increased computation time for the model fitting process.

6.2. MAPMORE damping force component model motion and pressure segmentation

Segmentation by pressure of MAPMORE damping force models significantly increased NRMSE, or at best only made a small decrease. One suggestion as to why, is the difference in the method of conducting the dynamic experiments performed in this work and those of Cao *et al* [49]. Cao *et al* maintained a constant pressure in the McKibben muscle under test as it was suddenly stretched and then allowed to contract by another actuator. This meant that for a given experiment the damping of the muscle was modeled for a constant pressure. Many experiments were carried out with a range of different test pressures to determine the effect of pressure on damping behavior.

In this work, the pressure was not held constant for a given experiment. Instead, a pressure step was used to contract the muscle against the inertial mass. Hence the pressure varied rapidly from 0 kPa up to the target pressure, remained constant for a time, and returned rapidly to 0 kPa. Model segments were then fitted to each experiment’s data. Thus a model segment for a target pressure of 80 kPa, for example, was assumed to model the muscle’s damping at 80 kPa. The flaw in this compared to Cao *et al*’s approach is that it does not acknowledge that the data captured a range of pressures (during the rise and fall of the pressure step from 0 kPa to 80 kPa and back to 0 kPa in this example). This could affect model segment accuracy.

The difference in the pressure conditions used in this work’s experiments and those in Cao *et al*’s is significant and could explain why the pressure segmented MAPMORE models in this work were not useful to significantly reducing NRMSE. A possible

improvement would be to create the model based on dynamic perturbation experiments like those of Cao *et al.*'s. In this case the muscle's damping model segments would be based on a dynamic experiment with a constant pressure. Fusion of these segments could then be used for prediction of damping force in muscles with a constant or varying pressure.

6.3. Damping force component models

MAPMORE's models proved to be the most accurate at predicting the damping force in the McKibben and Peano muscles. However, all damping models tested in this work failed to capture the timing and shape of the oscillations between the rise and fall peaks in the step response data. This could have been due to the fluid dynamics of the fluid column in the 0.5 m long tube connecting the pressure sensor to the muscle in the experiment's test rig. Hence, an improvement could be to mount the pressure sensor in the muscle itself.

Another issue with MAPMORE's models is that they tended to lag or lead the rise and fall peaks, or not quite estimate the peaks' durations correctly. A potential improvement to MAPMORE's models could be made by including time delayed (lag) terms of the variables in the dictionary [59] to enable time delayed dynamics to be incorporated in MAPMORE's models.

A specific limitation of the benchmark models, highlighted by the faster moving Peano muscle, is their inability to predict a nonzero damping force when $\dot{\epsilon}$ was zero. This happened when the muscle volume was small and its internal pressure increased rapidly before it began to move, as was the case with the low deadvolume of the Peano muscle. This is shown in the first 0.09s of the Peano muscle data (figure 13a). During this time the muscle began to produce a rapidly changing force as it was pressurized, but had not yet begun to move. Although the muscle was not moving, damping was still relevant because of flow restriction and fluid dynamics in the muscle and in the tubing supplying it with fluid. These dynamics explain, at least in part, the rapid changes in muscle pressure during this time. Hence, the benchmark damping models need to be used in tandem with a model of the pressurization system's dynamics to capture this kind of behavior. In contrast, MAPMORE correctly included a term independent of $\dot{\epsilon}$ (that is, the term $\dot{P}\epsilon$) in its Peano muscle model to account for the higher frequency dynamics of the Peano muscle. This allowed it to model the rise peak more accurately.

6.4. Dynamic force models

The combined MAPMORE static and damping force models were validated as more accurate than the most accurate of benchmark models for both the McKibben and Peano muscle. In the results it was observed that dynamic modeling of the rise and fall regions of the McKibben muscle could be improved. One possibility is addition of lag terms and more direct pressure measurements, as mentioned previously.

Dynamic modeling also showed that to model complete dynamic behavior of a muscle in a step response experiment (and not just the transient rise and fall dynamics as well as the static force), extra effects such as hysteresis need to be accounted for [60]. Possible hysteresis modeling approaches could include the Bilinear [59], Backlash operator [37] and Maxwell-slip virgin curve [36].

7. Conclusions and future work

Muscle-like actuators such as McKibben and Peano muscles have the potential to improve the ability of robots to interact with, and operate in real-world situations. However, their complex behavior makes them difficult to model and control. This work presented the validation of models generated by the MAPMORE algorithm as more accurate than existing accurate static and damping force models of Peano and McKibben models. In particular, its static model's NRMSE was 88 % of the most accurate static force benchmark model's NRMSE in McKibben muscles, and 69 % of the NRMSE of the most accurate static force benchmark model in Peano muscles. MAPMORE generated damping models that had an NRMSE of 71 % and 48 % respectively of the NRMSEs of the most accurate damping force models for McKibben and Peano muscles. These results demonstrate the accuracy of MAPMORE generated models and hint at their potential versatility to model different types of muscle-like actuators' static and dynamic behaviors.

This work had experimental and modeling limitations. Experimental limitations include the nonconstant pressure of the the dynamic and damping force model experiments. This possibly reduced model accuracy. Also, the lack of lag terms in MAPMORE's damping model dictionary and the lack of a hysteresis model were further factors that reduced the accuracy of MAPMORE's models.

As suggested by the results in this work, there are a number of future directions that could be taken to further improve MAPMORE. These include: MAPMORE's application to hysteresis modeling; validation of the usefulness of lag terms in MAPMORE's dictionary, particularly for modeling dynamic behavior; and implementation and performance validation of MAPMORE's models in a real-time control platform for a physical muscle-like actuator.

Acknowledgments

A. J. Veale is thankful for the financial support of The University of Auckland Doctoral Scholarship and the help of Dean Veale with testing and manufacture of the fluidic muscles.

References

- [1] Van Ham R, Sugar T G, Vanderborght B, Hollander K W and Lefeber D 2009 *IEEE Robot. Autom. Mag.* **16** 81–94

- [2] Gopura R A R C, Bandara D S V, Kiguchi K and Mann G K I 2016 *Robot. Auton. Syst.* **75** 203–20
- [3] Shorter K A, Xia J, Hsiao-Wecksler E T, Durfee W K and Kogler G F 2013 *IEEE/ASME Trans. Mech.* **18** 337–47
- [4] Mohammed S, Amirat Y and Rifai H 2012 *Adv. Robot.* **26** 1–22
- [5] DeDonato M, Dimitrov V, Du R, Giovacchini R, Knoedler K, Long X, Polido F, Gennert M A, Padr T, Feng S, Moriguchi H, Whitman E, Xinjilefu X and Atkeson C G 2015 *J. Field Robot.* **32** 275–92
- [6] Trevelyan J, Hamel W R and Kang S C 2016 Robotics in Hazardous Applications *Springer Handbook of Robotics* ed Siciliano B and Khatib O (Cham, Switzerland: Springer) pp 1521–48
- [7] Madden J D W, Vandesteeg N A, Anquetil P A, Madden P G A, Takshi A, Pytel R Z, Lafontaine S R, Wieringa P A and Hunter I W 2004 *IEEE J. Ocean. Eng.* **29** 706–28
- [8] Kellaris N, Venkata V G, Smith G M, Mitchell S K and Keplinger C 2018 *Science Robotics* **3** eaar3276
- [9] Haines C S, Lima M D, Li N, Spinks G M, Foroughi J, Madden J D W, Kim S H, Fang S, de Andrade M J, Göktepe F, Mirvakili S M, Naficy S, Lepro X, Oh J, Kozlov M E, Kim S J, Xu X, Swedlove B J, Wallace G G and Baughman R H 2014 *Science* **343** 868–72
- [10] Niino T, Higuchi T and Egawa S 1995 Dual excitation multiphase electrostatic drive *IEEE Ind. Appl. Conf. Thirtieth IAS Ann. Meeting (Orlando)* (IEEE) pp 1318–25
- [11] Yokota S, Yajima F, Takemura K and Edamura K 2010 *Adv. Robot.* **24** 1929–1943
- [12] Chou C P and Hannaford B 1996 *IEEE Trans. Robot. Autom.* **12** 90–102
- [13] Daerden F 1999 *Conception and realization of pleated pneumatic artificial muscles and their use as compliant actuation elements* Ph.D. thesis Vrije Universiteit Brussel
- [14] Mettam A R 1962 Inflatable servo actuators Tech. rep. Ministry of Aviation, Aeronautical Research Council, England
- [15] Yang D, Verma M S, So J H, Mosadegh B, Keplinger C, Lee B, Khashai F, Lossner E, Suo Z and Whitesides G M 2016 *Advanced Materials Technologies* **1** 1600055
- [16] Marchese A D, Katzschmann R K and Rus D 2015 *Soft Robotics* **2** 7–25
- [17] Mori M, Suzumori K, Takahashi M and Hosoya T 2010 *Adv. Robot.* **24** 233–54
- [18] Niiyama R, Sun X, Sung C, An B, Rus D and Kim S 2015 *Soft Robotics* **2** 59–70
- [19] Marchese A D, Onal C D and Rus D 2014 *Soft Robotics* **1** 75–87
- [20] Katzschmann R K, Marchese A D and Rus D 2016 Hydraulic autonomous soft robotic fish for 3D swimming *Experimental Robotics* ed Hsieh M A, Khatib O and Kumar V (Springer) pp 405–20
- [21] Noritsugu T, Yamamoto H, Sasakil D and Takaiwa M 2004 Wearable power assist device for hand grasping using pneumatic artificial rubber muscle *SICE 2004 Annual Conf. (Sapporo)* vol 1 (IEEE) pp 420–5
- [22] Sanan S, Lynn P S and Griffith S T 2014 *J. Mech. Robot.* **6** 031003
- [23] Ilijevski F, Mazzeo A D, Shepherd R F, Chen X and Whitesides G M 2011 *Angewandte Chemie* **123** 1930–5
- [24] Martinez R V, Fish C R, Chen X and Whitesides G M 2012 *Adv. Funct. Mater.* **22** 1376–84
- [25] Connolly F, Polygerinos P, Walsh C J and Bertoldi K 2015 *Soft Robotics* **2** 26–32
- [26] Gaylord R H 1958 Fluid actuated motor system and stroking device US Patent 2 844 126
- [27] Schulte H F 1961 The characteristics of the McKibben artificial muscle *The Application of External Power in Prosthetics and Orthotics* (Washington: National Academy of Sciences) pp 94–115
- [28] Caldwell D G, Tsagarakis N G, Kousidou S, Costa N and Sarakoglou I 2007 *Int. J. Human. Robot.* **4** 549–73
- [29] Niiyama R, Rus D and Kim S 2014 Pouch motors: Printable/inflatable soft actuators for robotics *IEEE Int. Conf. Robot. Autom.* (IEEE) pp 6332–7
- [30] Park Y L, Santos J, Galloway K G, Goldfield E C and Wood R J 2014 A soft wearable robotic device for active knee motions using flat pneumatic artificial muscles *IEEE Int. Conf. Robot. Autom.* (IEEE) pp 4805–10

- [31] Veale A J, Anderson I A and Xie S Q 2015 The smart Peano fluidic muscle: A low profile flexible orthosis actuator that feels pain *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems (San Diego)* vol 9435 (SPIE) pp 94351V1–11
- [32] Tondu B 1997 Analysis and modeling of the dynamic behaviour of the McKibben artificial muscle *IFAC Robot Control (Nantes)* vol 30 (Elsevier) pp 295–300
- [33] Tondu B 2012 *J. Intell. Material Syst. Struct.* **23** 225–53
- [34] Case J C, White E L and Kramer R K 2015 *Soft Robotics* **2** 80–7
- [35] Veale A J, Xie S Q and Anderson I A 2017 Accurate and versatile multivariable arbitrary piecewise model regression of nonlinear fluidic muscle behavior *IEEE Int. Conf. Mechatron.* (IEEE) pp 254–9
- [36] Vo-Minh T, Tjahjowidodo T, Ramon H and Van Brussel H 2011 *IEEE/ASME Trans. Mechatron.* **16** 177–86
- [37] Lin C J, Lin C R, Yu S K and Chen C T 2015 *Mechatronics* **28** 35–45
- [38] Van Damme M, Beyl P, Vanderborght B, Van Ham R, Vanderniepen I, Versluys R, Daerden F and Lefeber D 2008 Modeling hysteresis in pleated pneumatic artificial muscles *IEEE Int. Conf. Robot., Automat. Mechatron. (Chengdu)* (IEEE) pp 471–6
- [39] Tsagarakis N and Caldwell D G 2000 Improved modelling and assessment of pneumatic muscle actuators *IEEE Int. Conf. Robot. Autom. (San Francisco)* vol 4 (IEEE) pp 3641–6
- [40] Martens M and Boblan I 2017 *Actuators* **6** 33
- [41] Sárosi J 2012 New approximation algorithm for the force of fluidic muscles *IEEE Int. Symp. Appl. Comput. Intell. Informat.* (IEEE) pp 229–33
- [42] Tóthová M, Pitel’ J, Hošovský A and Sárosi J 2015 *Int. J. Math. Comput. Simulat.* **9** 228–33
- [43] Bertetto A M and Ruggiu M 2004 *Mech. Res. Commun.* **31** 185–94
- [44] Hošovský A, Pitel’ J, Židek K, Tóthová M, Sárosi J and Cveticanin L 2016 *Mech. Mach. Theory* **103** 98–116
- [45] Veale A J, Xie S Q and Anderson I A 2016 *Smart Mater. Struct.* **25** 065014
- [46] Reynolds D B, Repperger D W, Phillips C A and Bandry G 2003 *Ann. Biomed. Eng.* **31** 310–7
- [47] Tondu B 2013 Closed-loop position control of artificial muscles with a single integral action: Application to robust positioning of McKibben artificial muscle *IEEE Int. Conf. Mechatron. (Vicenza)* (IEEE) pp 718–23
- [48] Kerschler T, Albiez J, Zöllner J and Dillmann R 2006 Evaluation of the dynamic model of fluidic muscles using quick-release *IEEE/RAS-EMBS Int. Conf. Biomed. Robot. Biomechatron. (Pisa)* (IEEE) pp 637–42
- [49] Cao J, Xie S Q, Zhang M and Das R 2014 A new dynamic modelling algorithm for pneumatic muscle actuators *Int. Conf. Intell. Robot. Appl. (Guangzhou)* (Springer) pp 432–40
- [50] Sárosi J, Biro I, Nemeth J and Cveticanin L 2015 *Mech. Mach. Theory* **85** 25–34
- [51] Peternel L, Ugurlu B, Babič J and Morimoto J 2015 Assessments on the improved modelling for pneumatic artificial muscle actuators *Int. Conf. Adv. Robot. (Istanbul)* (IEEE) pp 34–9
- [52] Billings S A 2013 *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains* (Chichester: John Wiley & Sons, Ltd)
- [53] Billings S A, Chen S and Korenberg M J 1989 *Int. J. Control* **49** 2157–89
- [54] Rodríguez-Vázquez K and Fleming P J 1997 A genetic programming / NARMAX approach to nonlinear system identification *Genetic Algorithms in Eng. Syst.: Innovations and Applicat. (Glasgow)* (IET) pp 409–14
- [55] Guo Y, Guo L Z, Billings S A and Wei H L 2015 *Int. J. Syst. Sci.* **46** 776–89
- [56] Piroddi L and Spinelli W 2003 *Int. J. Control* **76** 1767–81
- [57] Solares J R A and Wei H L 2015 *Nonlinear Dyn.* **82** 201–15
- [58] Veale A J, Xie S Q and Anderson I A 2016 *Smart Mater. Struct.* **25** 065013
- [59] Zhang B, Lang Z Q, Billings S A, Tomlinson G R and Rongong J A 2013 *Mech. Syst. Signal Pr.* **39** 207–226
- [60] Vo-Minh T, Kamers B, Ramon H and Van Brussel H 2012 *Mechatronics* **22** 923–933