Accurate and Versatile Multivariable Arbitrary Piecewise Model Regression of Nonlinear Fluidic Muscle Behavior

Allan J. Veale and Sheng Q. Xie
Department of Mechanical Engineering
University of Auckland
Auckland, New Zealand
avea007@aucklanduni.ac.nz, s.xie@auckland.ac.nz

Iain A. Anderson
Biomimetics Laboratory
Auckland Bioengineering Institute
Auckland, New Zealand
i.anderson@auckland.ac.nz

***Abstract*—Wearable exoskeletons and soft robots require actuators with muscle-like compliance. These actuators can benefit from the robust and effective interaction that biological muscles' compliance enables them to have in the uncertainty of the real world. Fluidic muscles are compliant but difficult to control due to their nonlinear behavior. Precise control of these actuators needs accurate models that readily capture this behavior. Here we present the multivariable arbitrary piecewise model regression (MAPMORE) algorithm for automatically creating accurate data-driven, behavior-based models for fluidic muscles. MAPMORE integrates an arbitrary term dictionary based orthogonal forward regression algorithm with piecewise function fusion. We modeled the static and hysteresis force components of a McKibben pneumatic artificial muscle (PAM) and a Peano muscle with MAPMORE, Sárosi's empirical model, and a polynomial model. In all cases, MAPMORE's models had the best mean accuracy of below 15N. This shows it to be an easy to use, accurate, and versatile soft fluidic actuator modeling tool.**

Keywords—fluidic muscle; McKibben PAM; Peano muscle; MAPMORE; soft actuator; nonlinear behavior; modeling

I. Introduction

Soft, wearable robotic devices are moving out of the research lab and into the real world. Robotic exoskeletons can aid therapists in delivering rehabilitative treatment [1]. Soft robots have the resilience and adaptive interaction necessary to function in the uncertainty of real-world environments [2]. These robots require muscle-like actuators, artificial muscles, that are capable of robustly and controllably acting in everyday situations, like their biological counterparts [3]. Their inherent compliance, or ability to deform elastically under loads, promotes actuator efficiency and safety [4]. This unique property is enabling robots to bridge the gap from working in predictable workspaces to augmenting the abilities of humankind in the undefined and open nature of the real world.

The most popular subset of artificial muscles is fluidic muscles, in particular, McKibben pneumatic artificial muscles (PAMs). The McKibben PAM features an elastomer tube wrapped in a fiber braid. When inflated, Fig. 1(a) shows how the stiff fibers couple the tube's radial expansion to its ends, shortening the muscle. McKibben PAMs are frequently applied to robots for their large force generation and inherent compliance [5], but their use is limited by the difficulty in modeling and hence controlling them [6]. Another high-force, inherently compliant fluidic muscle is the Peano muscle, also known as a pouch motor, or flat PAM [7]-[12]. As shown in Fig. 1(b), it consists of a row of side-by-side inelastic tubes connected along their lengths. When inflated with a fluid, the

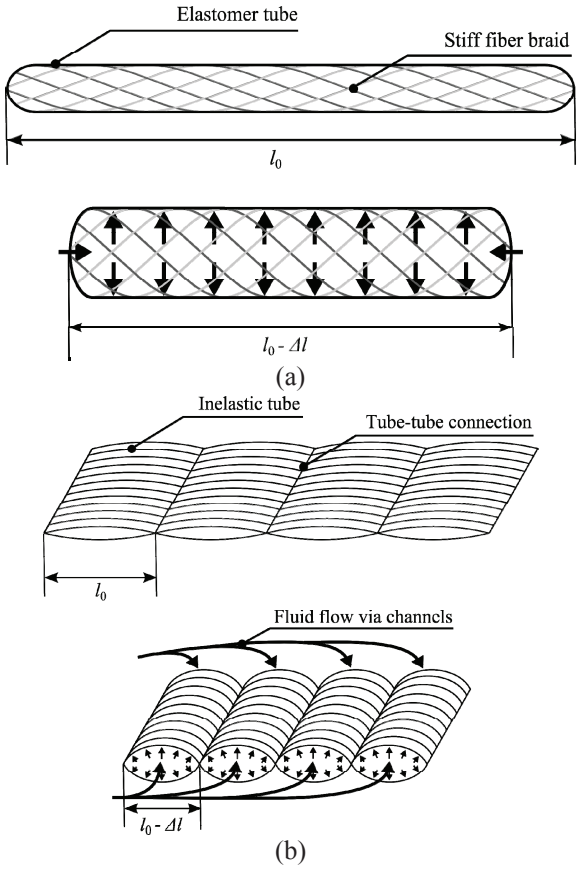

Fig. 1. The McKibben PAM (a) and Peano muscle (b) concepts. Both muscles' tubes contract from an initial length l_0 by a displacement Δl when inflated with a fluid. Fluid flow through the muscle is indicated by the arrows. In the McKibben PAM, fluid flows through the center of its fiber braid reinforced elastomer tube. In the Peano muscle, fluid flows through channels (not shown) connecting its multi-tube assembly.

flat, deflated tubes tend towards a circular cross-section so that the muscle shortens. Fluid flows between the tubes through channels fluidically connecting the tubes. The edges of the first and last tubes serve as mounting points to apply a force over a robot's joint. Unlike the McKibben PAM geometry, the Peano muscle's force is proportional to its planform, not cross-sectional area. Therefore, it is possible to construct actuators as thin as manufacturing and materials allow. This paves the way for fluidic muscles that can provide discreet and distributed actuation of soft-bodied robots [11].

Fundamentally, fluidic muscles are made of soft, viscoelastic materials that are difficult to characterize [6]. As such, they exhibit highly nonlinear behaviors including a threshold pressure, a distinct unpressurized force-strain curve, pressure dependent free-strain (maximum unloaded muscle contraction) [13], and hysteresis [14]. Models attempting to account for these nonlinearities become complex [6]. Ideally, the behavior of a fluidic muscle is easily and transparently captured by a model that has sufficiently low prediction error for controlling the motion of wearable and soft robots. In this study, we investigate the potential of an algorithm that automatically builds accurate and understandable models of nonlinear fluidic muscle behavior.

The following section highlights the achievements and limitations of related fluidic muscle modeling approaches. Section III provides an overview of the Multivariable Arbitrary Piecewise MOdel REgression (MAPMORE) algorithm. Next, we experimentally compare and validate MAPMORE against two accurate empirical models in section IV. Lastly, section V summarizes this study's key contributions and proposed future work.

II. Accurate Fluidic Muscle Modeling

Techniques from across the modeling spectrum have been applied to fluidic muscles. On one hand, physics-based models are derived from physical laws describing the fundamental operation of the muscle. The most accurate of McKibben PAM physics-based models have errors up to 10% [15] and still rely on some experimentally fitted coefficients. At the other end of the spectrum lie black-box models, in which the model form and parameters are characterized by experimental data. Hošovský *et al.* [16] and Jamwal *et al.* [17] have achieved modeling errors less than 3% using neural network and fuzzy logic black-box models. Although accurate, they are not transparent. Their generic and redundant model structures hide the physical meaning of the muscle behavior they approximate. Others have combined physics and empirical knowledge in lumped parameter [18], finite element method [19], and empirical models [20]. Notably, fluidic muscles with highly nonlinear static characteristics are more accurately modeled by techniques based on experimental data. Reasons for this include the inability of current physical laws to describe fluidic muscles' complex behavior [6] and the variability in fluidic muscles' unit-to-unit material properties [21].

An example of an accurate empirical model is Sárosi's [20] static McKibben PAM model in (1). Its model form also has potential to model the Peano muscle's static behavior [12]. It predicts muscle force, F, as a function of pressure, P, and strain, ε. Coefficients a_1 - a_6 are used to fit the model to empirical data.

$$F(P,\varepsilon) = (a_1 P + a_2)e^{a_3\varepsilon} + a_4\varepsilon P + a_5 P + a_6 \qquad (1)$$

Tóthová *et al.* [22] proposed another accurate empirical model. It predicts McKibben PAM force as the function of a 21 coefficient fifth order polynomial of pressure and strain. Although not as compact as Sárosi's model, it is more accurate [22] and may be able to model nonlinearities in the Peano muscle's behavior.

The muscle behavior captured by these empirical models is seen in their mathematical form, particularly in Sárosi's simpler model. This gives them a degree of transparency. However, neither model has been applied to other types of fluidic muscle or different muscle nonlinearities, such as the hysteresis component. Furthermore, if these models are not suitable, new model terms must be manually tried until a good fit is found. A more versatile solution would be a modeling tool that automatically selected, fitted, and pieced together model terms.

III. The MAPMORE Algorithm

MAPMORE is a modeling algorithm that automatically builds multivariable piecewise functions. Fig. 2(a)-(g) shows how it works when creating a static model of fluidic muscle force. Note that all symbols used in the MAPMORE algorithm are summarised at the end of this Section in Table I. Referring to Fig. 2, first, in (a), we measure muscle force over the muscle's contraction range at various pressures. MAPMORE uses this set of force-strain curves to build its model. Each curve is split (b) into segments. Then for each segment, we generate (c) a function, f_i, of the independent variables $\mathbf{x}$. These piecewise functions are constructed with Billings *et al.*'s [23], [24] forward regression orthogonal least squares estimator (FROLS) algorithm. We supply this algorithm with a dictionary of candidate function terms $\mathbf{D}(\mathbf{x})$. FROLS returns the indices of the selected terms, $\mathbf{s}$, whose linear combination best describe the segment data, along with those terms' coefficients, $\boldsymbol{\theta}$. After generating segment functions, we define (d) and generate (e) a fusion weight function w_{ij} for each independent variable with index j, of each segment function with index i. The weight functions smooth the transition between adjacent segment functions by gradually turning one function off and the next on. Each weight function defined in (d) turns on between the rise start $x_{\mathrm{rs},ij}$ and rise end $x_{\mathrm{re},ij}$. It turns off between the fall start $x_{\mathrm{fs},ij}$ and fall end $x_{\mathrm{fe},ij}$. We calculate the width of the rise and fall zones as a proportion of the segment length. In this example, these functions are calculated for strain and pressure. The pressure weight functions are triangular to interpolate between segment functions we have generated at discrete pressure values. Lastly, we combine the fusion weight and segment functions to create a fused piecewise model. The weight functions fuse the force-strain segment functions at a given pressure (f) and fuse these together to form the final model $y(\mathbf{x})$ (g).

MAPMORE is versatile, transparent, and accurate. The model generated by MAPMORE can have an indefinite number of variables and is not limited to static models. Its flexibility stems from its dictionary of arbitrary terms. For

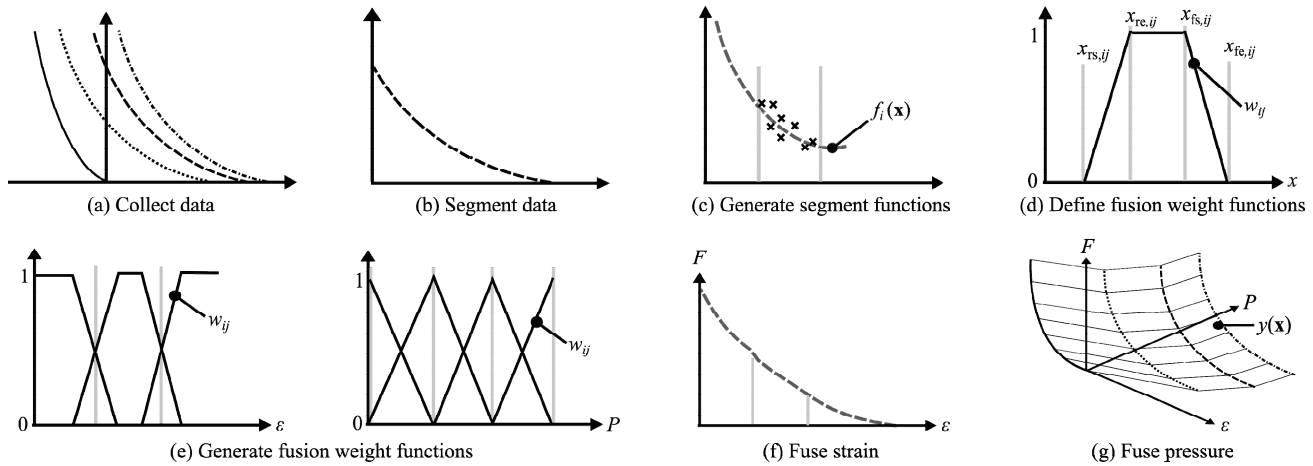

Fig. 2. The MAPMORE algorithm applied to static modeling of fluidic muscles. First, experiments are carried out on a fluidic muscle to determine the force it produces at different strains and pressures (a). Next, the data is split into segments (b) and piecewise functions created for each segment (c). We calculate fusion weight functions as a function of strain or pressure that switch these segment functions on and off. Finally, the piecewise segment functions for each static experiment are fused together with the fusion weight functions over strain (f), and then pressure (g), to form a smooth, three-dimensional surface. This surface represents the static force generation of the muscle as a function of its pressure and strain. Refer to Table I for definitions of the symbols.

example, they can be polynomials, physics-based, autoregressive, or with a local memory. Regardless, the creation of the final model from a linear combination of these user-supplied building blocks maximises its transparency. MAPMORE's models are not only easy to understand, they can be accurate. Although it is difficult to capture nonlinear behavior, the task is simplified by segmenting that behavior into regions that are modeled by independent functions. Thus, MAPMORE's use of fused piecewise functions enables accurate modeling of complex behavior, and it does so automatically. This last point is important because it takes the effort out of the trial and error process of selecting a suitable empirical model form and fitting its coefficients.

A. Piecewise Function Generation

MAPMORE uses a modified version of FROLS [23], [24] to generate piecewise functions. It builds these functions by selecting and fitting dictionary terms to segments of data following these steps:

1) The independent and dependent variable data are linearly scaled so both variables lie in the domain [−1,0] or [0,1] and the data trend passes through the origin. The independent variables' scaling slopes and intercepts are $\mathbf{m}_{\mathrm{x},i}$ and $\mathbf{c}_{\mathrm{x},i}$ and the dependent variable's scaling slope and intercept $m_{\mathrm{y},i}$ and $c_{\mathrm{y},i}$.

2) The scaled independent variable datapoints $\mathbf{X}_{\mathrm{t}}$ are evaluated for each of the N_{D} candidate terms in the dictionary. The result is N_{D} term vectors $\mathbf{p}_m$.

3) FROLS iteratively selects the candidate model terms and their coefficients that best explain the variation in the dependent variable data $\mathbf{y}_{\mathrm{t}}$. In each iteration, the unselected model terms are orthogonalized against the previously selected model terms (using the Gram-Schmidt method, for example). We then calculate the terms' potential coefficients g_m from the orthogonalized unselected model terms $\mathbf{q}_m$:

$$\mathrm{g_m} = \left(\mathbf{y}_{\mathrm{t}}{}^{\mathrm{T}}\mathbf{q}_{\mathrm{m}}\right) \Big/ \left(\mathbf{q}_{\mathrm{m}}{}^{\mathrm{T}}\mathbf{q}_{\mathrm{m}}\right) \tag{2}$$

Next we calculate how well they fit, measured by the error reduction ratio (ERR):

$$\mathrm{ERR_m} = \left(\mathrm{g_m}^2 \left(\mathbf{q}_{\mathrm{m}}{}^{\mathrm{T}}\mathbf{q}_{\mathrm{m}}\right)\right) \Big/ \left(\mathbf{y}_{\mathrm{t}}{}^{\mathrm{T}}\mathbf{y}_{\mathrm{t}}\right) \tag{3}$$

The model term with the highest ERR is selected. It is saved to previously selected orthogonalized term vectors, its index is saved to $\mathbf{s}$, and its coefficient to $\boldsymbol{\theta}$. When the selected model terms explain the variation in the dependent variable to within a specified tolerance, FROLS stops.

4) The segment function produced by FROLS is then a linear sum of the selected terms:

$$f_i(\mathbf{x}) = m_{\mathrm{y},i}\left(\sum_{k=1}^{N_{\mathrm{D},i}} \theta_k D_{s_k}\left(\mathbf{m}_{\mathrm{x},i}{}^{\mathrm{T}}\mathbf{x} + \mathbf{c}_{\mathrm{x},i}\right)\right) + c_{\mathrm{y},i} \tag{4}$$

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

Steps two and three summarize the FROLS algorithm, which is described in detail by Billings [24] and Wei *et al.* [25].

B. Piecewise Function Fusion

After generating segment functions, they are combined with the fusion weight functions w_{ij} to build the final model:

$$y(\mathbf{x}) = \sum_{i=1}^{N_{\mathrm{s}}}\left(f_i(\mathbf{x})\prod_{j=1}^{N_{\mathrm{x}}} w_{ij}\left(x_j, x_{\mathrm{rs},ij}, x_{\mathrm{re},ij}, x_{\mathrm{fs},ij}, x_{\mathrm{fe},ij}\right)\right) \tag{5}$$

Where N_{s} is the total number of segments and N_{x} is the number of independent variables. The weight function has a shape like that in Fig. 2(d). It switches $f_i(\mathbf{x})$ on in the rise

transition zone $x_{\mathrm{rs,ij}} < x_{\mathrm{j}} < x_{\mathrm{re,ij}}$ and off in the fall transition zone $x_{\mathrm{fs,ij}} < x_{\mathrm{j}} < x_{\mathrm{fe,ij}}$. We can model this zone with quadratic and cubic transitions, but here we use a linear function for simplicity:

$$w_{ij}\left(x_j, x_{\mathrm{rs},ij}, x_{\mathrm{re},ij}, x_{\mathrm{fs},ij}, x_{\mathrm{fe},ij}\right) = \begin{cases} 0 & x_j \le x_{\mathrm{rs},ij} \\ \dfrac{x - x_{\mathrm{rs},ij}}{x_{\mathrm{re},ij} - x_{\mathrm{rs},ij}} & x_{\mathrm{rs},ij} < x_j < x_{\mathrm{re},ij} \\ 1 & x_{\mathrm{re},ij} \le x_j \le x_{\mathrm{fs},ij} \\ \dfrac{x_{\mathrm{fe},ij} - x}{x_{\mathrm{fe},ij} - x_{\mathrm{fs},ij}} & x_{\mathrm{fs},ij} < x_j < x_{\mathrm{fe},ij} \\ 0 & x_{\mathrm{fe},ij} \le x_j \end{cases} \quad (6)$$

TABLE I. MAPMORE SYMBOL DEFINITIONS

Parameter	Definition
$\mathbf{c}_{x,i}$	independent variables' scaling intercepts for segment *i*
$c_{y,i}$	dependent variable's scaling intercept for segment *i*
D	dictionary of candidate function terms
ERR_m	orthogonalized term error reduction ratio
f	segment function
g_m	orthogonalized term potential coefficient
i	segment function index
j	independent variable index
k	selected dictionary term index
$\mathbf{m}_{x,i}$	independent variables' scaling slopes for segment *i*
$m_{y,i}$	dependent variable's scaling slope for segment *i*
N_{D}	number of dictionary candidate function terms
$N_{\mathrm{D},i}$	number of selected dictionary terms in segment *i*
N_{s}	total number of segments
N_{x}	number of independent variables
$\mathbf{p}_m$	term vector
$\mathbf{q}_m$	orthogonalized term vector
s	indices of selected dictionary terms
w	fusion weight function
x	independent variable(s)
x_{re}, x_{fe}	transition zone rise and fall end
x_{rs}, x_{fs}	transition zone rise and fall start
$\mathbf{X}_{\mathrm{t}}$	scaled independent variable training data
y	MAPMORE generated model
$\mathbf{y}_{\mathrm{t}}$	dependent variable segment training data
$\boldsymbol{\theta}$	coefficients of selected dictionary terms

IV. EXPERIMENTS

Given the potential accuracy and versatility of the MAPMORE algorithm, our aim is to apply it to modeling of different types of nonlinearities common to McKibben PAM and Peano muscle behavior. These nonlinearities are the static and hysteresis force components of the muscle during constant pressure contraction and extension experiments.

A. Method

We pressurized the muscle under test, a Festo DMSP-20-88N-RM-RM McKibben PAM or a four-tube Peano muscle prototype (as in [10]), and using the test-rig described in [12], measured its force and strain as it quasistatically contracted and extended. We used pressures of 0, 80, 150, 250, and 350kPa for the McKibben PAM and 0, 80, 200, 350, and 500kPa for the Peano muscle. The muscles' hysteresis yielded separate contraction and extension force-strain curves, which we averaged to find the static force curves. The differences between the upper extension curves and the static curves gives the positive hysteresis component curves. The differences between the contraction and static curves are identical to the positive hysteresis component curves, but negative. We used all the static and hysteresis component curves for model training except for the 250kPa curves for the McKibben PAM and the 350kPa curves for the Peano muscle. The 250kPa and 350kPa curves were used to validate MAPMORE. We also used them to validate the accuracy of the similarly trained Sárosi and fifth order polynomial models. These models provided an accuracy benchmark for comparing to MAPMORE's performance. The accuracy metric we were interested in is the validation fit root mean square error (RMSE).

In this preliminary study, we setup MAPMORE to have three segments, a fourth order polynomial term dictionary ($\mathbf{D}(\mathbf{x}) = \{1, x, x^2, x^3, x^4\}$), and a 25% transition zone for strain fusion weight functions. These settings were not optimized, but were chosen as a reasonable starting point given the nonlinearities in the data. The number of segments and order of dictionary terms appeared sufficient to manually construct a piecewise model; and the 25% transition zone size is halfway between a step transition (0%) and the most gradual transition (50%). The segments were evenly spaced in the static model; in the hysteresis model we tried a gradient segmentation method that located the segments at the greatest discontinuities in the data.

B. Results

Fig. 3 shows that MAPMORE and Sárosi's models fitted both muscles' static data well while the polynomial model wildly overestimated the static force. MAPMORE and Sárosi's models underestimated force, with the lowest prediction errors in the 0 to 0.1 strain range and increasing error beyond that. The polynomial model approximately followed the trend of the Peano muscle force, but diverged to large positive error for McKibben PAM strains over 0.075. These observations are reflected by the RMSEs in Fig. 4 of at least 170N for the polynomial model and below 20N for the other models. MAPMORE's models proved most accurate with RMSEs of 13.7N and 8.4N compared to Sárosi's 18.7N and 11.4N for the McKibben PAM and Peano muscle respectively. These results followed the fit expected from the training errors except for the polynomial model. While the polynomial model had the best training RMSEs at under 1N, it had poor accuracy with the validation data.

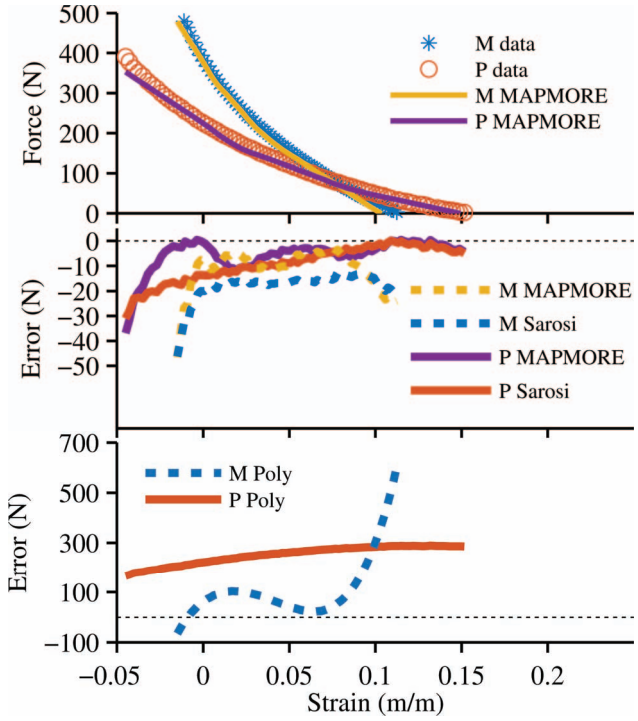

Fig. 3. The accuracy of the MAPMORE, Sárosi, and polynomial (Poly) models applied to modeling the static force component of a McKibben PAM (M) and Peano muscle (P). The top plot shows the static force validation data for the muscles overlaid with the MAPMORE model. The fit of Sárosi's model is very similar. The lower plots show the prediction error of the three models.

Fig. 5 reveals the same trends when the models are applied to the muscles' hysteresis component. Both MAPMORE and Sárosi's models fit well, but the polynomial model consistently overestimates despite having the lowest training RMSEs of no more than 1N. As for the static component modeling, MAPMORE and Sárosi's validation RMSEs (Fig. 6) are less for the Peano muscle. Also notable is how MAPMORE's gradient segmentation method splits the data around the maximum force in Fig. 5, following the decreasing force at low strains trend particularly well for the Peano muscle. In contrast, Sárosi's model (not shown in Fig. 5) does not do this, instead fitting a nearly straight line through the data.

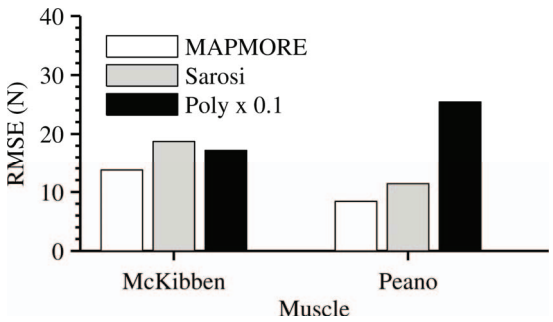

Fig. 4. The RMSEs for the static model validation plots in Fig. 3. The MAPMORE and Sárosi models both have RMSEs below 20N while the polynomial model (Poly) has RMSEs above 170N. Note the polynomial RMSEs are downscaled by a factor of 10.

C. Discussion

Compared to existing PAM empirical models, MAPMORE is an accurate and versatile modeling algorithm. It automatically produced static and hysteresis force component models for a McKibben PAM and Peano muscle. These models were transparent, with piecewise polynomial terms up to the second order being sufficient to capture the muscles' nonlinear behavior. Modeling this behavior automatically is easier than manually fitting empirical models such as the polynomial model and Sárosi's model. These latter models fitted the training data well, but needed trial and error selection of their coefficients' initial values to get a good fit. This can take some time when there are a large number of coefficients. These problems can be overcome with genetic training algorithms, but at the expense of significant computation and no guarantee of a good and generalizable fit. Another issue with fitting higher order polynomials is that the fitting algorithm may have trouble with very large or very small coefficients and require scaling of the training data.

Higher order terms also introduce other problems if they are redundant to the trends in the data they are fitted to. This is the reason the polynomial model had a low validation accuracy. It did not need pressure terms up to the fifth order. With only four pressures to fit to, these were poorly fitted for generalizing to other pressures. With these redundant higher order pressure terms the polynomial model requires more force-strain curves at different pressures to fit properly and get the accuracy in [22].

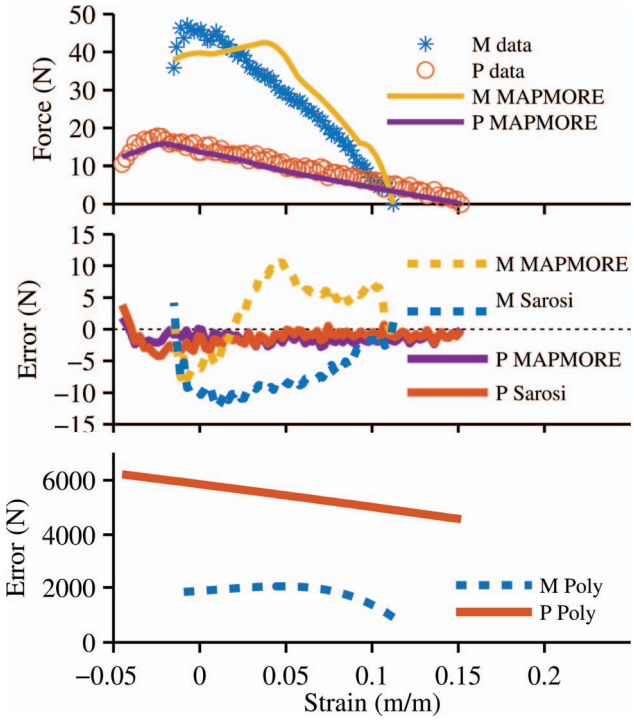

Fig. 5. The accuracy of the MAPMORE, Sárosi, and polynomial (Poly) models applied to modeling the hysteresis force component of a McKibben PAM (M) and Peano muscle (P). The top plot shows the static force validation data for the muscles overlaid with the MAPMORE model. The main trend and fit of Sárosi's model is similar although it does not capture the left hand dip in force. The lower plots show the prediction error of the three models.

MAPMORE can also be improved. Although its models were accurate, they still had errors due to repeatability of the actuators' behaviors and limitations of MAPMORE itself. For example, MAPMORE could be improved in the robustness of its gradient segmentation method, the type and transparency of its model terms, and completeness of its hysteresis modeling. The gradient segmentation method worked well in locating segmentation points for the Peano muscle's hysteresis component. However, it was less successful in segmenting the McKibben muscle's hysteresis component. It misplaced segments because of small, local data discontinuities. This could be resolved by averaging data before locating the discontinuities. The average window size could be set by the desired model feature resolution. Other improvements include the use of physics-based static terms to increase model transparency and extending the hysteresis component model to model hysteresis loops, as in [14].

V. Conclusion and Future Work

MAPMORE is a versatile modeling algorithm that can automatically produce accurate models of fluidic muscle behavior. MAPMORE had lower RMSEs than Sárosi's model and the fifth order polynomial model when modeling static and hysteresis force components in a McKibben PAM and Peano muscle. For example, MAPMORE's static model for a McKibben PAM had an RMSE of 13.7N compared to Sárosi's model's 18.7N.

Next, we plan to improve the robustness of MAPMORE's gradient segmentation method and extend it to model hysteresis loop behavior. When we implement its models on a real-time system, we also intend to compare its computation time with those of existing PAM models. This will see MAPMORE mature as an easy to use empirical modeling tool for fluidic muscle behavior.

Acknowledgment

A. J. Veale gratefully acknowledges the financial support of The University of Auckland Doctoral Scholarship, Jesus Christ for the inspiration behind the concept of the MAPMORE algorithm, and Dean Veale for help with testing the fluidic muscles.

References

[1] V. Klamroth-Marganska, J. Blanco, K. Campen, A. Curt, V. Dietz, T. Ettlin, M. Felder, B. Fellinghauer, M. Guidali, A. Kollmar, A. Luft, T. Nef, C. Schuster-Amft, W. Stahel, and R. Riener, "Three-dimensional, task-specific robot therapy of the arm after stroke: A multicentre, parallel-group randomised trial," *Lancet Neurol.,* vol. 13, no. 2, pp. 159-166, 2014.

[2] R. V. Martinez, A. C. Glavan, C. Keplinger, A. I. Oyetibo, and G. M. Whitesides, "Soft actuators and robots that are resistant to mechanical damage," *Adv. Funct. Mater.,* vol. 24, no. 20, pp. 3003-3010, 2014.

[3] J. D. W. Madden, N. A. Vandesteeg, P. A. Anquetil, P. G. A. Madden, A. Takshi, R. Z. Pytel, S. R. Lafontaine, P. A. Wieringa, and I. W. Hunter, "Artificial muscle technology: Physical principles and naval prospects," *IEEE J. Ocean. Eng.,* vol. 29, no. 3, pp. 706-728, 2004.

[4] K. W. Hollander, T. G. Sugar, and D. E. Herring, "Adjustable robotic tendon using a 'jack spring'™," in *Proc. 2005 IEEE 9th Int. Conf. Rehabilitation Robotics,* Chicago, IL, Jun. 28-Jul. 1, 2005, pp. 113-118.

[5] D. G. Caldwell, N. G. Tsagarakis, S. Kousidou, N. Costa, and I. Sarakoglou, "'Soft' exoskeletons for upper and lower body rehabilitation - design, control and testing," *Int. J. Hum. Robot.,* vol. 4, no. 3, pp. 549-573, 2007.

[6] B. Tondu, "Modelling of the McKibben artificial muscle: A review," *J. Intell. Mater. Syst. Struct.,* vol. 23, no. 3, pp. 225-253, 2012.

[7] S. Sanan, P. S. Lynn, and S. T. Griffith, "Pneumatic torsional actuators for inflatable robots," *J. Mechanisms Robotics,* vol. 6, no. 3, pp. 1-7, 2014.

[8] R. Niiyama, X. Sun, C. Sung, B. An, D. Rus, and S. Kim, "Pouch Motors: Printable soft actuators integrated with computational design," *Soft Robotics,* vol. 2, no. 2, pp. 59-70, 2015.

[9] Y. Park, J. Santos, K. G. Galloway, E. C. Goldfield, and R. J. Wood, "A soft wearable robotic device for active knee motions using flat pneumatic artificial muscles," in *IEEE Int. Conf. Robot. Autom.,* Hong Kong, May 31-Jun. 7, 2014, pp. 4805-4810.

[10] A. J. Veale, I. A. Anderson, and S. Q. Xie, "The smart peano fluidic muscle: A low profile flexible orthosis actuator that feels pain," in *SPIE Smart Structures and Materials NDE,* San Diego, CA, Mar. 8-12, 2015, pp. 94351V-1-11.

[11] A. J. Veale, S. Q. Xie, and I. A. Anderson, "Modeling the Peano fluidic muscle and the effects of its material properties on its static and dynamic behavior," *Smart Mater. Struct.,* vol. 25, no. 6, pp. 1-16, 2016.

[12] A. J. Veale, S. Q. Xie, and I. A. Anderson, "Characterizing the Peano fluidic muscle and the effects of its geometry properties on its behavior," *Smart Mater. Struct.,* vol. 25, no. 6, pp. 1-14, 2016.

[13] M. A. Meller, M. Bryant, and E. Garcia, "Reconsidering the McKibben muscle: Energetics, operating fluid, and bladder material," *J. Intell. Mater. Syst. Struct.,* vol. 25, no. 18, pp. 2276-2293, 2014.

[14] T. Vo-Minh, T. Tjahjowidodo, H. Ramon, and H. Van Brussel, "A new approach to modeling hysteresis in a pneumatic artificial muscle using the Maxwell-slip model," *IEEE Trans. Mechatron.,* vol. 16, no. 1, pp. 177-186, 2011.

[15] N. Tsagarakis and D. G. Caldwell, "Improved modelling and assessment of pneumatic muscle actuators," in *IEEE Int. Conf. Robotics and Automation,* San Francisco, CA, Apr. 24-28, 2000, pp. 3641-3646.

[16] A. Hošovský, J. Piteľ, K. Židek, M. Tóthová, J. Sárosi, and L. Cveticanin, "Dynamic characterization and simulation of two-link soft robot arm with pneumatic muscles," *Mech. Mach. Theory,* vol. 103, pp. 98-116, 2016.

[17] P. K. Jamwal, S. Hussain, and S. Q. Xie, "Dynamic modeling of pneumatic muscles using modified fuzzy inference mechanism," in *2009 IEEE Int. Conf. Robotics and Biomimetics,* Guilin, China, Dec. 19-23, 2009, pp. 1451-1456.

[18] C. Ferraresi, W. Franco, and G. Quaglia, "A novel bi-directional deformable fluid actuator," *Proc. Inst. Mech. Eng., Part C: J. Mech. Eng. Sci.,* vol. 228, no. 15, pp. 2799-2809, 2014.

[19] A. M. Bertetto and M. Ruggiu, "Characterization and modeling of air muscles," *Mech. Res. Commun.,* vol. 31, no. 2, pp. 185-194, 2004.

[20] J. Sárosi, "New approximation algorithm for the force of fluidic muscles," in *7th IEEE Int. Symp. Appl. Comput. Intell. Informat.,* Timisoara, Romania, May 24-26, 2012, pp. 229-233.

[21] J. C. Case, E. L. White, and R. K. Kramer, "Soft material characterization for robotic applications," *Soft Robotics,* vol. 2, no. 2, pp. 80-87, 2015.

[22] M. Tóthová, J. Piteľ, A. Hošovský, and J. Sárosi, "Numerical approximation of static characteristics of McKibben pneumatic artificial muscle," *Int. J. Math. Comput. Simulat.,* vol. 9, pp. 228-233, 2015.

[23] S. A. Billings, S. Chen, and M. J. Korenberg, "Identification of MIMO non-linear systems using a forward-regression orthogonal estimator," *Int. J. Control,* vol. 49, no. 6, pp. 2157-2189, 1989.

[24] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains,* Chichester, England: John Wiley & Sons, Ltd, 2013.

[25] H. Wei, S. Billings, and J. Liu, "Term and variable selection for non-linear system identification," *Int. J. Control,* vol. 77, no. 1, pp. 86-110, 2004.