

Article

A Piecewise Particle Swarm Optimisation Modelling Method for Pneumatic Artificial Muscle Actuators

Dexter Felix Brown and Shengquan Xie * 

School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, UK; el18dfb@leeds.ac.uk

* Correspondence: s.q.xie@leeds.ac.uk

Abstract: Pneumatic artificial muscles (PAMs) possess compliant properties desirable for certain applications such as prosthetics and robotic structures. However, this compliance along with their inherent nonlinear dynamics make them difficult to accurately model and as such accurately control under certain control architectures. Common approaches to this problem include measuring the actuator's physical properties and approximating a model based on these parameters or using deep learning methods to train a model with the actuator's behaviours. This paper introduces an optimisation-based modelling approach based on a particle swarm optimisation (PSO) algorithm using a mass–spring–damper approximation for the PAM, as well as a piecewise modelling method that accounts for nonlinear dynamics. The use of optimisation to estimate model parameters removes the need to measure physical properties, and the three-element approximation allows for fast model generation with low computational complexity and training data requirements. Through multiple tests comparing model behaviour with real PAM motion, the accuracy of these models is confirmed to be promising for future work. Dynamic nonlinearities are properly accounted for using the piecewise modelling method, including both hysteresis and disproportionate input/output relationship across the stroke length of the actuator. Compared with other PAM modelling techniques, this method has improved generation time, lower computational load requirements, and complexity and can be applied to actuators for which the phenomenological mass–spring–damper model is a good approximation.

Keywords: pneumatic artificial muscles; dynamic modelling; particle swarm optimisation; control systems

**Citation:** Brown, D.F.; Xie, S.A Piecewise Particle Swarm Optimisation Modelling Method for Pneumatic Artificial Muscle Actuators. *Actuators* **2024**, *13*, 286. <https://doi.org/10.3390/act13080286>

Academic Editor: Alessio Merola

Received: 28 June 2024

Revised: 26 July 2024

Accepted: 27 July 2024

Published: 29 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Pneumatic artificial muscles (PAMs) differ from more conventional actuators due to the inherently compliant behaviour and nonlinear dynamics arising from their construction of flexible materials and actuation method. These properties make them desirable for applications such as medical robotics in rehabilitation scenarios and prosthetics as they more closely mimic biological muscle than conventional actuators. This allows for a more comfortable human–robot interaction, as the compliance allows the user some give in their voluntary movement when under the external influence of a robot, as well as making the systems safer as it is less likely that unexpected forces or motion will injure the user or cause damage to the robot. They are also applicable to robotic platforms where a human-like motion is desired.

Various types of PAMs exist, suitable for different purposes based on differing characteristics, but all operate under the same principle. An elastic bladder is sealed and fixed at two ends, and upon pressurisation by compressed air will expand, applying force to the fixed ends of the actuator. A less elastic material surrounding the bladder, such as a harder rubber shell or braided fibre mesh, directs the force to cause the PAM to contract, expand, or bend about an axis depending on the construction [1]. The original fluidic

artificial muscles used in robotics were developed by Joseph McKibben in the 1950s for orthotics [2,3]. A more detailed review of the development of PAMs is presented in [1].

While the compliant behaviour of PAMs is desirable in many branches of industry and research, this and their nonlinear dynamics makes accurate motion control a challenge. Standard linear control systems such as proportional integral derivative (PID) are less effective than in more conventional applications as the friction present in the contraction/expansion of PAMs causes a hysteresis and changing behaviour with time, as well as the uncertainties caused by the fluid dynamics of pressurised air. Model-based control schemes can alleviate this problem by mapping the behaviour of the actuator to allow for more accurate control, but a similar problem arises with the complexity of these models. Many different approaches have been developed and studied in order to model PAMs that attempt to balance complexity with functionality, to find a solution that is accurate but does not take an exceedingly long time or large amount of computation to produce. The review presented in [4] describes some of the different approaches to the modelling problem of PAMs. The review suggests that geometric modelling is less suitable for these actuators as their parameters are difficult to accurately measure during use; however, their simplicity still makes the approach suitable for certain applications. The phenomenological modelling approach, which approximates the muscle as a mass–spring–damper system, is more suitable as this is a close approximation to the dynamics of PAMs based on their construction and flexible materials, and the simplicity makes them fairly easy to calculate. However, the hysteresis and nonlinearity over time is often not taken into consideration in these models. A simplified empirical model is also mentioned in which the PAM is simplified to that of a mechanical spring. Polynomial approximations of the parameters are used that allow for easy adjustment of the complexity/accuracy by increasing the degree of the model; however, again, these models suffer from not properly modelling the nonlinear behaviour of the PAMs.

Applications of these physical/mathematical models of PAMs are present throughout the literature. An early study on the phenomenological model for a McKibben muscle is presented in [5]. Experiments were performed to determine the changing parameters of the PAM with pressure, resulting in a model with 15% root mean squared error (RMSE), with the largest error and the peak and trough of the actuator's motion. An empirical model based on geometric constants of a McKibben muscle is presented in [6] including experimental parameter estimation. This model is used in model-based PID control of the actuator, with performance deemed satisfactory and within tolerance levels. A nonlinear numerical model for a McKibben muscle is shown in [7], showing a maximum error in experimental comparisons of 3% of the total displacement. Various models for contractile length, pulling force, and a specific application of four McKibben muscles in an arm configuration are shown in [8]. A mathematical model for an extensor PAM is presented in [9], differing from more commonplace contraction PAMs as these increase in length with increased pressure rather than decreasing.

PAMs developed by FESTO are popular in industrial applications as their construction makes them more reliable. These PAMs have longer operating lives than typical nylon-braided McKibben muscles and a greater force output capability; thus, they are also studied in many research applications including modelling. A piecewise three-element model of a FESTO PAM is presented in [10] for improved model predictive control (MPC). The piecewise model uses multiple three-element systems for different pressure levels in the PAM in order to overcome the nonlinearity of the actuator. The MPC developed with this piecewise model, in comparison with PID control, has much better performance. Mathematical models of FESTO PAMs are presented in [11,12], and phenomenological models are shown in [13,14].

In recent years, more research has gone into applying machine intelligence in the form of learning and optimisation to the problem of PAM modelling. This gives the advantage of potentially more accurate models with more sophisticated parameters as well as alleviating some of the need to measure the actuator's complex parameters and dynamics. Neural

networks are a popular algorithm used in this application; neural networks can be tailored to the specific needs of the model, and their complexity can be varied easily. A modelling method using an artificial neural network as well as hybrid backpropagation algorithm and genetic algorithm for the learning stage is presented in [15] for modelling a novel pleated PAM with promisingly accurate results obtained in experiments. Various other neural network-based modelling approaches have been applied to FESTO PAMs, including a recurrent fuzzy-neuro approach in [16] and neural network approaches for antagonistic pairs of PAMs in [17,18]. The architecture of each neural network used in these studies is widely similar, with multiple input nodes, a similar number of hidden nodes, and a single output.

Another common approach in applying intelligence in modelling is optimisation, with particle swarm optimisation (PSO) algorithms being a popular choice. PSO uses a swarm of particles that interact with each other as they move around a search space attempting to minimise some cost function. By using a large number of particles in the swarm, solutions can be found much faster. In the case of actuator modelling, each particle has a number of parameters that describe both its position in the search space as well as the parameters used in the equation describing the dynamics of the actuator. The objective function is typically some combination of errors between the actual behaviour of the actuator and the approximated behaviour based on each particle's parameters. Using this method, the actual parameters can be approximated rapidly and accurately, with the calculation time being largely invariant to the complexity of the model. An empirical model of a McKibben muscle with parameters generated using PSO is presented in [19] with the aim of characterising the hysteresis behaviour of the muscle. The model was verified to be accurate; however, the nonlinearities of the PAM are mentioned for causing some inaccuracies, and more intelligent control strategies are suggested as a solution. An inverse model for a PAM is presented in [20], also based on an PSO algorithm for system identification.

Limitations exist for all modelling practices. As mentioned in [4], geometric modelling is difficult to apply to PAMs accurately as online measuring of their nonlinear dynamic properties is difficult to achieve accurately. The standard phenomenological model using mass–spring–damper approximation also suffers lower accuracy due nonlinearity, as the linear model approximation does not account for hysteresis and disproportionate motion–input relationships. Intelligent learning algorithms, such as neural networks and genetic algorithm as presented in [15–18] and to an extent optimisation algorithms like PSO, can achieve highly accurate models of even complex nonlinear systems, but better approximations rely on sufficiently complex model parameterisation and learning datasets that can greatly increase the computational load and model generation time. This trade-off in model accuracy, complexity, and computation time must be accounted for.

This paper will present a piecewise PSO modelling method for a FESTO DMSP-20-400N PAM based on the phenomenological modelling approach. A three-parameter model based on the dynamics of the mass–spring–damper approximation for the PAM will be used as a template for the model, used for its good approximation of PAMs as well as its simplicity and low parameter count, reducing model complexity and improving generation time. In order to compensate for the nonlinearity of the PAM, a piecewise modelling method will be used with multiple different models for different sections of the actuator's stroke length. Models generated in this manner will be compared experimentally with the PAM under different loads and inputs to validate their accuracy.

2. Materials and Methods

2.1. Phenomenological Model

The phenomenological model is commonly used in modelling PAMs due to their approximately similar behaviour to mass–spring–damper systems. A three-element model

based on this will be used as a template for optimisation in this study. The equation of motion for this model is given as follows:

$$M\ddot{x}(t) + D\dot{x}(t) + Cx(t) = F_c - F_l \quad (1)$$

where M is the mass of the PAM, D is the damping element, C is the spring constant, F_c is the contractile force element applied by the internal pressure of the PAM, F_l is the force applied by external load to the system, and $x(t)$ is the displacement of the actuator from its nominal position at a given time step. This model is shown in Figure 1.

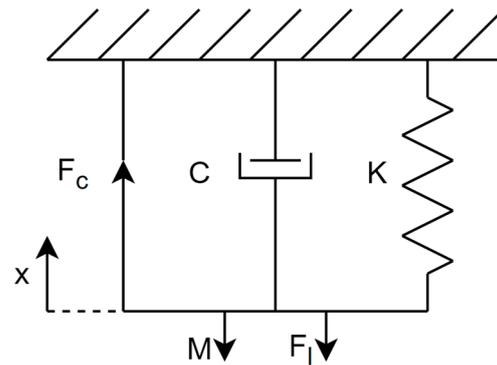


Figure 1. Phenomenological model of the PAM showing displacement x ; contractile force element F_c ; external load element F_l ; and PAM parameters M , C , and K .

The contractile force applied by the PAM is proportional to the input pressure, making F_c proportional to the control input of the system, which will be henceforth denoted with U .

Using the displacement and velocity of the system as state variables, the following equations are obtained:

$$x_1(t) = x(t) \quad (2)$$

$$x_2(t) = \dot{x}(t) \quad (3)$$

$$\dot{x}_1(t) = x_2(t) \quad (4)$$

The equation of motion can be re-written in terms of these states:

$$U(t) = M\dot{x}_2(t) + Cx_2(t) + Kx_1(t) \quad (5)$$

$$\dot{x}_2(t) = -\frac{K}{M}x_1(t) - \frac{C}{M}x_2(t) + \frac{U(t)}{M} \quad (6)$$

The output of the system is the displacement of the actuator, so the output of the model $y(t)$ is expressed as follows:

$$y(t) = x_1(t) \quad (7)$$

The state-space representation of this model in matrix form is given by the following:

$$\dot{X}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{C}{M} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix} U(t) \quad (8)$$

$$y(t) = [1 \quad 0] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + [0]U(t) \quad (9)$$

The unknown parameters of the system based on Equation (8) are $-\frac{K}{M}$, $-\frac{C}{M}$, and $\frac{1}{M}$. These three elements will be approximated through optimisation using the PSO algorithm and will be referred to henceforth as P_1 , P_2 , and P_3 respectively. The model equations for each state-space solution are given by Equations (9) and (10).

$$\dot{X}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ P_1 & P_2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ P_3 \end{bmatrix} U(t) \quad (10)$$

2.2. Particle Swarm Optimisation Algorithm

The classical PSO algorithm is referenced extensively in the literature [19–22]. It uses a group of potential solutions (particles) to an optimisation function that iteratively updates their parameters in response to some fitness function, with these parameters acting as coordinates in a search space. The particles base their movement around the search space on both their own best fitness performance and the overall best fitness found in the swarm. This sharing of data between solutions allows for fast optimisation convergence. The following equations describe the velocity and position updates for the particles at each iteration of the algorithm:

$$V_i(k+1) = WV_i(k) + C1Rand(X_{i\text{best}}(k) - X_i(k)) + C2Rand(G\text{best}(k) - X_i(k)) \quad (11)$$

$$X_i(k+1) = X_i(k) + V_i(k+1) \quad (12)$$

where i is the current particle; k is the current iteration of the algorithm; V is the particle's velocity in the search space; X is a particle's position in the search space quantified by its parameters; $Rand$ is a random value between 0 and 1; $X_{i\text{best}}$ is the position of the current particle which achieved the best (lowest) fitness; $G\text{best}$ is the global best position of all particles in the swarm; and W , $C1$, and $C2$ are algorithm parameters that are tuned manually.

The parameters to be optimised using PSO algorithm in this study are P_1 , P_2 , and P_3 as described in Equation (10). As such the search space is three-dimensional. The optimisation function used to determine the fitness of each particle is the root mean squared error (RMSE) between the expected displacement of the PAM based on the model generated by a particle's parameters and the actual displacement of the PAM, measured over a period of one repeated sin wave motion, as described in Equation (13).

$$F_i = \sqrt{\frac{\sum_{t=0}^N (x - y_i(t))^2}{N}} \quad (13)$$

where F_i is the fitness value for the current particle; t is the time step; N is the number of time steps in the measured motion sequence; x is the measured displacement of the PAM; and $y_i(t)$ is the output of the model based on Equations (9) and (10) using the three parameters of particle i as P_1 , P_2 , and P_3 at time step t .

Using this method, generated model solutions are iteratively improved based on how accurately they match the motion of the PAM. The update parameters of the algorithm are $W = 0.9$ to prevent divergent swarm motion and $C1 = C2 = 2$ to increase the efficacy of the algorithm and reach solutions in fewer iterations, while maintaining low enough velocities to keep the swarm coherent. The number of particles in the swarm is set to 50 as a balance between the time taken for each iteration and convergence time to a solution.

2.3. Piecewise Model

The nonlinear nature of PAMs makes them difficult to accurately approximate with linear models such as the three-element model described in Section 2.1. Nonlinear models of this nature are much more computationally complex and not feasible to generate using optimisation methods such as PSO. As such, a compromise is made in the form of piecewise models. Multiple models will be generated using different reference target motions of the PAM, as well as different input ranges within those motions. These models are then put together, each operating during the specific range of inputs to the system as specified by how they were generated. The resulting piecewise model comprised of multiple linear models should better approximate the nonlinearity of the PAM than a single linear model.

The nature of the PAM's nonlinearity is twofold. Internal friction caused by the braided fibres in the outer mesh cause a hysteresis to occur, and the method of actuation in inflation by pressurised air makes the displacement not linearly proportional to input pressure, with displacement velocity gradually decreasing along the stroke length of the actuator. To account for the hysteresis, different models can be generated for the actuator's inflation and deflation. The disproportionate motion across the stroke length can again be accounted for by generating multiple models for different input ranges.

Each piecewise model is therefore comprised of multiple individual three-parameter solutions as described in Equation (10). In order to generate the whole piecewise model solution, the PSO algorithm is run once for each model section, generating three parameters per section.

The model output for each part of a piecewise model solution would therefore be described using Equations (14) and (15).

$$\dot{X}_j(t) = \begin{bmatrix} \dot{x}_{j1}(t) \\ \dot{x}_{j2}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ P_{j1} & P_{j2} \end{bmatrix} \begin{bmatrix} x_{j1}(t) \\ x_{j2}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ P_{j3} \end{bmatrix} U(t) \quad (14)$$

With the specific model used based on the input $U(t)$ according to the following:

$$y(t) = \begin{cases} x_{11}(t) & \text{for } 0 \leq U(t) < K_1 \\ x_{21}(t) & \text{for } K_1 \leq U(t) < K_2 \\ x_{31}(t) & \text{for } K_2 \leq U(t) < K_3 \\ \dots & \dots \\ x_{n1}(t) & \text{for } K_{n-1} \leq U(t) \end{cases} \quad (15)$$

As well as differentiation of the inflating and deflating states according to the following:

$$y(t) = \begin{cases} x_{inf1}(t) & \text{for } \dot{U}(t) > 0 \\ x_{def1}(t) & \text{for } \dot{U}(t) \leq 0 \end{cases} \quad (16)$$

where j is the model section, n is the total number of linear model sections of the piecewise model, and K_i is the input bound determining the upper bound of the model section $X_j(t)$ and the lower bound of the model section $X_{j+1}(t)$. Each model section j will have a set of P_{j1} , P_{j2} , and P_{j3} that are independently generated for each j using the PSO algorithm for a total parameter count of $3j$ per piecewise model.

In cases where both the input ranges and inflating/deflating state differentiations are used, each model section $X_j(t)$ will constitute both an inflating and deflating state, resulting in both $X_{jinf}(t)$ and $X_{jdef}(t)$, such that $y(t)$ would be given by the following equation:

$$y(t) = \begin{cases} \begin{cases} x_{1inf1} & \text{for } \dot{U}(t) > 0 \\ x_{1def1} & \text{for } \dot{U}(t) \leq 0 \end{cases} & \text{for } 0 \leq U(t) < K_1 \\ \begin{cases} x_{2inf1} & \text{for } \dot{U}(t) > 0 \\ x_{2def1} & \text{for } \dot{U}(t) \leq 0 \end{cases} & \text{for } K_1 \leq U(t) < K_2 \\ \begin{cases} x_{3inf1} & \text{for } \dot{U}(t) > 0 \\ x_{3def1} & \text{for } \dot{U}(t) \leq 0 \end{cases} & \text{for } K_2 \leq U(t) < K_3 \\ \dots & \dots \\ \begin{cases} x_{ninf1} & \text{for } \dot{U}(t) > 0 \\ x_{ndef1} & \text{for } \dot{U}(t) \leq 0 \end{cases} & \text{for } K_{n-1} \leq U(t) \end{cases} \quad (17)$$

However, there then exists a trade-off between accuracy and computational efficiency in using piecewise modelling. A greater number of models across the stroke length will ultimately cause the model to be more accurate as the nonlinearity is better approximated; however, this also results in the model being far more complicated, taking much longer

to generate, and having far more parameters. There is also the issue of reference motions becoming too small to properly approximate the overall motion of the PAM, potentially causing each individual model to be less accurate within its own input range. As such the piecewise models generated in this study will comprise of six input ranges, spaced across the stroke length of the actuator to best model the nonlinearity of motion, as well as different models generated for the inflating and deflating state.

2.4. Experimental Setup

The actuator to be used in this work is a FESTO DMSP-20-400N PAM. As previously discussed, PAMs provide compliant motion, which is desirable in a rehabilitation robot. They are also controlled with a single pressurised air input, meaning the only necessary control value is the air pressure flowing into the PAM. The main output measure of the setup is the actuator's displacement to be used to generate target references for the optimisation function as well as measure the error and calculate accuracy of the models. The load force applied to the PAM will also be measured to test the effects of external loads on the accuracy of the models. To measure each of these, a linear encoder and a load cell will be connected to the actuator.

The load cell used is a FUTEK LCM300 model and has a 300 Lb load limit and sufficient precision for any force changes required by the system. The displacement encoder connected to the bottom of the linear actuator is a FESTO MLO-POT-300-TLF with a resolution of 0.01 mm and stroke length of 300 mm, both of which are sufficient for the system. The proportional pressure regulator used is a FESTO VPPM-6L-L-1-G18-0L6H. The system is held in place with a frame consisting of slotted aluminium extrusion, and the PAM is positioned vertically to ensure motion and forces are not adversely affected by the actuator's intrinsic compliance regarding gravity.

To control the system, a RoboRIO is connected to a desktop PC and coded using National Instruments LabVIEW. The outputs of the load cell and the displacement encoder are input to the RoboRIO to be measured on the display. The load cell is connected to a FUTEK IAA100 strain gauge voltage amplifier before the output is taken by the RoboRIO to allow for easier signal processing.

Figure 2 shows a diagram of the experimental setup alongside the circuitry and computer communication systems used for data acquisition and actuator control. Figure 3 shows the PAM in the experimental setup.

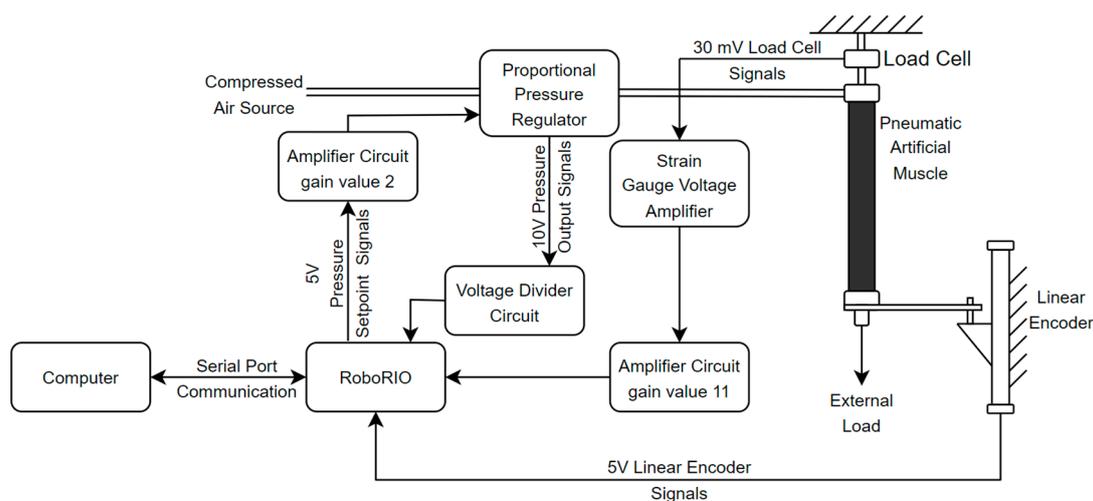


Figure 2. A diagram of the experimental setup, circuitry, and computer communications.

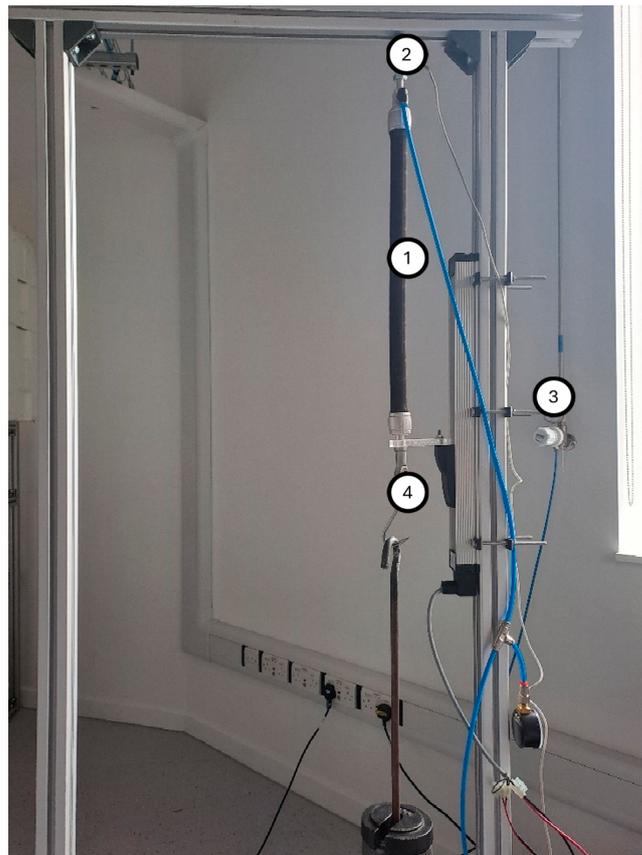


Figure 3. The experimental setup, showing the FESTO DMSF-20-400N PAM (1), FUTEK LCM300 load cell (2), FESTO MLO-POT-300-TLF displacement encoder (3), and external load attachment (4).

3. Results

3.1. Model Generation

To properly compare successes of the piecewise PSO modelling method, several different models were generated. As a benchmark comparison point, a linear phenomenological model was used. To properly justify the benefits of the piecewise modelling method, separate models were also generated using separate parameters for the inflating and deflating states, as well as six distinct ranges within the PAM's stroke length differentiated by input pressure. Four total models were generated for experiments and comparison. Each individual state-space solution, comprised of the three optimised parameters, was generated using 20 iterations of the PSO algorithm, taking approximately five min per solution. The total number of iterations required for each model is shown in Table 1.

The linear model was generated using a single reference trajectory as comparison for the PSO's fitness function. This effectively gives Equation (15) a value of $n = 1$, and no K values are needed as model bounds. This reference was the displacement of the PAM measured using the displacement encoder as shown in Figure 3, over a single sin wavelength with an amplitude of the full stroke length of the actuator—approximately 10 cm total displacement from the resting point—and frequency of 0.25 Hz. The same reference trajectory was used to generate a model differentiating the inflating and deflating states of the PAM, hereafter referred to as the inflate/deflate model, using Equation (16) to describe the model section separation. In order to generate this model, the fitness function of the PSO algorithm was only calculated during periods where the reference trajectory was increasing or decreasing, thus two state-space solutions were generated.

The piecewise models used six separate reference trajectories, giving a value of $n = 6$ for Equation (15), chosen at intervals that would improve the model’s approximation of the PAM’s nonlinear motion across its stroke length. As the motion of the actuator becomes less proportional to input as input pressure increases, the density of these intervals was chosen to increase across the stroke length. The full stroke length of the PAM is achieved using a range of input pressures from 0 to 6.3 Bar. In order to best split the piecewise model sections across the stroke length, a range of pressures were decided for this experiment of [0–1.5], [1.5–3], [3,4], [4,5], [5,6], and [6–6.3] Bar. At steady state, these pressure ranges are achieved by the following input voltages to the pressure regulator, $K_1 = 1.05$, $K_2 = 2.35$, $K_3 = 3.15$, $K_4 = 3.95$, and $K_5 = 4.75$. These values were therefore used as the piecewise model range boundaries in the model generation software.

Two piecewise models were generated using these input pressures as the operating ranges for each separate state-space solution. Each range was generated using a reference trajectory similar to that of the linear model, using a sin wave motion of the PAM, with the motion bounded by the lower and upper limits of each interval. Of the two models, one also used the inflation/deflation differentiation method described previously, resulting in what will hereafter be referred to as a piecewise model and a piecewise inflate/deflate model. The piecewise model has a total of six state-space solutions, one for each input interval, and the piecewise inflate/deflate model has a total of twelve state-space solutions. As previously discussed, this greater number of solutions is expected to result in better accuracy in comparison with the other models.

Table 1 shows the three parameters generated for each separate part of all four models. These parameters are then applied to Equation (10) to create a state-space solution for the input range and inflating/deflating state as specified by the reference used for comparison in the PSO algorithm fitness function. Total iterations used for each model are included to show computational costs.

Table 1. Model parameters as described in Equation (10) and Section 2.2, generated using the PSO optimisation algorithm for each different model. Included are the total PSO algorithm iterations required to generate each of the four models.

Model	Total PSO Iterations	Model Section	P_1	P_2	P_3
Linear	20	Linear	0.451259	0.305468	0.519089
Inflate/Deflate	40	Inflate	0.204797	0.401668	0.821284
		Deflate	0.591312	0.224405	0.421155
Piecewise	120	1	0.0419824	0.717121	0.385334
		2	0.230325	0.428379	0.970208
		3	−0.194232	0.441082	2.05713
		4	0.129964	0.416894	1.11129
		5	0.343061	0.207203	1.00254
		6	−0.0147093	0.151612	1.74169
Piecewise Inflate/Deflate	240	Inflate 1	0.233221	0.522204	0.417219
		Deflate 1	0.04244	0.588327	0.628349
		Inflate 2	0.112168	−0.0255462	2.36332
		Deflate 2	0.0872913	0.681367	0.651491
		Inflate 3	0.0986908	0.638282	0.771687
		Deflate 3	0.632625	−0.0197651	1.0512
		Inflate 4	0.370783	0.399432	0.596057
		Deflate 4	0.616219	0.0665999	0.762527
		Inflate 5	0.480481	0.112959	0.900554
		Deflate 5	0.168957	0.648523	0.38119
		Inflate 6	−0.123881	0.612445	1.03806
		Deflate 6	0.651356	−0.126021	0.961433

3.2. Model Accuracy Experiments

As PAMs are used in several small-scale and industrial applications, multiple experiments were performed to determine each of the four models' accuracy under different use cases. These experiments involved distinct repeated waveforms used as inputs to the pressure regulator controlling the PAM as well as to each of the four models in turn in order to measure the accuracy of the model to the PAM's actual motion. The parameters of each waveform were chosen to best demonstrate different use cases of the PAM. These included a sin wave spanning the full length of the PAM's stroke length, both unloaded and with a 5 kg external load attached as shown in Figure 3. The same full stroke sin wave at a higher frequency was used to show the model's response to rapidly changing motion. Square and triangle waveforms, also spanning the full stroke length of the PAM, were used to test other possible operating procedures. Finally, two sin waveforms spanning the lower and upper ends of the PAM's stroke were used to validate the models' accuracy in smaller motions. The details of waveforms used for each experiment are outlined in Table 2. Each waveform is repeated for four wavelengths to ensure consistency and to mitigate potential unwanted disturbances, except for the rapid sin wave experiment which was repeated for eight wavelengths.

Table 2. Details of waveforms used in the model accuracy experiments. As these waveforms constitute input sequences to the proportional pressure regulator, amplitude, and offset are measured in voltage.

Experiment	Waveform Type	Amplitude (V)	Offset (V)	Frequency (Hz)	Additional Notes
Full Stroke Sin Wave	Sinusoidal	5	0	0.25	The maximum input giving a measurable output for the pressure regulator was 5 V
Lower Stroke Sin Wave	Sinusoidal	3	0	0.25	To determine accuracy at the lower stroke of the PAM
Upper Stroke Sin Wave	Sinusoidal	3	2	0.25	To determine accuracy at the upper stroke of the PAM
Triangle Wave	Sinusoidal	5	0	0.25	To determine model response to constant velocity and rapid changes in velocity
Square Wave	Sinusoidal	5	0	0.25	To determine model response to rapid changes in displacement
Rapid Sin Wave	Square	5	0	1	Waveform repeated 8 times due to shorter wavelength to ensure consistency between experiments
Weighted Sin Wave	Triangle	5	0	0.25	A 5 kg external load is applied to the unaffixed end of the PAM, as shown in Figure 3

The motion tracking results for the full stroke sin wave motion is shown in Figure 4. The displacement error is shown for each of the models in each experiment in Figures 5–11. Specific error measures from each experiment are shown in Table 3.

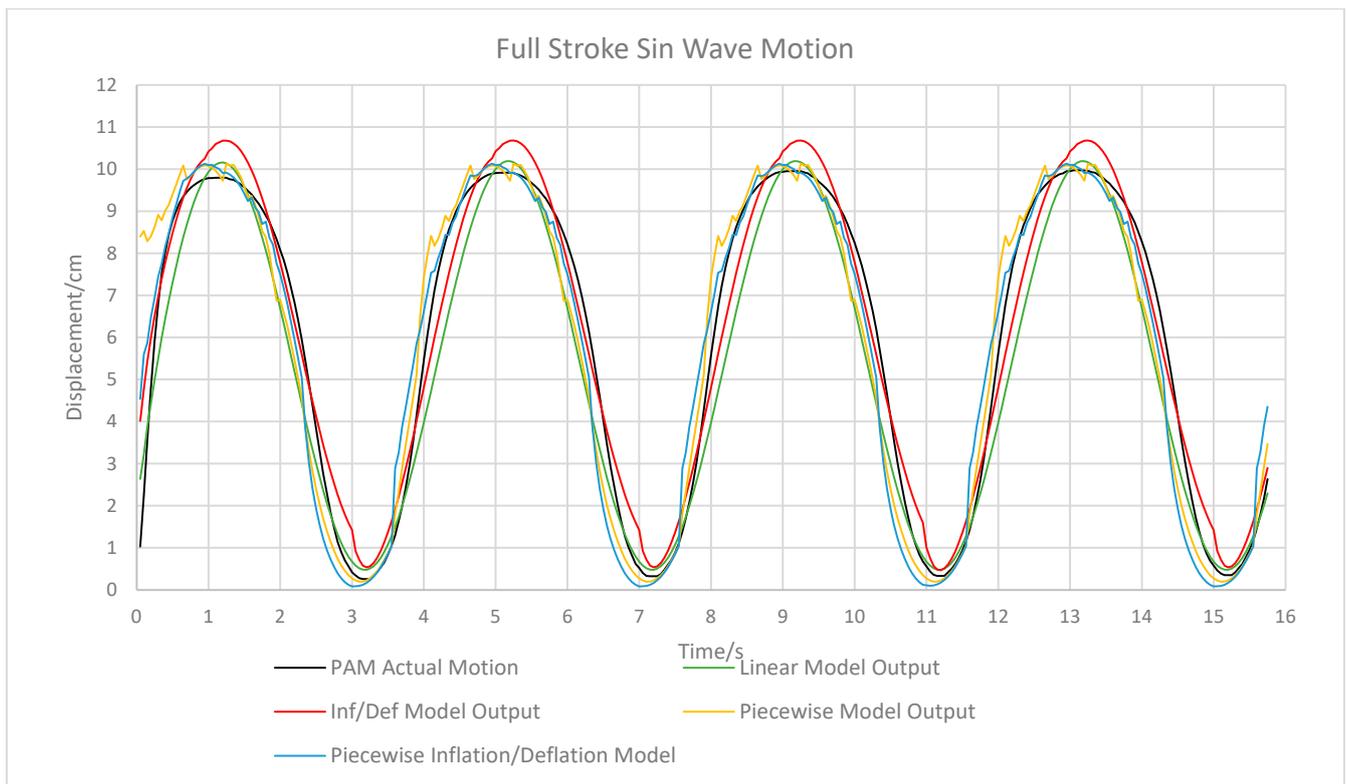


Figure 4. Graph showing the PAM displacement during the full stroke sin wave experiment and estimated displacement for each model based on the same input.

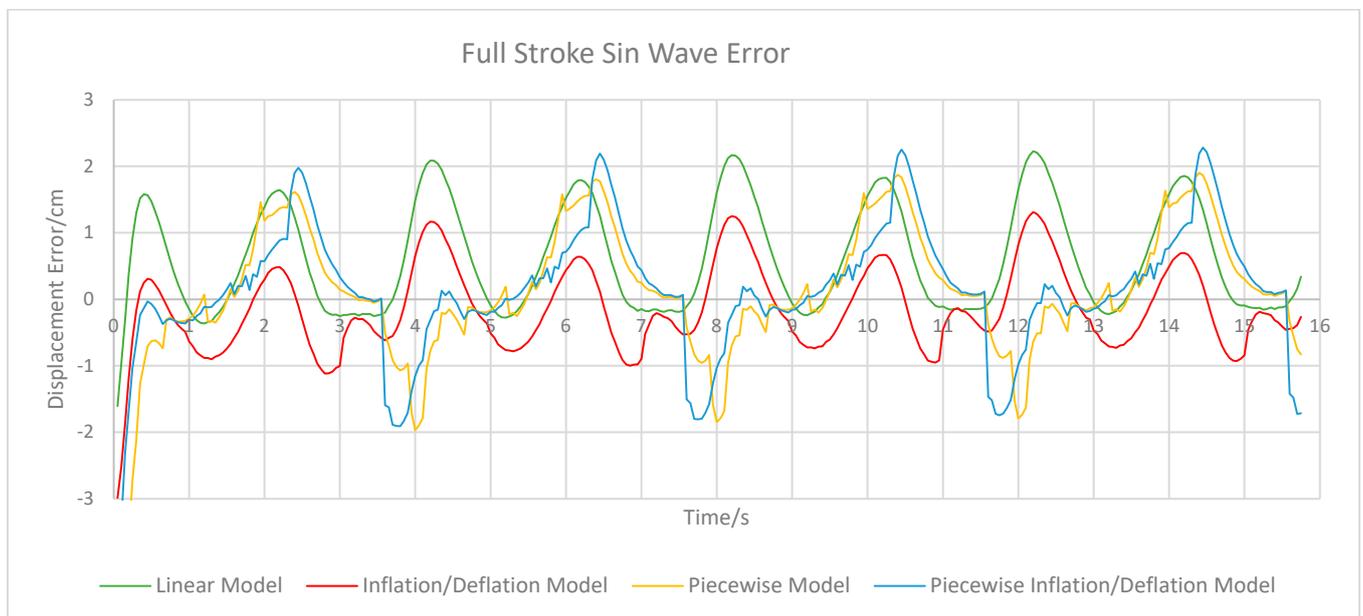


Figure 5. Graph showing displacement error for each model in the Full Stroke Sin Wave accuracy experiment.

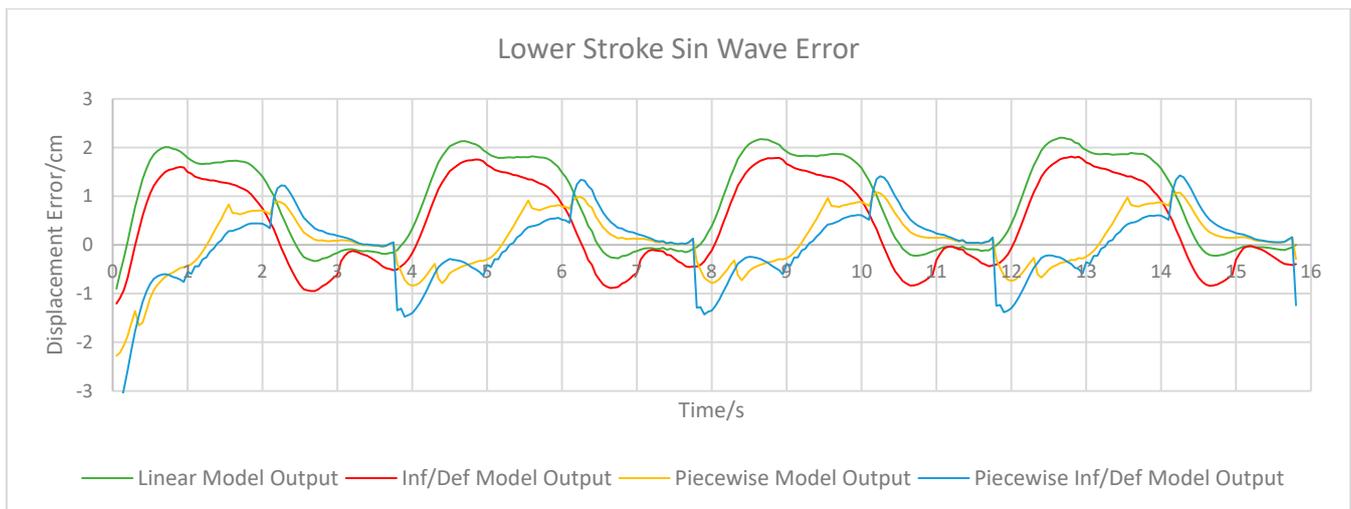


Figure 6. Graph showing the displacement error for each model in the Lower Stroke Sin Wave accuracy experiment.

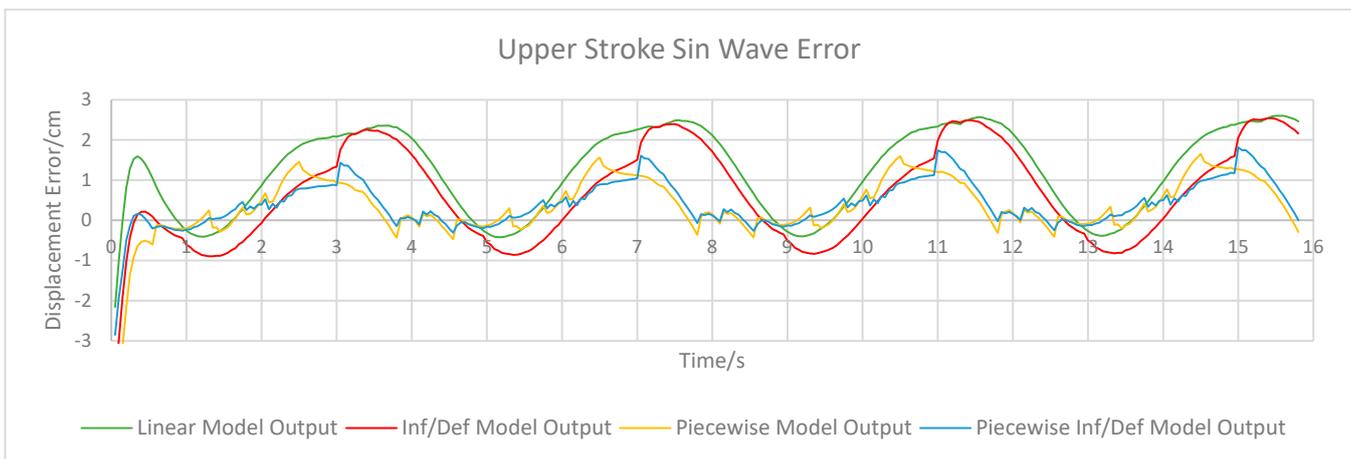


Figure 7. Graph showing the displacement error for each model in the Upper Stroke Sin Wave accuracy experiment.

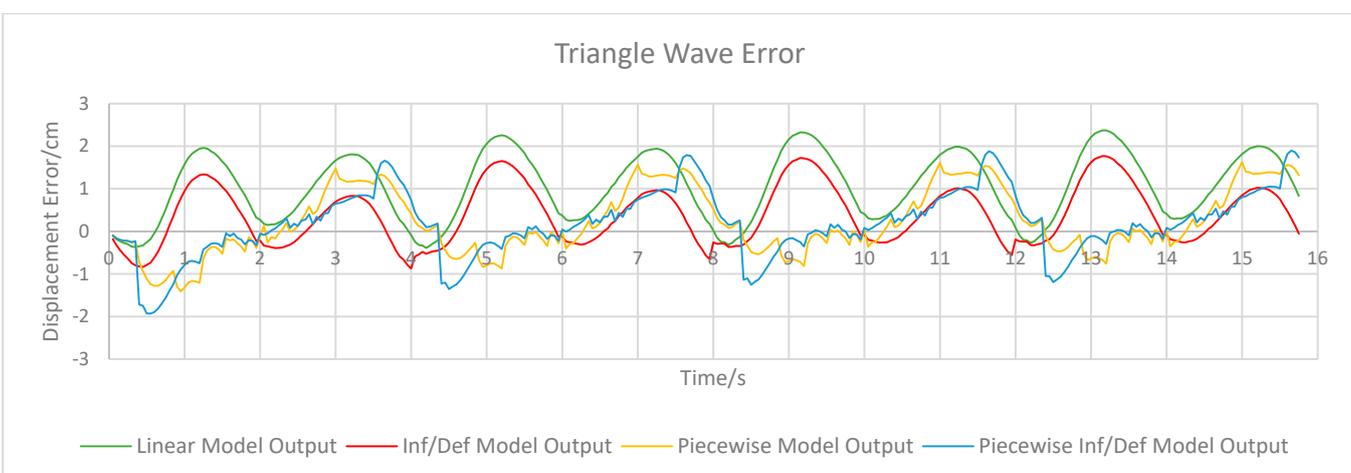


Figure 8. Graph showing the displacement error for each model in the Triangle Wave accuracy experiment.

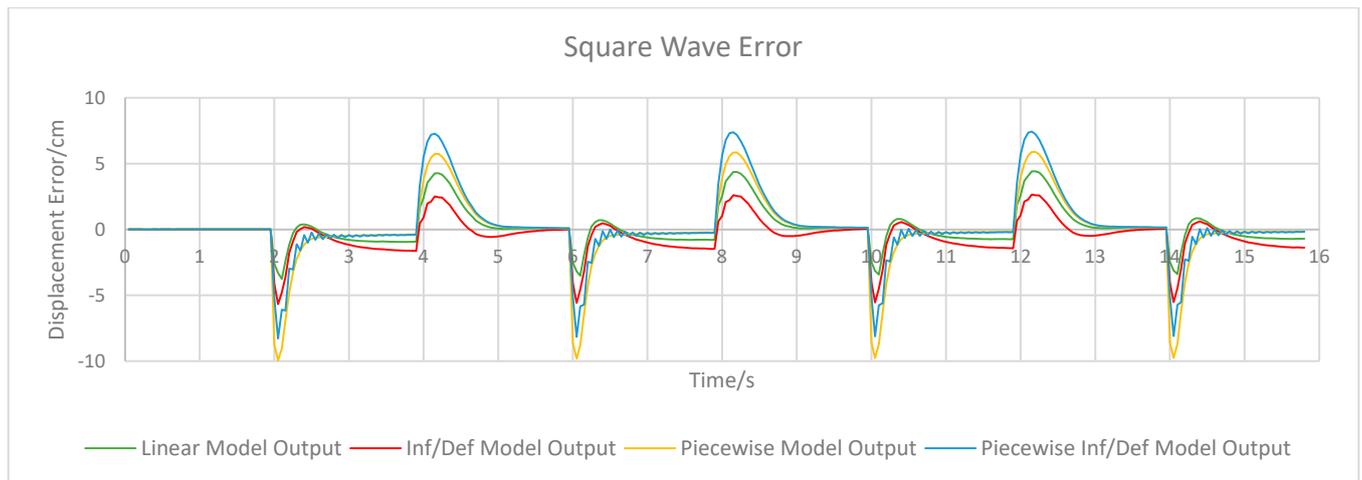


Figure 9. Graph showing the displacement error for each model in the Square Wave accuracy experiment.

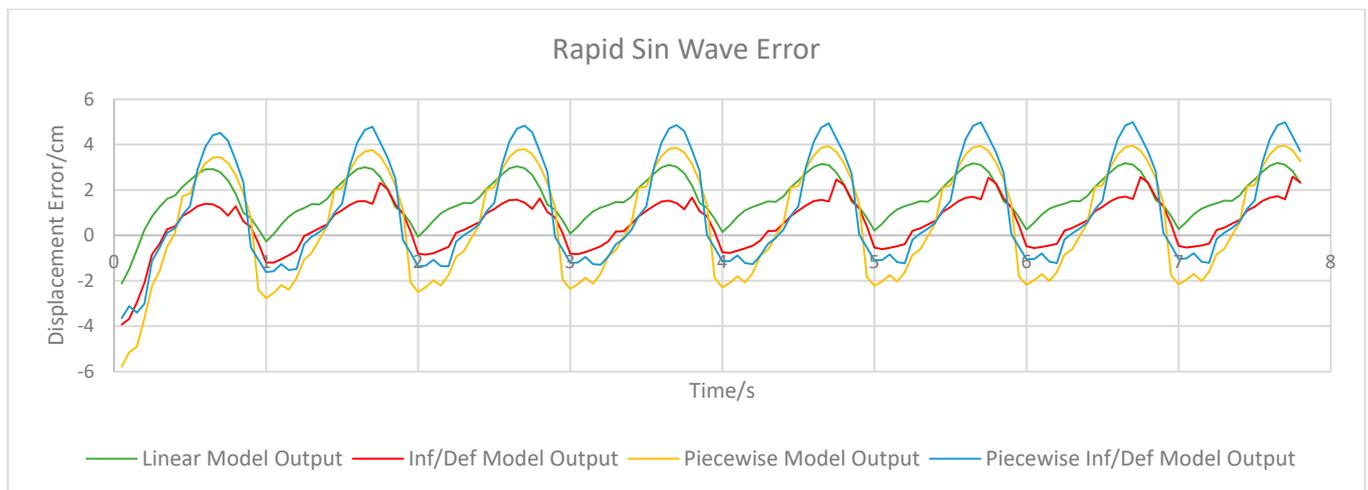


Figure 10. Graph showing the displacement error for each model in the Rapid Sin Wave accuracy experiment.

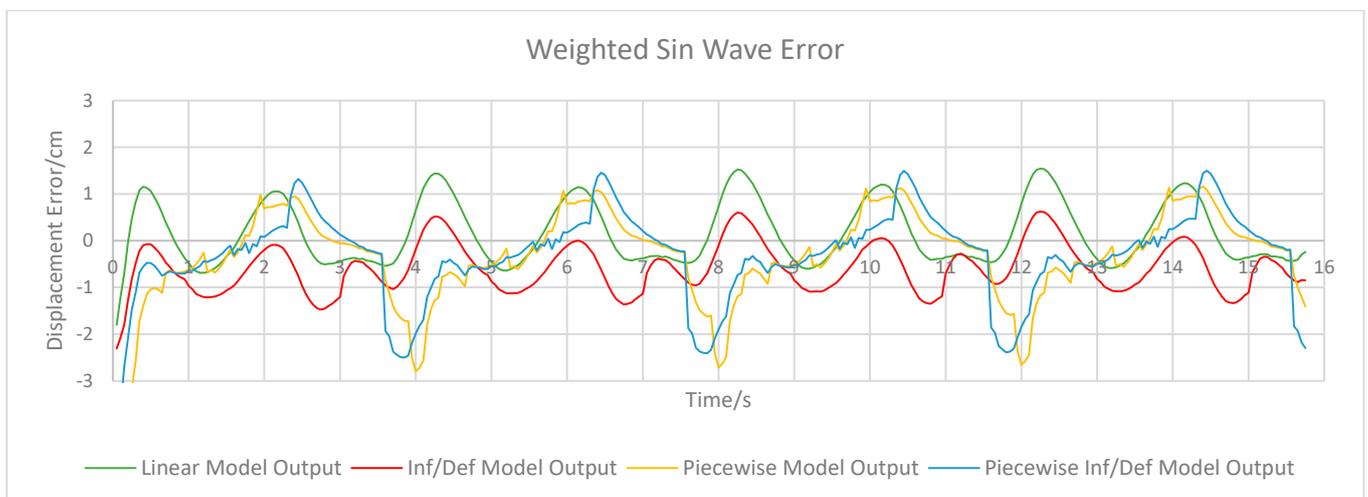


Figure 11. Graph showing the displacement error for each model in the Weighted Sin Wave accuracy experiment.

Table 3. Root mean squared error (RMSE) values for displacement and velocity calculated over a single wavelength for each experiment. Maximum displacement error values exclude the initial transient motion.

Experiment	Error Measure	Linear Model	Inflate/Deflate Model	Piecewise Model	Piecewise Inflate/Deflate Model
Full Stroke Sin Wave Motion	Displacement RMSE	6.13633	4.56575	5.83597	5.65194
	Velocity RMSE	0.923337	0.864623	1.03032	1.01383
	Maximum Displacement Error	2.22815	1.31099	1.96861	2.28073
	Maximum Velocity Error	0.668612	0.981179	1.2318	1.30308
Lower Stroke Sin Wave	Displacement RMSE	7.92829	7.22548	3.99538	4.34249
	Velocity RMSE	0.530426	0.673029	0.497157	0.644981
	Maximum Displacement Error	2.20256	1.8153	1.08325	1.47484
	Maximum Velocity Error	0.401167	0.626442	0.751128	0.703432
Upper Stroke Sin Wave Motion	Displacement RMSE	10.9469	9.13923	4.26431	4.12044
	Velocity RMSE	0.641451	0.71235	0.74694	0.592489
	Maximum Displacement Error	2.6029	2.53934	1.6502	1.81496
	Maximum Velocity Error	0.907793	1.21929	1.55914	1.05378
Triangle Wave Motion	Displacement RMSE	9.16222	5.40654	5.33678	4.92422
	Velocity RMSE	0.906614	0.784657	0.840442	0.854697
	Maximum Displacement Error	2.37337	1.76919	1.62597	1.8976
	Maximum Velocity Error	0.220352	0.231343	0.381985	0.753218
Square Wave Motion	Displacement RMSE	8.83882	8.25559	13.2914	13.5085
	Velocity RMSE	2.0656	2.25246	3.37061	3.24101
	Maximum Displacement Error	4.40405	5.5969	9.74115	8.26315
	Maximum Velocity Error	1.67346	2.81359	4.95761	4.13536
Rapid Sin Wave Motion	Displacement RMSE	7.00625	4.20318	9.08059	7.81292
	Velocity RMSE	1.09764	1.25888	2.54982	2.60158
	Maximum Displacement Error	3.19252	2.58384	3.95251	4.98838
	Maximum Velocity Error	0.994197	1.51908	2.13284	1.89554
Weighted Sin Wave Motion	Displacement RMSE	4.89103	5.96969	6.03493	5.67446
	Velocity RMSE	0.824754	0.804333	1.00429	0.954394
	Maximum Displacement Error	1.54481	1.47573	2.80113	2.50223
	Maximum Velocity Error	0.668612	1.04407	1.62628	1.30308

4. Discussion

Performance criteria for real-time operating models of actuators such as the ones presented here are numerous; thus, a consensus on the best performing model is difficult to reach. These criteria include accuracy to the physical system, robustness to external forces including both load and environmental noise and to different operating modes and input sequences, and computational complexity in generation and during runtime.

Displacement error of the linear PSO and piecewise PSO models are shown in Figures 5–11, while maximum error values for displacement and velocity and RMSE values are shown in Table 3. In general, the absolute error values in each experiment reach a maximum of around 2 cm for all models, except during the square wave experiment in which the maximum error was much higher due to the very rapid change in displacement of the PAM that exacerbates the model's slight lag/lead of the real motion. The rapid motion experiment also showed larger error values, see Figure 10, again due to rapid motion of the PAM but also due to the hysteresis effect in the actuator altering the dynamics during faster motion. The linear and inflate/deflate models have similar error functions with time in most experiments, with the inflate/deflate model having generally lower error values. However, as particularly evident in the upper stroke sin wave, see Figure 7, and rapid sin wave experiments, see Figure 10, the changing between inflating and deflating states can cause the inflate/deflate model to have a sharp spike in error as it switches

between parameters. Both of these models do not properly follow the nonlinearity of the PAM towards the peak of its displacement. They also suffer much larger error values during the lower- and upper stroke tests than in the full stroke test, as the use of a single reference trajectory in the model generation causes the PSO algorithm to overfit to the specific motion used to calculate fitness. The use of piecewise models reduces this effect as both the piecewise and piecewise inflate/deflate models have generally lower error values during the upper- and lower stroke tests. In terms of displacement RMSE, the piecewise model has the lowest error in each experiment as shown in Table 3 except during the rapid motion and square wave tests, where both the RMSE and maximum displacement error is greater than that of the linear and inflate/deflate models. This is slightly mitigated by the piecewise inflate/deflate model. Figure 4 shows the jumps between input ranges in the piecewise model's output; while this model does exhibit lower error values than the piecewise inflate/deflate model, these sharp jumps are undesirable in online operation. Thus, the piecewise inflate/deflate model may be more suited to real time applications.

In comparison with other PAM models in literature, especially those pertaining to the FESTO PAMs, the piecewise PSO modelling method shows similar motion prediction behaviour to other mathematical models. Results from [11] show similar model accuracy in online comparative displacement tests, with models overshooting at higher displacement values of the PAM, similar to that of the linear and inflate/deflate models presented here. Hysteresis is also not properly accounted for. The phenomenological model presented in [14] shows lower RMSE values and closer fitting motion prediction than the piecewise PSO models here; however, the experiments were conducted over much smaller contraction ranges, making accurate comparisons difficult. Models presented in [12,13] are also difficult to compare as experiments focus on force output estimation over pressure input range rather than displacement over time. Neural network-based models, such as those presented in [15–18], have very low error values in motion estimation experiments; however, the results in [17,18] utilise an antagonistic pair of PAMs and measure angle rather than displacement. The algorithm used in [15] for model generation uses a hybrid genetic algorithm and backpropagation neural network. There are several mentions of both algorithms having slow convergence times, and the model used in experiments required 3000 iterations of the backpropagation learning function. In [16], the error values converged to a satisfactory value after 250 iterations of the fuzzy neural network training. Exact generation time is not mentioned; however, the large number of iterations required for neural network-based modelling alongside the computational complexity and required training data make the piecewise PSO modelling method presented here a favourable option in terms of the model generation stage. The PSO-based empirical model presented in [19] shows similar motion tracking results to those presented here, although accuracy is generally better. However, this empirical model is based on a McKibben muscle, and, as mentioned in the paper, the model would not apply to other compliant or soft actuators. Therefore, the simple three-element estimation used here may be more generally useful in modelling various PAMs and other similar soft actuators. Table 4 shows a brief comparison of important points between the proposed piecewise PSO modelling method and those more commonly used modelling techniques.

The error values in the weighted test are generally similar to those in the unweighted full stroke test. This is mainly due to the FESTO PAM's high load tolerance causing the weight to have little effect on the motion. In order to properly determine the effects of larger loads on the model's accuracy, a more sophisticated experimental setup is required. As the piecewise PSO modelling method does not include any real-time measurement in the parameters, the models do not change dynamics based on external factors and do not respond to loads and noise that would impact their robustness in an industrial setting. The inclusion of measures from the attached load cell to the parameter calculation could improve this and have the models respond to loads and environmental factors affecting the PAM's motion.

Table 4. Comparison between the piecewise PSO modelling method and commonly used modelling methods for PAMs. Quantitative data are not provided due to different accuracy measures and experiment types between studies, making qualitative comparison difficult or potentially misleading. Comparisons are kept to those methods applied to FESTO PAMs to ensure similar results where possible.

Modelling Method	Comparison References	Accuracy Comparison	Computational Cost	Model Complexity
Mathematical Model	[10,11]	Similar accuracy, the piecewise model shows smaller overshoots at higher stroke lengths	Similar or slightly higher than the PSO method	Similar or slightly higher than the PSO method
Phenomenological Model	[14]	Better accuracy than the PSO method, different experiment procedures make comparison difficult	More measurements required than the PSO method	Similar model complexity
Neural Network-Based Model	[15–18]	Much higher than the PSO method	Much higher than the PSO method	Much higher than the PSO method
PSO-Base Empirical Model	[19]	Similar motion tracking results, generally better accuracy than the PSO method	More measurements required than the PSO method	More model parameters and templates required than the PSO method

The three-element phenomenological model used as the template of the piecewise PSO modelling method makes it computationally simple, and the use of state-space solutions for each individual model mode makes for fast runtime operation and simple implementation when compared to more involved methods such as deep learning modelling. The use of an optimisation function for parameter estimation also removes the need to measure these parameters manually, and as such this modelling method could be applied to other linear actuators for which the mass–spring–damper approximation is a good estimation of their dynamic properties. Under a different dynamic model template, the proposed PSO modelling method could apply to other types of actuators, but this would require further testing to properly validate. The PSO algorithm has fast optimisation time and requires very little training data when compared to deep learning methods, as only a single trajectory is required to calculate model fitness for each set of parameters. The piecewise inflate/deflate model, being the most complicated with 12 separate state-space solutions, takes approximately one hour to generate using the algorithm parameters mentioned previously. With further tuning of the C1 and C2 parameters, as well as swarm size and number of iterations used, this time can be reduced at little cost to the model’s accuracy.

5. Conclusions

A piecewise PSO modelling method for pneumatic artificial muscles is presented in this paper. A phenomenological model was adopted as a model template, based on the approximation of the PAM as a mass–spring–damper system, with three parameters based on the spring constant, damping constant, and mass of the muscle. Models were generated using reference trajectories from a displacement encoder connected to a FESTO PAM with sin waveform inputs to a pressure regulator forming a constant sin motion, and the RMSE calculated between each particle solution was used as the fitness function for the PSO algorithm. Four different models were generated for testing using this method: a linear model using a single three-element solution, an inflate/deflate model with two solutions differentiating between the PAM’s inflating and deflating states, a piecewise model using six pressure ranges for six different solutions, and a piecewise inflate/deflate model using both the inflation/deflation modes as well as input pressure ranges. Experiments show that these models have good accuracy, with the piecewise PSO modelling method effectively accounting for the nonlinearity present in the PAM. Compared to other PAM models in

literature, results show similar motion behaviour and similar or slightly worse prediction accuracy to mathematical and physical models. When comparing to neural network and genetic algorithm-based models, the error values from the piecewise PSO modelling method are noticeably higher; however, the use of deep learning algorithms greatly increases the time to generate models as well as the computational requirements and complexity. The use of the phenomenological model approximation of the PAM could also allow for more generalised use cases than other model examples, including other PAM types and some other actuators with similar operating principles for which the phenomenological model used here is applicable.

Improvements can be made to the robustness of these models. Measurements of external forces applied to the PAM during operation can be incorporated into the model parameters to allow for dynamic adjustment of the model in response to load and environmental factors. Improvements to the accuracy, as well as generation time, can be made in fine-tuning the PSO algorithm parameters. Accuracy can also be improved using a greater number of pressure ranges for piecewise model generation, as well as more efficient spacing of these ranges within the PAM stroke length. This does come with a trade-off of modelling time and complexity as each additional model requires further optimisation, but the simple phenomenological model template makes the increase in complexity minimal.

Author Contributions: Conceptualisation, D.F.B.; methodology, D.F.B.; software, D.F.B.; validation, D.F.B.; formal analysis, D.F.B.; investigation, D.F.B.; resources, D.F.B.; data curation, D.F.B.; writing—original draft preparation, D.F.B.; writing—review and editing, D.F.B. and S.X.; visualisation, D.F.B.; supervision, S.X.; project administration, S.X.; funding acquisition, D.F.B. and S.X.. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded in part by U.K. Engineering and Physical Sciences Research Council (EPSRC) under Full-Time Doctoral Training Partnership (DPT) Scholarship.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Kalita, B.; Leonessa, A.; Dwivedy, S.K. A Review on the Development of Pneumatic Artificial Muscle Actuators: Force Model and Application. *Actuators* **2022**, *11*, 288. [[CrossRef](#)]
2. Reva, E.J.; Jonathon, W.S. Chapter Two—Actuator Technologies. In *Handbook of Biomechanics*; Academic Press: Cambridge, MA, USA, 2019; pp. 31–59. [[CrossRef](#)]
3. Lizehn, W.; Peng, G.; Yao, W.; Biggar, S.; Hu, C.; Yin, X.; Fan, Y. 10—Soft robotics for hand rehabilitation. In *Intelligent Biomechanics in Neurorehabilitation*; Academic Press: Cambridge, MA, USA, 2020; pp. 167–176. [[CrossRef](#)]
4. Kelasidi, E.; Andrikopoulos, G.; Nikolakopoulos, G.; Manesis, S. A survey on pneumatic muscle actuators modeling. In Proceedings of the 2011 IEEE International Symposium on Industrial Electronics, Gdansk, Poland, 27–30 June 2011; pp. 1263–1269. [[CrossRef](#)]
5. Reynolds, D.B.; Repperger, D.W.; Phillips, C.A.; Bandry, G. Modeling the Dynamic Characteristics of Pneumatic Muscle. *Ann. Biomed. Eng.* **2003**, *31*, 310–317. [[CrossRef](#)] [[PubMed](#)]
6. Shameek, G.; Akash, G.; Akshay, P.; Dwivedy, S.K. Control of pneumatic artificial muscle system through experimental modelling. *Mechatronics* **2012**, *22*, 1135–1147. [[CrossRef](#)]
7. Antonelli, M.G.; Zobel, P.B.; Durante, F.; Raparelli, T. Numerical modelling and experimental validation of a McKibben pneumatic muscle actuator. *J. Intell. Mater. Syst. Struct.* **2017**, *28*, 2737–2748. [[CrossRef](#)]
8. Al-Ibadi, A.; Nefti-Meziani, S.; Davis, S. Valuable experimental model of contraction pneumatic muscle actuator. In Proceedings of the 2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 29 August–1 September 2016; pp. 744–749. [[CrossRef](#)]
9. Al-Ibadi, A.; Nefti-Meziani, S.; Davis, S. Novel models for the extension pneumatic muscle actuator performances. In Proceedings of the 2017 23rd International Conference on Automation and Computing (ICAC), Huddersfield, UK, 7–8 September 2017; pp. 1–6. [[CrossRef](#)]
10. Andrikopoulos, G.; Nikolakopoulos, G.; Manesis, S. Pneumatic artificial muscles: A switching Model Predictive Control approach. *Control Eng. Pract.* **2013**, *21*, 1653–1664. [[CrossRef](#)]

11. Kotkas, L.; Zhurkin, N.; Donskoy, A.; Zharkovskij, A. Design and Mathematical Modeling of a Pneumatic Artificial Muscle-Actuated System for Industrial Manipulators. *Machines* **2022**, *10*, 885. [[CrossRef](#)]
12. Sarosi, J.; Biro, I.; Nemeth, J.; Cveticanin, L. Dynamic modeling of a pneumatic muscle actuator with two-direction motion. *Mech. Mach. Theory* **2015**, *85*, 25–34. [[CrossRef](#)]
13. Cao, J.; Xie, S.Q.; Zhang, M.; Das, R. A New Dynamic Modelling Algorithm for Pneumatic Muscle Actuators. In *Intelligent Robotics and Applications. ICIRA 2014. Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2014; Volume 8918. [[CrossRef](#)]
14. Serres, J.L.; Reynolds, D.B.; Phillips, C.A.; Gerschutz, M.J.; Repperger, D.W. Characterisation of a phenomenological model for commercial pneumatic muscle actuators. *Comput. Methods Biomech. Biomed. Eng.* **2009**, *12*, 423–430. [[CrossRef](#)] [[PubMed](#)]
15. Jamwal, P.K.; Xie, S.Q. Artificial Neural Network based dynamic modelling of indigenous pneumatic muscle actuators. In Proceedings of the 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, Suzhou, China, 8–10 July 2012; pp. 190–195. [[CrossRef](#)]
16. Chavoshian, M.; Taghizadeh, M. Recurrent neuro-fuzzy model of pneumatic artificial muscle position. *J. Mech. Sci. Technol.* **2020**, *34*, 499–508. [[CrossRef](#)]
17. Ahn, K.K.; Anh, H.P.H. A new approach for modelling and identification of the pneumatic artificial muscle manipulator based on recurrent neural networks. *Proc. Inst. Mech. Eng. Part I J. Syst. Control. Eng.* **2007**, *221*, 1101–1121. [[CrossRef](#)]
18. Ahn, K.K.; Anh, H.P.H. Comparative study of modeling and identification of the pneumatic artificial muscle (PAM) manipulator using recurrent neural networks. *J. Mech. Sci. Technol.* **2008**, *22*, 1287–1298. [[CrossRef](#)]
19. Mat Dzahir, M.A.; Yamamoto, S.-i. Dynamic Modeling of McKibben Muscle Using Empirical Model and Particle Swarm Optimization Method. *Appl. Sci.* **2019**, *9*, 2538. [[CrossRef](#)]
20. Dzahir, M.A.M.; Yamamoto, S.-i. Inverse Modeling of Nonlinear Artificial Muscle Using Polynomial Parameterization and Particle Swarm Optimization. *Adv. Mater. Sci. Eng.* **2020**, 8189157. [[CrossRef](#)]
21. Du, W.; Li, B. Multi-strategy ensemble particle swarm optimization for dynamic optimization. *Inf. Sci.* **2008**, *178*, 3096–3109. [[CrossRef](#)]
22. Huang, J.; Qian, J.; Liu, L.; Wang, Y.; Xiong, C.; Ri, S. Echo state network based predictive control with particle swarm optimization for pneumatic muscle actuator. *J. Frankl. Inst.* **2016**, *353*, 2761–2782. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.