

DEPARTMENT OF  
ENGINEERING MATHEMATICS

UNIVERSITY OF BRISTOL

---

---

# Computing with Real Muscles

---

---



**Dan Bouchard**

Supervised by Dr Helmut Hauser

March 29, 2018

## Abstract

The required properties for morphological computation in a physical system are: non-linearity, complexity, compliance, and in some cases, noise. These are all properties in real muscles, and an investigation is proposed to assess whether the increased complexity with muscle models can increase the computational power for particular tasks. This report examines some passive mechanical models of muscles such as: Maxwell, Voigt, and two variations of the Kelvin model, as well as considering the active Hill muscle model. A reservoir or network of connected nodes is constructed using the muscle models presented to create a set of generic networks. The computational power of these networks are assessed by their performance on specific supervised learning tasks to evaluate whether real muscles could provide computational abilities.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Reservoir Computing and Morphological Computation . . . . .	2
1.3	Properties for Morphological Computation and Reservoir Setup . . . . .	3
1.4	Project Aims and Objectives . . . . .	4
1.5	Plan of Report . . . . .	5
<b>2</b>	<b>Muscle Models</b>	<b>6</b>
2.1	Viscoelastic material tests . . . . .	7
2.2	Maxwell Model . . . . .	8
2.3	Voigt Model . . . . .	11
2.4	Kelvin Model . . . . .	12
2.4.1	Model A . . . . .	13
2.4.2	Model B . . . . .	15
2.5	Hill Model . . . . .	17
2.6	Bouc-Wen Hysteresis . . . . .	22
<b>3</b>	<b>Implementation and Learning Tasks</b>	<b>24</b>
3.1	Setting up the recurrent network . . . . .	24
3.2	Specific muscle model implementation . . . . .	26
3.3	Learning Tasks . . . . .	27
<b>4</b>	<b>Network Results</b>	<b>29</b>
4.1	Memory Task . . . . .	29
4.2	NARMA Task . . . . .	33
4.3	Complexity Task . . . . .	34
4.4	Linear Model Results . . . . .	36
<b>5</b>	<b>Conclusion</b>	<b>37</b>
5.1	Further Work . . . . .	37

## 1 Introduction

### 1.1 Motivation

Classical robot design has often been to control rigid body parts using high torque servos with a separate controller undertaking the necessary computation of the whole system. Typically, for precise control, the number of degrees of freedom are kept as low as possible and each degree of freedom is kept fully actuated for all time, often requiring intensive computation and high energy consumption [8, 9, 22]. The robot design and overall control system are generally treated as separate entities with the former suppressing complex nonlinear dynamics through its rigidity in an attempt to simplify the task for the controller. However, these complex dynamics that are present in many systems can potentially be exploited to improve performance.

Rigid robots often exhibit jerky movements and any slight inaccuracies in the end effector are highlighted when interacting with a new or changing environment. In contrast, biological systems can utilise interactions with the environment to stabilise general motion, or particular control tasks. The complex dynamics resulting from interactions with the environment are not suppressed and instead can sometimes be viewed as input signals into the dynamic nonlinear body or system. Hence, any input dynamics generated from the interaction with the environment, which clearly are not generated explicitly from the body itself, become part of the overall nonlinear dynamical system encompassing the body and the environment it is within. So, these complex dynamics within the compliant material don't necessarily need to be explicitly controlled for all time [6], yet provide resourceful information for the overall computational task. Hence, the structure or body can be viewed as a computational resource which can outsource part of the control task from the brain or central controller [8].

Soft robotics is often bio-inspired and makes use of soft materials and soft actuators [6], to allow the system to passively interact with its environment. The body is said to be compliant, which means that it can be stretched and squeezed, whilst also being potentially more energy efficient. The morphology of the robotic structure or network is described together as its shape or form and its dynamic properties, which are usually nonlinear. This can be utilised as a computational resource, hence the term morphological computation. This on-going computation required to stabilise interactions with the environment occurring within the structure is very different from the traditional view of computation which takes in an input and maps it to a single output over a finite time period [6]. It has been shown, that the properties which are necessary for morphological computation in a physical system are: nonlinearity, complexity, compliance, and in some cases, noise [6].

One example of morphological computation is a soft silicone octopus inspired arm, that can generate a bending motion whilst immersed underwater from a single input command [15]. The dynamics of the arm were exploited to emulate particular functions requiring short-term memory. Another common example is the passive walker which makes use of gravity to walk down slope, using the environment to maintain balance and move along without the need of an external controller [14]. However, the parameters of this particular system are hand-tuned

or found by computationally intensive nonlinear optimisation algorithms. Therefore it cannot generalise to new environments or learn how to adapt its structure when faced with something slightly different [9]. This will become an important thing to note during the majority of this report when general morphological structures or networks will be used to learn multiple tasks.

## 1.2 Reservoir Computing and Morphological Computation

Morphological computation can be viewed as coming under the more general machine learning based area called reservoir computing [15], given the properties of nonlinearity and complex dynamics within the network. Reservoir computing uses a generic high dimensional connected network or topology to construct a complex dynamical system as seen in Figure 1. The nodes of the reservoir represent simple but nonlinear differential equations with random but fixed parameters. Random connections are made between nodes that encapsulate the complexity or dynamic coupling within the structure. This complexity allows the structure to nonlinearly map a low-dimensional input stream onto a high-dimensional state space. Hence, the structure can be thought of as a kernel, going from a low-dimensional input to a high-dimensional state space [7].

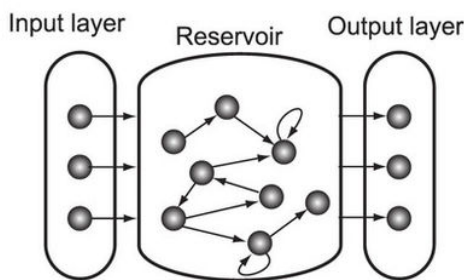


Figure 1: Classical RC setup with a reservoir of nodes representing simple nonlinear differential equations [1]

The reservoir setup can be used to do supervised learning as the low-dimensional input is mapped to a high-dimensional state space. The output is calculated as a weighted sum of the state space variables within the reservoir. These static output weights are the only parameters that are tuned or adapted to learn the given output. This means that different tasks can potentially be learnt using the same random reservoir by only adapting the network weights [7]. Reservoir computing is able to learn temporal problems given that the reservoir is composed of nodes representing differential equations. Thus, is applicable to soft robotics requiring on-going computation as the system interacts with a changing environment.

Aside from morphological computation, there exist different types of reservoir computing which use slightly different artificial reservoirs such as echo state networks or liquid state machines [7]. Echo state networks are randomly generated networks of analog neurons activated by a time input signal. Whereas liquid state machines are recurrent circuits of integrate and fire neurons excited by external input spike trains [11, 12, 19].

To have computational capabilities the reservoir must be a fixed nonlinear kernel [9], which has properties of input separability, fading memory and stable integration [7, 15]. The ability of the structure to separate out signals, is very important in order to gain computational power which would otherwise be lost with linearly dependant signals [7].

### 1.3 Properties for Morphological Computation and Reservoir Setup

In order to do morphological computation there are two properties that must be met. The first, is temporal integration of information meaning the system must show some form of dynamic memory. The output of the system  $y(t)$ , will depend not only on the current input  $u(t)$ , but also a particular set of input timepoints  $u(s)$ , where  $s < t$ . The second property is nonlinearity, this is relevant as nonlinear dynamics are present when using compliant bodies. The nonlinear computation can be executed by the morphology, leaving a linear learning task to produce the desired output [8].

Mass-spring systems can be used to provide dynamic temporal integration with different ways to incorporate the nonlinearity into the system. The first method considered (Figure 2a), uses a linear array of mass spring systems and an artificial neural network which behaves as a nonlinear static mapping to provide the necessary nonlinearity. The second method (Figure 2b), based on reservoir computing, uses a recurrently connected nonlinear mass-spring network, which provides temporal integration and also acts as a nonlinear kernel.

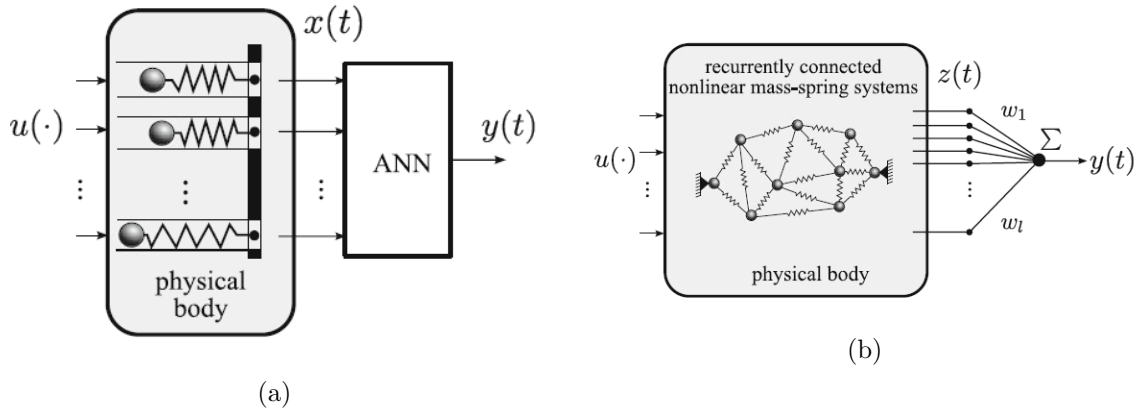


Figure 2: Two theoretical models of possible morphological structures, that have the necessary temporal integration and nonlinear combination required to do the morphological computation [8].

The resulting supervised learning tasks for the two proposed systems require a different amount of computational effort. The first system requires training the feedforward artificial neural network, which can be a computationally hard problem due to the backpropagation required [8]. The second system requires training the static weights  $w_i$  to learn the final output  $y(t)$ , as a weighted sum of the higher dimensional network state space  $\mathbf{z}(t)$ , as in Equation 1.

$$y(t) = \sum_i w_i z_i(t) \quad (1)$$

The network state space  $\mathbf{z}(t)$  could be represented as connection lengths or node positions for a given input  $u(t)$ . As discussed in Section 1.2, the weights  $w_i$  are the only parameters adjusted in this learning process and can be found efficiently by linear regression.

Certain limitations exist with this system. Firstly, that the network will only allow particular computations that are in the frequency range inherent to the physical body or network. Also, the mass-spring system acts as a nonlinear low-pass filter, which naturally suppresses very high frequency inputs as the signal propagates through the network. As such, it makes sense to only outsource computations that are in the frequency range inherent to the physical body as analog computation.

#### 1.4 Project Aims and Objectives

The aim of this project is to use and adapt current mass-spring recurrent networks that have been documented in [8] to incorporate muscle models within the network. The reason why this may be possible and interesting, is that muscles show some amount of nonlinearity, complexity and compliance which have previously been shown to be properties required for morphological computation. These models are more complex than the mass-spring systems currently considered. The objective is to investigate whether using a recurrent network of muscular models can increase the computational power or performance when learning multiple supervised learning tasks.

Models to be considered include passive spring-damper models that reproduce the passive mechanical behaviour of muscles without any active forcing or input. These models produce muscle tissue response under both compressive and tensile loads [17]. The Hill's model introduces an active contractual element within the existing passive model framework. The contractual element determines the amount of tension in the muscle from a stimulus or input to the muscle. Nowadays, the Hill-type muscle model is the most used in bio-mechanical studies involving muscular coordination [17].

Muscles also show a property called hysteresis where the tension is different when the muscle is loading and unloading. Hysteresis is a generic property that is not confined to muscles and describes how the muscular restoring force lags behind the change in muscle length that is causing it. This produces hysteresis loops, where the restoring force is dependant on whether the muscle is extending or contracting. Hysteresis is present due to energy dissipation, as the energy transferred during the unloading phase is not equal to the energy stored during the loading phase [16]. The Bouc-Wen hysteresis model can be introduced to all the mechanical models of muscles to be considered [10].

The aim is to consider different learning tasks and discover if muscle model networks can produce increased performance to emulate functions that require memory and complexity. These two important properties are required to do morphological computation. Any improved performance from using muscle models will provide a fundamental building block for using real muscle tissue in soft robotics.

### 1.5 Plan of Report

To achieve the above aim and objectives we will first discuss and explore the muscle models which are to be implemented into the recurrent network in Section 2. The current setup for morphological computation adapted from [5], and what specific learning tasks will be considered, will be discussed in Section 3. The exact implementation of the muscle models will also be highlighted, so that performance comparisons can be made across the different models. The network results and performance will be discussed in Section 4 and will conclude with plans for further work and how the system could be improved.



## 2 Muscle Models

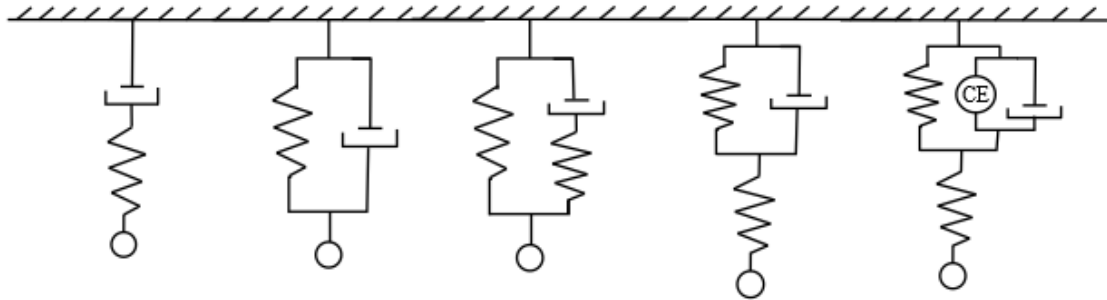


Figure 3: Schematic diagrams of the mechanical muscle models to be considered which from left to right are: Maxwell Model, Voigt Model, Kelvin Model (A), Kelvin Model (B) and Hill Model. The Hill Model contains a contractual element, CE which is the active element of the muscle model.

A characteristic of muscle tissue is that when given an active contraction, it produces a tension force due to the shortening of the muscle. The Hill-type muscle models are lumped parameter models where an arrangement of spring and damping elements can re-produce the mechanical behaviour of the tissue [17]. Other type of muscle models include the Huxley-based models which builds upon the Hill model with the same three element setup. However, the active element is represented as being produced due to the interaction of the actin and myosin filaments within the muscle fibre. These interactions, called cross-bridges, produce a force and the total contractive force of one muscle fibre is the sum of forces produced by all cross-bridges at one moment in time [4]. The number of cross-bridges within a muscle fibre is related to the fibre length, the active state of the fibre and the velocity of the contractual element. The activation of the contractual element is measured at the chemical level as the amount of  $Ca^{2+}$  ions that bind to troponin, or the concentration of free  $Ca^{2+}$  ions compared to resting fibre concentration [4]. This model correctly predicts the force-length and force-velocity relationships for the contractual element (from experimental data), but is very computationally expensive. Comparatively, the Hill model of the contractual element is described as a ‘black box’ model. It simply reproduces these known relationships without the mechanism of how they are produced [2].

All the models used within this report are represented in Figure 3 with the Hill muscle model on the right being the most widely used in bio-mechanical studies involving muscular coordination. The Huxley model is not considered as it is too computationally expensive and does not offer much advantage over the Hill model in this investigation. All the models are formed as a combination of basic spring and viscous dashpot elements to model the viscoelastic behaviour of the soft tissues [13].

The first four models considered are deemed to be passive because they contain no active contractual element, and in order to simulate their behaviour, an external load is required. These models are designed to model the physiological response of a passive muscle with no active

component [17]. Mechanical characterisation often involves uniaxial tensile tests which are usually confined to the creep test, stress relaxation (loading and unloading) and dynamic sinusoid loading given by  $f(t) = B \sin(pt)$  [18]. Stress and strain relationships are analogous to force and extensions which can be easily calculated. When implementing the models into the recurrent network, force and extension will be used over stress and strain.

For the springs and viscous dashpot elements in the models both the linear and polynomial possibilities are considered. However, it is noted for morphological computation a certain amount of nonlinearity is a required property. One advantage of the linear models is that the force can usually be expressed as a single linear differential equation reducing the amount of computation required to run the simulation. In other papers, exponential springs are used to represent the nonlinear passive mechanical behaviour of muscles, however this is not considered [4].

## 2.1 Viscoelastic material tests

The creep and stress relaxation tests are used to study the material response over prolonged durations. For a more instantaneous response, sinusoidal stress (or strain) can be applied to find the short-term response in strain (or stress). The strain is defined as

$$\varepsilon = \frac{x}{l},$$

where  $x$  is the extension from natural length  $l$ . The stress is defined as,

$$\sigma = \frac{F}{A},$$

where  $F$  is the force within the material and  $A$  is the cross-sectional area.

The creep test measures the time dependant strain,  $\varepsilon(t)$  from a steady uniaxial stress,  $\sigma_0$  or constant force,  $F$ . The stress relaxation test measures the time-dependent stress  $\sigma(t)$  from a constant strain  $\varepsilon_0$ . Dynamic sinusoidal stress (or strain) gives the instantaneous response of the system which approaches a sinusoidal steady state with the same frequency for stress and strain. However, the strain will always lag behind the stress in any viscoelastic material [18].

In order to evaluate the force,  $F(x, \dot{x})$  from the extension  $x$  and the rate of change of extension  $\dot{x}$  there are some basic principles that can be recalled. Firstly, it is assumed that the mechanical elements are of zero mass and the only mass in the system is the point mass connected at the end of the muscle. This means that the contact points where elements join together serially have no resultant force as there is no mass,  $F = ma$ . Additionally, the two serial extensions  $x_1$  and  $x_2$  sum together to give the overall extension  $x$ , Equation 5. Elements joined together in parallel are the opposite as the forces sum together but the extensions are equal. This is also analogous for stress and strain whereby for parallel elements, stresses sum together but strains are equal and the opposite is true for serial elements. A detailed investigation will now be presented into how each of the models behave.

## 2.2 Maxwell Model

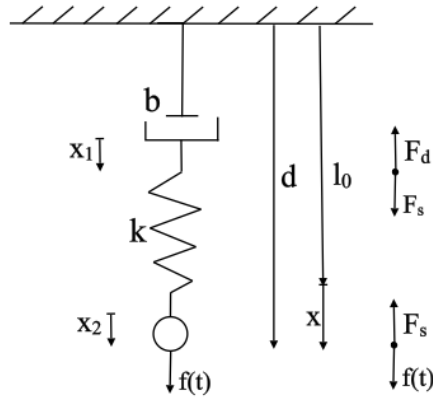


Figure 4: Maxwell Model of a muscle

The Maxwell model is a very simple model where the the dashpot and spring are attached serially as shown in Figure 4. The force from the damper is equal and opposite to the force from the spring  $F_d = F_s$ , and the total tension force produced from the muscle,

$$F = F_s = F_d.$$

In the linear model,

$$F_s = kx_2 = F, \quad (2)$$

where  $x_2$  is the extension of the spring and

$$F_d = b\dot{d}_1 = b\dot{x}_1 = F \quad (3)$$

where  $d_1$  and  $x_1$  are the length and extension of the damper respectively. Rearranging (2) for  $x_2$  and differentiating you get

$$\dot{x}_2 = \frac{\dot{F}}{k}. \quad (4)$$

Given that,

$$x = d - l_0 = (d_1 + d_2) - (l_1 + l_2) = (d_1 - l_1) + (d_2 - l_2) = x_1 + x_2, \quad (5)$$

rearrangement and differentiation gives

$$\dot{x}_1 = \dot{x} - \dot{x}_2. \quad (6)$$

Substituting (4) into (6) and back into (3) gives

$$F = b\left(\dot{x} - \frac{\dot{F}}{k}\right),$$

which can be rearranged for

$$\dot{F} = k\dot{x} - \frac{k}{b}F. \quad (7)$$

This equation can be used to calculate the change in force in the muscle purely in terms of the rate of change of extension and the current force without having to know  $x_1$  or  $x_2$ . When the external forcing is applied, and by assuming unit mass ( $m = 1$ ),  $\ddot{x} = f(t) - F$ . So, the system can be integrated numerically using Euler integration

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ F \end{bmatrix} = \begin{bmatrix} \dot{x} \\ f(t) - F \\ k\dot{x} - \frac{k}{b}F \end{bmatrix}, \quad (8)$$

given an input forcing  $f(t)$  or stress  $\sigma_0$ .

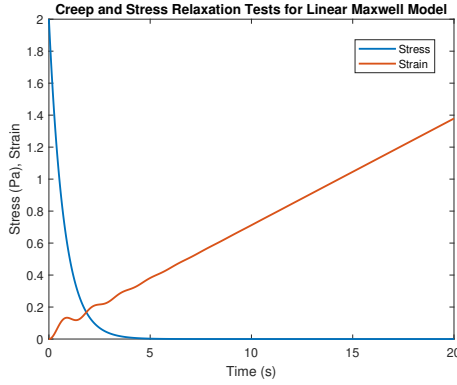


Figure 5: Creep test and stress relaxation test for the linear Maxwell model with  $k = 20$  and  $b = 15$ .

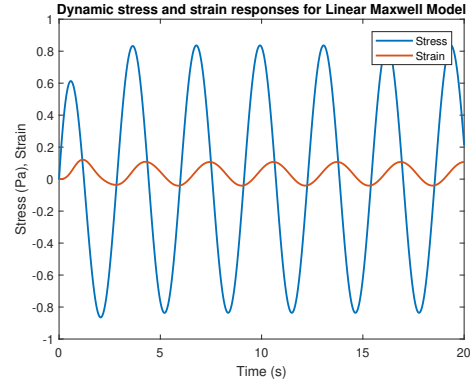


Figure 6: Dynamic creep test and stress relaxation test for the linear Maxwell model with  $k = 20$  and  $b = 15$ .

The creep and stress relaxation tests, which were mentioned in Section 2.1, by measuring the time dependant stress (or strain),  $\sigma(t)$  ( $\varepsilon(t)$ ) from a constant strain (or stress)  $\varepsilon_0$  ( $\sigma_0$ ), are summarised in Figure 5 for the linear Maxwell Model. The stress decays exponentially at a rate  $k/b$ , as seen from Equation 7. The force from the spring tends to zero as its extension  $x_2$  decreases exponentially whilst the extension in the damper  $x_1$  increases to keep the total extension or strain  $\varepsilon_0$  constant. The initial force from the spring acts on the damper to produce an extension  $x_1$  and within 5 seconds, all the energy stored in the spring has been dissipated by the viscous damper.

The creep test with a constant load shows that the extension will continue to increase at a constant rate once the force from the spring is zero, at  $t \approx 4$ . The damping force is not sufficient to prevent the extension  $x$  from increasing because it will only act when  $\dot{x}_1 = \dot{x} > 0$ . Hence, there can never be an equilibrium. The dynamic test in Figure 6 shows that the steady state for stress and strain is reached fairly quickly for the instantaneous response, which is advantageous when the model is implemented in the recurrent network to run a stable simulation.

For the nonlinear model there is no direct expression for the force,  $F$  in terms of the extension  $x$ .

Instead, given the expression in Equation 9,  $\dot{x}_2$  can be solved for using root finding algorithms using Equation 10 given that  $x_2$  and  $\dot{x}$  are found from Euler integrations.

$$F = F_d = F_s = b_3\dot{x}_1^3 + b_1\dot{x}_1 = k_3x_2^3 + k_1x_2 \quad (9)$$

$$b_3(\dot{x} - \dot{x}_2)^3 + b_1(\dot{x} - \dot{x}_2) - k_3x_2^3 - k_1x_2 = 0 \quad (10)$$

Note, that the creep and stress relaxation tests for the nonlinear system are identical to Figures 5-6. The model behaves in a similar way in both linear and nonlinear cases so it will be interesting if this has an affect on the morphological computation performance as nonlinearity is a required property.

Figures 7-8 show how the nonlinear Maxwell model behaves to both sinusoidal and step input forcing. Interestingly, both the linear and nonlinear models are indistinguishable which is understandable given the previous tests undertaken. It must be noted here that real muscles only undergo compression, and when the muscle length is less than the slack length, there is zero force applied but for simplicity this is not currently implemented. Figure 7 shows a clear hysteresis property as the tension force is dependant on whether the model is extending or contracting due to whether the input force is exerting a load on the system or not. The reason this occurs is that these are not purely elastic models that retain all their potential energy when deformed, but viscoelastic models that dissipate energy through the viscous damper. This means there is less kinetic energy returned to the system and so the resulting extension  $x$  of the mass follows a different path when unloading to when it was previously loading the input force.

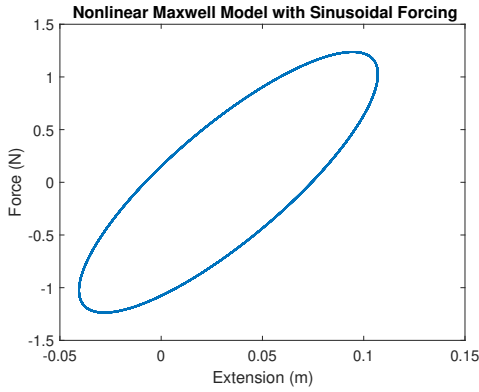


Figure 7: Force-extension relationship for the nonlinear Maxwell model with  $k_1 = 20$ ,  $k_3 = 10$ ,  $b_1 = 15$  and  $b_3 = 5$  and sinusoidal forcing.

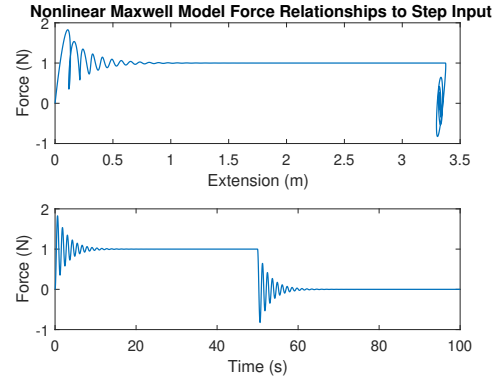


Figure 8: Force-extension relationship and the force over time for the nonlinear Maxwell model with  $k_1 = 20$ ,  $k_3 = 10$ ,  $b_1 = 15$  and  $b_3 = 5$  with step force input,  $f(t)$  at time  $t = 0$  and no input force at  $t = 50$  to show loading and unloading behaviour.

When a step function is used as the input force in Figure 8, the Maxwell model is instantly elastic and responds instantaneously. This scenario also exhibits hysteresis as when the force is taken away the extension does not return to zero along the same path. There is no restoring force from the damper to restore the extension.

### 2.3 Voigt Model

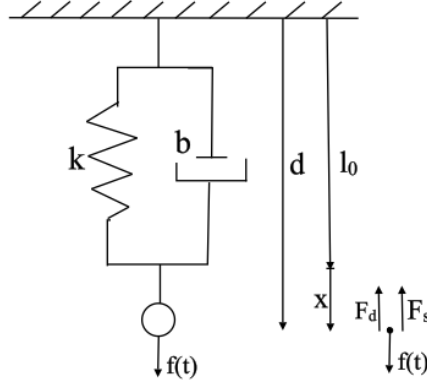


Figure 9: Voigt Model of a muscle

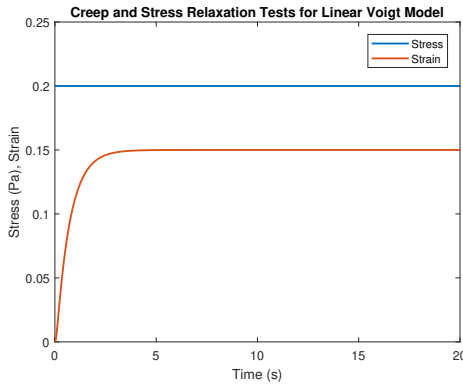
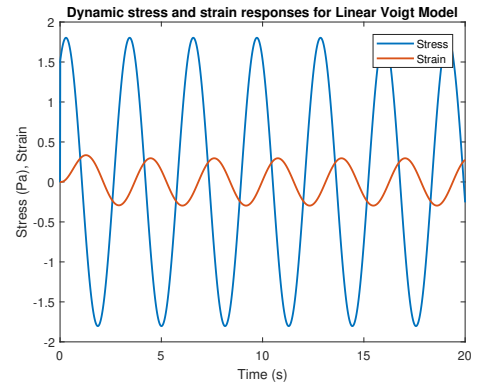
This very simple model can be seen in Figure 9 which is a spring and viscous damper in parallel thus the extensions are the same and the forces sum together. This is the opposite to the Maxwell model where the extensions sum together and the forces are the same at the contact point of the two elements. The linear model is given in Equation 11 and the nonlinear model is given in Equation 12. The nonlinear Voigt model has been used in the past when using mass-spring damper networks to do morphological computation so as such is the baseline to improve upon.

$$F = F_s + F_d = kx + b\dot{x} \quad (11)$$

$$F = F_s + F_d = k_3x^3 + k_1x + b_3\dot{x}^3 + b_1\dot{x} \quad (12)$$

It can simply be implemented using Equation 13 using numerical integration.

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ f(t) - F \end{bmatrix} \quad (13)$$

Figure 10: Creep test and stress relaxation test for the linear Voigt model with  $k = 20$  and  $b = 15$ .Figure 11: Dynamic creep test and stress relaxation test for the linear Voigt model with  $k = 20$  and  $b = 15$ .

As seen with the viscoelastic tests on the Maxwell model, the Voigt model also performs in an identical manner for both the linear and nonlinear systems. From the stress relaxation test in Figure 10, given a constant strain the system remains at constant stress. This is because there is a spring in parallel to the rest of the system, the rest of the system being one damper. This means that there will always be a tension force at any positive extension. As a result, given a constant force with the creep test, the strain tends to a stationary state due to the resisting force from the spring always being present. This model is likely to be more stable than the Maxwell model as the maximum strain is limited for a given input stress.

The dynamic response to a sinusoidal stress and strain can be seen in Figure 11 and again this is representative for both the linear and nonlinear systems. This model seems to have greater stability than the Maxwell model in its instantaneous response as it reaches the steady state almost within the first second of the simulation. As long as there is enough nonlinearity in this model, it should perform well given the response of the system is fairly instantaneous.

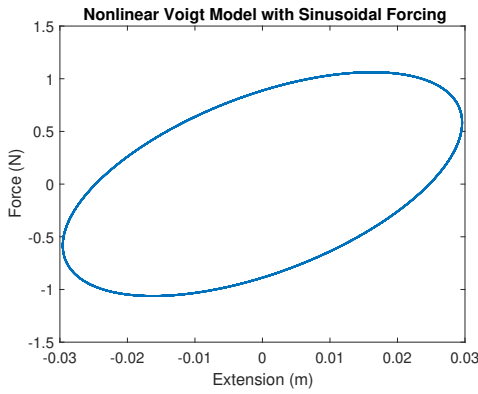


Figure 12: Force-extension relationship for the nonlinear Voigt model with  $k_1 = 20$ ,  $k_3 = 10$ ,  $b_1 = 15$  and  $b_3 = 5$  and sinusoidal forcing.

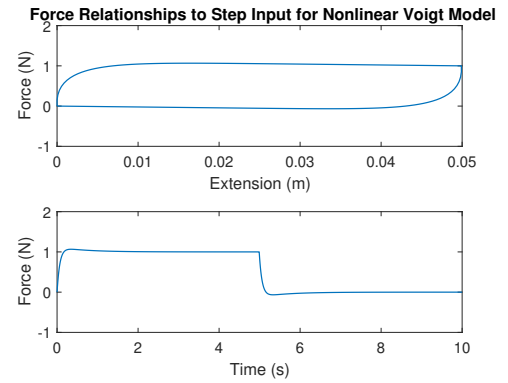


Figure 13: Force-extension relationship and the force over time for the nonlinear Voigt model with  $k_1 = 20$ ,  $k_3 = 10$ ,  $b_1 = 15$  and  $b_3 = 5$  with step force input,  $f(t)$  at time  $t = 0$  and no input force at  $t = 5$  to show loading and unloading behaviour.

As seen in Figure 13, the Voigt model responds instantaneously to a step input and undergoes symmetric hysteresis behaviour when the load is removed. The area between the force-extension curves represents the energy that is dissipated from the system by the viscous damper. The maximum extension for the sinusoidal forcing in Figure 12 is far lower than that for the Maxwell model, but this is to be expected as the strain is limited for a given stress value.

## 2.4 Kelvin Model

The Kelvin Model is a three element model that shows the passive mechanical response for a Hill-type muscle model with no active or contractual element. The two spring coefficients are represented as  $k_p$  and  $k_s$  to represent the parallel and series elastic elements. The viscous damping is a series damping element.

Two arrangements of the Kelvin model are proposed as shown in Figures 14-15. The advantage

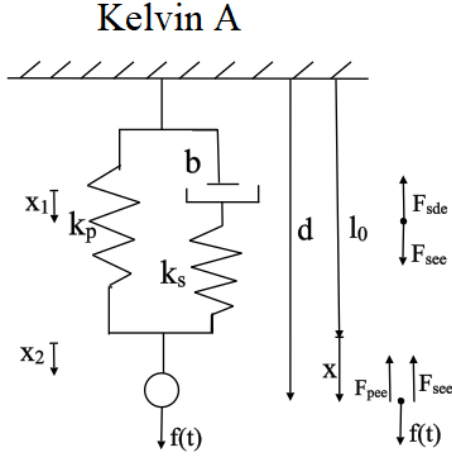


Figure 14: Kelvin Model A of a Muscle

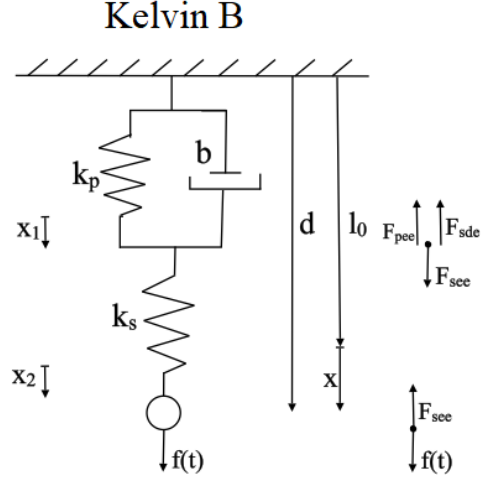


Figure 15: Kelvin Model B of a Muscle

of Model B is that with the linear model, the force can be expressed as a linear differential equation in terms of extension  $x$  and  $\dot{x}$  similar to Equation 7 for the Maxwell model. This simplifies the computational task in order to simulate the model.

#### 2.4.1 Model A

For model A, the force from the series damping element equals the force from the series elastic element,

$$F_{sde} = F_{see}. \quad (14)$$

The total force is given as

$$F = F_{pee} + F_{see}, \quad (15)$$

with the addition of the parallel elastic element. For the linear model from (14) and using (6)

$$b\dot{x}_1 = b(\dot{x} - \dot{x}_2) = k_s x_2,$$

which can be rearranged to

$$\dot{x}_2 = \dot{x} - \frac{k_s}{b}x$$

The tensile force in the muscle is found using (15) and is given as

$$F = k_p x + k_s x_2. \quad (16)$$

The whole system can be integrated numerically using

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ x_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ f(t) - F \\ \dot{x} - \frac{k_s}{b}x \end{bmatrix}, \quad (17)$$



where the value of  $F$  is taken from (16) and  $f(t)$  is the input force. The nonlinear model is very similar except  $\dot{x}_2$  has to be solved for using a root finding algorithm using Equation 18 and the resulting force is

$$F = k_{p3}x^3 + k_{p1}x + k_{s3}x_2^3 + k_{s1}x_2.$$

$$b_3(\dot{x} - \dot{x}_2)^3 + b_1(\dot{x} - \dot{x}_2) - k_{s3}x_2^3 - k_{s1}x_2 = 0 \quad (18)$$

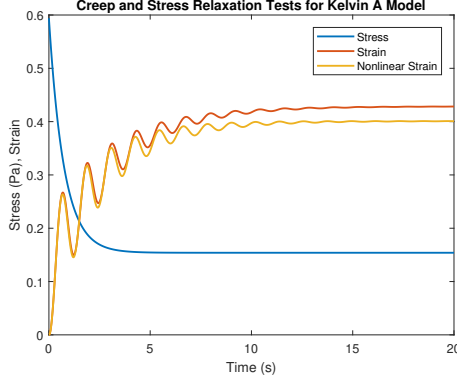


Figure 16: Creep test and stress relaxation test for the Kelvin A model with  $k_{s1} = 20$ ,  $k_{s3} = 10$ ,  $b_1 = 15$ ,  $b_3 = 5$ ,  $k_{p1} = 7$ ,  $k_{p3} = 3$ .

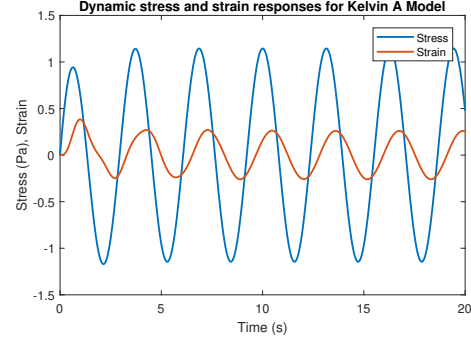


Figure 17: Dynamic creep test and stress relaxation test for the Kelvin A model with  $k_{s1} = 20$ ,  $k_{s3} = 10$ ,  $b_1 = 15$ ,  $b_3 = 5$ ,  $k_{p1} = 7$ ,  $k_{p3} = 3$ .

The viscoelastic tests for the Kelvin A Model can be seen in Figures 16-17 and this time there is a difference between the linear and nonlinear models. For the creep test, where the stress or force is kept constant, the final value for the nonlinear strain is lower than that of the linear system. This is due to the addition of the parallel spring which in the equilibrium state does not have a non-zero force. The nonlinear system introduces an additional cubic term which produces a stiffer spring. As a result, the final extension is lower to produce the same amount of force.

The behaviour of the model in terms of its creep stress relaxation dynamics could be seen as a composite of the previous two models. The creep strain reaches a steady value like the Voigt model, however produces some oscillator behaviour due to the dashpot being attached serially with the spring. The stress relaxation also reaches a steady state rather than decaying to zero from a high initial stress. The dynamic tests with a sinusoidal input seems to behave the same as the previous two models with the instantaneous response appearing to be very similar.

It must be noted that the linear and nonlinear results for Figures 18-19 are almost identical. Also, for the step input, the spring constants for the parallel elastic elements need to be very small otherwise the whole system is too stiff, the system will not produce much extension and will act as a single spring with no damping. Figure 19 shows an interesting hysteresis property as the amount of force oscillates as extension increases and decreases due to the serially connected damper in the system.

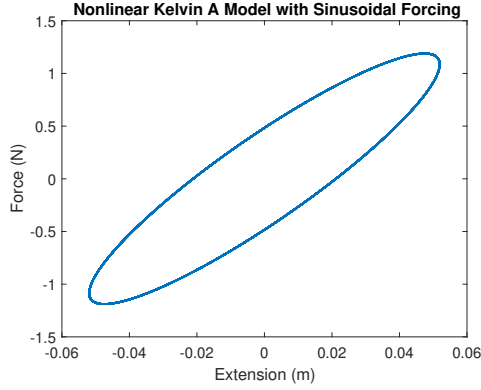


Figure 18: Force-extension relationship for the nonlinear Kelvin A model with  $k_{s1} = 20$ ,  $k_{s3} = 10$ ,  $b_1 = 15$ ,  $b_3 = 5$ ,  $k_{p1} = 7$  and  $k_{p3} = 3$  and sinusoidal forcing.

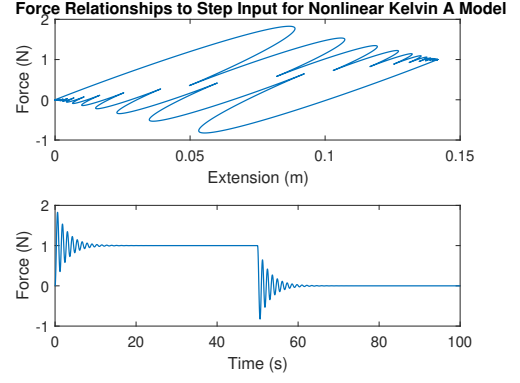


Figure 19: Force-extension relationship and the force over time for the nonlinear Kelvin A model with  $k_{s1} = 20$ ,  $k_{s3} = 10$ ,  $b_1 = 15$ ,  $b_3 = 5$ ,  $k_{p1} = 7$  and  $k_{p3} = 3$  with step force input,  $f(t)$  at time  $t = 0$  and no input force at  $t = 50$  to show loading and unloading behaviour.

#### 2.4.2 Model B

For Model B, the force can be expressed as a linear differential equation when using the linear model. Using Equation 19 which comes from the contact point where the three elements join, there is no resultant force because there is no mass and the tension force  $F$  equals the force in the series elastic element.

$$F = F_{sde} + F_{pee} = F_{see} \quad (19)$$

Using the force definitions,

$$F = k_s x_2 = k_p x_1 + b \dot{x}_1, \quad (20)$$

and substituting  $x_1$  and  $\dot{x}_1$  from (5) and (6) to get

$$F = k_p(x - x_2) + b(\dot{x} - \dot{x}_2). \quad (21)$$

But from (20), we already know  $x_2 = F/k_s$  and  $\dot{x}_2 = \dot{F}/k_s$  which can be substituted into (21) to get

$$F = k_p\left(x - \frac{F}{k_s}\right) + b\left(\dot{x} - \frac{\dot{F}}{k_s}\right),$$

which can be rearranged for  $\dot{F}$ .

$$\dot{F} = \frac{k_s}{b}(k_p x + b \dot{x} - (1 + \frac{k_p}{k_s})F) \quad (22)$$

The whole system can be integrated numerically using

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ F \end{bmatrix} = \begin{bmatrix} \dot{x} \\ f(t) - F \\ \frac{k_s}{b}(k_p x + b\dot{x} - (1 + \frac{k_p}{k_s})F) \end{bmatrix}. \quad (23)$$

For the nonlinear model,  $\dot{x}_2$  can be found using root finding algorithms given that  $x$ ,  $\dot{x}$  and  $x_2$  are found using Euler integrations. Equation 24 can be used to find  $\dot{x}_2$ .

$$F = F_{see} = F_{pee} + F_{sde} = k_{s3}x_2^3 + k_{s1}x_2 = k_{p3}(x - x_2)^3 + k_{p1}(x - x_2) + b_3(\dot{x} - \dot{x}_2)^3 + b_1(\dot{x} - \dot{x}_2)$$

$$k_{s3}x_2^3 + k_{s1}x_2 - k_{p3}(x - x_2)^3 - k_{p1}(x - x_2) - b_3(\dot{x} - \dot{x}_2)^3 - b_1(\dot{x} - \dot{x}_2) = 0 \quad (24)$$

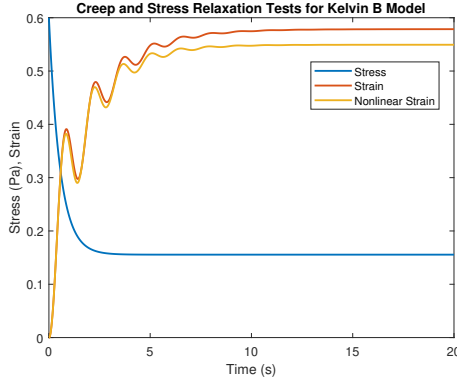


Figure 20: Creep test and stress relaxation test for the Kelvin B model with  $k_{s1} = 20$ ,  $k_{s3} = 10$ ,  $b_1 = 15$ ,  $b_3 = 5$ ,  $k_{p1} = 7$ ,  $k_{p3} = 3$ .

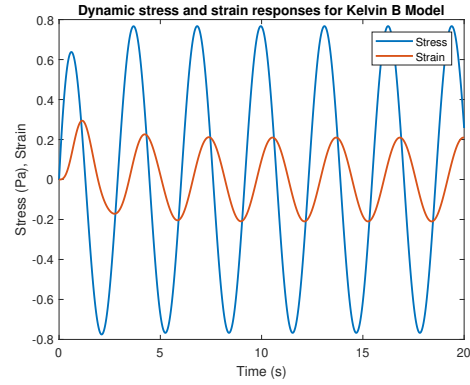


Figure 21: Dynamic creep test and stress relaxation test for the Kelvin B model with  $k_{s1} = 20$ ,  $k_{s3} = 10$ ,  $b_1 = 15$ ,  $b_3 = 5$ ,  $k_{p1} = 7$ ,  $k_{p3} = 3$ .

The stress relaxation and creep tests seen in Figure 21 are very similar to Model A seen in Figure 16 however, this model approaches a steady state strain in less time. For this figure, different values for the constant strain  $\varepsilon_0$  and  $\sigma_0$  are used hence the final values of stress and strain are not comparable to Figure 16. The dynamic response of the system, as seen in Figure 21 is very similar to Kelvin A with the instantaneous response approaching the steady state within 3-4 seconds.

The results for the force-length characteristics for the nonlinear Kelvin Model B can be seen in Figures 22-23. This seems to match to the dynamics of the Kelvin A model and in some papers they suggest that they are both identical, however a mathematical proof is required to be certain of this. Hence, the report will continue to treat them as two separate models. The model contains a similar hysteresis property to Kelvin A when subject to a step input force in Figure 23. However, there is less oscillatory behaviour along the force-extension curve.

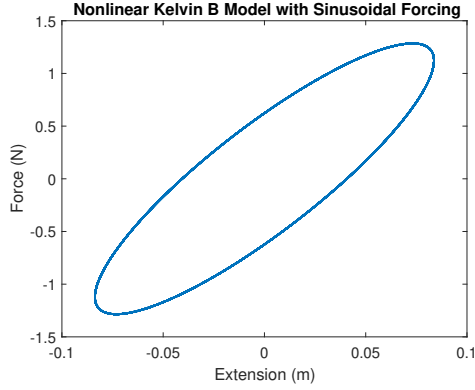


Figure 22: Force-extension relationship for the nonlinear Kelvin B model with  $k_{s1} = 20$ ,  $k_{s3} = 10$ ,  $b_1 = 15$ ,  $b_3 = 5$ ,  $k_{p1} = 7$  and  $k_{p3} = 3$  and sinusoidal forcing

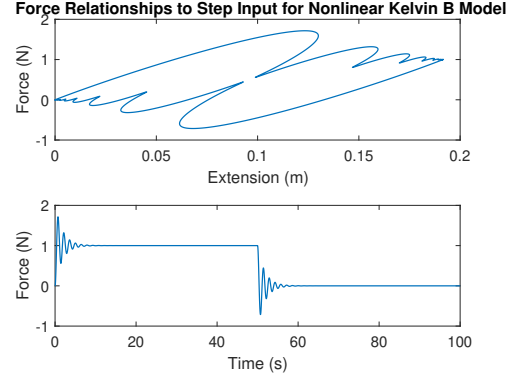


Figure 23: Force-extension relationship and the force over time for the nonlinear Kelvin B model with  $k_{s1} = 20$ ,  $k_{s3} = 10$ ,  $b_1 = 15$ ,  $b_3 = 5$ ,  $k_{p1} = 7$  and  $k_{p3} = 3$  with step force input,  $f(t)$  at time  $t = 0$  and no input force at  $t = 50$  to show loading and unloading behaviour.

## 2.5 Hill Model

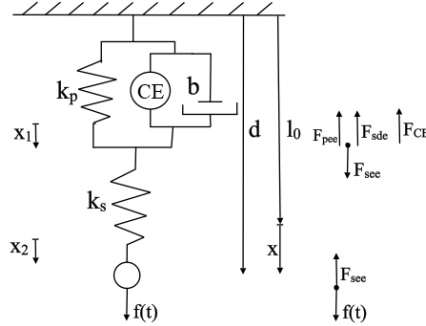


Figure 24: Hill Model of a muscle

The Hill Model is the only active muscle model that will be considered in this report and its mechanical topology is a three element mechanical model as seen in Figure 24. It is an extension of the Kelvin B model and hence direct comparisons can be made between using an active muscle model as opposed to passive muscle models. The linear version of the model which is composed of linear springs and dashpots is similar to Equation 22 with an added force term,  $F_{CE}$  for the contractual element. As mentioned previously, the Hill model uses established force-length and force-velocity relationships for the contractual element which have been obtained through experimental results on real muscles. However, the model makes no attempt to reproduce the mechanism of how these occur, like the computationally intensive Huxley model.

$$\dot{F} = \frac{k_s}{b} (k_p x + b \dot{x} - (1 + \frac{k_p}{k_s}) F + F_{CE}) \quad (25)$$

Equation 25 represents the linear Hill three element model which is a function of  $x$  and  $\dot{x}$ . The force from the contractual element is given by Equation 26 which depends on: the maximum isometric force the muscle can produce  $F_0^M$  (when  $\dot{x} = 0$ ), the activation of the muscle  $a \in [0, 1]$ , the force-length relationship  $f_L$  and the force velocity relationship  $f_v$  for the contractual element.

$$F_{CE} = F_0^M a f_L(d_1) f_v(\dot{x}_1) \quad (26)$$

The force-length relationship is given by Equation 27, where  $c = -\frac{1}{\text{width}^2}$  and  $\text{width} = 0.56$  [21], and the relationship can be seen in Figure 25. Alternative functions include the exponential bell curve which can be slightly different along the ascending and descending curves [3]. The implementation that was chosen was deemed to be the simplest and represented a suitable starting place for the investigation with active muscle models.

$$f_L(d_1) = c \left( \frac{d_1}{l_{CE_{\text{opt}}}} \right)^2 - 2c \left( \frac{d_1}{l_{CE_{\text{opt}}}} \right) + c + 1 \quad (27)$$

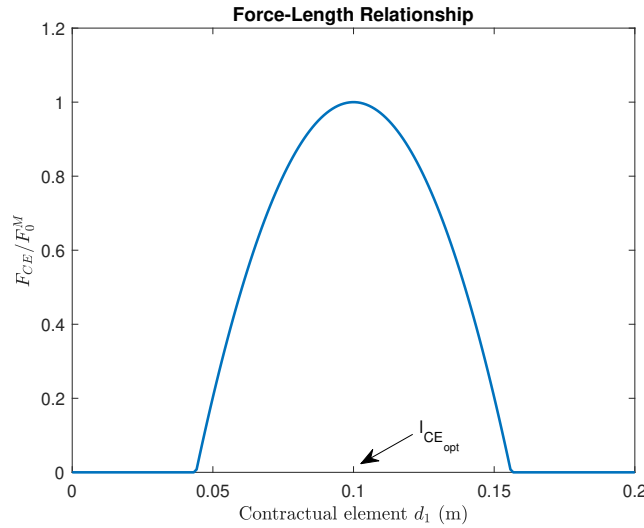


Figure 25: Force length relationship for the active contractual element in the Hill model. The start of the curve is at:  $\text{width} \times l_{CE_{\text{opt}}}$  and the maxima is at  $l_{CE_{\text{opt}}}$  [21]

The force velocity relationship is given by Equation 28 where  $f_{v \text{ max}}$  is the maximum contraction velocity, set to 1.5 [3].  $v_{\text{max}} = \frac{l_{CE_{\text{opt}}}}{\tau_c}$  is the maximum possible velocity of the contractual element with  $\tau_c = 0.1$ , a time constant [3]. The parameters  $k_{CE_1}$  and  $k_{CE_2}$  are the force-velocity shape factors which are 0.25 and 0.06 respectively [3, 17]. At zero velocity, the slope of the force-velocity curve for lengthening contractions was set to be twice as large as the slope for shortening contractions [20].

The force is different when the muscle undergoes eccentric and concentric contraction which is represented by the force-velocity relationship, Figure 26. However, this relationship still preserves Hill's equation from 1938 showing that a hyperbolic relationship exists between force and velocity when undergoing concentric contraction, Equation 29.  $F_0$  is the maximum isometric

tension of the muscle which occurs when the muscle, under a constant loading force, has no velocity, hence equals  $F_0^M$  in our representation.

$$\tilde{\dot{x}}_1 = \frac{\dot{x}_1}{v_{\max}}$$

$$f_v(\tilde{\dot{x}}_1) = \begin{cases} 0 & \tilde{\dot{x}}_1 \leq -1 \\ \frac{1+\tilde{\dot{x}}_1}{1-\tilde{\dot{x}}_1/k_{CE1}} & -1 < \tilde{\dot{x}}_1 \leq 0 \\ \frac{1+(\tilde{\dot{x}}_1 f_{v \max})/k_{CE2}}{1+\tilde{\dot{x}}_1/k_{CE2}} & \tilde{\dot{x}}_1 > 0 \end{cases} \quad (28)$$

$$(v + b)(F + a) = b(F_0 + a) \quad (29)$$

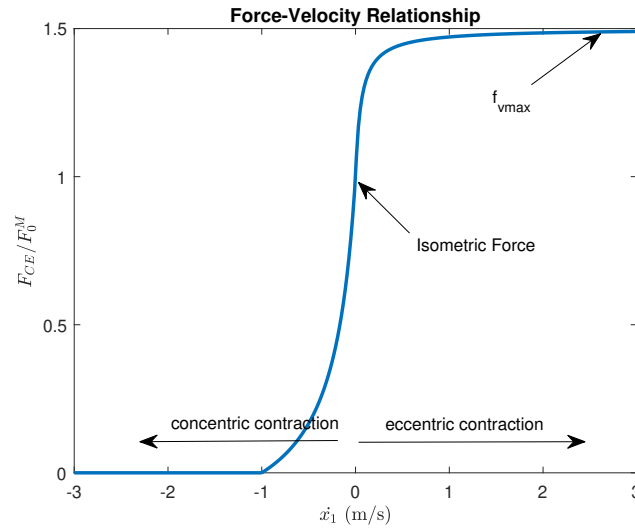


Figure 26: Force velocity relationship for the active contractual element in the Hill model. When the muscle is shortening (concentric contraction  $\dot{x}_1 < 0$ ), the faster the speed the lower the force. When the muscle is lengthening (eccentric contraction  $\dot{x}_1 > 0$ ), the faster the speed the higher the force up to a particular limit,  $f_{v \max}$ .

For the three passive elements, they behave similarly to the Kelvin Model but are slightly adapted to behave more like an actual muscle. The main difference is that the passive springs produce no force when the length is less than slack, whereas the current Kelvin model can produce both compressive and tensile forces.

For the parallel elastic element the slack length  $l_{PEE}$  is a function of  $l_{CE_{opt}}$  and has a maximum force that it can produce  $F_{PEE} = 1.3F_0^M$ , as seen in Figure 27.

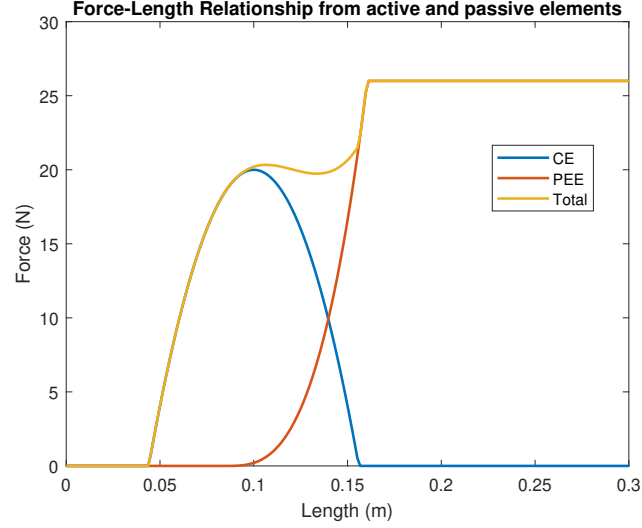


Figure 27: The force-length relationship of the Hill's muscle model with the active and passive element

For the series elastic element  $F_{SEE} = k_s(d_2 - l_2)^3 = k_s(x_2)^3$  where  $k_s$  is defined in Equation 30.  $U_0 = 0.04$  is the strain at maximal isometric force [21], and the force-length relationship can be seen in Figure 28.

$$k_s = \frac{F_0^M}{(U_0 x_2)^3} \quad (30)$$

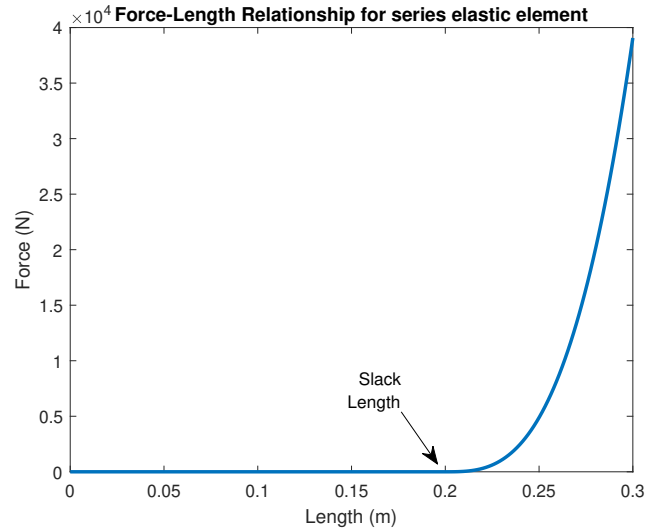


Figure 28: The force-length relationship for the series elastic element

The calculation for  $a$  in Equation 26 can be found from [3] represented in Equation 31 where  $A = -4$  and  $u(t) \in [0, 1]$  is the rescaled input to be inside the desired range. This activation

relationship which can be seen in Figure 29, represents taking input EMG data which can be directly measured and returning  $a$  which is the proportion of total activation of the muscle [17]. EMG data can be measured easily using electrodes on the surface of the skin and from this the muscle activation level can be calculated.

$$a(t) = \frac{e^{Au(t)} - 1}{e^A - 1} \quad (31)$$

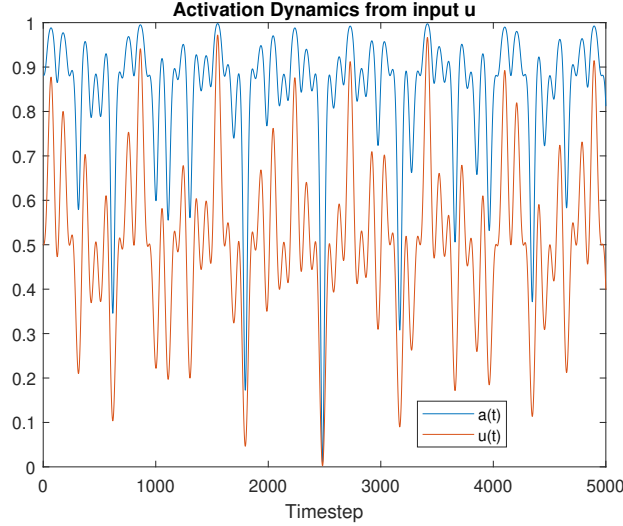


Figure 29: Activation dynamics

Although the Hill model described above has a strain relationship for the series elastic element to define the amount of extension of this element, and the total force from the parallel elastic element is limited at a particular value of  $1.3F_0^M$ , this has not been simulated into the network. The network must be random and generic, and if too many limitations are put on the model, all the connections will behave in a too similar manner. The slack behaviour of the model still remains for the two elastic elements but when the extension is positive, the spring acts nonlinearly with a cubic and linear term and no upper bound as described previously. The serial damper element behaves in the same way as for all models,  $F_{SDE} = b_3\dot{x}_1^3 + b_1\dot{x}_1$ .

The model can be simulated by solving for  $\dot{x}_2$  in the equilibria equation (with the slack extensions producing no force) given that  $x$ ,  $\dot{x}$  and  $x_2$  are already known,

$$F_{CE} + F_{SDE} + F_{PEE} = F_{SEE}$$

$$F_0^M a f_L((x - x_2) + l_1) f_v(\dot{x} - \dot{x}_2) + b_3(\dot{x} - \dot{x}_2)^3 + b_1(\dot{x} - \dot{x}_2) + k_{p3}(x - x_2)^3 + k_{p1}(x - x_2) = k_{s3}x_2^3 + k_{s1}x_2$$



and integrating numerically with

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ x_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ f(t) - F \\ \dot{x}_2 \end{bmatrix}.$$

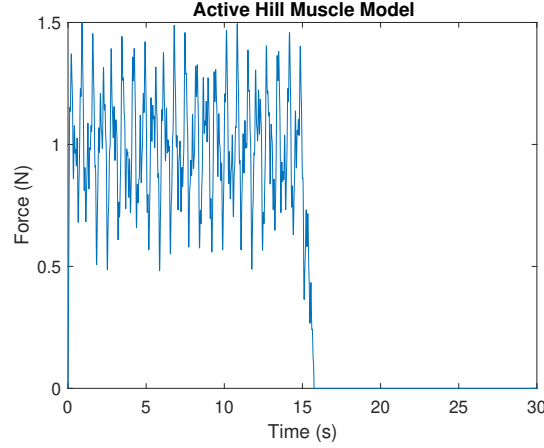


Figure 30: Force over time for the the Hill model with the activation input given in Equation 37 with randomly initialised parameters. Step force input at time  $t = 0$  and no input force at  $t = 15$  to show the loading and unloading behaviour.

The behaviour of the muscle to a step input forcing can be seen in Figure 30. The model still requires a load to produce a force even though this is an active model because muscles are always under some form of load in order to move the limbs that they are connected to. However, it could be that the damping is too large in the system as some papers suggest using linear damping for the damping element [4].

## 2.6 Bouc-Wen Hysteresis

A Hysteresis model was used in addition to the mechanical models proposed in order to accentuate the hysteresis property that all muscles show. The structure has a hysteric restoring force given in Equation 33 which is a function of the hysteretic displacement  $z$ , which lags behind the extension from natural length. The system outputs a different force if it is in the state of loading or unloading, so it dissipates energy because the path along the curve is different for muscle extension and contraction. The resisting force  $F$  is given in Equation 34, where  $\alpha$  is the post-to-pre yield ratio and  $k$  is the spring stiffness. Equation 35 governs the dynamics of the auxiliary variable  $z$ , the hysteresis displacement,  $A$  and  $n$  govern the size and smoothness of the hysteresis loop and  $\beta$ ,  $\gamma$  govern the general shape of the loop [10]. The muscle will generate stronger forces whilst lengthening compared to shortening which gives rise to this hysteresis loop. It is believed that in locomotion, hysteresis could be a useful property. In this model, the hysteresis is added onto the nonlinear Voigt model which is the standard mass spring damper system.

$$F_s = \alpha(k_3x^3 + k_1x) + b_3\dot{x}^3 + b_1\dot{x} \quad (32)$$

$$F_z = (1 - \alpha)(k_3z^3 + k_1z) \quad (33)$$

$$F = F_s + F_z \quad (34)$$

$$\dot{z} = A\dot{x} - \beta|\dot{x}||z|^{n-1}z - \gamma\dot{x}|z|^n \quad (35)$$

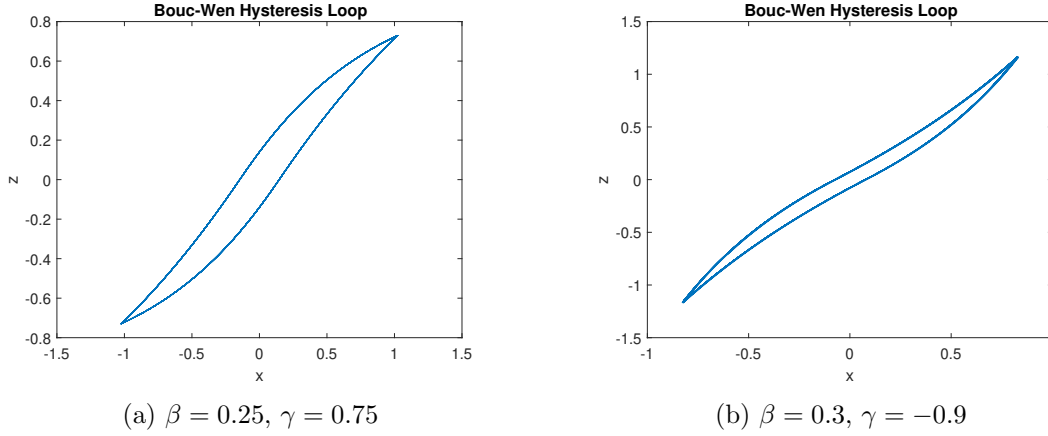


Figure 31: Examples of Bouc-Wen hysteresis models of muscles using different parameters of  $\beta$  and  $\gamma$  and integrated numerically with Euler. All other parameters are set to:  $A = 1$ ,  $B = 1$ ,  $d_1 = 0.5$ ,  $d_3 = 0.5$ ,  $k_1 = 1$ ,  $k_3 = 1$ ,  $\alpha = 0.5$ ,  $n = 1.2$ ,  $p = 1$ .

Figures 31a and 31b have been simulated with Euler numerical integration and slightly different  $\beta$  and  $\gamma$  parameters to get different hysteresis loops. When implemented into the recurrent networks  $\alpha = 0$  so that the force produced is purely coming from this hysteresis of the spring and not from the non-hysteretic spring that the Voigt model uses. This is so that the addition of hysteresis can be directly compared to the model without hysteresis to see if an improvement in performance has occurred.

### 3 Implementation and Learning Tasks

#### 3.1 Setting up the recurrent network

The setup that will be used to implement morphological computation with the muscle models discussed in Section 2 extends current work done adapted from [5, 8, 9]. The required properties from the network is that it is random and generic, so that it can be used for multiple tasks, contains nonlinearity and temporal dynamics, which is inherent with the muscle models used, and contains complexity and dynamic coupling.

The positions of the nodes of the network are set randomly within the range  $[0,10]$  for both  $x$  and  $y$  directions. The left-most and right-most nodes are selected as the fixed nodes in order to hold the network in place. A certain percentage of input nodes are chosen to receive an input signal  $u(t)$ , indicated in green in Figure 32, chosen to be a third of all nodes. Before the simulation starts, the weights for the input nodes  $\mathbf{w}_{\text{in}} = [w_{\text{in},1}, w_{\text{in},2}, \dots, w_{\text{in},n}]^T$  are fixed randomly from the range  $[-1,1]$  and each input node  $i$  receives a weighted input,

$$F_{\text{input},i} = w_{\text{in},i} \cdot u(t). \quad (36)$$

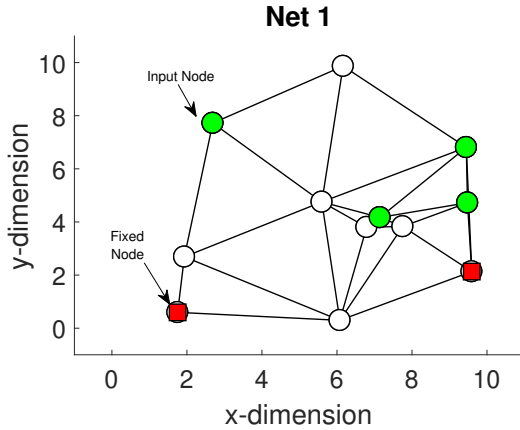


Figure 32: Example of a random recurrent network of mass-spring systems, adapted from [5].

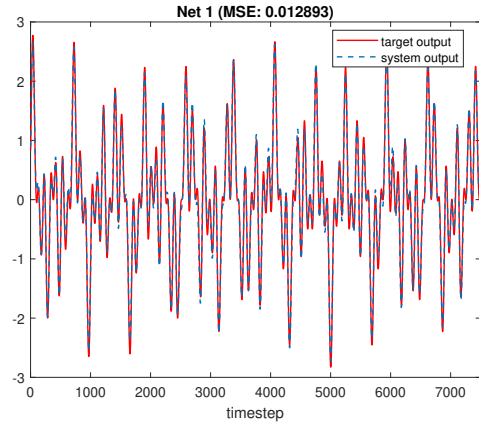


Figure 33: The system from Figure 32 output against desired output performance for learning the memory task given in Equation 36 with  $N = 5$ .

Additionally, the parameters from the muscle models are chosen randomly from a set range which will be discussed further in detail.

The connections between the nodes are set randomly using Delauney triangles, as a result different networks with the same number of nodes may have a different number of connections. Initially, the resting length of the connections are set as the current length so that no connection produces no force and the network is in equilibria. All the connection lengths contribute to the

final output as a weighted sum so for time  $t$ ,

$$y(t) = \sum_j X_{jk} \cdot w_{\text{out},j},$$

where  $\mathbf{X}$  is a matrix storing the simulated connection lengths for all time, with  $X_{jk}$  being the length of the  $j^{\text{th}}$  connection at timepoint  $k$  with  $t = k\Delta t$ .  $\mathbf{w}_{\text{out}} = [w_{\text{out},1}, w_{\text{out},2}, \dots]^T$  is a vector of output weights that are learnt for a specific learning task. The performance of these recurrent networks can be quantified using the mean-squared error between the system output  $y(t)$  and the desired output  $y^*(t)$ .

In order to run the simulation, the input nodes are given a horizontal force calculated from Equation 36. At each time step of the simulation, the horizontal and vertical forces on each node are calculated resulting from extensions of the connections from natural length. Both the positions of the nodes and the dynamic models, as discussed in Section 2, for the connections are integrated numerically using Euler's method with a fixed timestep,  $\Delta t$ . The desired output weights  $\mathbf{w}_{\text{out}}$  are learnt using linear regression given that the desired output  $\mathbf{y} = \mathbf{X}\mathbf{w}_{\text{out}}$  where  $\mathbf{X}$  is the simulated data connection lengths for all time. The transient, or washout, needs to be removed from  $\mathbf{X}$  at the start of simulation as it is not representative for how the whole system responds to the given input. Figure 34 shows that the first 15 seconds of the simulation does not behave in the same manner as the rest.

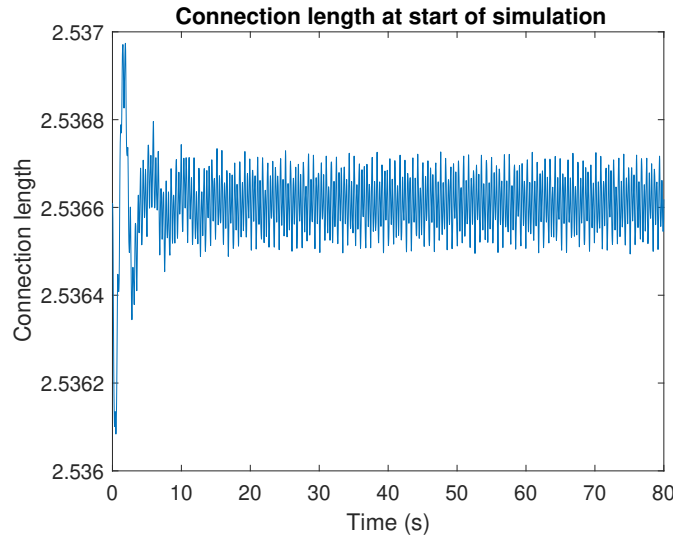


Figure 34: Connection length over time for an example connection at the start of the simulation.

With the static output weights  $\mathbf{w}_{\text{out}}$  set for the desired output task, the performance of the network can be calculated by running an input test set through the network using the learnt weights to calculate the actual system output  $y(t)$ . This can be compared to the desired output  $y^*(t)$  using the mean-squared error metric. A result of this can be seen in Figure 33 which gives a mean squared error of 0.12.

### 3.2 Specific muscle model implementation

For the Hill's muscle model for which an activation is required,  $a(t)$  was chosen to be  $a(t) = u(t)w_{in,i}$  for each active connection  $i$ , inputting  $u(t)$  into the connections. Hence, no horizontal forcing on the input nodes occurred and the input  $u(t)$  was transformed into the network through the connections rather than the nodes. The activation  $a(t)$  was appropriately scaled so that it would always be within the desired range  $a(t) \in [0, 1]$ . The input connections were chosen at random with a third of all connections being chosen as input connections, as seen in Figure 35. However, as the chosen connections did not take into account where the previous input nodes were placed, comparing the two methods would be harder. When using the Hill's muscle model, the remainder of the connections were chosen as the Hill's model without any activation, which is equivalent to the passive Kelvin B model.

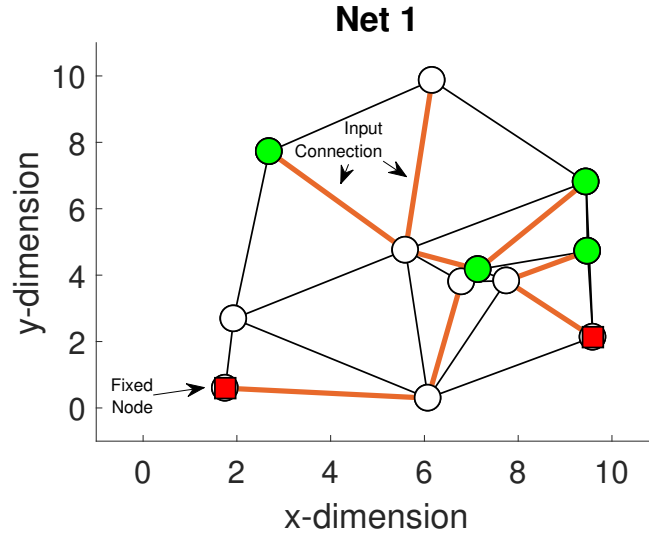


Figure 35: The activation muscles are given in orange that will use the input  $u(t)$  to form the activation  $a(t)$ .

In order to randomly assign the parameters for each connection in the muscle models, a desired range was selected for each parameter. This was based loosely on what had been done before in [8], however certain parameter combinations resulted in unstable simulations when having to find the value of  $\dot{x}_2$  for the nonlinear models using root finding algorithms. The parameter ranges can be found in Table 1 with values chosen from a random exponential distribution.

Parameter	Range	Parameter	Range
$k_1$ or $k_{s1}$	[10,100]	$k_{p1}$	[10,100]
$k_3$ or $k_{s3}$	[1,100]	$k_{p3}$	[1,100]
$d_1$	[1,10]	$\beta$	[-10,10]
$d_3$	[1,10]	$\gamma$	[-10,10]

Table 1: Parameter ranges chosen to randomly assign to the muscle models

Initially, 30 nodes were chosen for the network as adapted from [5], but when the performance to the first memory task, which can be seen in (38), was assessed, it was found that all the networks had very good performance, so the networks may be too complex for the task given. As a consequence, it was decided to use 12 nodes so that a better evaluation of the performance of the different muscle models could be obtained.

### 3.3 Learning Tasks

As mentioned previously in Section 1.2, there are two main properties required of a reservoir in order to do reservoir computing. These are the temporal integration of information, requiring dynamic memory, and nonlinearity or complexity. With this in mind the supervised learning tasks proposed cover aspects requiring memory and complexity or nonlinearity. The input for all three tasks  $u(t)$  is given as a product of three sinusoidal functions with different frequencies (37). This results in a function with a large timeperiod of 100 seconds when the frequencies are chosen as ( $f_1 = 2.11$ ,  $f_2 = 3.73$  and  $f_3 = 4.33$  Hz) as in [8].

$$u(t) = \sin(2\pi f_1 t) \sin(2\pi f_2 t) \sin(2\pi f_3 t) \quad (37)$$

The first task designed is a memory task and is a very simple task where the output is the sum of previous  $N$  inputs of the discretised input.

$$y_N = \sum_{i=0}^N u[t - i] \quad (38)$$

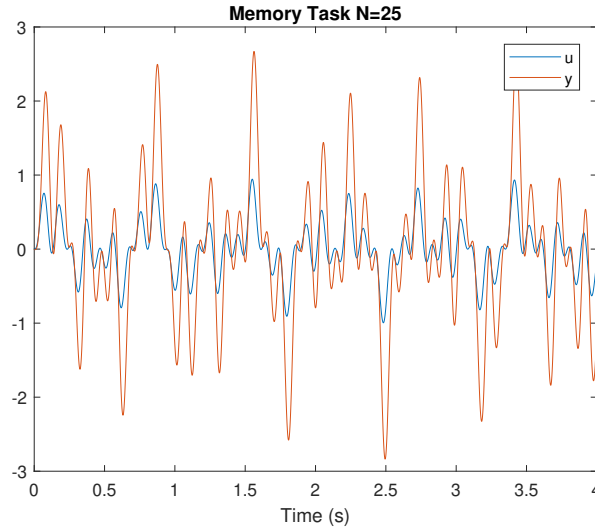


Figure 36: The input and output memory learning task from Equation 38 with  $N = 25$ .

The NARMA (nonlinear auto-regressive moving average) task as described in [15] is a second-order nonlinear dynamical system which is often a standard benchmark test in the context of

machine learning. NARMA is another memory task which is defined recursively for order  $N$  in Equation 39.

$$y_N[t+1] = 0.3y[t] + 0.05y[t] \left( \sum_{j=0}^{N-1} y[t-j] \right) + 1.5u[t-N+1]u[t] + 0.1 \quad (39)$$

In order to measure the ability of the muscle models to execute complex tasks, a task was created where the output is a polynomial of order  $n$ . The coefficients were set randomly within the range  $[-1,1]$  and set constant for each task,  $n = 2, 3, 4, 5$ .

$$y = a_5u^5 + a_4u^4 + a_3u^3 + a_2u^2 \quad (40)$$

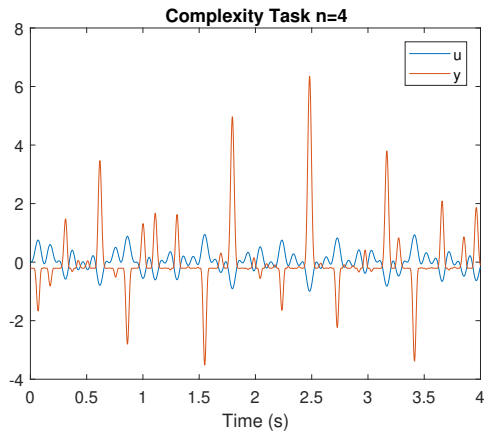


Figure 37: The complexity learning task from Equation 40 with  $n = 4$ .

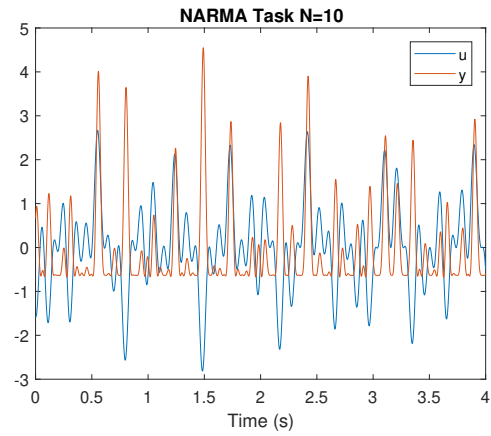


Figure 38: The standard NARMA memory task from Equation 39 with  $N = 10$ .

## 4 Network Results

The results were calculated using the same ten random networks that were all initialised in the same way as outlined in the previous section. These ten networks were run on the three different tasks with the amount of memory or complexity given by the variables  $N$  and  $n$  remaining consistent. The aim was to see an improvement in mean-squared error performance when compared to the nonlinear Voigt model which had been used previously as a standard nonlinear mass spring damper system. The results were plotted with error bars that show one standard deviation from the mean. For clarity, with some of the results which contain a log scale, the error bar was not plotted below the mean as it would have passed through zero so would not show any useful information.

The simulations were run for 200 seconds with the first 60 seconds being removed as washout. The remaining data was used to calculate output weights for the given task through linear regression. A further 15 seconds of input-output data was used to test the network performance for the given task once the output weights have been learnt.

### 4.1 Memory Task

All the results for the memory task follow similar characteristics for differing  $N$  values. The prior expectation was that the performance would decrease as  $N$  increases but there seems to be a plateau in mean-squared error around  $N = 40$ . The results for  $N = 60$  and  $N = 100$  show improved performance with a significantly lower variance value for the  $N = 100$  task. One possible explanation for this may be that the output given when  $N = 100$  could become more periodic as the increasing number of delayed functions to sum causes the output to act as a moving average. This moving average could start to show periodic behaviour if  $N$  becomes too large.

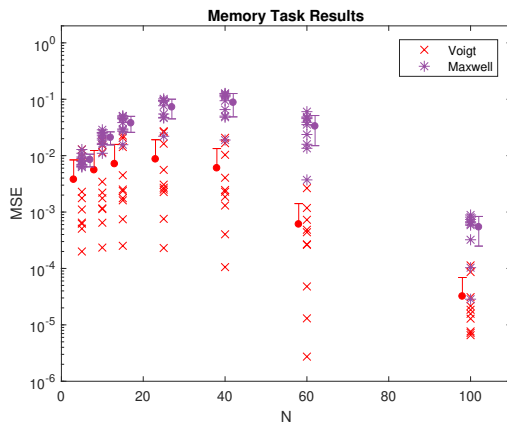


Figure 39: Memory task results comparing the Voigt and Maxwell models for differing  $N$  values.

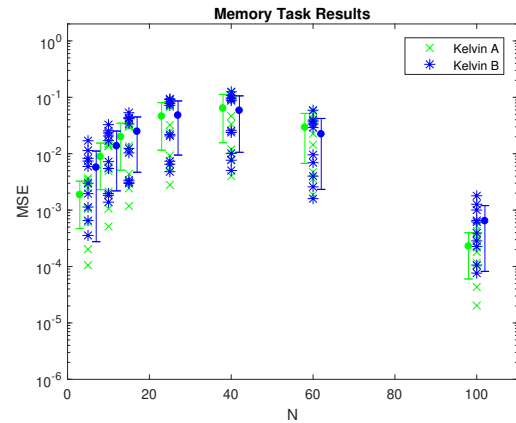


Figure 40: Memory task results comparing the Kelvin A and B models for differing  $N$  values.



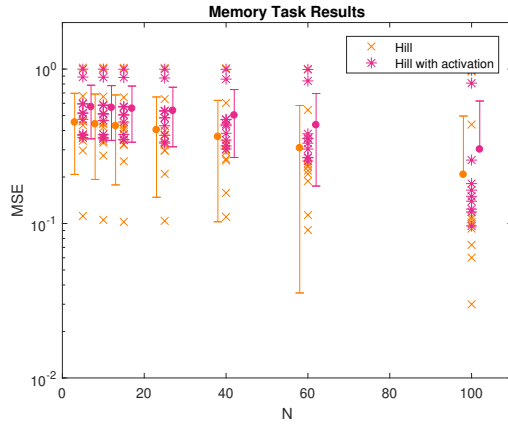


Figure 41: Memory task results comparing the Hill model with and without the activation function given in Equation 31 for differing  $N$  values.

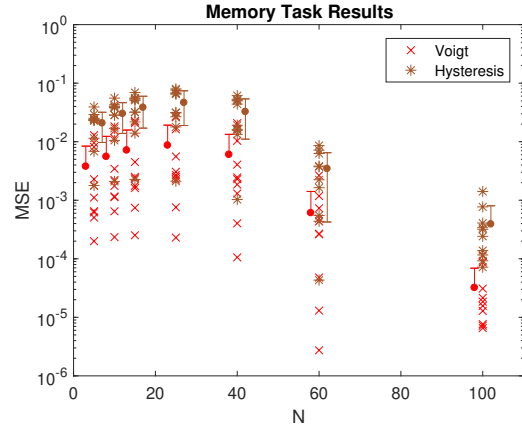


Figure 42: Memory task results comparing the Voigt and Hysteresis models for differing  $N$  values.

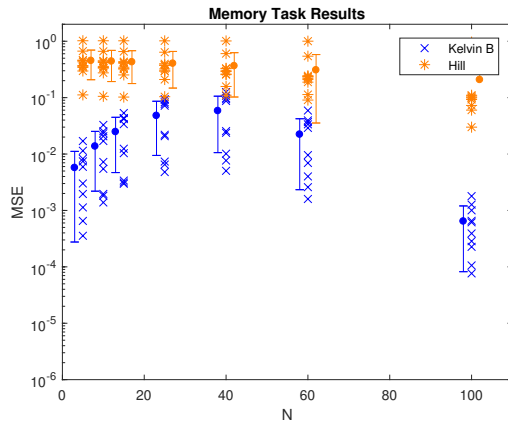


Figure 43: Memory task results comparing the Hill with activation function and Kelvin B models for differing  $N$  values.

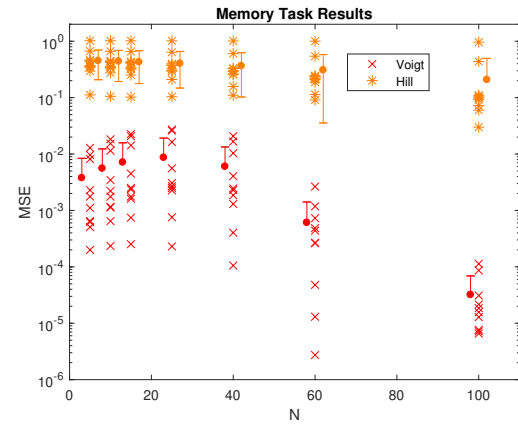


Figure 44: Memory task results comparing the Voigt and Hill without activation function models for differing  $N$  values.

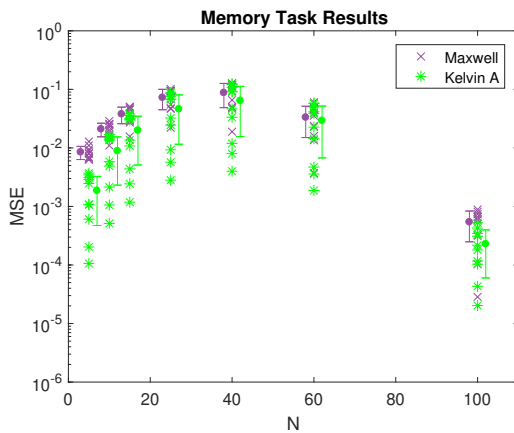


Figure 45: Memory task results comparing the Maxwell and Kelvin A models for differing  $N$  values.

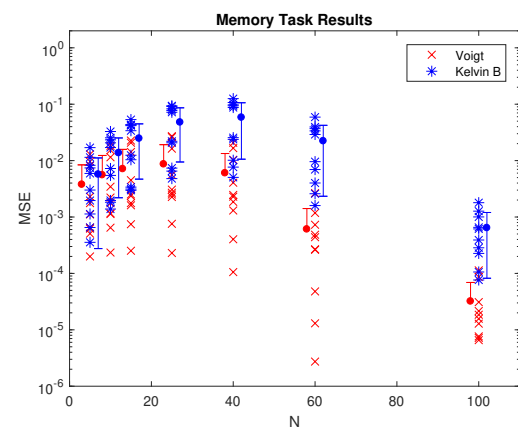


Figure 46: Memory task results comparing the Voigt and Kelvin B models for differing  $N$  values.

As can be seen in Figures 39-46, the Voigt model has the best performance when looking only at nonlinear models. In an attempt to improve performance, Bouc-Wen hysteresis is added to the Voigt model to accentuate the hysteric property that all muscles show. However, as Figure 42 shows the mean-squared error is clearly lower for the Voigt model.

Additionally, all four of the passive models follow a similar pattern in results along increasing  $N$  values with an increasing mean-squared error up to  $N = 40$ , but better performance for  $N = 60$  and  $N = 100$ . The active Hill model however doesn't seem to show this as can be seen in Figure 44, with very little change in mean-squared error across different  $N$  values. The Hill model seems to perform worse than all the passive models which can be seen in Figures 43-44. Additionally, in an attempt to make the Hill model more representative of real muscles that use an input stimulus to activate the contractural element, the activation relationship, which can be found from Equation 31, was implemented. The result can be seen in Figure 41, but this added activation function seems to decrease performance because the input has been nonlinearly transformed before going into the network.

Kelvin A and B models show very little difference in performance as can be seen from Figure 40, which would be expected as some papers suggest that they are identical. As seen from Figure 48, the Kelvin B network produces a stable simulation with the connection length operating within a specific range after a short washout period. This does not occur with the Maxwell model, Figure 47 shows that the connection length gradually decreases over time. This is due to the extension of the damper which will continue to either extend or shorten as the resistive force is not sufficient to return it to its resting length. The resulting simulation, if run for long enough, will likely be unstable. Therefore, it is not unusual to see that the Maxwell model performs worse than the Kelvin A model (similar to Kelvin B model) in Figure 45.

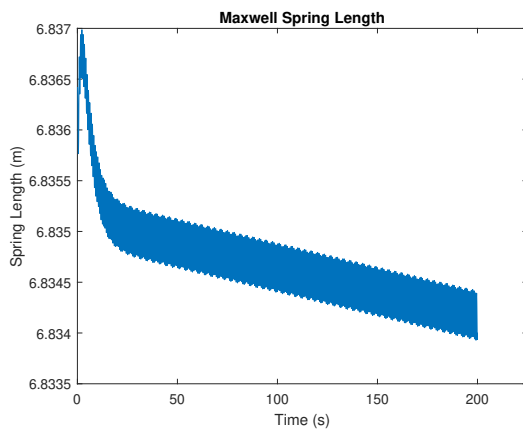


Figure 47: The connection length for a typical Maxwell model connection over the whole simulation time.

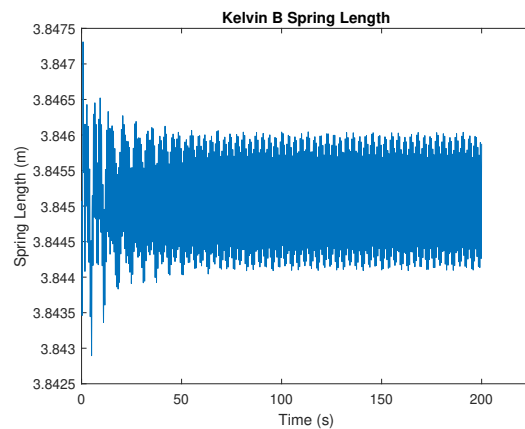


Figure 48: The connection length for a typical Kelvin B model connection over the whole simulation time.

An example of the active input connection lengths for the Hill model can be seen in Figure 49. An example passive connection within the same network can be seen in Figure 50. The active muscles show a high frequency response which may not propagate through the network as the

passive connection shows very little response. This may not be a problem if this only happens to a few connections as they will be given very low output weights and will not contribute to the final output. However, if too many connections behave in this way, the network will not be able to effectively learn the desired output requiring more information from the network.

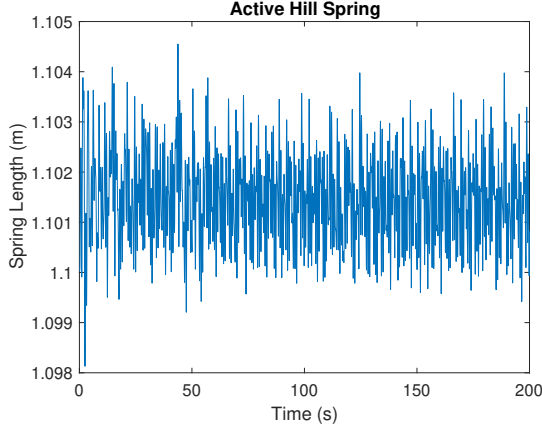


Figure 49: The connection length for an active Hill model muscle (indicated with orange connections in Figure 35) over the whole simulation time.

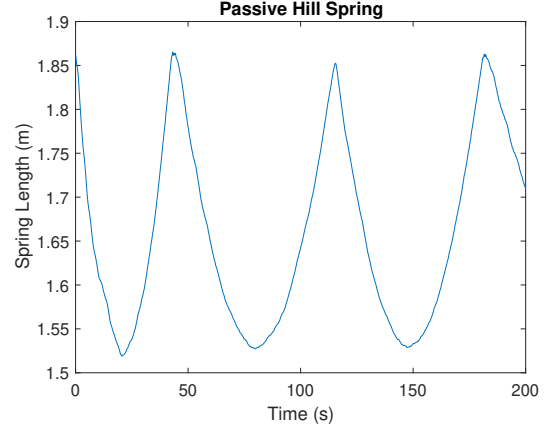


Figure 50: The connection length for a passive Hill model muscle (indicated with black connections in Figure 35) over the whole simulation time.

As a result, the Hill model was altered to get more of a dynamic response from the system. The first alteration was to shorten the resting length of the contractual element  $l_1 = l_{CE_{opt}}$ . This would mean there would be more variation of force from the contractual element so that for the same positive extension  $x_1$ , the value of  $\frac{d_1}{l_{CE_{opt}}}$  would increase. Therefore, along the force-length relationship, the force would be significantly lower as it moves further away from the peak in Figure 25. This variation in force should help to propagate the input through the network. Initially, the resting length of the contractual element was set to  $l_1 = 0.5l_0$  where  $l_0$  is the total resting length of the connection, but for the small contractual element model,  $l_1 = 0.2l_0$ . The second alteration was to use two thirds of the total connections as active input connections, keeping  $l_1 = 0.5l_0$ . Lastly, the final alteration was to use all the connections as active Hill model connections.

The results of these alterations can be seen in Figure 51 for the memory task with  $N = 5$ . This result is representative for all values of  $N$  chosen and it can be seen that these alterations only worsened performance.

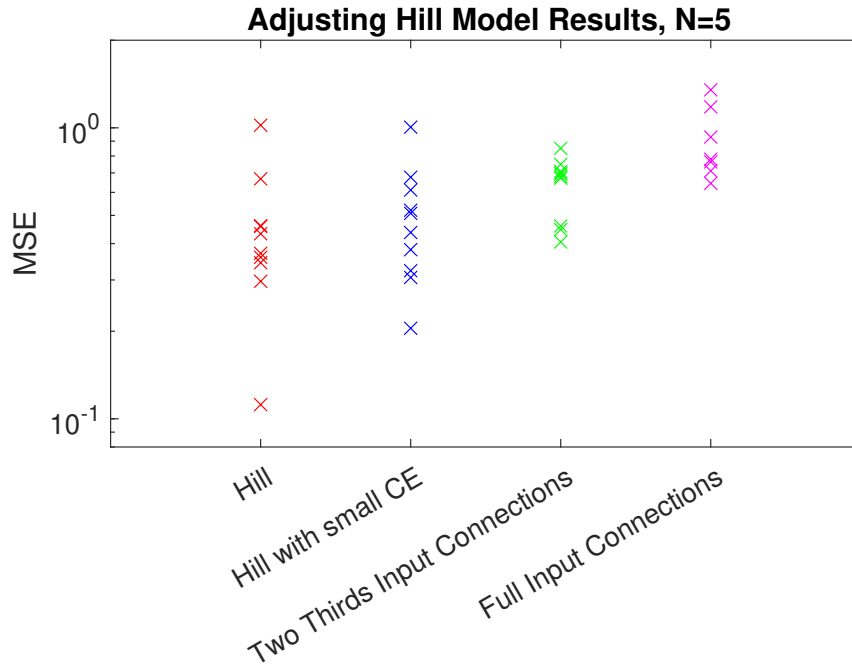


Figure 51: Memory task results with  $N = 5$  for the Hill model with new alterations.

## 4.2 NARMA Task

The results for the NARMA have not been shown for hysteresis as it only ran 20% of the time, the rest of the simulations were unstable. Additionally, the results are not shown for the Hill model with activation because it performed very poorly for the memory task.

As seen in Figures 52-55, the performance generally decreases with an increasing  $N$  value which is as expected for such a memory task. Figure 53 shows that the Hill and Maxwell models do not perform as well as with the previous memory task. However, Figure 54 shows that the passive Kelvin B model significantly improves performance compared with the Voigt model, which has previously been used as the standard nonlinear mass-spring damper system. This improved performance suggests that a complex passive muscle model such as the Kelvin model, could improve morphological computation performance. However, caution must be taken as it is difficult to compare these two models using only ten networks and with the Kelvin model having two additional parameters that define it.

Comparing the models Kelvin A and B, Kelvin B seems to perform slightly better which is promising given that the more realistic Hill's muscle model is most similar in its mechanical topology to the Kelvin B model.

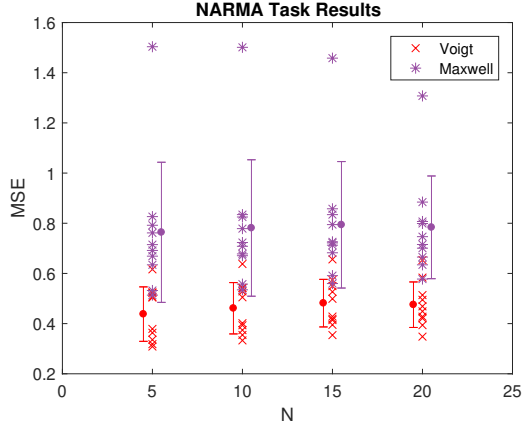


Figure 52: NARMA task results comparing the Voigt and Maxwell models for differing  $N$  values.

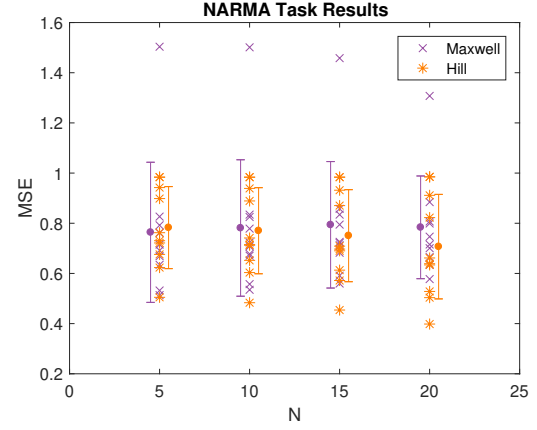


Figure 53: NARMA task results comparing the Maxwell and Hill models for differing  $N$  values.

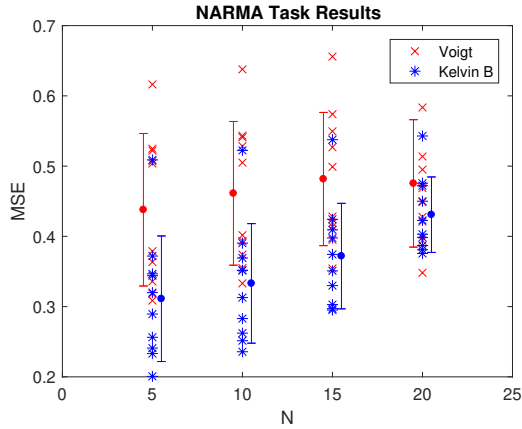


Figure 54: NARMA task results comparing the Voigt and Kelvin B models for differing  $N$  values.

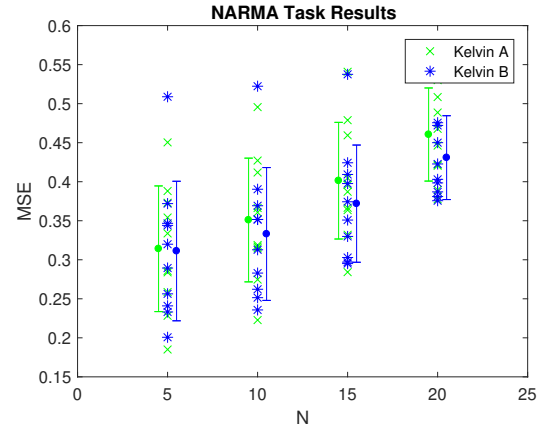


Figure 55: NARMA task results comparing the Kelvin A and B models for differing  $N$  values.

### 4.3 Complexity Task

For the complexity task, the Hill results are not shown because they performed particularly poorly. The four tasks defined were initialised from a randomly assigned polynomial of order  $n$ , and Figures 56-61 all show similar trend with an increasing mean-squared error with increasing complexity.

As with the previous tasks, the Maxwell model doesn't perform very well when compared to the Voigt and Kelvin models, shown in Figure 56. The Maxwell model is very simple and lacks stability as was shown in Section 2.2 with the creep test, where the strain will continue to increase. An interesting find is that both Kelvin A and B models seem to outperform the Voigt model for lower polynomial orders (Figures 57-58) with the performance of Kelvin A model being particularly good, Figure 60.

When hysteresis is added to the Voigt model in Figure 59, it again seems to perform much worse and as such the addition of hysteresis to other models shouldn't be considered in further work.

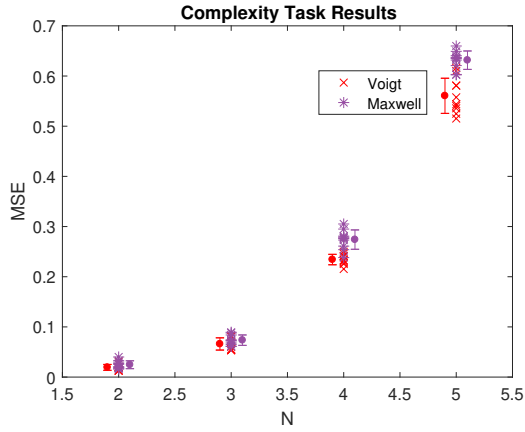


Figure 56: Complexity task results comparing the Voigt and Maxwell models for differing  $n$  values.

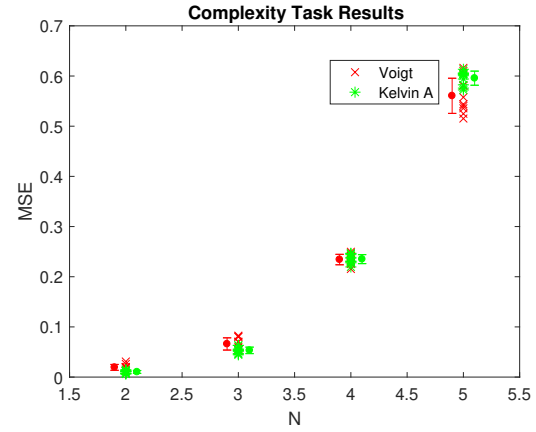


Figure 57: Complexity task results comparing the Voigt and Kelvin A models for differing  $n$  values.

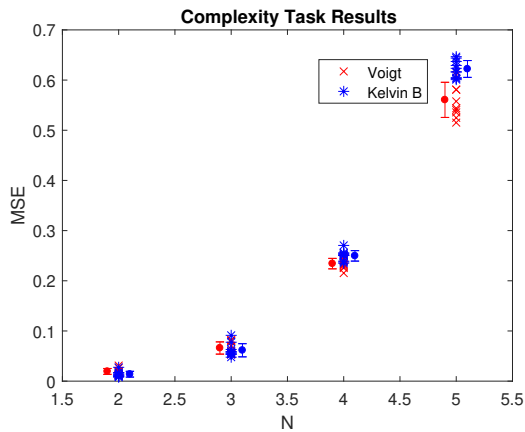


Figure 58: Complexity task results comparing the Voigt and Kelvin B models for differing  $n$  values.

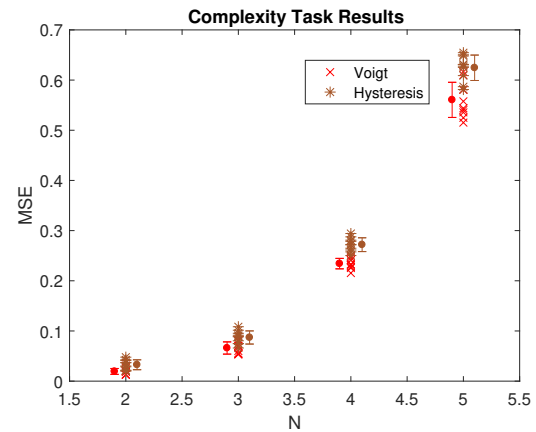


Figure 59: Complexity task results comparing the Voigt and Hysteresis models for differing  $n$  values.

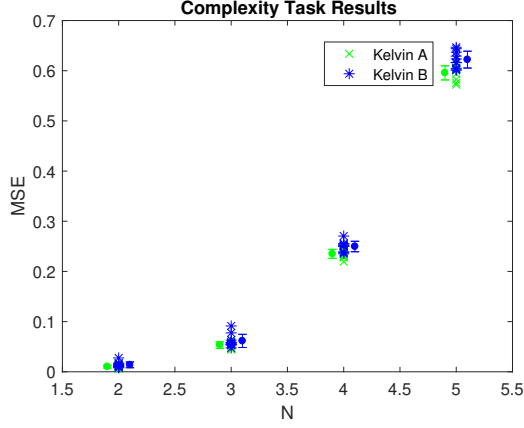


Figure 60: Complexity task results comparing the Kelvin A and B models for differing  $n$  values.

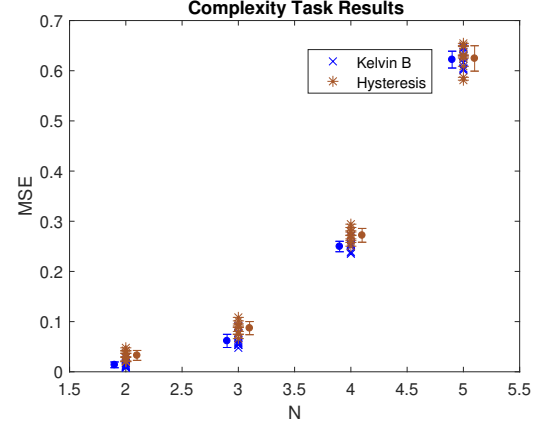


Figure 61: Complexity task results comparing the Kelvin B and Hysteresis models for differing  $n$  values.

#### 4.4 Linear Model Results

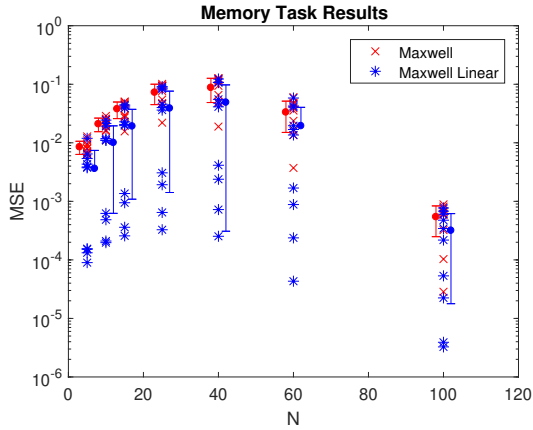


Figure 62: Linear and Nonlinear Models for Maxwell performing on the memory Task.

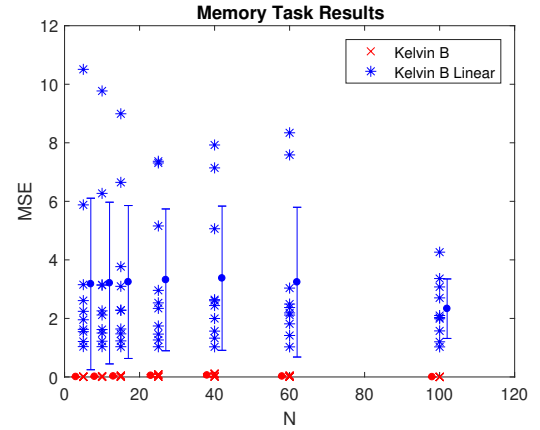


Figure 63: Linear and Nonlinear Models for Kelvin B performing on the memory Task.

A comparison of linear and nonlinear models was undertaken in order to validate the statement that nonlinearity is a required property for morphological computation. Figures 62-63 show the results for the memory task on both the Maxwell and Kelvin B models respectively. The  $y$ -axis has different scales in the two graphs with Figure 62 using a logarithmic scale. The linear Kelvin B model clearly performs much worse with the mean-squared error being at least three orders of magnitude higher than what is seen in Figure 40. However, the linear Maxwell model doesn't perform as one would expect, which again seems to suggest that it is not a good model to use for morphological computation.

## 5 Conclusion

This report summarises an examination into mechanical muscle models that can be incorporated into a morphological computational network. A proposed method for using the connections of the network to feed in the input  $u(t)$ , across the network when using the active Hill model was implemented however, improved performance was not found. Possible adjustments to the Hill model were briefly made however, a more detailed investigation would be required. Overall, some promising results have been shown from the Kelvin A and B models particularly for the NARMA and complexity tasks defined. The passive Kelvin models build upon the basic Voigt and Maxwell models, and more understanding of how the Kelvin model operates and its sensitivity to changes in parameters is required to advance further towards gaining computational power with such a model.

The response of the models to a constant stress, in the creep test, seems to be an important factor to the extent at which the models perform within the network. The Voigt model reaches a steady state within 4 seconds as can be seen in Figure 10, whereas the Maxwell model never reaches a steady state, and the Kelvin models reach a steady state within 10-15 seconds, as seen in Figures 5, 16 and 20. If the parameter ranges used for each model were adapted to increase the response in strain for a constant stress, the models may show an increase in performance.

Additionally, some papers on muscle models suggest that the damping element should be linear whilst the two elastic spring elements are nonlinear [4]. Although this wasn't considered, it may be worth noting as the damping range used in Table 1 had to be kept lower than other parameters to ensure stable simulations.

### 5.1 Further Work

A very simple version of the Hill model has been proposed as an introduction to using this active muscle model in a morphological computational context. Variations and increasing complexity of this model exist and extensive work has been carried out with the model in a range of different contexts. One such proposed model of interest from [3] could be implemented, which is constructed using a slightly different mechanical element combination, and a more rigorous definition of the force-length and force-velocity curve with more shaping parameters defined.

Additionally, an extensive parameter search to find some better suited parameter ranges for the models, which were chosen in Table 1, could definitely improve the performance of the models, however this would be very computationally intensive. The chosen ranges would have to be sufficiently large so that the randomly generated networks would remain generic in order to effectively learn new supervised learning tasks.



## References

- [1] Lennert Appeltant, Guy Van der Sande, Jan Danckaert, and Ingo Fischer. Constructing optimized binary masks for reservoir computing with delay systems. *Scientific Reports*, 4:3629, Jan 2014. Article.
- [2] A J van den Bogert, G K Cole, and K G M Gerritsen. Human muscle modelling from a user ' s perspective. 8:119–124, 1998.
- [3] D. F.B. Haeufle, M. Günther, A. Bayer, and S. Schmitt. Hill-type muscle model with serial damping and eccentric force-velocity relation. *Journal of Biomechanics*, 47(6):1531–1536, 2014.
- [4] H. Hatze. A myocybernetic control model of skeletal muscle. *Biological Cybernetics*, 25(2):103–119, 1977.
- [5] Helmut Hauser. Matlab morphological computation simulation. [https://github.com/helmuthauser/Matlab\\_Morphological\\_Computation\\_Simulation](https://github.com/helmuthauser/Matlab_Morphological_Computation_Simulation). Accessed: 2017-11-23.
- [6] Helmut Hauser. Shanghai lectures 2013 - morphological computation and soft robotics.
- [7] Helmut Hauser, Rudolf M Fuchslin, and Kohei Nakajima. Morphological computation - the physical body as a computational resource. In *Opinions and Outlooks on Morphological Computation*, chapter 20, pages 227–244. 2014.
- [8] Helmut Hauser, Auke J Ijspeert, Rudolf M Fuchslin, Rolf Pfeifer, and Wolfgang Maass. Towards a theoretical foundation for morphological computation with compliant bodies. *Biol Cybern*, 105:355–370, 2011.
- [9] Helmut Hauser, Auke J Ijspeert, Rudolf M Fuchslin, Rolf Pfeifer, and Wolfgang Maass. The role of feedback in morphological computation with compliant bodies. *Biol Cybern*, 106:595–613, 2012.
- [10] Hong-guang Li and Guang Meng. Nonlinear dynamics of a SDOF oscillator with Bouc Wen hysteresis. *Chaos, Solitons and Fractals*, 34:337–343, 2007.
- [11] M Lukoševičius. A Practical Guide to Applying Echo State Networks. *Neural Networks: Tricks of the Trade*, pages 659–686, 2012.
- [12] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*, 14(11):2531–2560, 2002.
- [13] M. Mailah, H. Jahanabadi, M. Z.M. Zain, and G. Priyandoko. Modelling and control of a human-like arm incorporating muscle models. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 223(7):1569–1577, 2009.
- [14] Tad McGeer. Passive Dynamic Walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990.
- [15] Kohei Nakajima, Helmut Hauser, Tao Li, and Rolf Pfeifer. Information processing via physical soft body. *Nature Publishing Group*, 2015.
- [16] C Paetsch, B A Trimmer, and A Dorfmann. A constitutive model for activepassive transition of muscle fibers. *International Journal of Non-Linear Mechanics*, 47:377–387, 2012.
- [17] F. Romero and F. J. Alonso. A comparison among different Hill-type contraction dynamics formulations for muscle force estimation. *Mechanical Sciences*, 7(1):19–29, 2016.

- [18] David Roylance. Engineering viscoelasticity, 2001.
- [19] Benjamin Schrauwen, David Verstraeten, and Jan Van Campenhout. An overview of reservoir computing: theory, applications and implementations. *European Symposium on Artificial Neural Networks*, 2007.
- [20] Darryl G Thelen. Adjustment of Muscle Mechanics Model Parameters to Simulate Dynamic Contractions in Older Adults. 125(February 2003), 2003.
- [21] Arthur van Soest and Maarten Bobert. The contribution of muscle properties in the control of explosive movements. *Biological Cybernetics*, 69:195–204, 1993.
- [22] Qian Zhao, Kohei Nakajima, Hidenobu Sumioka, Helmut Hauser, and Rolf Pfeifer. Spine dynamics as a computational resource in spine-driven quadruped locomotion. *IEEE International Conference on Intelligent Robots and Systems*, pages 1445–1451, 2013.