

Adaptive Filter Model of Cerebellum for Biological Muscle Control with Spike Train Inputs

Emma Wilson¹

¹School of Computing and Communications, Lancaster University

Keywords: Cerebellum; Adaptive Filter; Functional Electrical Stimulation; Muscle Control

Abstract

Prior applications of the cerebellar adaptive filter model have included a range of tasks within simulated and robotic systems. However, this has been limited to systems driven by continuous signals. Here, the adaptive filter model of the cerebellum is applied to the control of a system driven by spiking inputs, by considering the problem of controlling muscle force. The performance of the standard adaptive filter algorithm is compared with the algorithm with a modified learning rule that minimise inputs and a simple Proportional-Integral-Derivative (PID) controller. Control performance is evaluated in terms of the number of spikes, the accuracy of spike input locations, and the accuracy of muscle force output. Results show the cerebellar adaptive filter model can be applied without change to the control of systems driven by spiking inputs. The cerebellar algorithm results in good agreement between input spikes and force outputs and significantly improves on a PID controller. Input minimisation can be used to reduce the number of spike inputs, but at the expense of a decrease in accuracy of spike input location and force output. This work both extends the applications of the cerebellar algorithm, but also demonstrates the potential of the adaptive filter model to be used to improve functional electrical stimulation (FES) muscle control.

1 Introduction

The cerebellum contains up to 80% of all neurons in the human brain, highlighting its importance (Herculano-Houzel, 2010; Azevedo et al., 2009). The structure of the cerebellum is highly uniform, with a discrete cerebellar microcircuit that is repeated across the cerebellar cortex (Ito, 1984; Albus, 1971; Eccles, Ito, & Szentágothai, 1967; Marr, 1969). This uniformity has led to the concept of a ‘cerebellar chip’ where a single cerebellar algorithm is implemented in the control of different tasks (Porrill, Dean, & Anderson, 2013; E. D. Wilson et al., 2021). With the function of each microcircuit determined both by the uniform internal algorithm, but distinguished by the unique external connections. Despite the uniformity of its structure, the cerebellum is implicated in a wide range of tasks. It is traditionally regarded as a motor control structure and strongly associated with adaptive control and skilled movement (Ito, 1984; Albus, 1971; Marr, 1969; Dean, Porrill, Ekerot, & Jörntell, 2010). However, there is accumulating evidence that the cerebellum is additionally involved in sensory and cognitive tasks, including emotion, language and map calibration (Gao et al., 1996; Blakemore, Frith, & Wolpert, 2001; Murdoch, 2010; Baumann et al., 2015; E. D. Wilson, Anderson, Dean, & Porrill, 2019). This range of tasks, carried out by a brain region with such a uniform structure, is remarkable and suggests a highly adaptive, general purpose algorithm is being implemented. Such an algorithm has great potential in the control of robotic and hybrid systems.

The adaptive filter model of the cerebellum (further details provided in the related work section) has been applied to the control of a number of simulated, and experimental robotic systems. Applications include: biological control problems (Dean, Porrill, & Stone, 2002; Porrill & Dean, 2007a, 2007b, 2008), sensory processing problems (Anderson et al., 2012; Dean, Anderson, Porrill, & Jörntell, 2013; E. D. Wilson et al., 2019), and neurobotic tasks (Lenz et al., 2009; Anderson et al., 2010; E. D. Wilson

et al., 2016, 2021). To date these applications of the cerebellar adaptive filter model to a range of different tasks have thus far been limited to robotic and simulated systems driven by continuous signals.

A key difference with biological systems is the way in which information is coded and transmitted. The brain uses spikes to transmit information between neurons. It is thought that neurons encode information either via their firing rate, or the precise timing of spikes, or a combination (Masuda & Aihara, 2002; Yu, Li, & Tan, 2018; Panzeri, Brunel, Logothetis, & Kayser, 2010; Gütig, 2014; Brette, 2015). Although there is debate to whether the rate or temporal code is dominant for processing information, the essential role that spikes play in information transmission is well established. However, to date, the cerebellar adaptive filter model has not been tested in systems driven by spiking inputs. This is the key gap that this contribution aims to address. In this paper, we consider if the adaptive filter model of the cerebellum can be applied without change to the control of a system driven by spike inputs. In particular, the problem of controlling biological muscle using spiking inputs is considered. This work establishes if the cerebellar chip concept of the cerebellar adaptive filter algorithm can be extended to systems activated by spike inputs, thus increasing the generalisability of the adaptive filter algorithm of cerebellar function. In addition, the cerebellar algorithm has promise for hybrid systems control. For example, it could potentially be applied to improve Functional Electrical Stimulation (FES) control, where muscles of those with paralysed or weakened function are stimulated via electrodes (Peckham, Knutson, et al., 2005). To date, a number of control approaches have been utilised for FES control, as summarised in Barbosa, Temporão, & Meggiolaro, 2021. These include classic control techniques such as the PID controller (Rouhani, Same, Masani, Li, & Popovic, 2017); intelligent extensions to the PID controller, where techniques are used to update boudville2018development, qiu2014intelligent or augment (Chen, Chen, Hsiao, Kuo, & Lai, 2005) the PID controller ; other intelligent control techniques, such as neural networks (Griffis, Le, Stubbs, & Dixon, 2021; Sharma, Gregory, Johnson, & Dixon, 2011) and fuzzy controllers (Poboroniuc et al., 2018) and Modern and Robust Control Techniques (Mohammed, Poignet, Fraisse, & Guiraud, 2012; Kirsch, Alibeji, & Sharma, 2017; Oliveira et al., 2017). However, the cerebellar algorithm has not been considered. The biological plausibility, simplicity and adaptive nature of the cerebellar adaptive filter algorithm make it a potential candidate for use in FES control.

This contribution assess whether we can apply the cerebellar adaptive filter algorithm, without change, to the problem of controlling biological muscle. To do this, we use a simulated model of muscle that takes a spike train as an input and gives the isometric muscle force as an output. An integrate-and-fire model is used to generate spiking inputs from a continuous signal. These spiking inputs are used as an input to the muscle model. The cerebellar adaptive filter, in combination with a fixed brainstem controller, is used to control the force response to match a desired response. We evaluate the performance, both in terms of the estimated spike input numbers and location, and in terms of the accuracy of the controlled muscle force. The performance of the cerebellar controller is compared to using a PID feedback controller. The PID feedback controller was chosen as a comparator for consistency with other papers on advanced FES control. Although PID controllers have generally been shown to under perform in comparison to other more intelligent and adaptive control techniques (Boudville et al., 2018; Qiu et al., 2014; Wannawas, Shafti, & Faisal, 2022), it is common practice to compare developed FES control algorithms to the basic PID controller as a benchmark. There are numerous other control algorithms that have been applied to FES control, making the choice of controller to use in comparisons difficult and somewhat arbitrary, so we choose to use a PID controller for comparison, as is standard.

Results show that the cerebellar algorithm performs better in terms of both the accuracy of spike input locations and the accuracy of muscle force output. We show that the cerebellar algorithm can also be easily adapted to reduce inputs to the muscle. The amount of input is critically important to minimise in FES applications, which are highly effected by muscle fatigue (Ibitoye, Hamzaid, Hasnan, Abdul Wahab, & Davis, 2016). Including input minimisation shows that for similar output force errors, the number of spike inputs could be reduced by around 20% when comparing the cerebellar adaptive filter algorithm to a conventional PID controller. This work demonstrates the promise and potential application of the cerebellar adaptive filter model to FES control. Due to the close synergies with

the biological counterpart, a cerebellar inspired algorithm to improve FES control has great potential. The work also extends the generalisability of the adaptive filter model of the cerebellum and is the first step in applying the cerebellar algorithm to systems driven by spiking inputs.

2 Related Work

Key existing approaches to modelling cerebellar function and their usages are described in this section.

2.1 Computational Approaches to Modelling Cerebellar Function

A number of computational approaches have been suggested and utilised to model cerebellar function. Generally, these approaches can be categorised as: (i) *Descriptive models* which focus on neural dynamics and are often based on compartmental models (Bower & Beeman, 2012; De Schutter & Bower, 1994; Gleeson, Steuber, & Silver, 2007); (ii) *Look-up tables*, such as the cerebellar model articulation computer (CMAC) (Albus, 1971), this historic approach is now little used; (iii) *Olivary models*, where cerebellar output is driven by inputs from the inferior olive to the Purkinje cell (Jacobson, Rokni, & Yarom, 2008; Torben-Nielsen, Segev, & Yarom, 2012) and; (iv) *Marr–Albus models* (Marr, 1969; Albus, 1971), which include many variations (Kawato & Gomi, 1992; Schweighofer, Arbib, & Dominey, 1996; Medina & Mauk, 2000), and in particular, the adaptive filter model of cerebellar function (Fujita, 1982; Dean et al., 2010) and cerebellum-inspired spiking neural network models (Luque, Garrido, Carrillo, Tolu, & Ros, 2011; Vijayan & Diwakar, 2022).

2.2 Adaptive Filter Model of the Cerebellum

The adaptive filter model of the cerebellum is emerging as a leading computational model of cerebellar function, capable of representing both forward and inverse models (Porrill et al., 2013).

The adaptive filter model has been utilised for motor control by embedding it in a recurrent loop architecture. In this architecture, a desired reference signal is passed to a fixed element B (representing the brainstem) to give a motor command. This motor command is used as an input to the adaptive cerebellar element (C). The output of C is used as a correction to the input to B (as in Figure 1e). The advantages of such a recurrent connectivity, in contrast with a forward architecture, are discussed in (Porrill & Dean, 2007b) and (Porrill et al., 2013). To summarise, using the recurrent architecture enables the available sensory error to be used directly to drive adaptation, as opposed to needing to estimate the motor error, which is usually unavailable. This motor error signal is usually unavailable as it requires having the desired motor command as well as the actual motor command. Although, in the simulations presented here the desired motor signal is available as we know the plant model exactly, this is not commonly the case. For example, if the controller was being used for FES application the desired motor command is the one that would produce the exact force or movement required so that the plant output matches the desired output. However, to know this, we would need an exact model of the plant, which we are unlikely to have. Especially for non-linear, biological, real-world systems.

A key concept of the adaptive filter model of cerebellar function, is the chip metaphor. That is, the model can be implemented without change in a number of different tasks, with the different functionality afforded by different external connections, but with the internal algorithm kept the same. As stated in the introduction, this model has been applied to a number of different tasks within simulated and robotic systems.

In summary, advantages of the adaptive filter model of cerebellar function include: (i) that it is a simple homogenous representation which is easy to implement and analyse; (ii) the learning rule is biologically plausible; (iii) when used within a biologically inspired recurrent architecture, rather than use the unavailable motor-error signal to drive adaptation, it uses the available sensory error.

The adaptive filter model, used in this contribution, is a high-level functional systems model of the cerebellum. How in practice the signals are generated by the neural system at the spiking level is not

considered here. However, it has been shown that the granular-layer recursive neural network can in principle generate the outputs necessary to implement an adaptive filter (Rössert, Dean, & Porrill, 2015). There is also existing evidence supporting the idea that the granule layer performs complex signal decomposition and the computational principles that underlie many functional models of the adaptive filter (Dean et al., 2010).

2.3 Spiking Cerebellar Models

Cerebellum-inspired spiking neural network models have been developed and applied to several different tasks (Fruzzetti et al., 2022; Vijayan & Diwakar, 2022; Vijayan et al., 2017; Luque et al., 2011). Applications of such models include pattern classification and trajectory prediction, particularly robot arm trajectory prediction. These spiking models consider how signals are reconstructed from the neuronal behaviour. In the model input stimuli (Mossy Fibres (MF)) and cerebellar neurons (Granule Cells (Grc), Golgi Cells (CoC) and Purkinjia Cells (PC)) are reconstructed using extended integrate and fire models. For each feature, a set of n -neurons is used. For example, in (Vijayan & Diwakar, 2022) for a dataset with 4 features the network consisted of 28MFs, 371 GrCs, 1 GoC and 1 PC. As a learning mechanism, error based learning, as in the adaptive filter is used in which the predicted output is compared to actual output to find errors. However, in contrast, these spiking models have considered using both sensory and motor errors (Luque et al., 2011).

The core ideas behind adaptive filter models and spiking cerebellar models are not dissimilar. The key difference between the models being the level of detail in the implementation. The adaptive filter model, used in this contribution, is a high-level functional system models of the cerebellum. In comparison, spiking cerebellar models, consider the low level behaviour of the network. A comparison between the adaptive filter models and spiking cerebellar models with their advantages and disadvantages is provided in the discussion.

3 Methodology

The methodology is split into two main sections. The first of these describe the background and existing work that the methodology is based upon and the second extensions and the algorithm developed and implemented in this contribution as well as the parameters and settings used.

3.1 Existing Techniques

This section describes the existing methods that are utilised to develop an adaptive cerebellar algorithm that can be used to control systems driven by spiking inputs. It is split up into several parts. Section 3.1.1 describes the muscle model, 3.1.2 the adaptive filter model of the cerebellum, 3.1.4 the method for generating spikes and 3.1.4 the method for evaluating the accuracy of timing of spike inputs.

3.1.1 Muscle Model

In order to simulate the muscle force response, Wilson *et als.* (E. Wilson, Rustighi, Newland, & Mace, 2012) simplified adapted model was used. This model takes a spike train $u(t)$ as an input and produces the isometric force response $F(t)$ as an output. The model was chosen as it has some physiological basis and explanation, but is simple to implement and has been fully parameterised. It also performed well in a comparison of existing models that convert an input spike train into an isometric force output. Using the model, the muscle force is described as

$$v(t) = \sum_{j=1}^{n_p} \delta(t - t_j) \quad (1)$$

$$\dot{C}_N(t) + \frac{C_N(t)}{\tau_c} = v(t) \quad (2)$$

$$x(t) = \frac{C_N(t)^m}{C_N(t)^m + \kappa^m} \quad (3)$$

$$\dot{F}(t) + \frac{F(t)}{\tau_1} = Ax(t) \quad (4)$$

Here $v(t)$ is an input with n_p input pulses located at times t_j , and $F(t)$ the muscle force output. The model has 5 parameters and parameter values were set according to the fit to experimental data from the locust hind leg extensor muscle from (E. Wilson et al., 2012) ($\tau_c = 0.071$, $\tau_1 = 0.13$, $m = 2.5$, $\kappa = 0.75$, $A = 7.4$).

3.1.2 Adaptive Filter Model of Cerebellum

This section describes the adaptive filter algorithm that is thought to be implemented by each instance of the repeated cerebellar microcircuit. This algorithm remains unchanged from prior versions of the cerebellar adaptive filter model of cerebellar function. Figure 1a), b) gives a highly simplified representation of the cerebellar microcircuit and its interpretation as an adaptive filter.

This model has one output, the Purkinje cell output $z(t)$. This is modelled as a weighted combination of parallel fibre signals

$$z(t) = \sum_{i=1}^{n_w} w_i(t)p_i(t) = \mathbf{w}(t)^T \mathbf{p}(t) \quad (5)$$

here, n_w gives the number of weighted parallel fibre connections, $w_i(t)$ the i^{th} weight at time t , $p_i(t)$ the i^{th} parallel fibre signal at time t , $\mathbf{w}(t) = [w_1(t) \ w_2(t) \ \dots \ w_n w(t)]^T$ a $n_w \times 1$ vector of parallel fibre weights and $\mathbf{p}(t) = [p_1(t) \ p_2(t) \ \dots \ p_n w(t)]^T$ a $n_w \times 1$ vector of parallel fibre signals.

The parallel fibre signals are obtained from filtering the mossy fibre inputs ($u(t)$) to the cerebellum at time t . The filtering represents processing by the granule cell layer, in which the adaptive filter model is represented by a bank of time-invariant basis filters. We model these basis filters as a bank of n_w alpha filters with log spaced time constants between T_{min} and T_{max} . The Alpha basis filters are given in the Laplace domain as

$$G_i(s) = \frac{1}{(T_i s + 1)^2} \quad i = 1, 2, 3, \dots n_w \quad (6)$$

where G_i denotes the i^{th} Alpha basis filter, n_w the total number of basis filters, and T_i the time constant of the i^{th} filter. Details of how the filter parameters values were set are provided in 3.2.3 and Figure 1c) shows the impulse response of the alpha basis filters.

Adaptive filter weights are learnt (from initial values of zero) using the decorrelation learning rule (Sejnowski, 1977) (identical to the LMS rule from adaptive control theory (Widrow & Stearns, 1985)).

$$\dot{w}_i(t) = -\beta e(t) \bar{p}_i(t) \quad (7)$$

where β is the learning rate and is a constant. The error signal carried by the climbing fibre inputs to the cerebellum at time t is denoted $e(t)$, and $\bar{p}_i(t)$ is the parallel fibre signal $p_i(t)$ filtered through reference model M (see (E. D. Wilson et al., 2015) and Section 3.2.1 for more detail on using a reference model). The error signal for our control problem is described in the section 3.2.1. A learning rule, which penalised inputs, was also considered. The modified learning rule is,

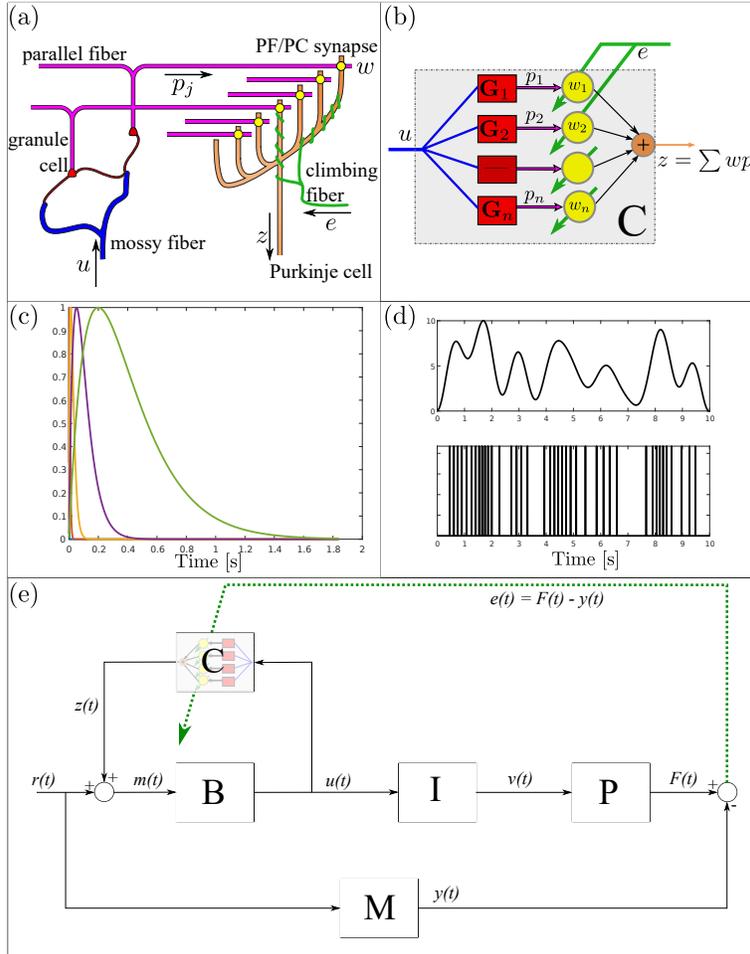


Figure 1: Cerebellar control, adapted from (E. D. Wilson et al., 2015). (a) Highly simplified diagram of the cerebellar microcircuit. (b) The cerebellar microcircuit as an adaptive filter. Mossy fibre inputs are passed through a bank of fixed filters (basis functions) and weighted and recombined to give the Purkinje cell output. Weights are learned using the error signal, (c) Basis function impulse responses (d) Example input (top plot) and output (bottom plot) of the integrate-and-fire neuron model (e) Control architecture with the control provided by both a fixed brainstem element **B** and an adaptive cerebellar element **C**. A reference model **M** is used to specify the desired response of the system, **P** represents the muscle model and **I** the integrate and fire model

$$\dot{w}_i(t) = -\beta e(t)\bar{p}_i(t) - \lambda u(t)\bar{p}_i(t) \quad (8)$$

where λ is another learning rate and is a constant. When $\lambda = 0$, the rule is equivalent to Equation (7).

3.1.3 Spike Generation

Spikes are generated according to a simple leaky integrate and fire neuron model. Although more detailed neuronal models exist, the exact nature of the spike generation is not the focus of this paper. The leaky integrate and fire model was used as it is a widely used, simple, effective method for generating spiking inputs.

In the leaky integrate and fire model, when the membrane potential exceeds a threshold, a spike is fired and the membrane potential is set back to g_{rest} . Here, we use a resting potential of zero ($g_{\text{rest}} = 0$), with the leak conductance set to 1. The leaky integrate and fire model used here is governed by the following differential equation

$$T_{LI}\dot{g}(t) + g(t) = u(t) \quad (9)$$

$$v(t) = \begin{cases} 0 & \text{if } g(t) < \text{th} \\ 1/dt & \text{if } g(t) > \text{th and reset } g(t) \text{ to zero.} \end{cases} \quad (10)$$

Where T_{LI} defines the leaky integrator time constant, $u(t)$ defines the input to the leaky integrate and fire filter and $g(t)$ the membrane potential. Once the membrane potential exceeds a threshold, a spike is fired and the membrane potential is set back to zero. We use $v(t)$ to represent the spiking input and a spike with magnitude $1/dt$ and time dt . The transformation from continuous input $u(t)$ to spiking input $v(t)$ for an example signal is shown in Figure 1d).

3.1.4 Spike Timing Evaluation

The Spike Time Tiling Coefficient (STTC) was used to evaluate how similar the actual spiking input (obtained during control) was to the desired spiking input. This metric effectively quantifies the degree of correlation between the two spike trains (desired and actual spike input trains). Quantifying correlations between spike trains is challenging both due to the fact that spiking is sparse in comparison to the sampling frequency and because correlated neurons may not fire precisely together, but still fire at very similar times, meaning correlations need to be defined with reference to a timescale within which spikes can be said to be correlated (Cutts & Egle, 2014). This makes traditional correlations metrics such as Pearson's correlation coefficient unsuitable. Instead, a number of options exist for detecting correlations between spike trains, these are detailed and compared by (Cutts & Egle, 2014), who find that the STTC has several advantages. It was therefore chosen to use the STTC to compare spike trains in this work. The STTC is calculated as,

$$STTC = \frac{1}{2} \left(\frac{PA - TB}{1 - PA * TB} + \frac{PB - TA}{1 - PB * TA} \right) \quad (11)$$

where PA is the proportion of spikes from train 1 that are within $\pm sdt$ of any spike from train 2, divided by the total number of spikes in train 1. PB is the equivalent metric, but for the spikes in train 2. TA is the proportion of total recording time that lies within $\pm sdt$ of any spike in train 1. Again, TB is the equivalent metric for train 2. Where TA=PB=1, or TB=PA=1, this would result in 0/0. These instances are replaced with 1, because in this case, every spike from the train is within $\pm sdt$ of a spike of the other train. The STTC metric is always a value between -1 to 1, with positive values indicating correlations between the two spike trains and negative values indicating there is less correlation than expected by chance. A STTC of 1 indicates that the locations of spikes in the two trains are equivalent (within $\pm sdt$). In this evaluation $sdt = 0.02s$.

3.2 Developed Algorithm

Section 3.2.1 describes the overall cerebellar adaptive filter control algorithm that utilises the parts described in Sections 3.1.1 to 3.1.3 and works to control a muscle model driven by a spiking input. A PID control scheme is described in 3.2.2 which is used as a comparison to the cerebellar adaptive filter controller. The settings used for simulations are provided in section 3.2.3, this includes, the data used both to train the adaptive filter weights and to evaluate the control performance, the parameters used in simulations and a summary of evaluation metrics.

3.2.1 Adaptive Filter Controller for Muscle

In this section, the overall control architecture and algorithm is described. This uses the muscle model, the cerebellar adaptive filter algorithm and leaky integrate and fire model described in the prior sections. The external connections to the adaptive filter model which enable the desired control functionality are described in this section. Figure 1e) describes the overall control architecture and shows how the different elements link together.

A reference model, M is used to generate the desired force profile from reference input. This is necessary to ensure stable control and that the plant inverse (that our controller learns to approximate) is proper (See (E. D. Wilson et al., 2015) for further details). The reference model used here is defined as,

$$M(s) = \frac{1}{(\tau s + 1)^2} \quad (12)$$

with time constant $\tau = 0.1$.

The fixed brainstem controller, B and adaptive cerebellar filter, C provide inverse control of the muscle plant, P . The fixed brainstem filter, B , was set to be an approximate compensator for the muscle plant, with the cerebellar adaptive filter fine-tuning the control, as in the biological system. The brainstem filter was designed to provide perfect compensation for an approximate plant model \hat{P} , such that

$$B(s) = \hat{P}(s)^{-1}M(s) \quad (13)$$

where $\hat{P}(s)^{-1}$ is the inverse of the approximate plant model. For the approximate plant model, the linear part of the muscle model was used, such that

$$\hat{P}(s) = \frac{\hat{A}}{(s + 1/\hat{\tau}_c)(s + 1/\hat{\tau}_1)} \quad (14)$$

where the estimated plant model parameters (\hat{A} , $\hat{\tau}_c$, $\hat{\tau}_1$) are based on the muscle model parameters, with a random noise added, such that $\hat{p} = p + pN$. Here N is a pseudorandom value drawn from the standard uniform distribution on the interval (0,0.5) and \hat{p} and p used to represent estimated and actual plant model parameters respectively. The resultant fixed brainstem controller is,

$$\hat{B}(s) = \frac{(s + 1/\hat{\tau}_c)(s + 1/\hat{\tau}_1)}{\hat{A}(\tau s + 1)^2} \quad (15)$$

The error signal, $e(t)$ is given as the difference between the desired force $y(t)$ (reference signal $r(t)$ filtered through reference model M) and actual force $F(t)$. This error signal is used to adjust the adaptive filter weights, as described in Equation (8).

The overall algorithm is described in pseudocode in Table 3.2.1. All codes and data used to produce the results in this paper can be found here: <https://osf.io/3s5rn/>.

—p10cm p5cm— Cerebellar Algorithm Description. Here, the notation x_k is used to denote signal x at time step k . Discrete time filters are used in the implementation, where for example, $M(q)$ is the discrete version of the continuous filter $M(s)$ calculated using zero-order hold and q is the shift operator ($qu_k = u_{k+1}$)

Initialise Inputs:

Set initial motor command, and sum signals to zero $u_0 = 0, us_0 = 0$
Set vector of first weights to zero $\mathbf{w}_1 = \mathbf{0}$
Set spiking input vector v to zero $v = \mathbf{0}$

Define reference signal r at each time step $r = \text{user defined}$

for each time step k : for $k=1$ to length(r)

Filter reference signal r_k through reference model filter \mathbf{M} $y_k = \mathbf{M}(q)r_k$
Filter motor command through bank of i basis filters \mathbf{G}_i $p_{i,k} = \mathbf{G}_i(q)u_{k-1}$
Calculate cerebellar output - sum weighted filter outputs $z_k = \mathbf{w}_k^T \mathbf{p}_k$
Sum reference signal and cerebellar output $(r_k + z_k)$
Filter sum through brainstem \mathbf{B} $u_k = \mathbf{B}(q)(r_k + z_k)$
Keep leaky running sum us_k of motor command u_k $us_k = u_k + (1 - dt/T_{LI})us_{k-1}$
if running sum us_k exceeds threshold if $us_k > 1/dt$:
Spike input on v_k $v_k = 1/dt$
Reset input sum if threshold exceeded $us_k = 0$
end if end

Get muscle force response F_k to spike train input, v_k Implement Eqs (2)-(4).
Calculate error signal e_k $e_k = F_k - y_k$
Filter parallel fibre signals through ref model $\bar{\mathbf{p}}_k = \mathbf{M}(q)\mathbf{p}_k$
Update cerebellar adaptive filter weights $\mathbf{w}_{k+1} = \mathbf{w}_k - \beta e_k \bar{\mathbf{p}}_k - \lambda u_k \bar{\mathbf{p}}_k$

11

Cerebellar Algorithm Description. Here, the notation x_k is used to denote signal x at time step k . Discrete time filters are used in the implementation, where for example, $M(q)$ is the discrete version of the continuous filter $M(s)$ calculated using zero-order hold and q is the shift operator ($qu_k = u_{k+1}$)

3.2.2 PID Controller

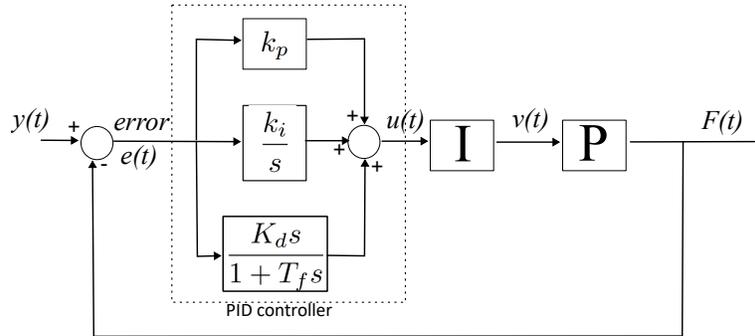


Figure 2: Control architecture for the PID controller, \mathbf{P} represents the muscle model and \mathbf{I} the integrate and fire model

The PID controller has been widely used in FES control studies as a benchmark comparison to other control algorithms (Boudville et al., 2018; Qiu et al., 2014; Wannawas et al., 2022). Here, we use heuristics to hand tune the PID parameters one at a time, particularly focusing on the response to a single pulse. If the PID parameters are too large, the amplitude of response to a single pulse is

amplified. We increased PID parameters, starting with k_p , then k_i , then k_d in increments of 10 until just before the amplitude of the response to any of the single pulses was over 50% larger than the desired amplitude to determine PID parameters. The PID parameters used were $k_p = 360$, $k_i = 60$, $k_d = 10$, with a first order filter (time constant $T_f = 10$) applied to the derivative of the error. The control architecture for the PID controller is given in Figure 2.

The performance of the PID controller described here is compared to the cerebellar adaptive controller using the evaluation metrics detailed in the following section. The same training and test data sets were used as for the cerebellar controller, with the training set used to tune the PID parameters.

3.2.3 Simulation settings

The data, settings, parameter values and evaluation metrics used in simulations are described here.

Data

The muscle model is non-linear, with the force response saturating as the neural input increases. The response to a single, well separated input spike differs to the response of a spike with other closely spaced input spikes, with saturation occurring when many input pulses are close together. Therefore, any controller developed to control the muscle force must be able to operate over the full range of possible inputs, ranging from closely spaced to well separated spikes over a full spectrum of possible spike frequencies. Therefore, it is essential that the training data, used to train the model and learn adaptive filter weights contains such a range of input pulses, ensuring the learnt controller works well under all possible input conditions.

It was decided to use the same training and test data sets as were used to estimate the muscle model in E. Wilson et al., 2012. The reason for this is this data set has already been shown to be representative of a range of inputs, enabling good fits over a full range of test data, whilst keeping the overall size of the data set as small as possible. Furthermore, the test data includes physiologically relevant signals. An additional test signal, with input spikes occurring at pseudorandom times, was also used. This random input was generated as 384 spikes, with interpulse intervals randomly assigned to values between $2dt$ and 0.5s. Input spikes, used to generate the desired muscle force, and corresponding modelled force outputs, are shown in Figure 3

The desired muscle force was obtained by passing spike inputs to the muscle model described in Equations (2) to (4). For the cerebellar adaptive filter control (see Figure 1e) the desired muscle force is the reference input $r(t)$ filtered through the reference model M (Equation (12)). The reference input $r(t)$ was obtained from the desired muscle force by first using a second order low pass Butterworth filter with cut-off frequency 12.5Hz to smooth the signal, then passing the signal back through M

The training data set contains 156 input spikes and 42.2s or 91766 samples of data, during training this data set is repeated 4 times. The test1 data set contains 127 input spikes and 55.6s or 120868 samples of data. The test2 data set contains 202 input spikes and 74.4s or 161843 samples of data. The test3 data set contains 384 spikes and 100s or 217392 samples of data.

Parameters

Hyperparameters were estimated using the test data set and a heuristic approach where different values were tested in simulation. The aim was not finding optimal parameter values, but instead parameter values that work well enough. In particular, the aim was to find parameters that gave stable learning and small errors (final RMS errors < 0.2 without input penalisation and < 0.4 with input penalisation) within a reasonable time scale (learned within four repeats of the training data). Parameters for the cerebellar adaptive filter algorithm that were used in the simulations presented here are defined in Table 1.

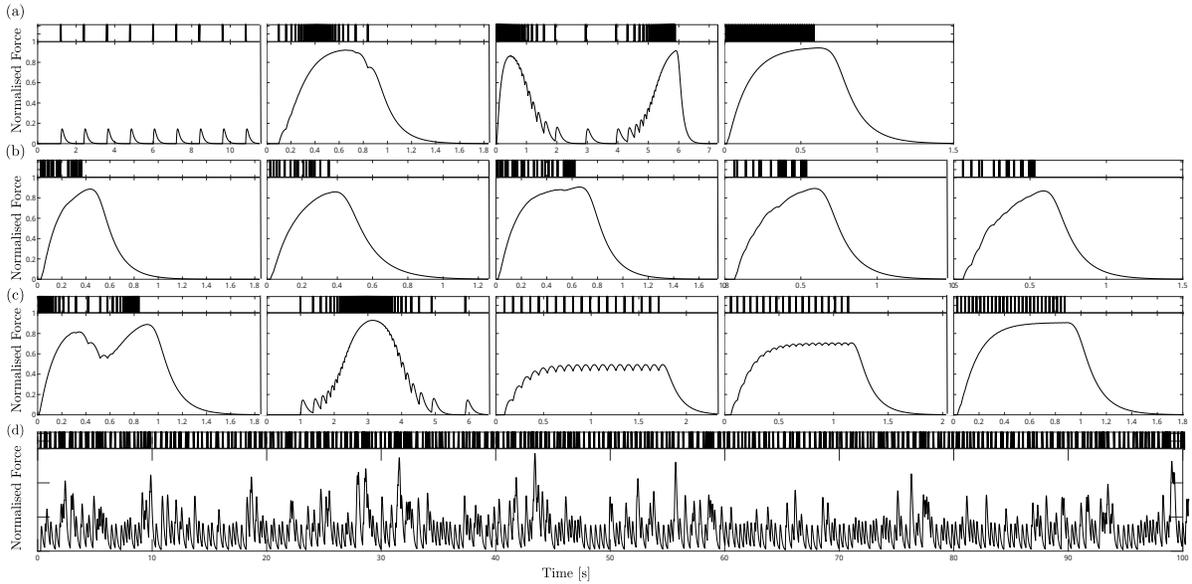


Figure 3: Data used to learn controller parameters and to test the performance of the learned controller. Note that all test data also contained the train of single pulses between each pulse train (a) Training data, (b) Test1 data: physiological, kick type test data, (c) Test2 data: constant pulse train and increasing/ decreasing frequency pulse train test data, (d) Test3 data: random pulse train test data set

Table 1: Summary of parameters used in control algorithms

Parameter	Value(s)	Brief Description
dt	$4.6e - 04$	Sampling time
τ	0.1	Reference Model Time Constant
n_w	5	Number of basis filters
T_{min}	$9.2e - 4$	Min basis time constant
T_{max}	0.2	Max basis time constant
β	$5e - 7$	Error learning rate
γ	0 and $4e - 9$	Input penalisation learning rate
Muscle model parameters	$\tau_c = 0.071$, $\tau_1 = 0.13$, $m = 2.5$, $\kappa = 0.75$, $A = 7.4$	Parameters for simulating muscle model
th		Threshold for determining whether to spike
T_{LI}	1	Leaky integrator time constant
sdt	0.02	For STTC calculation to give proportion of spikes that are within $\pm sdt$ of the desired spikes
k_p	360	Proportional gain in PID controller
k_i	60	Integral gain in PID controller
k_d	10	Derivative gain in PID controller
T_f	10	Time constant used in derivative of error filter in PID controller

Evaluation Metrics

The difference between both the desired and actual output and input was evaluated. In addition, due to the importance of fatigue and minimising inputs during FES, the total number of spikes was also calculated. The accuracy of the estimated output force was assessed using the RMS error given as

$$F_{RMSE} = \sqrt{\frac{1}{N} \sum_{i=0}^N (F(i) - y(i))^2} \quad (16)$$

The STTC (as described in section 3.1.4) was used to determine how similar the actual spiking input (obtained during control) was to the desired spiking input.

4 Results

4.1 Cerebellar Learning

The training data set (see Fig 3a) was not sufficiently long to ensure that learned weights converged to reasonably consistent values. Therefore, this training data was repeated four times to allow enough time for learning. Examples of signals obtained during cerebellar learning (with $\beta = 5e - 7$ and $\lambda = 0$) are given in Figure 4. The initial errors are large, indicating the approximate brainstem controller is not a very good controller. However, as cerebellar adaptive filter weights are learned over time, weights increase to an approximate steady value and errors reduce, giving good agreement between the desired and actual muscle force.

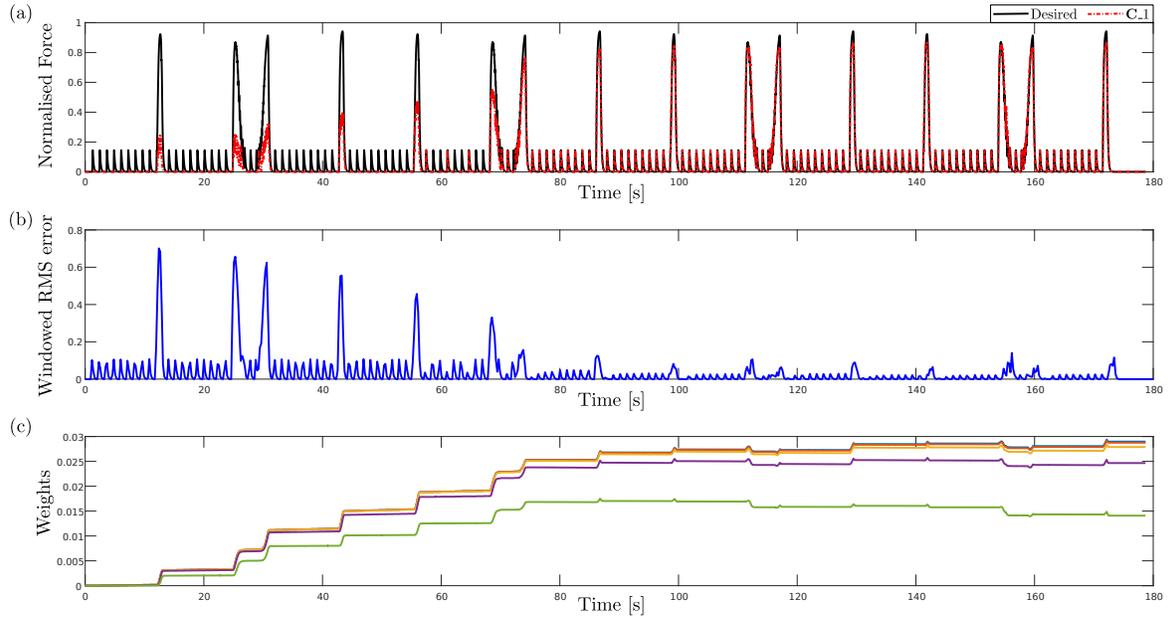


Figure 4: Results during cerebellar controller learning over time, with C_1 representing cerebellar learning with rates $\beta = 5e - 7$ and $\lambda = 0$.

The force responses and input spikes from the training data at the start and end of learning are shown at a larger time resolution in Figure 5. Here, results from cerebellar learning are shown with (C_2) and without (C_1) input penalisation. It is evident that for each controller when the force is larger the force response is slightly reduced from desired, with, as expected, this worse for the controller

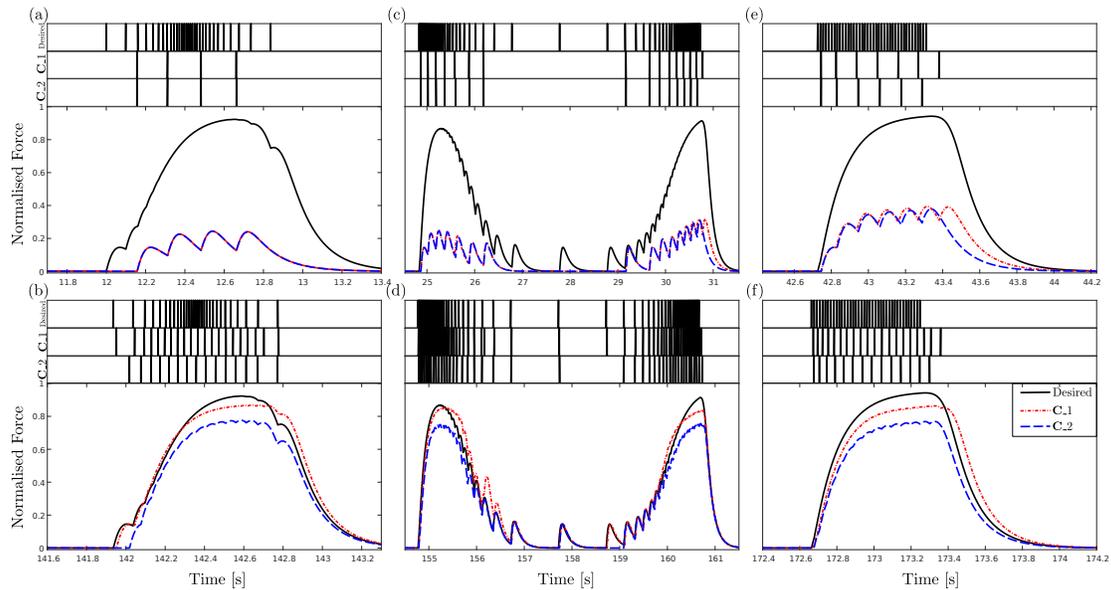


Figure 5: Cerebellar control algorithm response at the start (top) and end (bottom) of learning. Plots (a), (c) and (e) show the input pulses and normalised force response towards the start of learning; (b), (d) and (f) give the same desired input pulses and normalised force responses, but after cerebellar learning

with input penalisation (C_2). Some differences between actual and desired input spikes can also be seen, particularly at higher frequencies, with the controlled inputs generally more spaced out than the desired inputs.

4.2 Controller Evaluation

Control results for the cerebellar algorithm, both with $\lambda = 0$ (standard learning rule) and with $\lambda = 4e - 9$ (learning rule penalises large inputs) for the learned cerebellar controllers (e.g. weights fixed at those learned using the training data) are given here alongside PID control results. Overall metrics for each controller are summarised in Table 2. The cerebellar algorithm with $\lambda = 0$ gives the smallest errors and the best match between the desired and actual spike inputs. The input penalisation factor ($\lambda = 4e - 9$) was set as this gives similar performance to the PID controller in terms of overall errors. However, the penalised cerebellar controller uses fewer spikes (reduction of $\sim 20\%$) in more similar locations to the desired inputs in comparison to the PID controller. In comparison, the algorithm with $\lambda = 0$ uses a similar number of spikes to the PID controller, but these are more accurately located.

Control results are shown in Figures 6 to 8. In the response to single pulses (Fig. 6), both the PID controller and C_1 (cerebellar controller without input penalisation) after learning approximate the spike locations reasonably well. It is not evident from the resolution of the figure, but in general PID spikes are slightly delayed in comparison to the desired response. Results for the controller C_2 (cerebellar controller with input penalisation) show that occasionally a spike is not fired.

The controller inputs and force response to Test1 (physiological, kick type inputs), for each controller is shown in figure 7, with the inputs and force response for Test2 data shown in figure 8. In general, the cerebellar algorithm C_1 slightly over-estimates the force for lower forces, but under-estimates the higher forces. This is likely due to the fact that the muscle response is non-linear, but the controller used here is linear. The cerebellar algorithm C_2 and PID controller slightly underestimate all forces.

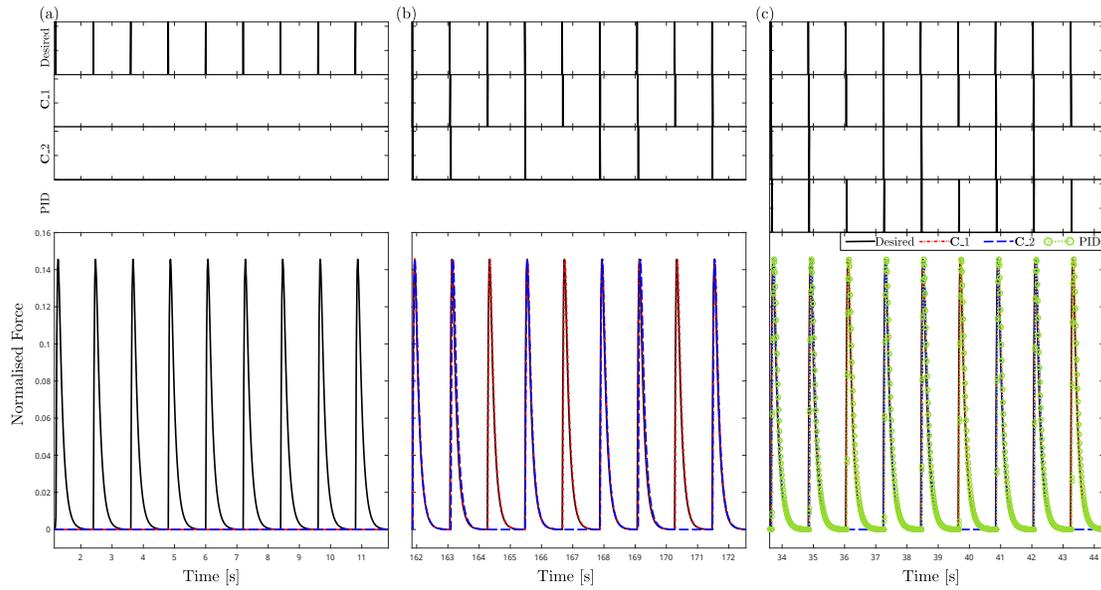


Figure 6: From fitting to well separated single twitch pulses. The top plots show the locations of input spikes (desired and actual for each different controller) and the bottom the force response. Results from three controllers are plotted, with C_1 representing cerebellar learning with rates $\beta = 5e - 7$ and $\lambda = 0$; C_2 representing cerebellar learning with rates $\beta = 5e - 7$ and $\lambda = 4e - 9$; and PID representing a PID controller with $k_p = 360$, $k_i = 60$, $k_d = 10$. (a) Cerebellar controllers at the start of learning, (b) Cerebellar controllers at the end of learning, (c) Example response of all controllers with cerebellar learned weights fixed

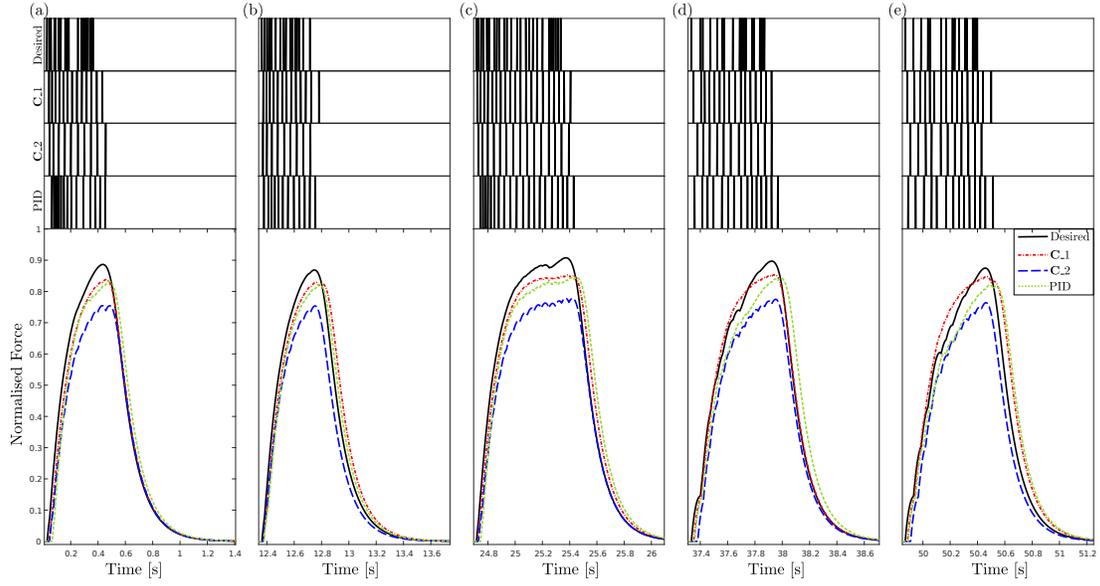


Figure 7: Example controller response to desired force generated from kick type inputs (Test1). The top plots show the locations of input spikes (desired and actual for each different controller) and the bottom the force response. Results from three controllers are plotted, with C_1 representing cerebellar learning with rates $\beta = 5e - 7$ and $\lambda = 0$; C_2 representing cerebellar learning with rates $\beta = 5e - 7$ and $\lambda = 4e - 9$; and PID representing a PID controller with $k_p = 360$, $k_i = 60$, $k_d = 10$.

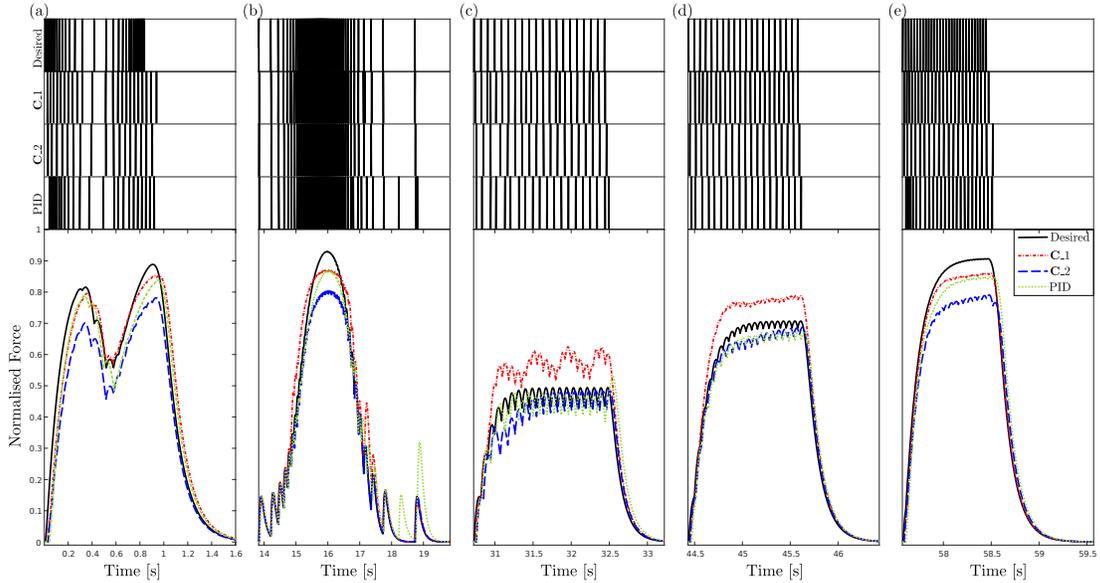


Figure 8: Example controller response to desired force generated from other test data (Test2). The top plots show the locations of input spikes (desired and actual for each different controller) and the bottom the force response. Results from three controllers are plotted, with C_1 representing cerebellar learning with rates $\beta = 5e - 7$ and $\lambda = 0$; C_2 representing cerebellar learning with rates $\beta = 5e - 7$ and $\lambda = 4e - 9$; and PID representing a PID controller with $k_p = 360$, $k_i = 60$, $k_d = 10$.

Controller Settings	Data set	F_{RMSE}	STTC	Total spikes
Cerebellum $\beta = 5e - 7$ $\lambda = 0$	Test1	0.017	0.91	114
	Test2	0.027	0.88	199
	Test3	0.018	0.96	386
	All	0.021	0.93	699
Cerebellum $\beta = 5e - 7$ $\lambda = 4e - 9$	Test1	0.034	0.57	84
	Test2	0.035	0.57	152
	Test3	0.053	0.72	317
	All	0.043	0.68	553
PID $k_p = 360$ $k_i = 60$ $k_d = 10$	Test1	0.027	0.47	112
	Test2	0.030	0.35	188
	Test3	0.066	-0.07	393
	All	0.048	0.16	693

Table 2: Controller performance for different test data and controllers

Figure 9 gives an example of part of the response to Test3 (random inputs). The agreement between force output and spike input for the controller C_1 is very good. This is supported by the large STTC and small error in fit. Fits for C_2 and the PID controller are less good, with the PID spike inputs in particular not being well-placed and giving a poor match with the desired inputs. In fact, the STTC indicates no correlation between the desired and actual spikes using the PID controller. From visual inspection, there does appear to be some correlation between the spikes. However, due to the way the STTC metric works, any pulses have to be within certain bounds ($\pm 0.02s$) to be considered as correlated.

5 Discussion

Results show that the cerebellar algorithm can be applied without change to the control of systems with spiking inputs. They also show how well such a controller can place input spikes to match a desired input response. The performance of the cerebellar algorithm with and without input penalisation and a PID controller was compared. The best performance, in terms of matching the desired force response and desired input, is achieved with the cerebellar algorithm without input penalisation. This algorithm does result in a large number of spikes (albeit ones that match the desired spikes). The input can be reduced by including input penalisation in the cerebellar algorithm. However, this results in slightly higher errors, caused by a corresponding decrease in the force output. The PID controller is the worst of the three tested, giving comparable errors to the cerebellar controller with input penalisation, but with a larger number of spike inputs.

5.1 Adaptive Control

The cerebellar adaptive filter algorithm is an adaptive controller. Using adaptive control makes sense for application to biological muscle control, as the muscle response can change over time, for example due to fatigue. However, there are also risks to using adaptive control as it can be harder to guarantee stability, convergence and robustness. Limits on the adaptation and controller design can help to improve these guarantees for adaptive control algorithms.

It has previously been shown that errors can be at least halved by using adaptive PID parameters in a FES situation (Qiu et al., 2014). Hence, it could be argued that a fixed PID controller is not a good comparator. However, it was chosen as a benchmark, as has been done in a number of prior studies. Even if PID errors were halved, they would still be comparable to the cerebellar algorithm.

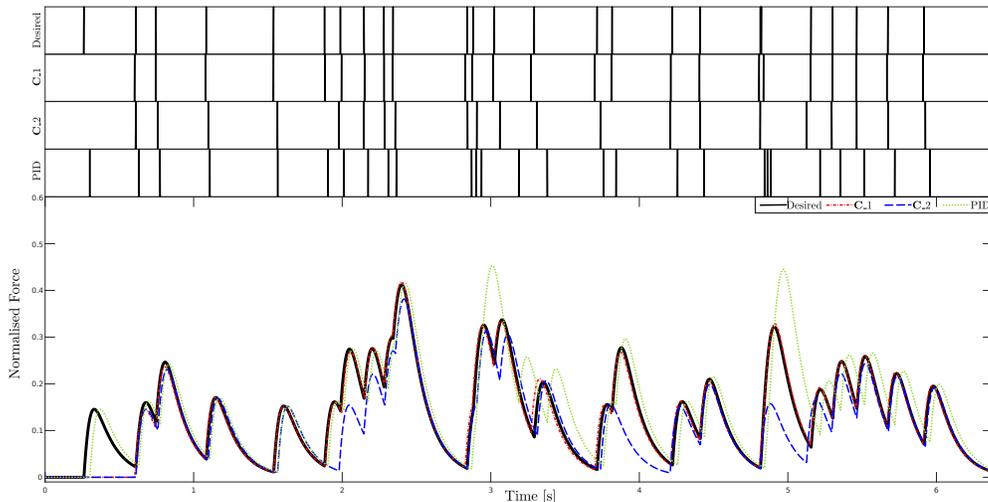


Figure 9: Example controller response to desired force generated from random pulse inputs (Test3). The top plots show the locations of input spikes (desired and actual for each different controller) and the bottom the force response. Results from three controllers are plotted, with C_1 representing cerebellar learning with rates $\beta = 5e - 7$ and $\lambda = 0$; C_2 representing cerebellar learning with rates $\beta = 5e - 7$ and $\lambda = 4e - 9$; and PID representing a PID controller with $k_p = 360$, $k_i = 60$, $k_d = 10$.

Also, in this study the muscle parameters were kept fixed so the dynamic response due to fatigue is not considered.

5.2 Cerebellar Algorithm Extensions

The cerebellar algorithm used here is relatively simple and was chosen to show that a basic cerebellar adaptive filter could be applied to improve control of a system driven by spiking inputs. A number of extensions have already been proposed for this algorithm for systems driven by continuous inputs. These include, transfer of learning (E. D. Wilson et al., 2016), using non-linear basis (E. D. Wilson et al., 2016) and orthonormalisation (E. D. Wilson et al., 2015). Investigating the use of these in controlling systems driven by spiking inputs, and in particular muscle, is a natural next step.

Although fatigue was not considered here, muscles fatigue and so their response can be variable over time. We have previously shown how transferring learning to the brainstem is necessary for stability in systems that change significantly over time (E. D. Wilson et al., 2016). Including such an extension is likely to be necessary in situations when fatigue significantly affects muscle outputs.

Including non-linear basis increases complexity and can quickly increase the number of filter basis. However, for a non-linear system such as that considered here, it would likely give better fit to response. The fact that the cerebellar controller under-estimates higher forces and overestimates lower forces suggests that including nonlinear basis would improve the controlled force output, albeit at the expense of increased complexity.

Although orthogonalisation has been used in the past within the cerebellar adaptive filter, it was not included here. The approach used previously has been to base the orthonormalisation on the brainstem output prior to learning to obtain a fixed matrix to orthonormalise signals (E. D. Wilson et al., 2016). However, here the brainstem compensator used is poor (show by the poor match between the estimated and actual force before learning) and so basing the fixed matrix on this output did not help to improve the learning rate. As this limited the learning rate that could be used training data was repeated to enable sufficient time to learn. An online solution to optimising basis outputs in the

cerebellar adaptive filter is an ongoing area of research.

5.3 Spike Timing

There is still disagreement between spike vs rate based theories of motor control. However, there is some recent research that does support the importance of spike timing and suggest that rate-based theories need revision (Sober, Sponberg, Nemenman, & Ting, 2018). The cerebellar control results show that the cerebellar controller is very good at optimizing the location of spikes and gives good agreement with desired spikes. If spike timing is important for motor control, then it is expected that it would also be important for FES applications. Therefore, the good performance in matching desired spikes makes the cerebellar algorithm attractive for use in FES control.

For the cerebellar controllers (*C.1* and *C.2*) the STTC is slightly lower for cases that contain higher frequency inputs (in Test1 and Test2). In recordings from kick data (E. Wilson et al., 2012), it is evident that the exact spike input differs between kicks, but the output force is relatively similar. This suggests that in cases of closely separated pulses, the rate may be more important than precise spike timing. In which case, the reduction in cerebellar algorithm performance is in line with this. However, this requires further investigation.

The match between desired and actual inputs for the PID controller is poor, particularly for the Test3 data, and slightly improved for the higher frequency inputs (in Test1 and Test2). This is the opposite pattern as seen with the cerebellar algorithm. Aside from the adaptive nature discussed in section 5.1, a core difference between the PID controller and the cerebellar algorithm is the control structure used. With the PID controller being a feedback controller, but the cerebellum and brainstem providing feedforward plant compensation. This feedforward compensation is necessary for the fast responses that are needed for certain biological reflexes, such as the Vestibular Ocular Reflex (VOR) (Porrill & Dean, 2007a). In contrast, the PID controller is a feedback controller, so corrections to control actions can only be made in response to errors. Looking at the controlled response to a single pulse, it can be seen that the PID response is delayed slightly in comparison to the cerebellar output and desired response. This delay could be reduced by increasing the PID controller parameters k_p , but increases beyond those used here also increased the errors.

Here, we use a hybrid spiking system. Spikes are generated and used to stimulate the muscle model. However, the timing of these is dependent on an continuous signal, which is then converted to spikes using an integrate and fire filter (see figure 1e). In this hybrid model, muscle is only stimulated with an input once the integral of the continuous output from the brainstem and cerebellar controller reaches a threshold. As such, this introduces a slight delay ($\approx 0.01s$) in the spiking. For the purpose of FES applications as considered here, this delay in generating the muscle force is small. However, it has been suggested that spike timing is critical and that the cerebellar cortex does rely on millisecond spike precision of PCs (Sedaghat-Nejad, Pi, Hage, Fakharian, & Shadmehr, 2022). Methods such as using an eligibility trace or selecting the reference model can be investigated as a way to remove such a delay from this hybrid model. Alternatively, a fully spiking cerebellar model would not introduce this delay, as no conversion between continuous and spiking signals is needed.

5.4 Comparison to Spiking Cerebellar Models

Generally, the adaptive filter model is more compact and computationally efficient model. The advantage of this is it is simple to implement and make changes. However, this comes at the expense of losing some of the biological detail. Spiking models consider the behaviour from the bottom up and how signals are generated. In many instances biological control appears superior to engineered control systems. In simplifying the model, some of this superiority may be lost. In particular the role of redundancy in the biological system. Notably as in practice, although there are a massive number of Cerebellar granule cells and they are the most numerous neurons in the brain, granule cell re-coding is sparse. Here a compact basis is used. The advantage of a compact basis is that it reduces the computational complexity. However, the likely benefits of a sparse granule cell layer, such as efficient

separation of inputs (Schweighofer, Doya, & Lay, 2001), as well as the benefits of a large-expansion recoding, such as redundancy and adaptability to different problems, are lost.

Overall, the adaptive filter model of cerebellar function is more similar to existing engineering control solutions, particularly adaptive controllers. This has both advantages and disadvantages. For FES control the similarity to existing solutions may make it easier to implement and accept in practice. However, on the flip side, as mentioned above, some of the biological superiority may be lost. Furthermore, for FES applications, a key advantage that spiking models may present is that they are usually power efficient (Wu, Yi, & Huang, 2022). If such advantages in power consumption can be exploited, then for FES applications such spiking cerebellar models may provide a core advantage over other solutions.

In this contribution we consider the adaptive filter model of cerebellar function. This is a first step in using cerebellar models for FES control. It enables the functionality of the adaptive cerebellar model to be explored, but does not consider in detail how such function is implemented in the biological system. An interesting avenue to explore in the future would be to extend this control to consider how well spiking cerebellar models can control muscle and so how suitable for FES control.

5.5 Muscle Control

This work is the first step in increasing the generalisability of the cerebellar algorithm to systems driven by spiking inputs and demonstrates its potential for use in muscle control and FES applications. There are a number of extensions that need to be considered before this algorithm can be utilised in practice. These are discussed here.

In this work, an isometric model was used. To extend to dynamic contractions, the force-length and force-velocity dependencies also need to be considered in generating the muscle force. One advantage of the cerebellar algorithm is that the brainstem need only provide very approximate control of the plant, with the cerebellar adaptive filter providing more accurate compensation, based on output errors. This is evident from the results presented here, which show that before cerebellar learning, the control response is poor. Therefore, in including force-length and force-velocity extensions to the muscle force generation, the controller design does not necessarily need to change. However, the conditions over which the controller is tested need to be extended. Furthermore, the model used here is parameterised based on the force response of the locust hind leg muscle. For FES applications, mammalian muscle must be considered. It is not expected that this change the control strategy used by the cerebellar algorithm, but again just increase the conditions over which the controller is tested and evaluated.

Here, only a single muscle is considered. In operation, muscles work as agonist-antagonist pairs to retract and extend the skeleton. In such a set up, there is redundancy in the system as muscles can co-contract, meaning the same output force could be achieved using different inputs. The cerebellar control algorithm could lend itself for controlling such a redundant agonist-antagonist muscle system. The cost function that the learning rule is based upon could be adapted in order to give inputs that, for example, minimised jerk, to optimise inputs across the two muscles.

The cerebellar algorithm is also an ideal candidate to move beyond purely the control of an agonist-antagonist pair to multi-joint systems, perhaps using similar extensions to the cost function as for the agonist-antagonist pair. The overall principle in this work, is that the internal algorithm remains largely unchanged, with external connections providing the different functionalities. This means that the core algorithm can be utilised in a number of different conditions, so long as suitable motor inputs, reference signals, and error signals are available and suitable target outputs can be found.

The suitability of the controller for applications to human muscle control needs to be considered. Extending this work to consider dynamic conditions with agonist-antagonist pairs and then multi-joint systems is a natural first step. However, the ultimate aim is to test the cerebellar controller in a real FES application. Given the adaptive nature of the algorithm this would likely require hard constraints for safety.

6 Conclusions

This contribution has shown that the adaptive filter model of the cerebellum can be extended to the application of systems driven by spike train inputs. The algorithm gives a good match with both the force response and input spikes in control simulations. It can also be easily adapted to reduce the inputs to the muscle, which are important to limit in FES situations to avoid fatigue. More work, as outlined section 5.5, is required before practical online implementation of this controller. However, the modular nature of the cerebellum and this algorithm lends itself to the extensions required.

References

- Albus, J. S. (1971). A theory of cerebellar function. *Math. Biosci.*, *10*(1), 25–61.
- Anderson, S. R., Pearson, M. J., Pipe, A., Prescott, T., Dean, P., & Porrill, J. (2010). Adaptive cancellation of self-generated sensory signals in a whisking robot. *IEEE Transactions on Robotics*, *26*(6), 1065–1076.
- Anderson, S. R., Porrill, J., Pearson, M. J., Pipe, A. G., Prescott, T. J., & Dean, P. (2012). An internal model architecture for novelty detection: implications for cerebellar and collicular roles in sensory processing.
- Azevedo, F. A., Carvalho, L. R., Grinberg, L. T., Farfel, J. M., Ferretti, R. E., Leite, R. E., ... others (2009). Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, *513*(5), 532–541.
- Barbosa, W. S., Temporão, G. P., & Meggiolaro, M. A. (2021). Control techniques for neuromuscular electrical stimulation: a brief survey. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 2998–3005).
- Baumann, O., Borra, R. J., Bower, J. M., Cullen, K. E., Habas, C., Ivry, R. B., ... others (2015). Consensus paper: the role of the cerebellum in perceptual processes. *The Cerebellum*, *14*(2), 197–220.
- Blakemore, S.-J., Frith, C. D., & Wolpert, D. M. (2001). The cerebellum is involved in predicting the sensory consequences of action. *Neuroreport*, *12*(9), 1879–1884.
- Boudville, R., Hussain, Z., Yahaya, S. Z., Abd Rahman, M. F., Ahmad, K. A., & Husin, N. I. (2018). Development and optimization of pid control for fes knee exercise in hemiplegic rehabilitation. In *2018 12th International Conference on Sensing Technology (ICST)* (pp. 143–148).
- Bower, J. M., & Beeman, D. (2012). *The book of genesis: exploring realistic neural models with the general neural simulation system*. Springer Science & Business Media.
- Brette, R. (2015). Philosophy of the spike: rate-based vs. spike-based theories of the brain. *Frontiers in systems neuroscience*, 151.
- Chen, Y.-L., Chen, W.-L., Hsiao, C.-C., Kuo, T.-S., & Lai, J.-S. (2005). Development of the fes system with neural network+ pid controller for the stroke. In *2005 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 5119–5121).
- Cutts, C. S., & Eglon, S. J. (2014). Detecting pairwise correlations in spike trains: an objective comparison of methods and application to the study of retinal waves. *Journal of Neuroscience*, *34*(43), 14288–14303.
- Dean, P., Anderson, S., Porrill, J., & Jörntell, H. (2013). An adaptive filter model of cerebellar zone c3 as a basis for safe limb control? *The Journal of physiology*, *591*(22), 5459–5474.
- Dean, P., Porrill, J., Ekerot, C.-F., & Jörntell, H. (2010). The cerebellar microcircuit as an adaptive filter: experimental and computational evidence. *Nature Reviews Neuroscience*, *11*(1), 30–43.
- Dean, P., Porrill, J., & Stone, J. V. (2002). Decorrelation control by the cerebellum achieves oculomotor plant compensation in simulated vestibulo-ocular reflex. *Proceedings of the Royal Society of London B: Biological Sciences*, *269*(1503), 1895–1904.
- De Schutter, E., & Bower, J. M. (1994). An active membrane model of the cerebellar purkinje cell. i. simulation of current clamps in slice. *Journal of neurophysiology*, *71*(1), 375–400.

- Eccles, J. C., Ito, M., & Szentágothai, J. (1967). *The cerebellum as a neuronal machine*. Oxford, UK: Springer-Verlag.
- Fruzzetti, L., Kalidindi, H. T., Antonietti, A., Alessandro, C., Geminiani, A., Casellato, C., ... D'Angelo, E. (2022). Dual stdp processes at purkinje cells contribute to distinct improvements in accuracy and speed of saccadic eye movements. *PLoS Computational Biology*, *18*(10), e1010564.
- Fujita, M. (1982). Adaptive filter model of the cerebellum. *Biol. Cybern.*, *206*, 195–206.
- Gao, J.-H., Parsons, L. M., Bower, J. M., Xiong, J., Li, J., & Fox, P. T. (1996). Cerebellum implicated in sensory acquisition and discrimination rather than motor control. *Science*, *272*(5261), 545–547.
- Gleeson, P., Steuber, V., & Silver, R. A. (2007). neuroconstruct: a tool for modeling networks of neurons in 3d space. *Neuron*, *54*(2), 219–235.
- Griffis, E. J., Le, D. M., Stubbs, K. J., & Dixon, W. E. (2021). Closed-loop deep neural network-based fcs control for human limb tracking. In *2021 60th IEEE conference on decision and control (cdc)* (pp. 360–365).
- Gütig, R. (2014). To spike, or when to spike? *Current opinion in neurobiology*, *25*, 134–139.
- Herculano-Houzel, S. (2010). Coordinated scaling of cortical and cerebellar numbers of neurons. *Frontiers in neuroanatomy*, *12*.
- Ibitoye, M. O., Hamzaid, N. A., Hasnan, N., Abdul Wahab, A. K., & Davis, G. M. (2016). Strategies for rapid muscle fatigue reduction during fcs exercise in individuals with spinal cord injury: a systematic review. *PloS one*, *11*(2), e0149024.
- Ito, M. (1984). *The cerebellum and neural control*. New York, Raven.
- Jacobson, G. A., Rokni, D., & Yarom, Y. (2008). A model of the olivo-cerebellar system as a temporal pattern generator. *Trends in neurosciences*, *31*(12), 617–625.
- Kawato, M., & Gomi, H. (1992). The cerebellum and vor/okr learning models. *Trends in neurosciences*, *15*(11), 445–453.
- Kirsch, N., Alibeji, N., & Sharma, N. (2017). Nonlinear model predictive control of functional electrical stimulation. *Control Engineering Practice*, *58*, 319–331.
- Lenz, A., Anderson, S. R., Pipe, A. G., Melhuish, C., Dean, P., & Porrill, J. (2009). Cerebellar-inspired adaptive control of a robot eye actuated by pneumatic artificial muscles. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *39*(6), 1420–1433.
- Luque, N. R., Garrido, J. A., Carrillo, R. R., Tolu, S., & Ros, E. (2011). Adaptive cerebellar spiking model embedded in the control loop: Context switching and robustness against noise. *International journal of neural systems*, *21*(05), 385–401.
- Marr, D. (1969). A theory of cerebellar cortex. *J. Physiol.*, *202*(2), 437–470.
- Masuda, N., & Aihara, K. (2002). Bridging rate coding and temporal spike coding by effect of noise. *Physical review letters*, *88*(24), 248101.
- Medina, J. F., & Mauk, M. D. (2000). Computer simulation of cerebellar information processing. *nature neuroscience*, *3*(11), 1205–1211.
- Mohammed, S., Poignet, P., Fraisse, P., & Guiraud, D. (2012). Toward lower limbs movement restoration with input–output feedback linearization and model predictive control through functional electrical stimulation. *Control Engineering Practice*, *20*(2), 182–195.
- Murdoch, B. E. (2010). The cerebellum and language: historical perspective and review. *Cortex*, *46*(7), 858–868.
- Oliveira, T. R., Costa, L. R., Catunda, J. M. Y., Pino, A. V., Barbosa, W., & de Souza, M. N. (2017). Time-scaling based sliding mode control for neuromuscular electrical stimulation under uncertain relative degrees. *Medical engineering & physics*, *44*, 53–62.
- Panzeri, S., Brunel, N., Logothetis, N. K., & Kayser, C. (2010). Sensory neural codes using multiplexed temporal scales. *Trends in neurosciences*, *33*(3), 111–120.
- Peckham, P. H., Knutson, J. S., et al. (2005). Functional electrical stimulation for neuromuscular applications. *Annual review of biomedical engineering*, *7*(1), 327–360.

- Poboroniuc, M.-S., Irimia, D.-C., Baciuc, A., Bondar, T., Nițică, I. A., & Piseru, A. E. (2018). A fuzzy controller to support fes-based sitting-down in paraplegia. In *2018 international conference and exposition on electrical and power engineering (epe)* (pp. 0523–0528).
- Porrill, J., & Dean, P. (2007a). Cerebellar motor learning: when is cortical plasticity not enough? *PLoS computational biology*, *3*(10), e197.
- Porrill, J., & Dean, P. (2007b). Recurrent cerebellar loops simplify adaptive control of redundant and nonlinear motor systems. *Neural computation*, *19*(1), 170–193.
- Porrill, J., & Dean, P. (2008). Silent synapses, ltp, and the indirect parallel-fibre pathway: computational consequences of optimal cerebellar noise-processing. *PLoS computational biology*, *4*(5), e1000085.
- Porrill, J., Dean, P., & Anderson, S. R. (2013). Adaptive filters and internal models: multilevel description of cerebellar function. *Neural Networks*, *47*, 134–149.
- Qiu, S., He, F., Tang, J., Xu, J., Zhang, L., Zhao, X., ... others (2014). Intelligent algorithm tuning pid method of function electrical stimulation using knee joint angle. In *2014 36th annual international conference of the IEEE engineering in medicine and biology society* (pp. 2561–2564).
- Rössert, C., Dean, P., & Porrill, J. (2015). At the edge of chaos: how cerebellar granular layer network dynamics can provide the basis for temporal filters. *PLoS computational biology*, *11*(10), e1004515.
- Rouhani, H., Same, M., Masani, K., Li, Y. Q., & Popovic, M. R. (2017). Pid controller design for fes applied to ankle muscles in neuroprosthesis for standing balance. *Frontiers in neuroscience*, *11*, 347.
- Schweighofer, N., Arbib, M. A., & Dominey, P. F. (1996). A model of the cerebellum in adaptive control of saccadic gain. *Biological cybernetics*, *75*(1), 19–28.
- Schweighofer, N., Doya, K., & Lay, F. (2001). Unsupervised learning of granule cell sparse codes enhances cerebellar adaptive control. *Neuroscience*, *103*(1), 35–50.
- Sedaghat-Nejad, E., Pi, J. S., Hage, P., Fakharian, M. A., & Shadmehr, R. (2022). Synchronous spiking of cerebellar purkinje cells during control of movements. *Proceedings of the National Academy of Sciences*, *119*(14), e2118954119.
- Sejnowski, T. J. (1977). Storing covariance with nonlinearly interacting neurons. *Journal of mathematical biology*, *4*(4), 303–321.
- Sharma, N., Gregory, C. M., Johnson, M., & Dixon, W. E. (2011). Closed-loop neural network-based nmes control for human limb tracking. *IEEE Transactions on Control Systems Technology*, *20*(3), 712–725.
- Sober, S. J., Sponberg, S., Nemenman, I., & Ting, L. H. (2018). Millisecond spike timing codes for motor control. *Trends in neurosciences*, *41*(10), 644–648.
- Torben-Nielsen, B., Segev, I., & Yarom, Y. (2012). The generation of phase differences and frequency changes in a network model of inferior olive subthreshold oscillations. *PLoS computational biology*, *8*(7), e1002580.
- Vijayan, A., & Diwakar, S. (2022). A cerebellum inspired spiking neural network as a multi-model for pattern classification and robotic trajectory prediction. *Frontiers in Neuroscience*, *16*.
- Vijayan, A., Nutakki, C., Kumar, D., Achuthan, K., Nair, B., & Diwakar, S. (2017). Enabling a freely accessible open source remotely controlled robotic articulator with a neuro-inspired control algorithm. *Int. J. Online Eng.*, *13*(1), 61–75.
- Wannawas, N., Shafti, A., & Faisal, A. A. (2022). Neuromuscular reinforcement learning to actuate human limbs through fes. *arXiv preprint arXiv:2209.07849*.
- Widrow, B., & Stearns, S. D. (1985). Adaptive signal processing prentice-hall. *Englewood Cliffs, NJ*.
- Wilson, E., Rustighi, E., Newland, P. L., & Mace, B. R. (2012). A comparison of models of the isometric force of locust skeletal muscle in response to pulse train inputs. *Biomechanics and modeling in mechanobiology*, *11*(3), 519–532.
- Wilson, E. D., Anderson, S. R., Dean, P., & Porrill, J. (2019). Sensorimotor maps can be dynamically calibrated using an adaptive-filter model of the cerebellum. *PLoS computational biology*, *15*(7), e1007187.

- Wilson, E. D., Assaf, T., Pearson, M. J., Rossiter, J. M., Anderson, S. R., Porrill, J., & Dean, P. (2016). Cerebellar-inspired algorithm for adaptive control of nonlinear dielectric elastomer-based artificial muscle. *Journal of The Royal Society Interface*, *13*(122), 20160547.
- Wilson, E. D., Assaf, T., Pearson, M. J., Rossiter, J. M., Dean, P., Anderson, S. R., & Porrill, J. (2015). Biohybrid control of general linear systems using the adaptive filter model of cerebellum. *Frontiers in neurorobotics*, *9*, 5.
- Wilson, E. D., Assaf, T., Rossiter, J. M., Dean, P., Porrill, J., Anderson, S. R., & Pearson, M. J. (2021). A multizone cerebellar chip for bioinspired adaptive robot control and sensorimotor processing. *Journal of the Royal Society Interface*, *18*(174), 20200750.
- Wu, D., Yi, X., & Huang, X. (2022). A little energy goes a long way: Build an energy-efficient, accurate spiking neural network from convolutional neural network. *Frontiers in neuroscience*, *16*.
- Yu, Q., Li, H., & Tan, K. C. (2018). Spike timing or rate? neurons learn to make decisions for both through threshold-driven plasticity. *IEEE transactions on cybernetics*, *49*(6), 2178–2189.