










Multimodal Parameter Inference for a Canonical Motor Microcircuit Controlling Rat Hindlimb Motion

Clayton Jackson¹ , Matthieu Chardon² , Y. Curtis Wang³ ,
Johann Rudi⁴ , Matthew Tresch⁵ , Charles J. Heckman⁶ ,
and Roger D. Quinn¹ 

¹ Department of Mechanical and Aerospace Engineering,
Case Western Reserve University, Cleveland, OH 44106-7222, USA
clayton.jackson@case.edu

² Interdepartmental Neuroscience, Northwestern University, Chicago, IL, USA

³ Department of Electrical and Computer Engineering, California State University,
Los Angeles, CA 90032, USA

⁴ Department of Mathematics, Virginia Tech, Blacksburg, VA 24060, USA

⁵ Department of Biomedical Engineering Northwestern University,
Evanston, IL 60208, USA

⁶ Department of Physical Therapy and Human Movement Sciences,
Northwestern University, Evanston, IL 60208, USA

Abstract. This work explored synaptic strengths in a computational neuroscience model of a controller for the hip joint of a rat which consists of Ia interneurons, Renshaw cells, and the associated motor neurons. This circuit has been referred to as the Canonical Motor Microcircuit (CMM). It is thought that the CMM acts to modulate motor neuron activity at the output stage. We first created a biomechanical model of a rat hindlimb consisting of a pelvis, femur, shin, foot, and flexor-extensor muscle pairs modeled with a Hill muscle model. We then modeled the CMM using non-spiking leaky-integrator neural models connected with conductance-based synapses. To tune the parameters in the network, we implemented an automated approach for parameter search using the Markov chain Monte Carlo (MCMC) method to solve a parameter estimation problem in a Bayesian inference framework. As opposed to traditional optimization techniques, the MCMC method identifies probability densities over the multidimensional space of parameters. This allows us to see a range of likely parameters that produce model outcomes consistent with animal data, determine if the distribution of likely parameters is uni- or multi-modal, as well as evaluate the significance and sensitivity of each parameter. This approach will allow for further analysis of the circuit, specifically, the function and significance of Ia feedback and Renshaw cells.

This work was supported by NSF DBI 2015317 as part of the NSF/CIHR/DFG/FRQ/UKRI-MRC Next Generation Networks for Neuroscience Program.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
F. Meder et al. (Eds.): Living Machines 2023, LNAI 14158, pp. 38–51, 2023.
https://doi.org/10.1007/978-3-031-39504-8_3

Keywords: Sensory-motor Control · Markov chain Monte Carlo ·
Neuromechanical Simulation

1 Introduction

Mammalian locomotion continues to be a much researched topic in the world of neuroscience. It is commonly accepted that the rhythmic activity during locomotion arises from descending commands from pattern generator networks in the spinal cord [16, 17]. The activity from pattern generators is then modulated by neural circuits and sensory feedback at the output stage before controlling muscle activity. We have designed one of these output stage neural circuits after the works of Hultborn et al. [10] which we refer to as the Canonical Motor Microcircuit (CMM). The CMM modeled in this work does not include gamma motoneurons as in the Hultborn model for simplicity [10]. This network, consisting of Ia interneurons, Renshaw cells, and motor neurons, is included in numerous other neural models of locomotor circuitry. [8, 9, 12, 18, 23]. However, these works primarily focus on the pattern generator networks and largely ignore the role of the CMM in modulating these commands. Aside from studies showing that the neurons in this model are active in mammals during locomotion, the role of the CMM and of the specific neurons within the CMM remains unknown [21].

The aim of this work is to enhance our understanding of the CMM through the use of neural and biomechanical modeling. While the ideal approach would involve recording neural activity in the spinal cord *in vivo*, this method can be challenging and resource-intensive. Therefore, we aim to use modeling as a complementary tool to aid in these experiments. By working with neuroscientists, we can use these models to test theories, design future experiments, and provide insights into experimental results.

Modeling neural and biomechanical systems is challenging, tuning these models is often nontrivial. Hand-tuning the networks is one possible method, but is often time consuming and impractical. Another possible method is the Functional Subnetwork Approach (FSA) [28]. This method describes the tuning of small networks to perform addition, subtraction, multiplication, division, integration, and differentiation. These smaller networks can then be joined to create large networks without the need for global optimization [28]. The FSA can be useful in designing neural circuits, though when modeling existing circuits, it may not be clear how to divide the network into subnetworks. In these cases, we can turn to more traditional optimization techniques. Gradient-based optimization is likely not an option as these systems tend to be highly nonlinear. Other methods, such as particle swarm optimization, are reliant on initial values and like gradient-based optimization, are susceptible to getting stuck in local minima [14, 15]. While genetic algorithms can use mutation to avoid local minima, they share one large downside with other methods in this application. These optimization methods produce one result and often tell little to nothing about the solution space. This paper examines a tuning method for the CMM using a Markov chain Monte Carlo (MCMC) approach to solve a parameter estimation

problem in a Bayesian inference framework. This parameter inference method has been used to tune a single Hodgkin-Huxley style neuron, but has not yet been used to tune a network of neurons [31]. The MCMC method is advantageous for this application because it provides a view of the solution space as a whole, showing the significance and sensitivity of each parameter. In this way, we can use the results to analyze the components of the network.

2 Methods

2.1 Modeling

We first create a biomechanical model of a rat hindlimb to be controlled by the CMM. The physics engine and biomechanical modeling for this work were done in Mujoco [29]. An existing model of the hindlimb with muscles to control the hip, knee, and ankle in three dimensions, modeled in Opensim, was taken and converted to the format needed for Mujoco [7, 11, 24]. The model was then simplified to only two muscles, a flexor-extensor muscle pair controlling the hip joint in the sagittal plane. In this model, the pelvis is fixed and the leg air steps without ground contact. These simplifications to the model were chosen as a starting point; future models will expand to include the complete array of muscles and control of all joints. The simplified model in Mujoco can be seen in Fig. 1.

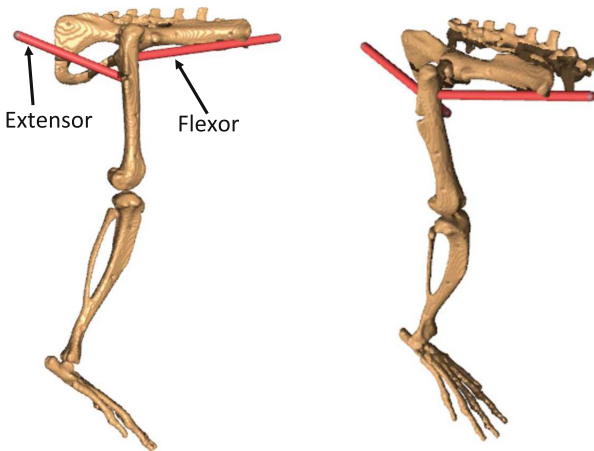


Fig. 1. Biomechanical model of a rat right hindlimb in Mujoco with a flexor/extensor muscle pair (red). (Color figure online)

Mujoco implements a Hill muscle model which is similar to that of Opensim. The key differences are that Mujoco assumes the tendons are inelastic and that

the pennation angle is zero [7, 24, 29]. For this work, the default muscle parameters in Mujoco were used. The muscle force, F_M , can be found using the equation [29]:

$$F_M = (F_L(L) \cdot F_{Vel}(Vel) \cdot act(V_{MN}) + F_P(L)) \cdot F_0 \quad (1)$$

where F_L is the active force as a function of length, L is the length of the muscle, F_{Vel} is the active force as a function of velocity, Vel is the velocity of the muscle, F_P is the passive force, F_0 is the peak active force at zero velocity and is computed in the model compiler, and act is the muscle activation as a function of the corresponding motor neuron potential, V_{MN} . In this model, we use a sigmoid muscle activation function. Examples of the F_L and F_{Vel} curves are shown in Fig. 2.

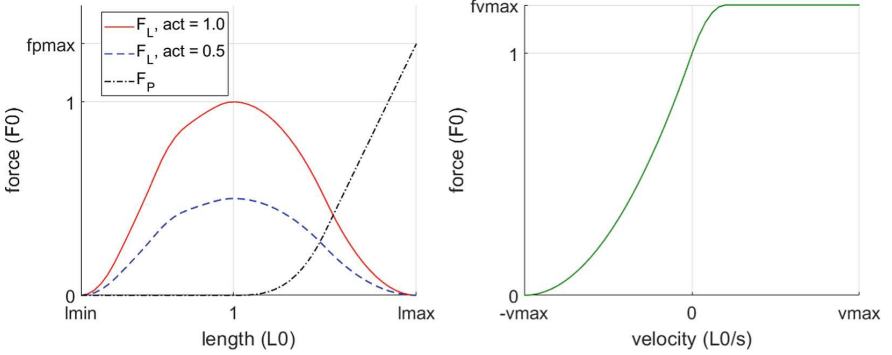


Fig. 2. Mujoco muscle modeling. Left: F_P curve and F_L curve with various activation. The activation function simply scales the F_L curve. The F_L curve is centered around L_0 , which is the resting length of the muscle as computed in the model compiler. Right: F_{Vel} curve. [29]

We then create the neural model of the CMM, shown in Fig. 3, to control muscle activation of the biomechanical model. The neural modeling and simulation in this work was done in Python using SNS-Toolbox [20]. The neurons were modeled using a non-spiking leaky-integrator model:

$$C \frac{dV}{dt} = I_{app} - I_{leak} + \sum_{i=1}^n I_{syn_i} \quad (2)$$

where C is the membrane capacitance, V is the membrane voltage, I_{app} is an applied current, I_{leak} is the membrane leak current, and I_{syn_i} are the synaptic currents. The benefit of using this neural model is that it allows for a single non-spiking neuron to approximate the average activity of populations of spiking neurons [32]. The synaptic currents are modeled using conductance based synapses:

$$I_{syn_i} = g_{max_i} \cdot \min \left(\max \left(\frac{V_{pre} - E_{lo_i}}{E_{hi_i} - E_{lo_i}}, 0 \right), 1 \right) \cdot (E_{syn_i} - V(t)) \quad (3)$$

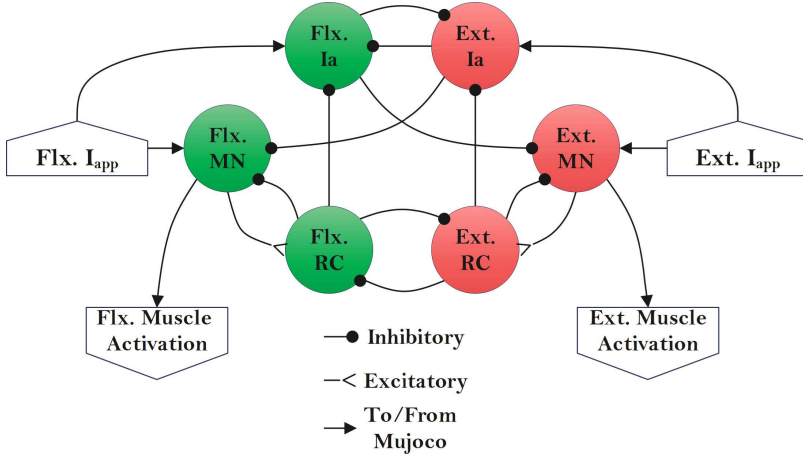


Fig. 3. CMM Neural Model in SNS-Toolbox. Ia: Ia inhibitory neuron. MN: Motor Neuron, RC: Renshaw cell.

where g_{max} is the maximum synaptic conductance, V_{pre} is the membrane potential of the pre-synaptic neuron, E_{hi_i} and E_{lo_i} are saturation and threshold parameters, respectively, which are properties of the pre-synaptic neuron, and E_{syn} is the synaptic reversal potential. The synaptic reversal potential was set to -100 mV for inhibitory synapses and 0 mV for excitatory synapses. The leak current in Eq. 2 estimates the net effect of sodium, potassium, and chloride channels with a net membrane conductance G , and a resting potential E_R :

$$I_{leak} = G \cdot (V(t) - E_R). \quad (4)$$

The neural model incorporates Ia afferent feedback from muscle spindles to the respective Ia inhibitory and motor neurons. We model this as a discharge rate from the muscle spindles. It was found that for the Hill muscle model, the discharge rate, is proportional to the tension of the muscle [25]. We can then model the afferent feedback as an applied current:

$$I_{app} = m \cdot T + b \quad (5)$$

where T is the muscle tension, m is a gain to convert from muscle tension to a current, and b is a y-offset.

Tables 3 and 2 show the parameter values used in this model for the parameters not explored in the MCMC search (Table 1).

In order to run the neural and biomechanical models together, we first initialize the neurons at their resting potentials. The biomechanical model is initialized with the hip in slight flexion. This causes the limb to fall due to gravity at the beginning of the simulation, initially exciting the neurons. Using the forward Euler method, we then repeat the following for a desired number of iterations:

Table 1. Neural Parameters. MN: Motor Neuron. Ia: Ia inhibitory Neuron. RC: Renshaw Cells

| | E_R (MV) | E_{hi} (MV) | E_{lo} (MV) |
|-----------|------------|---------------|---------------|
| <i>MN</i> | -62 | -54.53 | -78 |
| <i>Ia</i> | -60 | -40 | -62 |
| <i>RC</i> | -50.5 | -40 | -60 |

Table 2. Muscle Feedback Applied Current Parameters.

| | m | b |
|-----------------------------|----------|----------|
| l_{app} to Ext <i>Ia</i> | 0.4565 | -0.5617 |
| l_{app} to Flex <i>Ia</i> | 3.5385 | -0.7185 |
| l_{app} to Ext <i>MN</i> | 0.0106 | -0.9210 |
| l_{app} to Flex <i>MN</i> | 0.5036 | 0.0411 |

1. Run one timestep of the neural model in SNS-Toolbox using muscle feedback from Mujoco as inputs.
2. Feed the motor neuron potentials from the previous timestep into the sigmoidal muscle activation function.
3. Take one timestep in the biomechanical model in Mujoco using the calculated muscle activation.
4. Get the muscle tension from Mujoco to input to the next step of the neural model.

In earlier works, both the biomechanical and neural models were run in Animatlab2 [6]. While it was beneficial to have both models in the same software, the simulation time in Animatlab2 was slow, which greatly impacted the computation time for automating the parameter search. By utilizing newly available software tools, SNS-Toolbox and Mujoco, simulation time was decreased from 10.21 to 2.48 s on average when compared to Animatlab2.

2.2 Markov Chain Monte Carlo

Previous iterations of this work aimed to hand-tune similar models to produce oscillatory motion in the hip joint [13]. While this method proved to be successful, the results provided little information about the significance or sensitivity of parameters in the models. To address these shortcomings, we implement an automated parameter search for the maximum synaptic conductance, g_{max} , in the CMM shown in Fig. 3 using a Markov chain Monte Carlo (MCMC) method. This method is centered around Bayes theorem [5]:

$$P(g_{max}|y) = \frac{P(y|g_{max}) \cdot P(g_{max})}{P(y)} \quad (6)$$

where $P(g_{max}|y)$ is the posterior (conditional probability for parameters g_{max} , given the data y), $P(y|g_{max})$ is the likelihood (conditional probability of the data y given the parameters g_{max}), g_{max} is the prior (probability of the parameters), and $P(y)$ is called evidence (probability of the data y). The evidence term is a constant, therefore Bayes theorem is typically used in the form:

$$P(g_{max}|y) \propto P(y|g_{max}) \cdot P(g_{max}). \quad (7)$$

The prior term is designed to take into account known information about the parameters. As there is little known about the CMM, unbiased priors were used. These priors were uniform distributions with lower and upper bounds of 0 and 8 microSiemens, respectively. The likelihood term is based on a loss function which was designed to match the simulated data of the hip angle to data of the hip trajectory of a rat walking on a treadmill as collected by Alessandro et al. [2]. Rather than explicitly trying to match the simulated and animal data, the loss function aims to mimic certain aspects of the motion:

$$loss = l_{freq} + l_{sw/st} + l_{smooth} + l_{oscillate} \quad (8)$$

where l_{freq} is the percent error of the frequency of oscillations between the simulated and animal data, $l_{sw/st}$ is the percent error of the swing to stance ratio. The percent error is calculated as:

$$\%Error = \left| \frac{v_S - v_A}{v_A} \right| \quad (9)$$

where v_S is the simulated value and v_A is the value from the animal data. The smoothness term in the loss function, l_{smooth} , is the mean of the second derivative of the simulated hip angle squared.

$$l_{sw/st} = mean \left(\left(\frac{d^2\theta}{dt^2} \right)^2 \right) \quad (10)$$

The last term in the loss function is designed to promote solutions which oscillate, $l_{oscillate}$ is the inverse of the number of peaks in the simulated hip angle vs. time.

$$l_{oscillate} = \frac{1}{num_peaks} \quad (11)$$

The likelihood probability density is then defined as:

$$P(y|g_{max}) = e^{-loss} \quad (12)$$

where the loss depends on the data, y , and parameters, g_{max} : $loss = loss(g_{max}, y)$.

To begin the MCMC, we first choose a sampling method. In this work, we have chosen an adaptive Metropolis-Hastings algorithm [26]. The Metropolis-Hastings algorithm is typically a good choice when the conditional probability

$P(y|g_{max})$ cannot be drawn from directly [4]. As previously stated, we aim to determine if the distribution of likely parameters is uni- or multi-modal. To do this we run an adaptive parallel tempering MCMC using the PyPESTO library in python [19,27]. Parallel tempering is a technique used in MCMC to help chains traverse high posterior density regions and overcome local optima. Multiple chains are run in parallel where each chain has a unique starting point and runs its own Metropolis-Hastings random walk. Similar to simulated annealing algorithms, a temperature is computed for each chain and varies according to Vousden et al. [30]. We then assign a term, β where $0 < \beta < 1$ and is inversely related to the temperature of the chain: $\beta = 1/T$. This β term is then applied as an exponent to the posteriors of the given chain, resulting in $P^\beta(g_{max}|y)$. This newly scaled posterior is then what is used to calculate the acceptance ratio in the Metropolis-Hastings algorithm. The impact of the β term is that chains at higher temperatures can more easily explore the entire prior distribution for parameters, while chains at lower temperatures can sample from regions of high probability more efficiently [30]. Additionally, chains may perform swaps at certain intervals, swapping their parameters while maintaining their temperature.

3 Results

The parallel tempering MCMC with an adaptive Metropolis-Hastings random walk algorithm was successful in its ability to explore the multidimensional parameter space, finding that the posterior is multi-modal with two distinct regions of parameters with acceptable loss values. We present marginals of the posterior showing the relationships between parameters. Figure 4 shows four of these marginalized densities.

Here, we observe two modes in the posteriors. One mode showing the Renshaw cells inhibits Ia interneurons more strongly than they do motor neurons, while the other indicates that the synapse from the extensor Renshaw cell to the extensor Ia inhibitory neuron has a maximum synaptic conductance close to zero (Fig. 4C). This would suggest that there are no synapses between these neurons. We are able to dismiss these results as it is known that Renshaw cells inhibit Ia inhibitory neurons [21]. The marginal densities in Fig. 4 also show that the two modes in the posterior densities may overlap for certain parameters (Fig. 4B).

To further examine the results, we look at the hip angle for the simulated data and animal data. Figure 5 shows a variety of joint trajectories and their associated cost. The parameter values used in these plots can be found in Table 3. Note that the loss function in Eq. 8 only aims to match certain properties of the trajectory, and therefore, the joint trajectories from the simulation may not match the animal data completely. The results in Fig. 5 show a variety of joint trajectories, all of which indicate that the CMM is capable of producing oscillatory behavior without the input of a pattern generator.

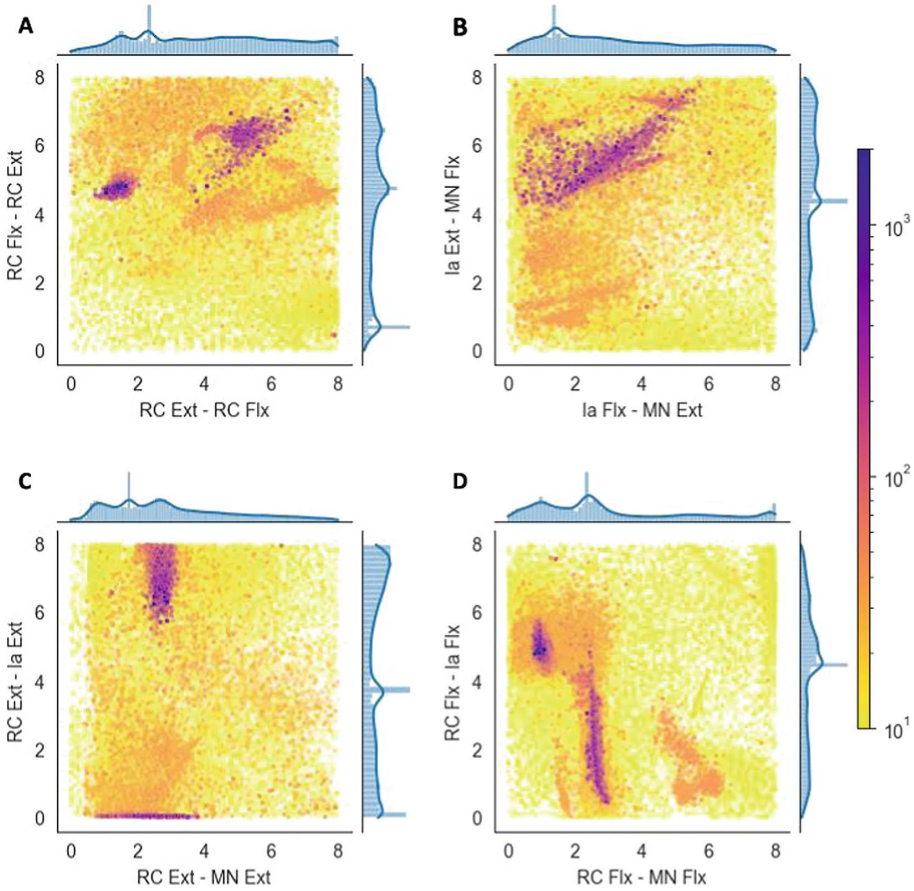


Fig. 4. Partial solution map of the posterior for the 12 parameter search of maximum synaptic conductance of the CMM model. Axes are labeled as: pre-synaptic - post-synaptic neurons. For example: RC Ext - Ia Ext represents the synapse where the RC Ext. is the pre-synaptic neuron and the Ia Ext. is the post-synaptic neuron. 1D marginals are shown above and to the right of each subplot, which tell the search range and the normalized distribution. The other portion of each subplot shows the 2D marginals of pairs of parameters with the x- and y-axes being the range of the respective parameter. Each point in the 2D marginals is a sample found in the MCMC search. The color of the point is associated with the posterior, with the darker colors indicating a large posterior, and therefore a lower loss. The darkness of the point also indicates the density of the area, with more prominent color indicating more points overlapping. This figure shows 4 of the 66 2D marginals, the complete set can be found at: <https://github.com/cxj271/Living-Machines-2023>

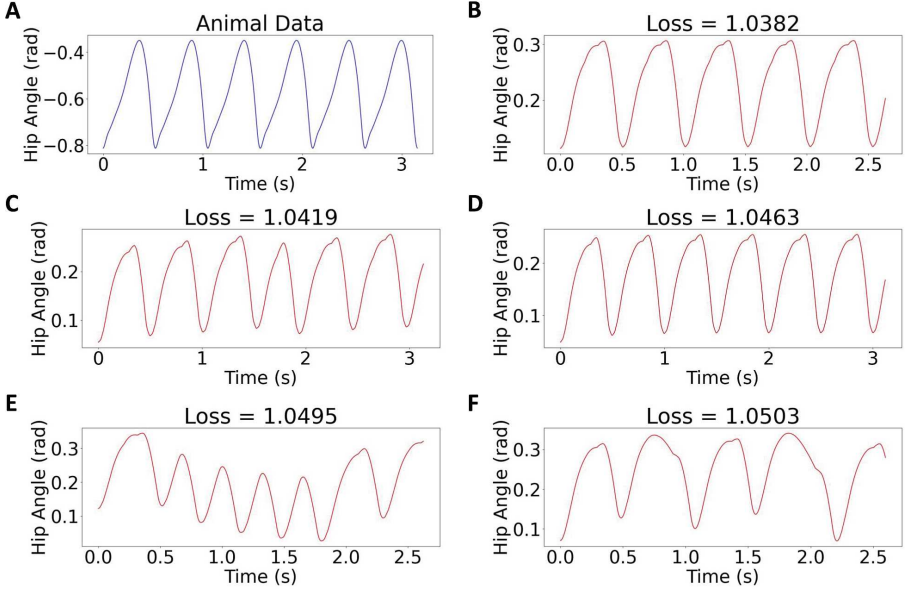


Fig. 5. Hip angle vs. time of a rat walking on a treadmill as it compares to the simulated data. A) Hip angle in the sagittal plane recorded from a rat walking on a treadmill. B-F) Solutions found from the MCMC search and their associated loss.

Table 3. Parameter Values for Simulations in Fig. 5

| Fig. 5 Subplot | Loss | RC Flx to RC Ext | RC Ext to RC Flx | Ia Ext to MN Flx | Ia Flx to MN Ext | RC Ext to Ia Ext | RC Ext to MN Ext | RC Flx to Ia Flx | RC Flx to MN Flx |
|----------------|--------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| B | 1.0382 | 4.7841 | 1.4200 | 6.3990 | 3.9774 | 6.9240 | 2.8318 | 5.0318 | 0.9442 |
| C | 1.0419 | 4.7357 | 1.0733 | 5.3191 | 2.3385 | 6.4873 | 2.5085 | 4.8937 | 0.8550 |
| D | 1.0463 | 4.8273 | 1.4041 | 5.0100 | 1.7797 | 6.0296 | 2.7741 | 4.8000 | 0.9119 |
| E | 1.0495 | 4.8319 | 1.4309 | 5.6222 | 3.0556 | 6.7807 | 2.5516 | 4.8108 | 1.0417 |
| F | 1.0503 | 4.1899 | 1.6269 | 4.9527 | 2.2418 | 6.3311 | 2.8697 | 4.9122 | 1.0798 |

4 Discussion

The results in Figs. 4 and 5 show that the parallel tempering MCMC method can successfully be used to automate the parameter search of a neural circuit to control a biomechanical model to exhibit a desired behavior. While the simulation results from the MCMC shown in Fig. 5 show difference in joint trajectory from the animal data, they share similarities in frequency of oscillations and swing to stance ratios. This is a result of the design of the loss function in Eq. 8. Many iterations of loss functions were tested to determine an appropriate equation to generate different results. Other loss functions included: mean squared error, mean absolute error, energy distance, Pearson Correlation, and Dynamic time warping [1, 3, 22]. Additionally, we explored the inclusion other

terms to the current loss function such as a root mean square error and errors in the amplitude of oscillation. These attempts produced results with a variety of undesirable characteristics such as sharp changes in velocity or results which did not oscillate the hip joint. Issues such as these prompted the inclusion of terms in the loss function which promoted smooth trajectories and oscillatory motion (l_{smooth} and $l_{oscillate}$). The differences in joint trajectories may also be a result of using a simplified model of the rat hindlimb with only two muscles controlling the joint motion as opposed to applying muscle synergies. Further testing and refining the loss function may provide better results in the sense of hip trajectories. However, designing and selecting a “good” loss function is a difficult process for automated parameter search and optimization.

Unlike traditional optimization methods, the MCMC results provide a posterior density (Fig. 4), allowing us to see the whole picture rather than a single solution of the parameter space. The MCMC method was also able to avoid getting stuck at local minima and found acceptable results with no need for initial guesses and no information about the parameters other than upper and lower bounds. Furthermore, the MCMC was able to locate multiple modes in the posteriors. We were then able to apply what we know in biology to dismiss the results from one of the modes. Another option would be to include this knowledge in the priors.

The final results confirmed some expectations. It was thought that the Renshaw cells more strongly inhibit Ia inhibitory neurons than motor neurons. This was found in previous iterations of the work through hand-tuning and confirmed through the posterior densities in Figs. 4C and 4D [13]. The posterior of the mutual inhibition of the Renshaw cells in Fig. 4A show a tight spread of losses. Therefore, it can also be inferred that these synapses are sensitive to change and play a significant role in muscle control. The sensitivity and significance of the remaining synaptic conductances can be examined in a similar fashion using the posteriors presented in Fig. 4. For example, the Ia-MN synapses (Fig. 4B), are insensitive compared to the RC-RC synapses (Fig. 4A) as the darker region in the posterior is more spread out.

It should be noted that the posteriors shown in Fig. 4 were found using animal data of a rat walking at one frequency and swing stance ratio. While it is likely that using animal data with different frequencies and swing stance ratios would cause changes in the posterior of the synaptic conductances, the results presented should not be dismissed. It is important to remember that this circuit receives a rhythmic input which was omitted in these experiments. Future works will further analyze the modes found in the posteriors, exploring how this system responds to the additional rhythmic input. We can then determine if tuning this circuit to oscillate at one frequency is beneficial when driven at other frequencies. It is also possible that small changes in the desired frequency and swing stance ratio would result in small shifts in the posteriors of the synaptic conductances. These small shifts may be due to synaptic plasticity, which may be accounted for by expanding the neural model to include synaptic plasticity.

The MCMC method for tuning neural circuits provides promising results which may help to make neural modeling more readily and widely used in computational neuroscience. In future works, the models will also be expanded to include control of the hip in three-dimensions as well as control of the knee and ankle joints. We will also explore how changes to the scale of the biomechanical model impact changes to the neural model. It is known that as the length scale of the biomechanical model increases, inertial forces become more dominant leading to changes in muscle activity during normal gait [33]. However, changes in the neural control which drives this activity have not yet been explored. The use of the MCMC method for parameter searching will allow us to develop more complex models and begin to answer these larger questions.

References

1. Müller, M.: Dynamic time warping. In: Müller, M. (ed.) *Information Retrieval for Music and Motion*, pp. 69–84. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74048-3_4
2. Alessandro, C., Rellinger, B.A., Barroso, F.O., Tresch, M.C.: Adaptation after vastus lateralis denervation in rats demonstrates neural regulation of joint stresses and strains. *eLife* **7**, e38215 (2018). <https://doi.org/10.7554/eLife.38215>
3. Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. In: Cohen, I., Huang, Y., Chen, J., Benesty, J. (eds.) *Noise Reduction in Speech Processing*. Springer Topics in Signal Processing, pp. 1–4. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00296-0_5
4. Calvetti, D., Somersalo, E.: Sampling: the real thing. In: Calvetti, D., Somersalo, E. (eds.) *Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*. STAMS, pp. 161–182. Springer, New York (2007). https://doi.org/10.1007/978-0-387-73394-4_9
5. Chen, Z.: An overview of Bayesian methods for neural spike train analysis. *Comput. Intell. Neurosci.* **2013**, 1–17 (2013). <https://doi.org/10.1155/2013/251905>. <http://www.hindawi.com/journals/cin/2013/251905/>
6. Cofer, D., Cymbalyuk, G., Reid, J., Zhu, Y., Heitler, W.J., Edwards, D.H.: AnimateLab: a 3D graphics environment for neuromechanical simulations. *J. Neurosci. Methods* **187**(2), 280–288 (2010). <https://doi.org/10.1016/j.jneumeth.2010.01.005>. <https://linkinghub.elsevier.com/retrieve/pii/S0165027010000087>
7. Delp, S.L., et al.: OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Trans. Biomed. Eng.* **54**(11), 1940–1950 (2007). <https://doi.org/10.1109/TBME.2007.901024>
8. Deng, K., et al.: Biomechanical and sensory feedback regularize the behavior of different locomotor central pattern generators. *Biomimetics* **7**(4), 226 (2022). <https://doi.org/10.3390/biomimetics7040226>. <https://www.mdpi.com/2313-7673/7/4/226>
9. Deng, K., et al.: Neuromechanical model of rat hindlimb walking with two-layer CPGs. *Biomimetics* **4**(1), 21 (2019). <https://doi.org/10.3390/biomimetics4010021>. <https://www.mdpi.com/2313-7673/4/1/21>
10. Hultborn, H., Lindström, S., Wigström, H.: On the function of recurrent inhibition in the spinal cord. *Exp. Brain Res.* **37**(2) (1979). <https://doi.org/10.1007/BF00237722>
11. Ikkala, A., Hämäläinen, P.: Converting Biomechanical Models from OpenSim to MuJoCo (2020). <https://doi.org/10.48550/ARXIV.2006.10618>

12. Ivashko, D., Prilutsky, B., Markin, S., Chapin, J., Rybak, I.: Modeling the spinal cord neural circuitry controlling cat hindlimb movement during locomotion. *Neurocomputing* **52–54**, 621–629 (2003). [https://doi.org/10.1016/S0925-2312\(02\)00832-9](https://doi.org/10.1016/S0925-2312(02)00832-9). <https://linkinghub.elsevier.com/retrieve/pii/S0925231202008329>
13. Jackson, C., Nourse, W.R.P., Heckman, C.J., Tresch, M., Quinn, R.D.: Canonical motor microcircuit for control of a rat hindlimb. In: Hunt, A., et al. (eds.) *Biomimetic and Biohybrid Systems*, vol. 13548, pp. 309–320. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20470-8_31
14. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of ICNN 1995 - International Conference on Neural Networks*, Perth, WA, Australia, vol. 4, pp. 1942–1948. IEEE (1995). <https://doi.org/10.1109/ICNN.1995.488968>. <http://ieeexplore.ieee.org/document/488968/>
15. Lee, K.Y., Park, J.: Application of particle swarm optimization to economic dispatch problem: advantages and disadvantages, pp. 188–192 (2006). <https://doi.org/10.1109/PSCE.2006.296295>
16. Lindén, H., Petersen, P.C., Vestergaard, M., Berg, R.W.: Movement is governed by rotational neural dynamics in spinal motor networks. *Nature* **610**(7932), 526–531 (2022). <https://doi.org/10.1038/s41586-022-05293-w>. <https://www.nature.com/articles/s41586-022-05293-w>
17. MacKay-Lyons, M.: Central pattern generation of locomotion: a review of the evidence. *Phys. Ther.* **82**(1), 69–83 (2002). <https://doi.org/10.1093/ptj/82.1.69>. <https://academic.oup.com/ptj/article/82/1/69/2837028>
18. McCrea, D.A., Rybak, I.A.: Organization of mammalian locomotor rhythm and pattern generation. *Brain Res. Rev.* **57**(1), 134–146 (2008). <https://doi.org/10.1016/j.brainresrev.2007.08.006>. <https://linkinghub.elsevier.com/retrieve/pii/S0165017307001798>
19. Miasojedow, B., Moulines, E., Vihola, M.: An adaptive parallel tempering algorithm. *J. Comput. Graph. Stat.* **22**(3), 649–664 (2013). <https://doi.org/10.1080/10618600.2013.778779>. <http://www.tandfonline.com/doi/abs/10.1080/10618600.2013.778779>
20. Nourse, W.R.P., Szczecinski, N.S., Quinn, R.D.: SNS-toolbox: a tool for efficient simulation of synthetic nervous systems. In: Hunt, A., et al. (eds.) *Biomimetic and Biohybrid Systems*, vol. 13548, pp. 32–43. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20470-8_4
21. Pratt, C.A., Jordan, L.M.: IA inhibitory interneurons and Renshaw cells as contributors to the spinal mechanisms of fictive locomotion. *J. Neurophysiol.* **57**(1), 56–71 (1987). <https://doi.org/10.1152/jn.1987.57.1.56>. <https://www.physiology.org/doi/10.1152/jn.1987.57.1.56>
22. Rizzo, M.L., Székely, G.J.: Energy distance. *Wiley Interdisc. Rev.: Comput. Stat.* **8**(1), 27–38 (2016). <https://doi.org/10.1002/wics.1375>. <https://onlinelibrary.wiley.com/doi/10.1002/wics.1375>
23. Rybak, I.A., Stecina, K., Shevtsova, N.A., McCrea, D.A.: Modelling spinal circuitry involved in locomotor pattern generation: insights from the effects of afferent stimulation: modelling afferent control of locomotor pattern generation. *J. Physiol.* **577**(2), 641–658 (2006). <https://doi.org/10.1113/jphysiol.2006.118711>. <https://onlinelibrary.wiley.com/doi/10.1113/jphysiol.2006.118711>
24. Seth, A., et al.: OpenSim: simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS Comput. Biol.* **14**(7), e1006223 (2018). <https://doi.org/10.1371/journal.pcbi.1006223>

25. Shadmehr, R., Wise, S.P.: *The Computational Neurobiology of Reaching and Pointing: A Foundation for Motor Learning*. Computational neuroscience, MIT Press, Cambridge (2005)
26. Spencer, S.E.: Accelerating adaptation in the adaptive Metropolis-Hastings random walk algorithm. *Aust. New Zealand J. Stat.* **63**(3), 468–484 (2021). <https://doi.org/10.1111/anzs.12344>. <https://onlinelibrary.wiley.com/doi/10.1111/anzs.12344>
27. Stapor, P., et al.: PESTO: parameter ESTimation TOolbox. *Bioinformatics* **34**(4), 705–707 (2018). <https://doi.org/10.1093/bioinformatics/btx676>. <https://academic.oup.com/bioinformatics/article/34/4/705/4562504>
28. Szczecinski, N.S., Hunt, A.J., Quinn, R.D.: A functional subnetwork approach to designing synthetic nervous systems that control legged robot locomotion. *Front. Neurobot.* **11**, 37 (2017). <https://doi.org/10.3389/fnbot.2017.00037>. <http://journal.frontiersin.org/article/10.3389/fnbot.2017.00037/full>
29. Todorov, E., Erez, T., Tassa, Y.: MuJoCo: a physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033 (2012). <https://doi.org/10.1109/IROS.2012.6386109>. ISSN 2153-0866
30. Vousden, W.D., Farr, W.M., Mandel, I.: Dynamic temperature selection for parallel tempering in Markov chain Monte Carlo simulations. *Monthly Notices R. Astron. Soc.* **455**(2), 1919–1937 (2016). <https://doi.org/10.1093/mnras/stv2422>. <https://academic.oup.com/mnras/article-lookup/doi/10.1093/mnras/stv2422>
31. Wang, Y.C., et al.: Multimodal parameter spaces of a complex multi-channel neuron model. *Front. Syst. Neurosci.* **16** (2022). <https://www.frontiersin.org/articles/10.3389/fnsys.2022.999531>
32. Wilson, H.R., Cowan, J.D.: Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.* **12**(1), 1–24 (1972). [https://doi.org/10.1016/S0006-3495\(72\)86068-5](https://doi.org/10.1016/S0006-3495(72)86068-5). <https://linkinghub.elsevier.com/retrieve/pii/S0006349572860685>
33. Young, F.R., Chiel, H.J., Tresch, M.C., Heckman, C.J., Hunt, A.J., Quinn, R.D.: Analyzing modeled torque profiles to understand scale-dependent active muscle responses in the hip joint. *Biomimetics* **7**(1), 17 (2022). <https://doi.org/10.3390/biomimetics7010017>. <https://www.mdpi.com/2313-7673/7/1/17>