A Dynamic Neural Network Designed Using Analytical Methods Produces Dynamic Control Properties Similar to an Analogous Classical Controller

Wade W. Hilts, Nicholas S. Szczecinski, Roger D. Quinn, and Alexander J. Hunt

***Abstract*—Human balance is achieved using many concurrent control loops that combine to react to changes in environment, posture, center of mass and other factors affecting stability. Though numerous engineering models of human balance control have been tested, no methods for porting these models to a neural architecture have been established. It is our hypothesis that the analytical methods we have developed, combined with classical control techniques will provide a reasonable starting point for developing dynamic neural controllers that can reproduce classical control capabilities. In previous work, we tested this hypothesis and demonstrated that a biologically constrained neural controller that replicates human balance control characteristics is achievable. The objective of the work presented in this letter was to further understand how parameters within the neural model affect stability and correspond to expected changes predicted by classical control theory. We compare the performance between the neural and classical engineering models for bipedal balance in an inverted pendulum balance experiment. We then carry out an extended analysis on the performance of the neural controller by varying neural parameters, observing the changes in system dynamics, and comparing these changes to those predicted by the classical model. Our methods generate compact neural systems with few parameters, all of which are correlated to classical engineering parameters for a given control model. This works serves as a basis for how to port classical control problems to neural control architectures.**

***Index Terms*—Biological control systems, biological neural networks, biological system modeling.**

I. Introduction

IN RECENT years, neuromorphic computing chips with promises to revolutionize computing technology have become available, however, there are few synthetic neural control algorithms that can utilize them. These chips effectively model neurons and synapses in a compact architecture that consumes orders of magnitude less power than a comparable digital system [1], [2]. Most synthetic neural research has been focused around pattern recognition, image processing, or decision making [3]–[5]. Almost all of these systems do not require quick reactions to external changes or interaction with an unpredictable environment.

For dynamic non-linear control situations, neural nets have been shown to provide effective control in the face of unknowns [6]–[8]. However, although these systems have abstract roots in biology, they are engineered solutions to control problems. When they are applied to models of biological control, they provide little to no insight into how an animal might be solving the control problem. To bridge this gap, more dynamic controllers based directly from neurobiology have been developed and applied to robotic systems. A few neural controllers that must dynamically interact with their environment have been developed for legged systems [9]–[11]. These controllers quickly process sensory data and take action to maintain effective interaction with the surrounding environment. However, these neural systems are built on individual case studies, and though insights can be gathered from how the control systems worked, they are not easily portable to new problems or systems.

To this end, tools that assist in creating neural controllers for new systems have been developed. For example, Nengo provides methods to set up spiking neural systems and then train them to produce specific desired outputs [12]. We have crafted methods in which parameters in small neural systems can be set analytically to perform mathematical operations such as addition, subtraction, multiplication, division, differentiation, and integration [13]. These different subnetworks can be developed and tuned independently and then connected together to perform complex, multipart mathematical or control operations.

Put into broad terms, the objective of this letter is to take an existing engineering model of biological control, and

Manuscript received May 31, 2018; revised August 23, 2018; accepted September 5, 2018. Date of publication October 30, 2018; date of current version November 12, 2018. This work was supported by NSF under Grant IIS-1608111. This brief was recommended by Associate Editor M. Arcak. *(Corresponding author: Wade W. Hilts.)*

W. W. Hilts and A. J. Hunt are with the Department of Mechanical and Materials Engineering, Portland State University, Portland, OR 97213 USA (e-mail: whilts@pdx.edu).

N. S. Szczecinski and R. D. Quinn are with the Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH 44106 USA.

Digital Object Identifier 10.1109/LCSYS.2018.2871126

reproduce this model's behavior using simulated biological circuitry (neurons and synapses). This approach is novel, no simulated neural network with biologically realistic parameters and analogous classical control model parameters has been demonstrated to perform control with similar dynamics observed in biological test subjects. The development of these networks is a detailed hypothesis of what types of networks may be required to generate this performance in the animal. We show how it is possible for simulated networks of neurons with biologically realistic parameters to reproduce behavior observed in biological testing.

It is our hypothesis that the analytical methods we have developed, combined with classical control techniques will provide a reasonable starting point for developing dynamic neural controllers that can reproduce classical control capabilities. In previous work, we tested this hypothesis by developing a neural controller that is analogous to a classical control model fit to human test subject data [14]. In this system, a PD controller with time delay and low-passed positive feedback uses corrective torque to keep an inverted pendulum system upright under perturbations. Parameters in the system are first calculated using classical control methods. Then, our analytical methods are applied to the system to determine neural network connectivity and parameters needed to replicate the classical control results. We present here an expansion of this letter by varying network parameters and demonstrating that network control results also vary as predicted by classical control theory. This opens the door for porting these systems to biologically analogous hardware and materials, such as a neuromorphic computing chip, or even a lab controlled growth of real neurons for use in organismal robotics [15], [16].

II. Methods for System Identification and Controller Design

A. A Linear Model for Human Balance

The human balance controller in this letter was based on a model derived by Peterka from human test subject data collected on a tilting platform [17]. The test subjects in Peterka's experiment had profound vestibular loss, and the data was collected with their eyes closed, thus only proprioceptive feedback was available. Additionally, the subjects were strapped to a fixture that allowed them to only use corrective torque at the ankle joint. This experiment effectively eliminated the contribution of vestibular and visual feedback while constraining the corrective output to torque at the ankle joint.

Frequency response data points were collected for these test subjects and Peterka proposed a control architecture to fit the test data. The plant model for the human body consisted of a simple inverted pendulum model, free of any damping effects. He also proposed a model for the control response that includes a positive torque feedback with a low-pass filter, a time delayed PD controller in the standard controller position, and a feedforward controller modeling the passive muscle dynamics (unaffected by the time delay) [17]. Peterka's results provide an engineering control model that has been tuned to match human test data in simulation on an inverted pendulum plant model. This engineering model was used as the basis for the neural control structure in this letter.

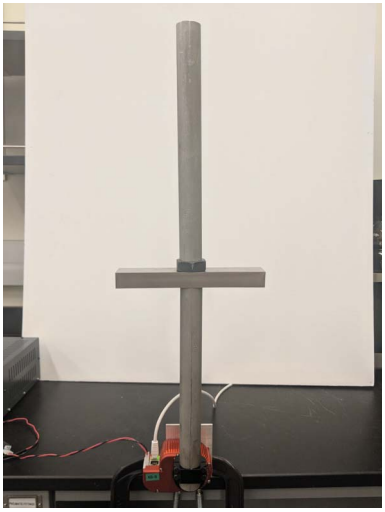

Fig. 1. Controlled system used in experimentation. This system models human balance control and is comprised of a several pieces of steel rigidly fastened together, with a torque controlled servomotor acting as the base joint.

B. Identifying the Plant Model Used in the Experiment

To model an upright human maintaining balance by applying torque at the ankles, we constructed a single jointed inverted pendulum model with a servomotor placed at the base joint (Fig. 1) modeled by the time domain differential equation:

$$J\ddot{\theta} + b\dot{\theta} - mgh\theta = T_c \tag{1}$$

where θ is the angular position, T_c is the commanded torque, J is the moment of inertia, b is the damping ratio and the mgh term is the destabilizing torque due to gravity. We assume an ideal motor that produces the commanded torque instantly, the HEBI servomotor behaves as such. We also use the small-angle approximation, $sin(\theta) = \theta$, to linearize the model for the controller design process.

System identification is performed using a closed loop controller because the inverted pendulum plant model is unstable for open-loop position control. The moment of inertia, destabilizing torque due to gravity of the system were measured before the experiment. These values were found to be $J = 0.44$ kg·m^2, and $mgh = 9.5$ kg·m^2/s^2 respectively. The damping ratio was derived from test data. A proportional controller was used to experimentally determine the gain and phase shift of the system output with a closed loop transfer function of the form:

$$\frac{\theta_{act}}{\theta_{des}} = \frac{K_p}{Js^2 + bs - mgh + K_p} = \frac{G_c G_p}{1 + G_c G_p} \tag{2}$$

where $G_c = K_p$ represents the proportional controller and G_p is the plant model. An 8-degree peak to peak sinusoidal commanded position signal was used. The proportional gain, K_p, was set to 30.

The servomotor used to control the system had more complex dynamics at torque values near zero and introduced damping to the system. A state space model of this system was constructed in MATLAB, and the theoretical form of the model enforced by using the *greyest* linear function fitting tool. The damping ratio parameter was set completely free and the

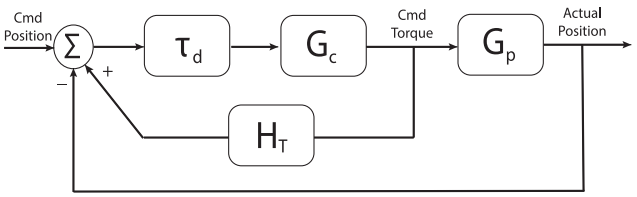

Fig. 2. Block diagram of human balance control engineering model. There are two nested feedback loops. The inner feedback loop consists of a time delayed controller receiving a low-passed positive torque feedback signal. The outer loop provides negative angular position feedback from the measured response of the plant model.

other known parameters were fixed to their measured values. The *greyest* function was used to optimize the damping ratio value to best match experimental results by minimizing the error between the model prediction and the experimental data. The damping ratio was found to be $b = 0.40$.

C. Designing a Controller to Produce a Closed Loop Response Similar to the Test Subject Data

After identifying a linear plant model for the inverted pendulum system, a controller that produces similar frequency response characteristics as the human test subjects was developed. The proposed control system takes a single input, the inverted pendulum's angular position, and outputs a corrective torque that is applied at the base joint. This control system can be represented by the block diagram in Fig. 2 and closed loop transfer function (eq. (3)):

$$\frac{\theta_{act}}{\theta_{des}} = \frac{\tau_d G_c G_p}{1 - \tau_d G_c H_T + \tau_d G_c G_p} \tag{3}$$

$$G_p = \frac{1}{Js^2 + bs - mgh} \tag{4}$$

$$G_c = K_p + K_d s \tag{5}$$

$$H_T = \frac{K_t \omega_c}{s + \omega_c} \tag{6}$$

$$\tau_d = \frac{(-\tau s + 2)}{(\tau s + 2)} \approx e^{-\tau s} \tag{7}$$

where K_p is proportional gain, K_d is derivative gain, K_t is positive torque feedback gain, ω_c is low-pass filter cutoff frequency and τ is a time delay. A first order Padé approximant was used to linearize the time delay (eq. (7)).

Using a similar method as in the plant model identification, the above transfer function was converted into state space form and plugged into the MATLAB *greyest* function. We set the time delay, low-pass filter cutoff frequency, and the proportional, derivative and torque positive feedback gains as free parameters and used the *greyest* function to minimize the error between the test subject data and the controlled system's forecasted closed loop frequency response. This process produced a controller for our inverted pendulum plant model that would emulate the response of a blindfolded human with vestibular loss on a tilting platform using only corrective torque at the ankle joints.

The classical controller was validated using MATLAB's system identification toolbox to derive the closed loop transfer function of the controlled system. We used time domain experimental data to determiine a transfer function. Filtered Gaussian white noise was sent as an input, with a low-pass filter applied that removed frequencies above 2 Hz. The max amplitude was set at 8 degrees peak to peak, which represented the majority of the operating space observed in the human data [17]. We took the input and output time domain data and fit it to a fourth order transfer function using the MATLAB System Identification toolbox. Based on the plant model and the controller design that was defined in (eq. (3)), the resulting closed loop system should be well represented by a transfer function with 4 poles and 3 zeros. The transfer function fit to this data was then compared to the human data and the theoretical prediction of the classical controller's performance.

III. Methods for Dynamic Neural Control System Design

The neural controller, shown in Fig. 3, was created by connecting a series of subnetworks to create a PD controller with a time delay and low passed positive torque feedback. The neurons and synapses in each subnetwork are assigned specific characteristics and connections to approximate the mathematical operations of the classical controller. This letter utilizes a leaky integrator non-spiking neuron model. Information is encoded in the neurons' membrane voltage, and is transmitted via synaptic connections. The membrane voltage of a neuron is governed by:

$$C_m \frac{dV}{dt} = I_{leak} + I_{syn} + I_{app}, \tag{8}$$

where C_m is the membrane capacitance, V is membrane voltage, and I_x are the various current sources and sinks. The leak current is:

$$I_{leak} = G_m \cdot (E_r - V) \tag{9}$$

where G_m is the membrane conductance. Neurons can transmit signals via synapses. This input current, I_{syn}, is defined as:

$$I_{syn} = \sum_{i=1}^{n} G_{s,i} \cdot (E_{s,i} - V) \tag{10}$$

where $G_{s,i}$ represents the synapse conductance of the ith synapse. The synapse conductance can be described by a piecewise function:

$$G_{s,i} = \begin{cases} 0 & V_{pre} < E_{lo} \\ g_{s,i} \frac{V_{pre} - E_{lo}}{E_{hi} - Elo} & E_{lo} \leq V_{pre} \leq E_{hi}. \\ g_{s,i} & V_{pre} > E_{hi} \end{cases} \tag{11}$$

Equation (11) parametrizes the range over which postsynaptic neurons receive increasing current from presynaptic neurons, after which the synapse is saturated at its maximum conductance, $g(s, i)$. E_{lo} and E_{hi} are the lower and upper thresholds of this conductance activation range. I_{app}, is an external stimulus current. For the purpose of this simulation, the external stimulus current is injected into a neuron to represent outside information, such as the angular position of the inverted pendulum model.

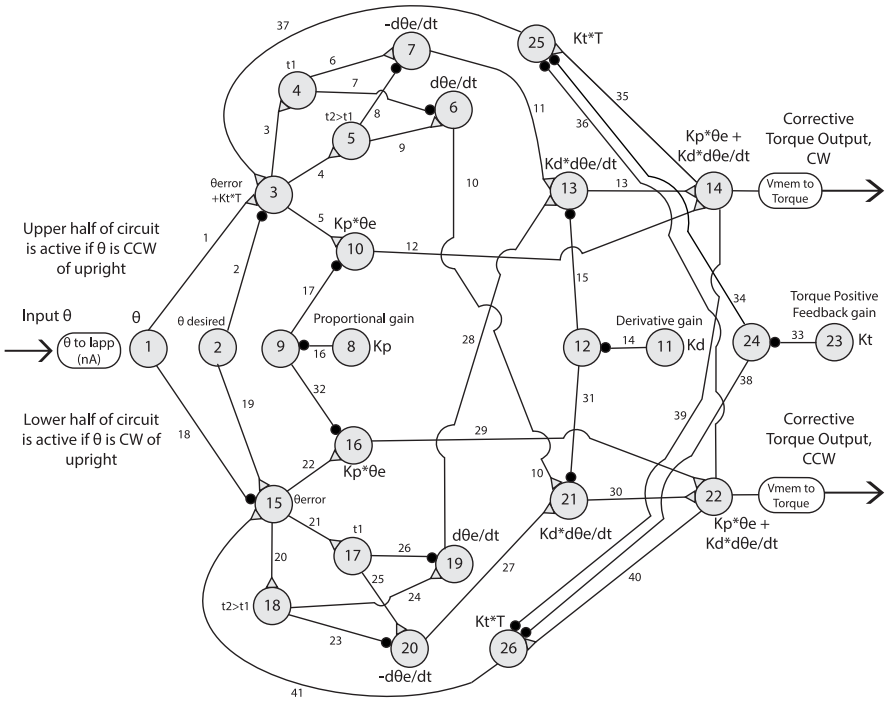

Fig. 3. A network of neurons and synapses that outputs a torque command based on a joint angle input signal. The circuit is a collection of interconnected addition, subtraction, multiplication and derivative subnetworks and is broken into two sections, each governing the clockwise and counterclockwise regions (about the marginally stable midpoint) of the pendulum system. CW and CCW torque response signals are manifest in the membrane potentials of neurons 14 and 22.

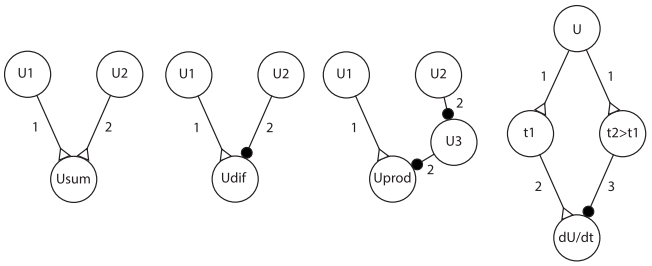

Fig. 4. Graphical representations of the neurons and synapses. From left to right: addition, subtraction, multiplication and derivative subnetworks. Synapses terminating in a triangle are excitatory, whereas the shaded circular terminals are inhibitory synapses. For details, see [13].

A. Subnetworks

A graphical representation of each subnetwork is shown in Fig. 4. For more details on the formulation of the network and parameter choices, see [13], [14], [18]. The parameters for the neural controller are for a network designed with an operating range of 20 mV. Since the selected control parameters called for large gain values that stretched the linear bounds of Szczecinski's multiplication subnetworks, we increased the synapse conductance to preamplify input signals to the multiplication network. This allows the multiplication subnetworks used in the proportional, derivative and positive feedback portions of the neural controller to function as modulating circuits, allowing a secondary gain adjustment between 0.1 and 1. Another useful tunable parameter in these subnetworks is the neuron time constant. In our model, neurons behave as RC low-pass filters where the cutoff frequency is determined by the membrane capacitance and conductance. We used this property to filter out high frequency noise from the motor position feedback signal and the commanded torque output signals. It also is used to filter the positive torque feedback signal.

The neural system control parameters were matched to the classical control parameters by hand tuning synapse conductances and multiplication neuron stimulus currents. This was validated by sending a test signal to the network and observing the output.

B. Simulating the Network in Animatlab

The subnetworks outlined above were assembled into a larger network (Fig. 3) that emulates the classical controller design. The network was simulated in Animatlab [19], an open source neuromechanical simulation tool. It provides a powerful environment based in C++ that can perform neural network simulation in real-time. It simulates the same leaky integrator non-spiking neuron model as used in Szczecinski's subnetworks [13]. Animatlab also allows the user to construct a visual model of the network and graphically represent the signal at different points throughout the circuit.

Animatlab has the ability to interface with external devices or software via a serial connection. In order to control the HEBI X8-9 torque control servomotor in the inverted pendulum system, the HEBI MATLAB API is used to send torque commands and pull feedback information from the servomotor (Fig. 5). The control loop timing was enforced by the HEBI MATLAB API, and the control loop was run at a sampling frequency of 150 Hz. Achieving faster sampling times was not limited by processing time, it was limited by the effects of 'jitter' in the Windows 10 OS. A constant $+/-$ 1 ms jitter

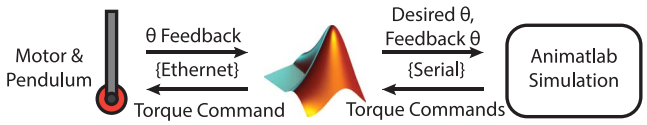

Fig. 5. Schematic representation of information flow between software and hardware platforms. MATLAB manages the communication between the motor/pendulum system and the neural controller in Animatlab.

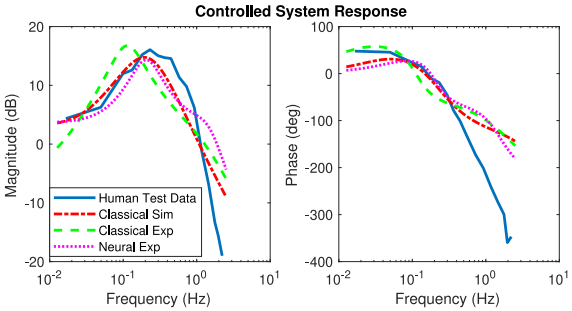

Fig. 6. Comparison of human test data with the responses of the classical controller design in both simulation and experiment, as well as the response of the neural controller in an experiment.

was observed for each sample duration, thus a slower sampling time reduced the error caused by jitter.

IV. Results

For the controllers in this letter, we collected time domain data on both the classical and neural closed loop controllers operating on the physical motor/pendulum system. Fitting a 4-poles, 3-zeros transfer function to this input/output data resulted in the highest degree of accuracy compared to other transfer function forms for both the neural and classical controllers. The controller parameters that were found to be the best fit using the *greyest* function are $K_p = 11.69$, $K_d = 1.90$, $K_t = 0.0548$, $\omega_c = 0.209$ and $\tau = 0.0774$. Tables displaying the neural system parameters selected to emulate the classical control performance are provided in the supplementary materials submitted with this letter.

The neural and classical controllers' gain and phase margin in the frequency domain matched well, having significant overlap and following similar trajectories (Fig. 6). All of the tested and simulated models agreed reasonably well with the phase and gain plots of the human test data, until they diverge near 0.4 Hz. The human data drops in gain and phase much quicker than the other models beyond this threshold. The neural system response exhibits a slight swell in gain and phase in the higher frequencies between 0.5 and 1.5 Hz that the classical controller simulation and experimental data do not show.

A. Neural Controller Parameter Sensitivity Testing Results

The neural controller has different behavior depending on the gains assigned within the network. We tested how changes in these gains affected the overall system and compared the results to the simulated classical model's prediction. The neural controller's gain was less than the simulated prediction in

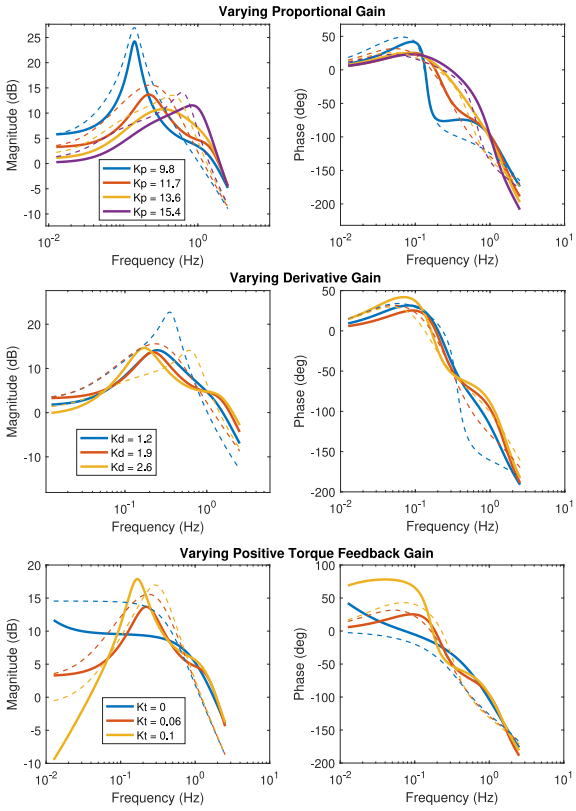

Fig. 7. The frequency response of the neural controller and motor/pendulum system is explored with various parameter gain values. The dashed lines correspond to the classical model simulation and the solid lines are the neural controller experimental results.

all cases, but the overall shape of the gain and phase plots were similar. The highest gain on the derivative gain plot disagreed most significantly from the experimental data. For clarity of understanding, the multiplication subnetwork gains controlled by a stimulus current within the network have been converted to their classical control parameter equivalents in Fig. 7.

V. Discussion

We hypothesized that the analytical methods we have developed, combined with classical control techniques will provide a reasonable starting point for developing dynamic neural controllers that can reproduce classical control capabilities. We demonstrated this by taking a classical control model fit to test subject data and capturing core dynamical features with a network of continuous time model neurons (demonstrated in the frequency domain in Fig. 6). Our experiment shows that continuous time neural systems are capable of emulating classical control for an inverted pendulum, however this process could be generalized to any physical system that can be linearly modeled. To do this, the control designer can specify the desired dynamics of the systems and perform system analysis in the well-mapped classical controls domain. The resultant controller can then be converted into a neural counterparts and tuned to produce the desired gains. These subnetworks can be connected together to produce the final desired control network with no further tuning. While the network designed in this letter succeeded in replicating control dynamics observed

in human test subjects, it also was designed with biologically realistic simulation parameters in the neural controller.

When fitting a transfer function to experimental data, it was found that a 4 poles, 3 zeros transfer function provided a better fit than other transfer function orders. This suggests that our overall theoretical transfer function (eq. (3)) for the system model was an accurate representation.

The controllers simulated and tested in this letter deviated from human test data more significantly at higher frequencies. This stems from the lack of muscle dynamics. A feedforward PD component was included in Peterka's model to account for passive muscle dynamics [17], [20], and it was intentionally omitted in this letter as it is an artifact of the physical constraints of the human body - not a deliberate control calculation.

The figures in Section IV-A that show varied gains within the network gave a qualitative display of the neural controller's versatility in modulating the inverted pendulum system's dynamics. The classical model simulation was able to predict the general behavior and effects of the neural controller parameter variation, although significant differences in magnitude were observed in some cases near the highest gains. As the system gains approach 10 dB, the $\sin(\theta) = \theta$ linearizing approximation used in the classical model begins to lose its accuracy. Most notably, this range of performance was achieved solely by adjusting the stimulus currents in the multiplication subnetworks. This provides the ability for the neural controller to update its internal gains, without changing any fundamental parameters such as synapse conductance, time constants and other biological analogs determined in the design process. Since the neural controller's behavior changed in a manner that correlated to the classical simulation, neural control design and tuning can be partially carried out using classical methods before being fine tuned in the neural form.

In future works, this neural control design process can be used to build large models of balance control with a cascaded control model built off of many neural subnetworks. Centralized vestibular and visual information can be used to determine the desired stabilizing maneuvers that are necessary to achieve the optimal posture for quality sensory feedback and minimize the likelihood of falling over [20]. Engineering models of human balance control that include vestibular and visual feedback loops, with the reliance on each of these sensory signals varying with their quality, have already been postulated [17], [21].

REFERENCES

[1] C. Mead, "Neuromorphic electronic systems," *Proc IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[2] C. D. Schuman *et al.*, "A survey of neuromorphic computing and neural networks in hardware," *CoRR*, vol. abs/1705.06963, 2017. Accessed: May 19, 2017. [Online]. Available: http://arxiv.org/abs/1705.06963

[3] M. Chu *et al.*, "Neuromorphic hardware system for visual pattern recognition with memristor array and CMOS neuron," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2410–2419, Apr. 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6894575

[4] J. A. G. Franco, J. L. D. V. Padilla, and S. O. Cisneros, "Event-based image processing using a neuromorphic vision sensor," in *Proc. IEEE Int. Autumn Meeting Power Electron. Comput. (ROPEC)*, Nov. 2013, pp. 1–6.

[5] F. Corradi, H. You, M. Giulioni, and G. Indiveri, "Decision making and perceptual bistability in spike-based neuromorphic VLSI systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 2708–2711.

[6] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

[7] J. G. Kuschewski, S. Hui, and S. H. Zak, "Application of feedforward neural networks to dynamical system identification and control," *IEEE Trans. Control Syst. Technol.*, vol. 1, no. 1, pp. 37–49, Mar. 1993.

[8] Y. Jiang *et al.*, "A brief review of neural networks based learning and control and their applications for robots," *Complexity*, vol. 2017, Oct. 2017, Art. no. 1895897. [Online]. Available: https://www.hindawi.com/journals/complexity/2017/1895897/

[9] A. Hunt, M. Schmidt, M. Fischer, and R. Quinn, "A biologically based neural system coordinates the joints and legs of a tetrapod," *Bioinspir. Biomimetic*, vol. 10, no. 5, 2015, Art. no. 055004. [Online]. Available: http://stacks.iop.org/1748-3190/10/i=5/a=055004

[10] W. Li, N. S. Szczecinski, A. J. Hunt, and R. D. Quinn, "A neural network with central pattern generators entrained by sensory feedback controls walking of a bipedal model," in *Proc. 5th Int. Conf. Living Mach. Biomimetic Biohybrid Syst.*, Jul. 2016, pp. 144–154, doi: 10.1007/978-3-319-42417-0_14.

[11] A. J. Hunt, N. S. Szczecinski, E. Andrada, M. Fischer, and R. D. Quinn, "Using animal data and neural dynamics to reverse engineer a neuromechanical rat model," in *Biomimetic and Biohybrid Systems* (Lecture Notes in Computer Science), S. P. Wilson, P. F. M. J. Verschure, A. Mura, and T. J. Prescott, Eds. Cham, Switzerland: Springer Int., Jul. 2015, pp. 211–222. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-22979-9_21

[12] T. Bekolay *et al.*, "Nengo: A Python tool for building large-scale functional brain models," *Front. Neuroinf.*, vol. 7, p. 48, Jan. 2014. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fninf.2013.00048/full

[13] N. S. Szczecinski, A. J. Hunt, and R. D. Quinn, "A functional subnetwork approach to designing synthetic nervous systems that control legged robot locomotion," *Front. Neurorobot.*, vol. 11, p. 37, Aug. 2017. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnbot.2017.00037/full

[14] W. W. Hilts, N. S. Szczecinski, R. D. Quinn, and A. J. Hunt, "Emulating balance control observed in human test subjects with a neural network," in *Biomimetic and Biohybrid Systems* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, Jul. 2018, pp. 200–212. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-95972-6_21

[15] B. C. Wheeler and G. J. Brewer, "Designing neural networks in culture: Experiments are described for controlled growth, of nerve cells taken from rats, in predesigned geometrical patterns on laboratory culture dishes," *Proc. IEEE Inst. Elect. Electron. Eng.*, vol. 98, no. 3, pp. 398–406, Mar. 2010.

[16] V. A. Webster-Wood, O. Akkus, U. A. Gurkan, H. J. Chiel, and R. D. Quinn, "Organismal engineering: Toward a robotic taxonomic key for devices using organic materials," *Sci. Robot.*, vol. 2, no. 12, Nov. 2017, Art. no. eaap9281. [Online]. Available: http://robotics.sciencemag.org/content/2/12/eaap9281

[17] R. J. Peterka, "Simplifying the complexities of maintaining balance," *IEEE Eng. Med. Biol. Mag.*, vol. 22, no. 2, pp. 63–68, Mar. 2003.

[18] W. W. Hilts, N. S. Szczecinski, R. D. Quinn, and A. J. Hunt, "Simulation of human balance control using an inverted pendulum model," in *Biomimetic and Biohybrid Systems* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, Jul. 2017, pp. 170–180. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-63537-8_15

[19] D. Cofer *et al.*, "AnimatLab: A 3D graphics environment for neuromechanical simulations," *J. Neurosci. Methods*, vol. 187, pp. 280–288, Mar. 2010.

[20] L. Asslände and R. J. Peterka, "Sensory reweighting dynamics following removal and addition of visual and proprioceptive cues," *J. Neurophysiol.*, vol. 116, no. 2, pp. 272–285, Aug. 2016. [Online]. Available: http://jn.physiology.org/content/116/2/272

[21] R. J. Peterka and P. J. Loughlin, "Dynamic regulation of sensorimotor integration in human postural control," *J. Neurophysiol.*, vol. 91, no. 1, pp. 410–423, Jan. 2004. [Online]. Available: http://jn.physiology.org/content/91/1/410