Improving a balance control neural model through numerical optimization

Stu McNeal[1], Alexander Hunt[1]

[1]Department of Mechanical and Materials Engineering, Portland State University

Portland State University

Agile & Adaptive Robotics Lab

Poster # 13348

C³NS NeuroNex

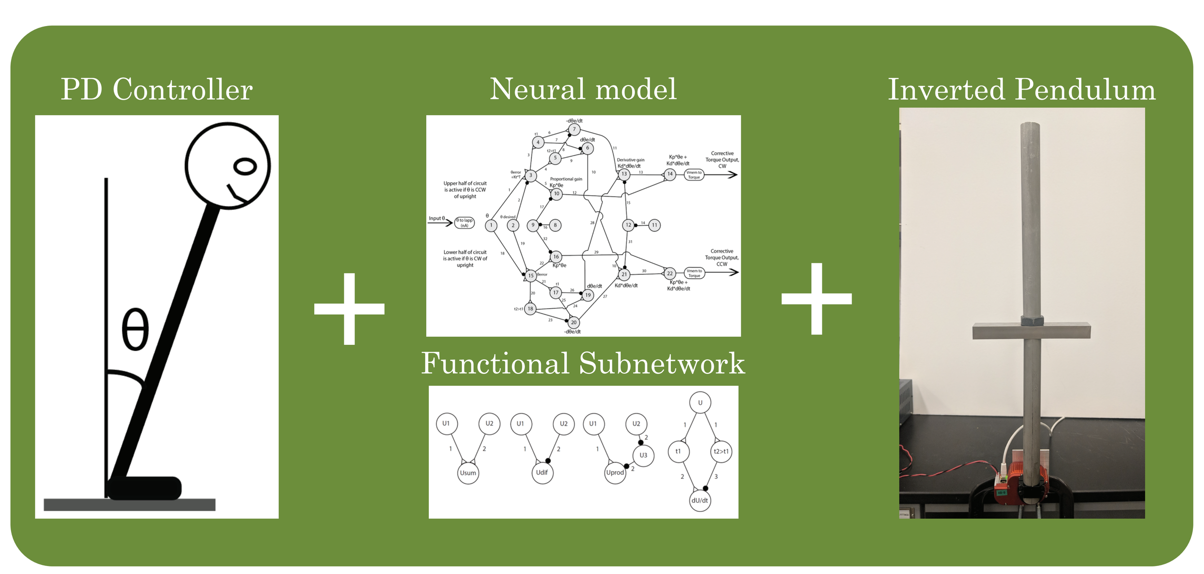
Abstract

A primary goal of neuromechanical models is to discern the relationships between neural networks and animal movement, including balance control. Previous work, based on a functional subnetwork approach, has generated and tested models that envision human balance control as an inverted pendulum. That work demonstrated that by pairing engineering control theory with trials of human subjects with severe vestibular deficiencies, different aspects of balance control can be decomposed and modeled by a proportional-derivative (PD) controller. Some subsequent work has demonstrated that this controller can be modeled with a relatively simple neural system. The effort used a hand-tuned real-time neural model to drive a single, torque-controlled motor, however, this previous work still fell short of fully matching the experimental data. It is our hypothesis that the existing neural model can be made to match the human data more accurately through improved parameter setting. We accomplish this using numerical optimization techniques in PyTorch to set connection strengths and neural timve constants within the model. Here we present our methods and an assessment of the impact of training by simulations and real-time simulations of the inverted pendulum.

Research Goal

Our work seeks to add to the body of knowledge by implementing numerical optimization to a neuromechanical AnimatLab [5] model that approximates human balance control from [1, 2]. This pathfinding exercise will allow us to implement optimization strategies in order to test/validate models used to control our biologically-inspired lab robots.

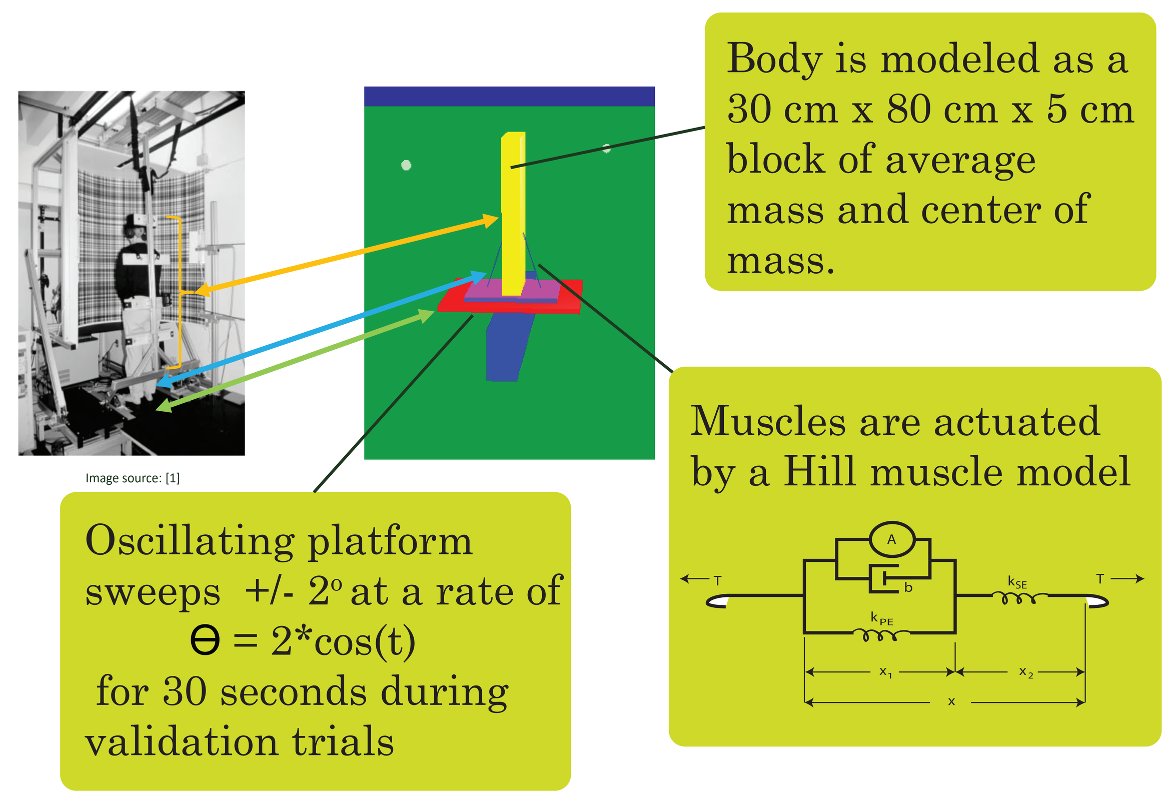
Previous Work



Our previous work [3] implemented the neural network shown using a functional subnetwork approach [4] that envisions neural clusters as mathematical operators. In this effort, the subnetwork approach provided a starting point for neuron/synapse parameterization. The model implements key elements of a proportional-derivative controller

Physical Model

A model of a human patient atop an oscillating platform was generated in AnimatLab in order to approximate experimental conditions described in [2], and to validate the optimized neural model.

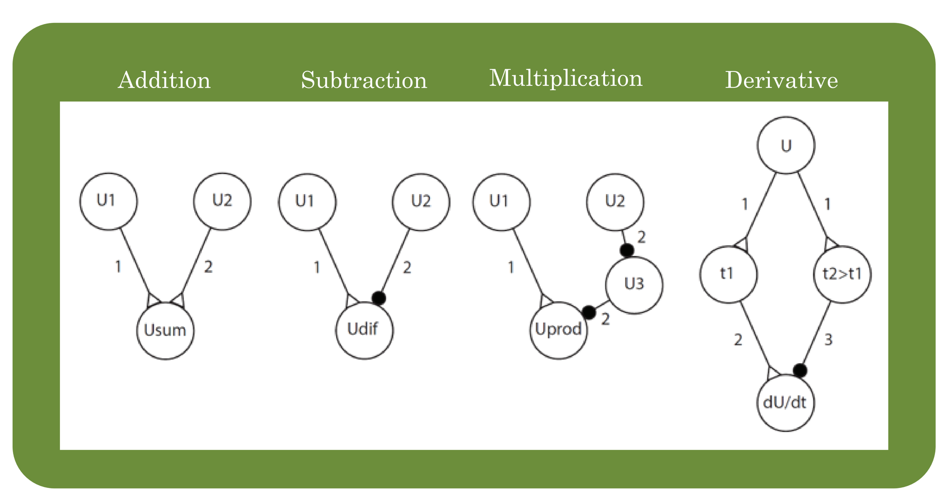


Optimizations

The central focus of this study is to add particle swarm optimization (PSO) to a neural model with parameter values derived from a functional subnetwork approach (FSN) [4].

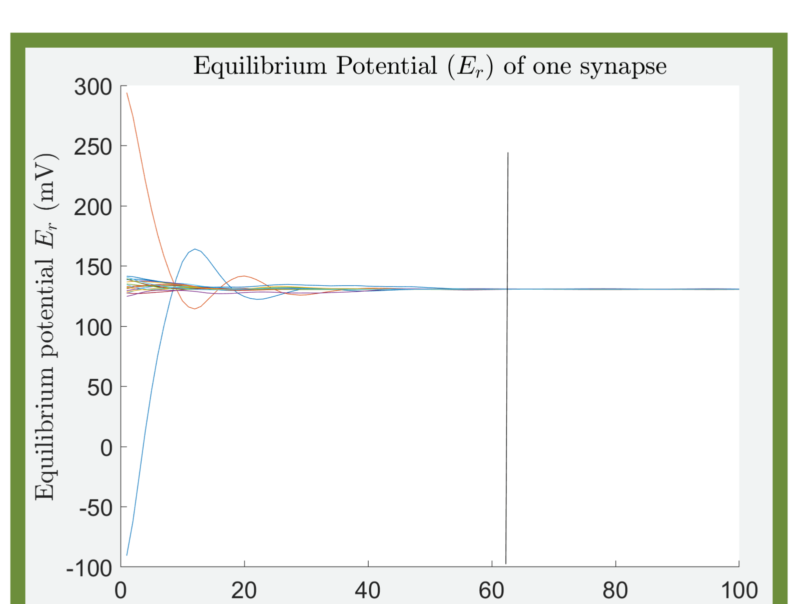
Functional Subnetworks (FSN)

FSN envisions small neural clusters as mathematical operators and leverages inherent constraints to establish neural/synaptic parameter values.



Particle Swarm Optimization (PSO)

Once the FSN approach is applied, PSO quickly fine-tunes the solution

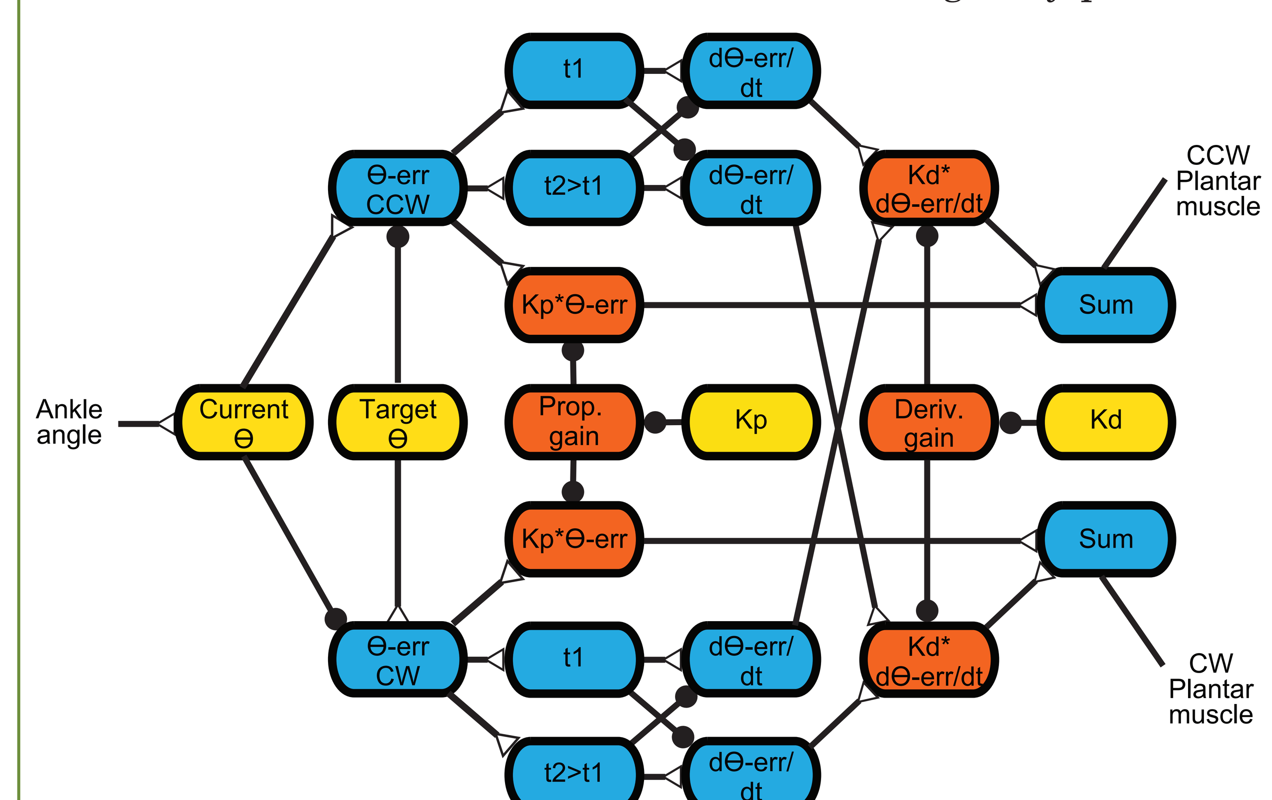


Each particle represents a discrete configuration of parameter values, adjusted incrementally with each epoch

Neural Model

This study adapts a previous synthetic nervous system (SNS) model [3] in AnimatLab based on the mathematical model distilled in [1]. The model recieves the Ankle angle as an input and outputs directional plantar musle actuation.

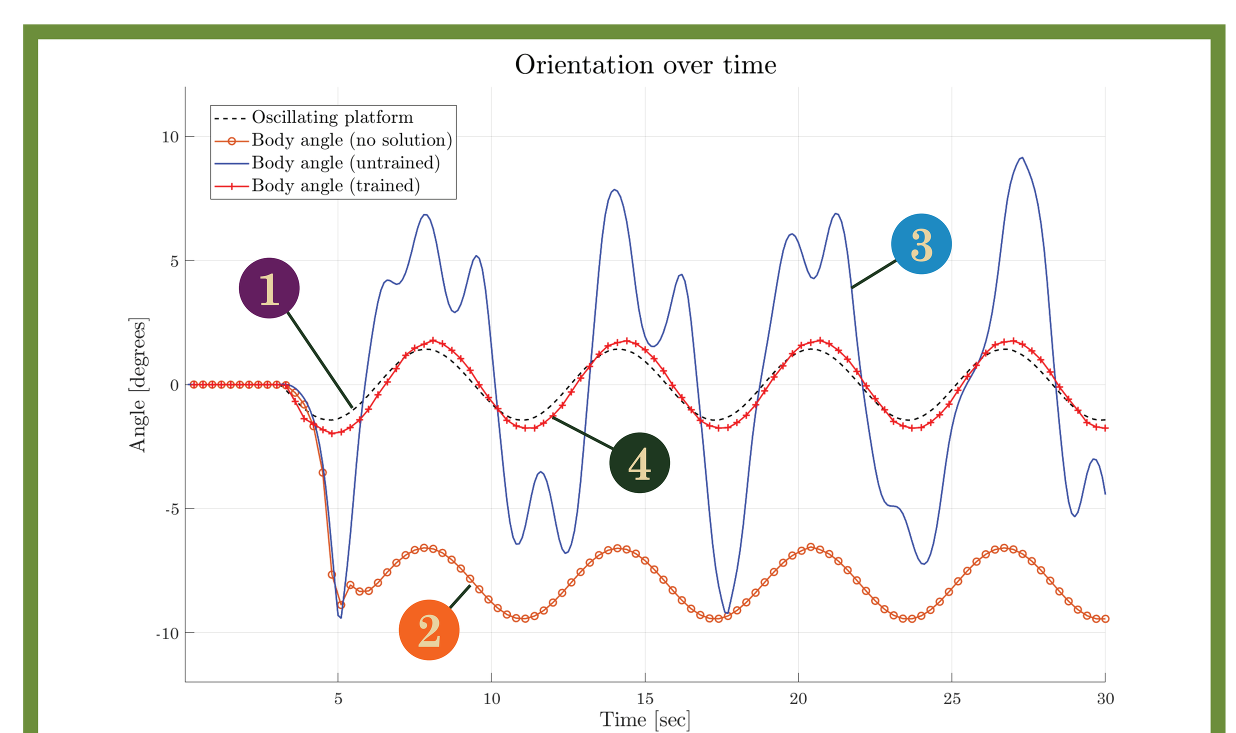
- Model symmetry divides the signal processing according to which muscle will be actuated
- Although the model is a pure representation of the control equations, it remains biologically plausible



- Target ϴ represents default (0-degrees) ankle angle that the model attempts to maintain throughout the simulations

Results

1. The error used by the PSO is the ankle angle, equivalent to the difference between the platform and the body
2. Prior to FSN, the model body falls to one side and remains pitched for the remainder of the simulation



3. FSN allows the model to actuate the muscles, indicating partial solution that overshoots the target response
4. The PSO solution closely follows the platform, likely more responsive than the human balance control

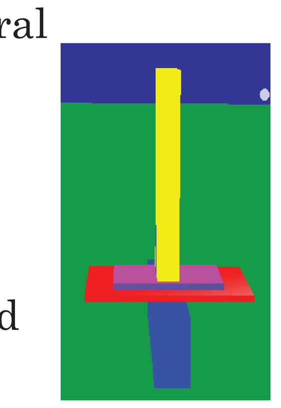
Conclusions

- A FSN-PSO optimization strategy can be applied to a neural model to improve throughput perfomance
- The FSN-PSO strategy overcomes uncertainty in FSN and PSO's need for proximal solutions
- FSN-PSO produces results at a fraction of the computational cost of gradient-centered strategies
- Human balance control data can serve to guide neuromechanical modeling efforts

Next Steps

We will incorporate positive force feedback to the neural model to more closely approximate the model deduced in [2].

The particle swarm optimization will be tuned so that the model performance matches human balance control data.



Alternative evolutionary optimizers, as well as Hebbian optimization strategies, will be deployed and compared with the results of this study.

Acknowledgements

This work was supported by NSF DBI 2015317 as part of the NSF/CIHR/DFG/FRQ/UKRI-MRC Next Generation Networks for Neuroscience Program. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Peterka, Robert J. "Sensorimotor integration in human postural control." Journal of neurophysiology 88.3 (2002): 1097-1118.
[2] Peterka, Robert J. "Simplifying the complexities of maintaining balance." IEEE Engineering in Medicine and Biology Magazine 22.2 (2003): 63-68.
[3] Hilts, Wade W., et al. "A dynamic neural network designed using analytical methods produces dynamic control properties similar to an analogous classical controller." IEEE Control Systems Letters 3.2 (2018): 320-325.
[4] Szczecinski, Nicholas S., Alexander J. Hunt, and Roger D. Quinn. "A functional subnetwork approach to designing synthetic nervous systems that control legged robot locomotion." Frontiers in neurorobotics 11 (2017): 37.
[5] Cofer, David, et al. "AnimatLab: a 3D graphics environment for neuromechanical simulations." Journal of neuroscience methods 187.2 (2010): 280-288.