SABR: Development of a Neuromorphic Balancing Robot

Alec Yen, Yaw Mensah, Mark Dean

Introduction

Background

DANNA 2 (Dynamic Adaptive Neural Network Array) is a brain-inspired computing model developed by the TENNLab at the University of Tennessee, Knoxville. It is the successor the original DANNA, a neuromorphic hardware implementation [1]. The model follows a spiking neural network model using neuromorphic elements of neurons and synapses [2].

The models are trained using evolutionary optimization (EO). A population of random networks are evaluated using a designed fitness function. Each iteration, the networks with the highest fitness are combined and mutated to create a new population [2].

Research Goals

Design a two-wheeled, self-adjusted balancing robot (SABR)

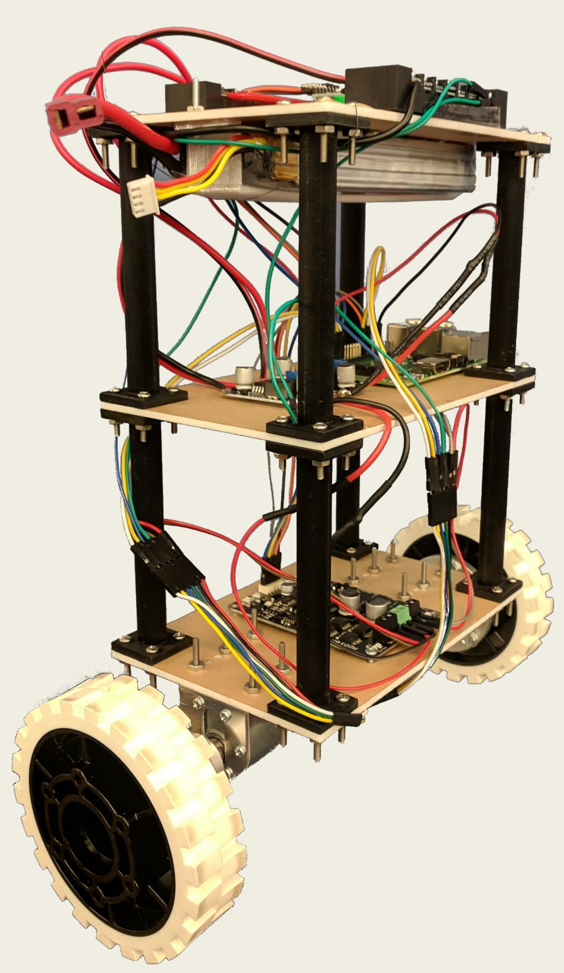
Balance the robot using traditional control algorithms

Balance the robot using a neuromorphic model (DANNA 2)

Design and Modelling

Components

- Lithium Polymer 12V Battery
- Inertial Measurement Unit
- 12V DC Motors
- Raspberry Pi 3B+
- PYNQ FPGA



Sensor Data

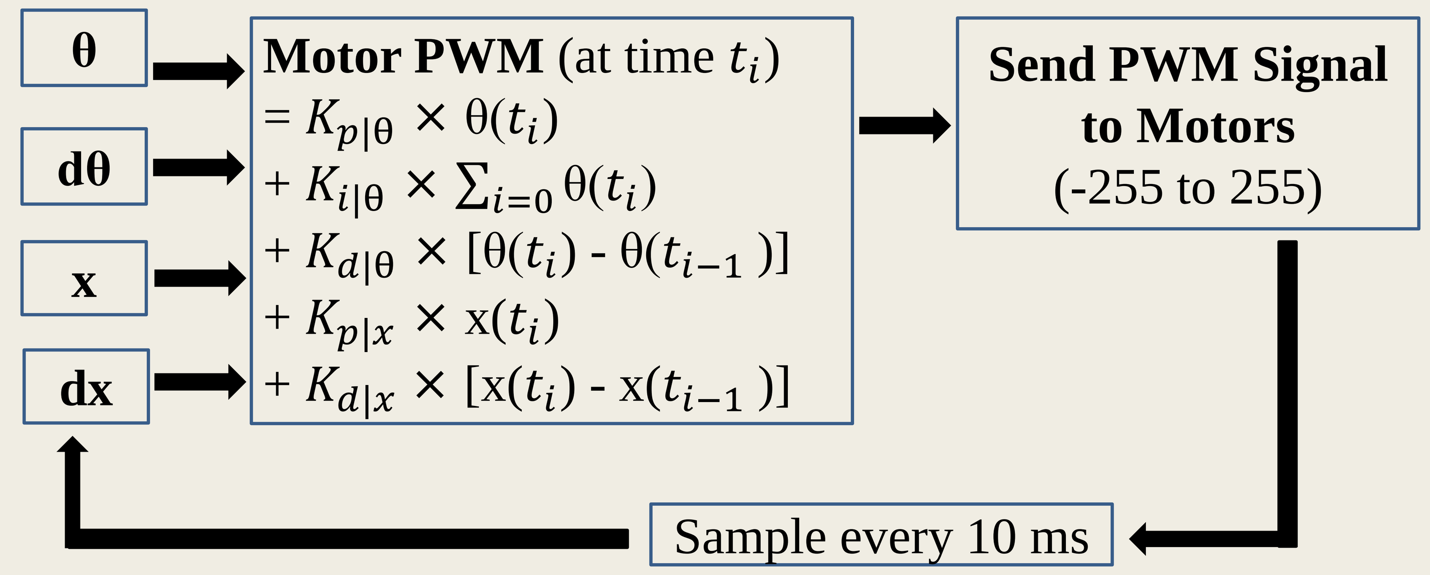
- Angular Position θ
- Angular Velocity dθ
- Translational Position x
- Translational Velocity dx

Mathematical Model

The two-wheeled balancing robot can be loosely categorized as an inverted pendulum, with the addition of DC motor characteristics.

The velocity and acceleration of the pendulum system and of the wheels were characterized based on balancing robot studies [3]. Using these and the Euler method, the system was simulated in code.

Traditional Implementation



PID Loop

The inverted pendulum problem is a classic problem from control theory. Like many control problems, it can be solved traditionally using a proportional-integral-derivative (PID) loop (depicted above).

The robot balances in a very stable manner, even returning to its original x-position after being pushed.

The Network

DANNA 2 Input

The network's input is rate-coded, meaning it uses multiple pulses of the same weight rather than a single pulse of different weights. Each input has 2 neurons (or bins). There is one bin for negative values and positive values. The raw scalar values are converted to 1 to 10 pulses.

DANNA 2 Output

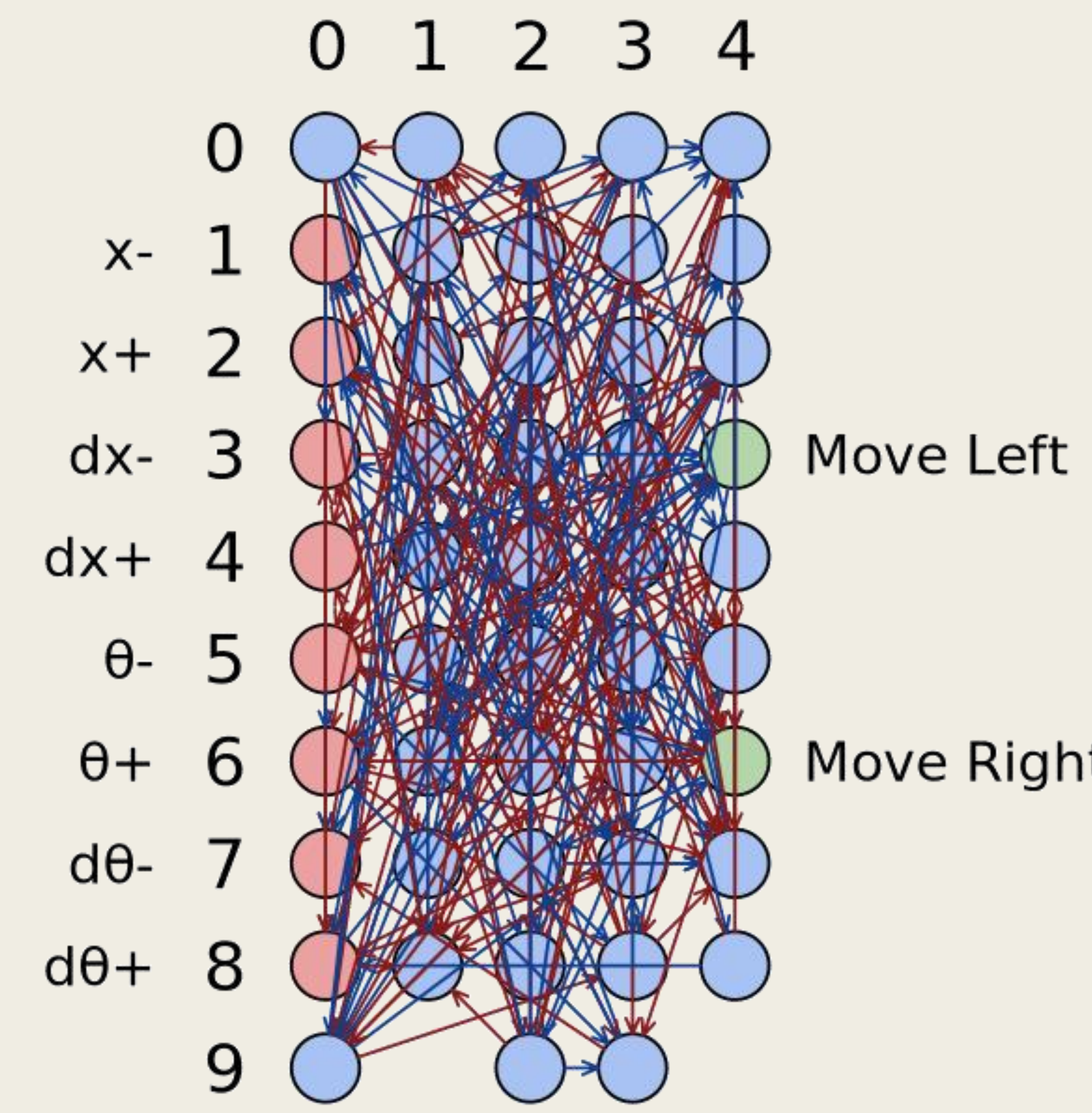
The motor speed is determined by 2 output neurons using a vote count. The number of times each neuron pulses in a given network run time indicates the number of "votes" to move left or move right. The difference is found and scaled to determine the duty-cycle (PWM) to drive the motors. There are 17 discrete bins of possible PWM outputs.

DANNA 2 Training

The network was trained to encourage 3 parameters:

- Survival - The network should balance in the model for as long as possible (a max of 5 minutes in simulation was established).
- Stability - The network should minimize oscillations.
- Symmetry - The network should provide opposite output given opposite input.

Neuromorphic Implementation



Simulated Network Implementation

Using the mathematical model, networks were trained to balance the system in simulation. Different networks were implemented and tested on a Raspberry Pi. The network above indicates the best model trained thus far. The network demonstrates correct balancing response and can balance for a brief period of time (around 5 seconds). However, the network has not provided strong enough motor output to keep the system from falling at larger angles.

Conclusions and Future Work

We successfully implemented control algorithms to balance a robot using a traditional implementation. Neuromorphic implementation has been trained to balance in simulation, however the physical application has not yet successfully balanced.

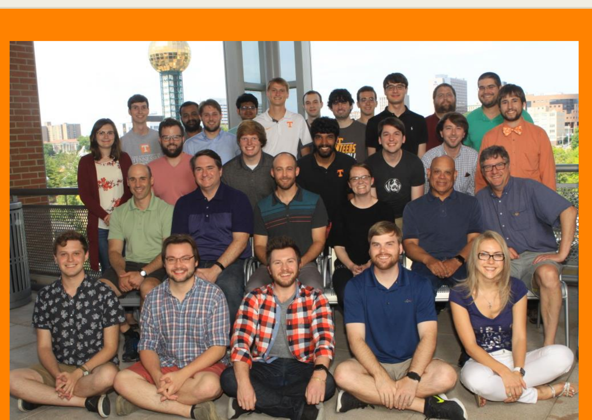
- The nature of the discrete output bins of the neuromorphic implementation limits the ability of the robot to balance itself with higher resolution.
- Refining the mathematical model to incorporate friction, slippage, and other imperfections could benefit the neuromorphic implementation's accuracy.

Future work will also include implementing the neuromorphic model on a PYNQ FPGA by generating and loading the FPGA with a Linux-based operating system.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Literature Cited

1. J. P. Mitchell, M. E. Dean, G. R. Bruer, J. S. Plank, and G. S. Rose. "DANNA 2: Dynamic adaptive neural network arrays." Proceedings of the International Conference on Neuromorphic Systems, July 2018. 10.
2. M. E. Dean, C. D. Schuman, and J.D. Birdwell. "Dynamic Adaptive Neural Network Array." Unconventional Computation and Natural Computation. Springer International Publishing, 2014. 129-141.
3. C. Sundin, and F. Torstensson. "Autonomous balancing robot." 2012.



Visit our website at neuromorphic.eecs.utk.edu