Rogues Gallery – Neuromorphic Team

James Wood, Hiren Kumawat, Jinchu Li, Diana Lalieva, Samay Chandna, Colten Webb, Austin Chemelli, Anthony Wong, Reetesh Sudhakar

Introduction & Goals

Introduction

The Neuromorphic Team specializes in neuromorphic, also known as brain-inspired, computing applications.

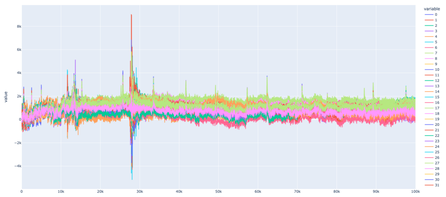
Semester Goals

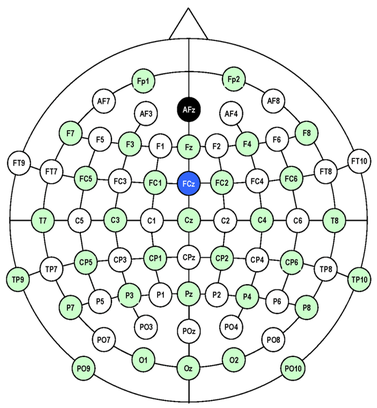
- Develop an architecture for an autonomous navigation system, with communication between components.
- Gather EEG datasets and research options for machine learning on EEG time-series data.
- Introduce new members to neuromorphic tools and concepts.

Key Concepts & Results

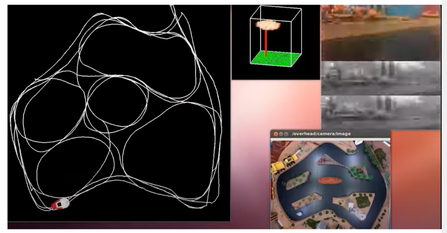
Motor Imagery Prediction via EEG Data Classification

- A low-power spiking neural network driven by an FPGA to offer real-time motor imagery classification through gathered EEG readings.
- After combing through and processing time series data, developing a simple RNN- or CNN-based DNN to then implement into a SNN to finally run on an FPGA.
- Low-power classification of motor imagery from EEG data in real-time will lead to more affordable prosthetics for amputees and paralyzed persons.





RatSLAM



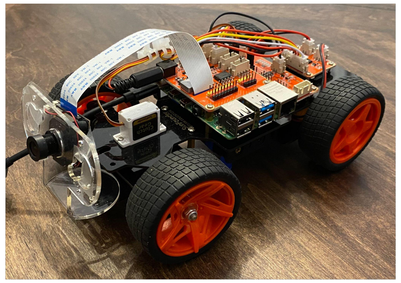
- Simultaneous Localization and Mapping (SLAM) algorithm inspired by rat brains; a neuromorphic method for mapping an environment and tracking an agent's location within it. Neuro-Car uses OpenRatSLAM.
- Developing spiking implementation of Posecell Network, representing beliefs of the current heading/location of agent, using a spiking CAN and recurrent SNN for Bayesian Filtering.

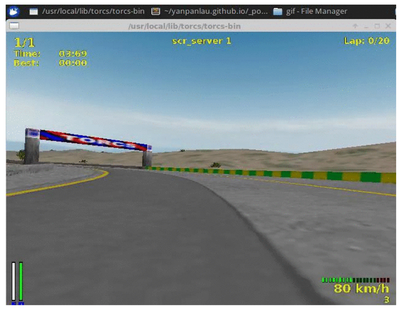
The Neuro-Car

A small-scale, low-power autonomous vehicle that utilizes spiking neural networks to power onboard decision-making.

Software / Hardware

- Raspberry Pi 4
- Raspian / Ubuntu (Docker)
- Python servo and throttle bindings from SunFounder.
- Raspberry Pi Camera Module





DDPG

(Deep Deterministic Policy Gradient)

- Takes in SLAM data and makes throttle / steering decisions.
- Training method involves the use of two pairs of networks, in order to stabilize learning.

ROS

(Robotic Operating System)

- OpenRatSLAM is written in ROS 1.
- Why need ROS2: ROS1's EOL (End of Life) is scheduled for 2025.
- ROS Bridge: compatibility between ROS 1 and 2.

Challenges & Next Steps

Challenges

- Exploring compatibility between ROS1 and ROS2.
- Containerizing our development with Docker.
- Establishing stable connection to Neuro-Car (for wireless controller, clients).

Next Steps

- Train a DDPG model to work on the NeuroCar based on OpenRatSLAM output.
- Fully implementing Posecell Network according to our architecture.
- Create an efficient network model to accurately predict movement imagined through EEG data classification.