IMPLEMENTATION OF BIOLOGICALLY INSPIRED SPIKING NEURAL NETWORK FOR CLASSIFICATION

A TECHNICAL REPORT

submitted by

ATHEENA LATISH AM.EN.U4ECE21012
GOVIND A R AM.EN.U4ECE21120
VINAYAK C M AM.EN.U4ECE21166

under the guidance of

DR.ASHA VIJAYAN , PROF.SHYAM DIWAKAR

submitted as part of

19ECE495 PROJECT PHASE II

in

ELECTRONICS AND COMMUNICATION ENGINEERING

AMRITA SCHOOL OF ENGINEERING
AMRITA VISHWA VIDYAPEETHAM
AMRITAPURI (INDIA)
May - 2025

AMRITA SCHOOL OF ENGINEERING
AMRITA VISHWA VIDYAPEETHAM
AMRITAPURI (INDIA)

BONAFIDE CERTIFICATE

This is to certify that the report entitled **”IMPLEMENTATION OF BIOLOGICALLY INSPIRED SPIKING NEURAL NETWORK FOR CLASSIFICATION”** submitted by **ATHEENA LATISH** (AM.EN.U4ECE21012), **GOVIND A R** (AM.EN.U4ECE21120) and **VINAYAK C M** (AM.EN.U4ECE21166) as part of the 19ECE495 PROJECT PHASE II is a bonafide record of the work carried out by them under my guidance and supervision at Amrita School of Engineering, Amritapuri.

Signature of the Supervisor:
DR. ASHA VIJAYAN
Assistant Professor,
Amrita Mind Brain Center, Amritapuri, India

Date: 26/05/2025

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

AMRITAPURI - 690 542

DEPARTMENT OF ECE

DECLARATION

We, **ATHEENA LATISH (AM.EN.U4ECE21012), GOVIND A R (AM.EN.U4ECE21120) and VINAYAK C M (AM.EN.U4ECE21166)** hereby declare that this technical report entitled **“IMPLEMENTATION OF BIOLOGICALLY INSPIRED SPIKING NEURAL NETWORK FOR CLASSIFICATION”,** is the record of our original work done by us under the guidance of **DR. ASHA VIJAYAN, PROF. SHYAM DIWAKAR**, Amrita Mind Brain Center. To the best of my knowledge, this work has not formed the basis for awarding any degree/diploma/ associateship/fellowship/or a similar award to any candidate in any university.

Place: Amritapuri **Signature of students**
Date: 26/05/2025

Chapter 1

Acknowledgement

It is a matter of pleasure for us to thank all the persons who have contributed directly or indirectly towards the successful completion of our project report titled “**”IMPLEMENTATION OF BIOLOGICALLY INSPIRED SPIKING NEURAL NETWORK FOR CLASSIFICATION”**”.

At the very outset, we would like to offer a token of heartfelt gratitude to the management and teachers at the Department of Electronics and Communication Engineering and the Amrita Mind Brain Center for providing us with all the necessary facilities and support for our project.

This acknowledgment would not be complete without expressing gratitude to our mentors, **DR.ASHA VIJAYAN and PROF.SHYAM DIWAKAR** of Amrita Mind Brain Centre, whose guidance and support were invaluable throughout the seminar. We also take this opportunity to thank our advisors for their care and support.

Furthermore, we extend our thanks to the staff of the Amrita School of Engineering (Amritapuri), whose support and resources facilitated our research. The conducive academic environment provided by the college has been instrumental in our intellectual growth and the successful execution of this project. We also express our gratitude to our friends and colleagues who provided encouragement, insightful discussions, and constructive feedback. Their contributions have enriched our understanding and improved the quality of our work.

We want to thank everyone who supported us during this project.

Contents

1 Acknowledgement **4**

List of Figures **iv**

2 Abstract **v**

3 Introduction **1**

4 Literature Survey **3**

4.1 Artificial cerebellum on FPGA: realistic real-time cerebellar spiking neural network model capable of real-world adaptive motor control 3

4.1.1 Key Contributions 3

4.1.2 Relevance to the Project 4

4.2 A Configurable CPG Controller using Connectome-based SNN on FPGA for Robot Locomotion 5

4.2.1 Key Contributions 5

4.2.2 Relevance to the Project 6

4.3 Towards neuromorphic FPGA-based infrastructures for a robotic arm . 6

4.3.1 Key Contributions 7

4.3.2 Relevance to the Project 7

4.4 Real-time implementation of the cerebellum neural network 8

4.4.1 Key Contributions 8

4.4.2 Relevance to the Project 8

4.5 Background of the proposed work 9

5 Proposed Methodology 12

5.1 System Overview 12

5.2 AdEx Model 14

5.3 Methodology 15

5.3.1 Development of Single Neurons 15

5.3.2 Multi-Layer SNN Construction 16

5.3.3 Spike Behavior Monitoring 18

5.3.4 Learning Algorithm Implementation 18

5.3.5 Pseudocode for Network Architecture 19

5.3.6 Pseudocode for STDP Learning Rule 20

5.3.7 Pattern Recognition and Encoding 21

5.3.8 Simulation and Analysis 21

5.4 Block Diagram of the System 22

6 Results and Discussions 25

6.1 Simulation Results 25

7 Conclusion and Future Work **33**

7.1 Scalability and Future Applications 34

List of Figures

4.1 Structure of the artificial cerebellum 5
4.2 Latency and utilization metrics on CPU and FPGA implementations . 6
4.3 Hardware platform block diagram for 6x DoF spike PID control 7
4.4 The structure of the neural network of the cerebellum model. 9

5.1 Structure of the network created 17
5.2 Block Diagram of the project 22

6.1 Spiking Pattern of the Granular cells of the network 25
6.2 Raster plot of spikes of Granular cells in the network 26
6.3 Firing rate vs Input current plot of Granular cells 26
6.4 Firing Pattern of Purkinje Cells 27
6.5 Raster plot of spikes of Purkinje Cells 27
6.6 Firing rate vs input current plot of Purkinje Cells 28
6.7 Network Output of Pre-Learning 28
6.8 Network Output of Post-Learning 29
6.9 Rastor Plot of Pre-Learning Network 30
6.10 Rastor plot of Post-Learning of the Network 31

Chapter 2

Abstract

A potent class of biologically inspired models that accurately replicate neuronal processes over time are Spiking neuronal Networks (SNNs). In this research, we suggest and put into practice a multi-layered SNN architecture for motor articulation control that is inspired by the cerebellum. In order to replicate the functional organization of the cerebellum, the network was first developed as a single adaptive exponential integrate-and-fire (AdEx) model neuron. This neuron was then grown into a three-layer hierarchical structure made up of 50 mossy fibers, 250 granular cells, and 5 Purkinje cells.

The system combines computational learning methods with spiking behavior that is biologically realistic. The network was able to identify and encode motor control patterns through the use of both supervised and unsupervised learning algorithms. Weight adaption and network-level spike simulations were used to investigate the system's plasticity and dynamic response to different motor inputs.

By utilizing SNNs' innate efficiency and learning capabilities, this architecture illustrates the promise of cerebellar modeling in neuromorphic motor control applications. Our findings provide a basis for future real-time robotic or prosthetic applications by demonstrating how physiologically inspired neuron models can aid in the creation of reliable, adaptive motor control systems.

Chapter 3

Introduction

Neuromorphic engineering has become an important field in recent years, using the structure and functions of the brain as inspiration to create computing systems that are effective, flexible, and scalable. The cerebellum is one of the brain's many regions that is essential for motor control, coordination, timing, and fine-tuning of both motor and cognitive activities. As such, it is a perfect model for designing neuromorphic systems, especially for tasks that need precise motor articulation.

The application of a cerebellar-inspired Spiking Neural Network (SNN) to regulate motor articulation is the main goal of this project. Compared to conventional artificial neural networks, SNNs simulate the temporal dynamics of biological neurons, enabling more realistic and effective neural processing. Using this biologically realistic method, we created a multi-layered SNN based on the Adaptive Exponential Integrate-and-Fire (AdEx) model. This was then developed into a structured neural network with five Purkinje cells, 250 granular cells, and fifty mossy fibers that closely resembles the architecture of the cerebellum.

The system consists of an output interface for motor control, input encoding, and a central processing layer of interconnected spiking neurons. The network was able to learn, identify, and encode motor patterns by utilizing both supervised and unsupervised learning techniques. This allowed the network to adapt to different motor inputs in a manner that was inspired by biology. In order to observe and evaluate dynamic spiking behavior and learning performance, the study also included significant weight adaptation and spike-based simulations.

In comparison with earlier hardware implementations that used simplified Verilog LIF models, this effort places a higher priority on biological accuracy by utilizing modified LIF neurons in conjunction with more intricate neuron models like AdEx and Izhikevich. These models offer better accuracy and learning potential, which are essential for simulating the cerebellum's complex role in motor function, despite their higher processing demands.

By showing how cerebellum-inspired spiking networks may be applied to precise motor control and adaptive movement creation, this work seeks to close the gap between neuroscience and robotics. Future neuromorphic systems in robotics, prosthetics, and brain-machine interfaces that require real-time learning and decision-making skills can benefit from the insights gathered from this effort.

Chapter 4

Literature Survey

4.1 Artificial cerebellum on FPGA: realistic real-time cerebellar spiking neural network model capable of real-world adaptive motor control

[?] An FPGA-based development of a biologically realistic cerebellum SNN with real-time motor control is presented in this study. Important components of the cerebellar microcircuitry, such as Purkinje cells, granule cells, and mossy/climbing fiber inputs, are replicated in the model.

4.1.1 Key Contributions

- Real-World Motor Adaptation: Utilizing robotic hardware for dynamic motor coordination and learning in a changing environment, the artificial cerebellum is evaluated in real-world adaptive control tasks.

- Real-Time Performance on FPGA: The author's implementation of the model on FPGA allowed for high-speed, low-latency spiking computation, which allowed for effective and continuous real-time adaptation that was appropriate for robotics' closed-loop control.

- Hardware Resource Optimization: A scalable design that supports accuracy and adaptability is made possible by the emphasis on the efficient use of FPGA resources, such as memory and logic blocks.

4.1.2 Relevance to the Project

This study demonstrates a biologically realistic cerebellar SNN implemented on FPGA for real-time adaptive motor control. Its architecture—featuring mossy fibers, granule cells, and Purkinje cells—closely aligns with the network structure used in the project. The successful hardware deployment underscores the practicality of cerebellum-inspired SNNs for efficient, low-latency motor control and supports the project's emphasis on combining biological accuracy with real-world applicability.

In addition, the use of FPGA in this study provides a practical framework for future hardware implementations. While the current project focuses on simulating learning algorithms and neuronal behavior, this work demonstrates how such models can be efficiently realized in hardware, bridging the gap between simulation and real-world application. It reinforces key design goals such as low-latency spike processing, scalability, and adaptive motor control based on cerebellar neuroscience principles.

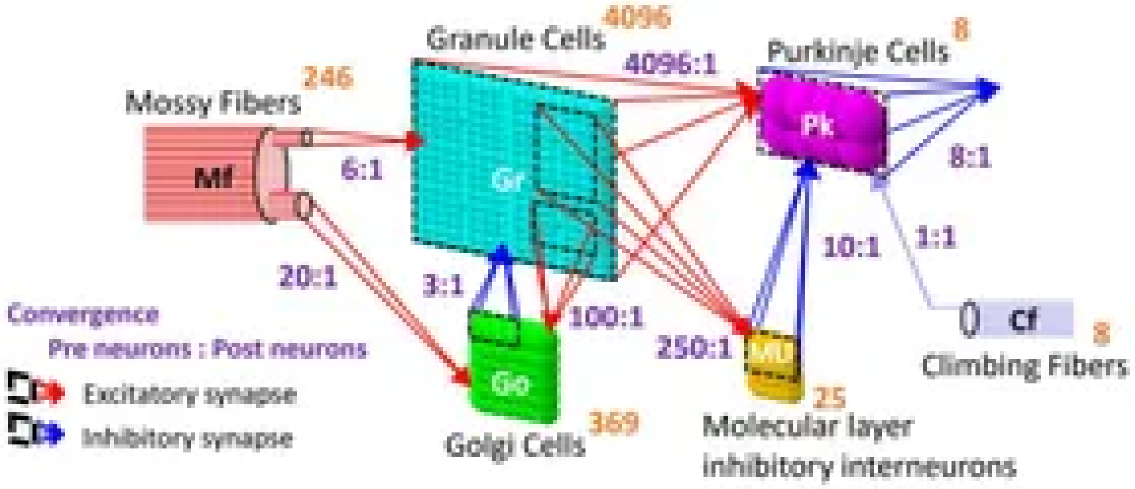

Figure 4.1: Structure of the artificial cerebellum

4.2 A Configurable CPG Controller using Connectome-based SNN on FPGA for Robot Locomotion

[?] Based on biological connectome data, the thesis proposes a spiking neural network (SNN) architecture to represent a Central Pattern Generator (CPG) controller that may be adjusted to enable rhythmic and coordinated movement for robotic locomotion.

4.2.1 Key Contributions

- FPGA Implementation: The controller is built on FPGA technology, emphasizing its low power consumption, reconfigurability, and real-time performance—all of which are essential for mobile robotic systems.

- Scalable and Modular Architecture: The architecture may be configured in a modular fashion, allowing it to be customized for robots with diverse locomotion patterns and degrees of freedom.

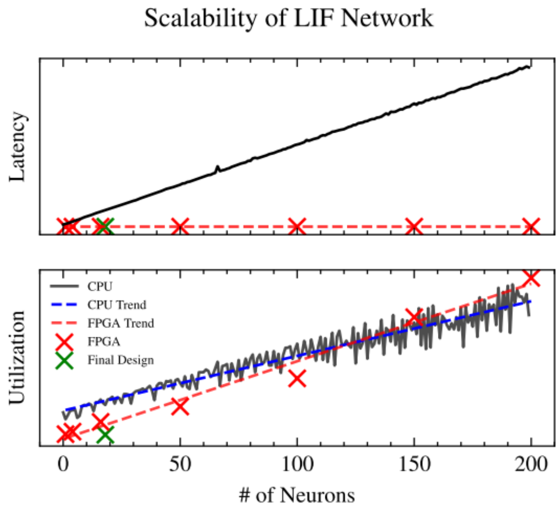

Figure 4.2: Latency and utilization metrics on CPU and FPGA implementations

4.2.2 Relevance to the Project

The development of biologically inspired motor control systems with SNNs and hardware acceleration is supported by this work. The project's objective of deploying cerebellum-inspired neural networks for motor articulation is in line with its focus on connectome-based modeling and FPGA realization. The thesis's real-time execution and adaptability offer important insights for creating scalable, effective SNN-based controllers for robotic systems.

4.3 Towards neuromorphic FPGA-based infrastructures for a robotic arm

[?] The study presents a neuromorphic FPGA-based system architecture that uses Spiking Neural Networks (SNNs) to operate robotic arms.

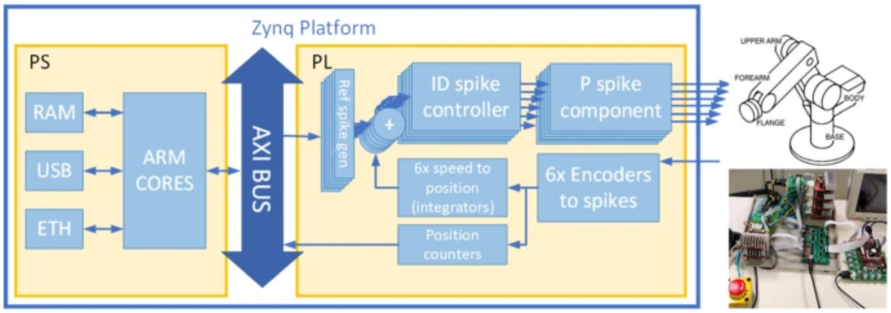

Figure 4.3: Hardware platform block diagram for 6x DoF spike PID control

4.3.1 Key Contributions

- In order to facilitate closed-loop motor control and adaptive behaviors, it highlights how sensory feedback is integrated with the spiking network.

- While preserving biologically realistic dynamics, the study offers a scalable, modular, and resource-efficient infrastructure that enables real-time data processing and motor response.

4.3.2 Relevance to the Project

The objectives of the cerebellum-inspired SNN implementation for motor articulation control are closely aligned with this work. The current project's hardware execution methodologies are supported by the usage of FPGA for the deployment of low-latency, spike-based networks. Furthermore, the focus on neuromorphic principles and closed-loop sensory feedback offers important insights into creating responsive and durable robotic control systems. The project's emphasis on scalability and biological realism in neuromorphic engineering is further supported by the modularity and real-time performance of the suggested infrastructure.

4.4 Real-time implementation of the cerebellum neural network

[?] In this paper, a cerebellum-inspired neural network designed for motor learning and correction is implemented in real-time.

4.4.1 Key Contributions

- The model uses physiologically inspired SNN structures to capture key cerebellar activities as timing, sensorimotor coordination, and error-driven adaptation.
- Simulating cerebellar plasticity and learning mechanisms, real-time hardware tests show how the system may learn and improve motor outputs dynamically in response to feedback.

4.4.2 Relevance to the Project

The study provides direct support for the use of motor control systems that are biologically based, especially those that draw inspiration from cerebellar function. Its real-time error correction and adaptive learning demonstration validates key elements of the current project's goals. The project's multi-layer network architecture and learning-based control techniques are mirrored in the usage of cerebellum-inspired SNNs for precise motor articulation. The fundamental ideas for converting cerebellar computing into efficient neuromorphic motor systems are presented in this paper.

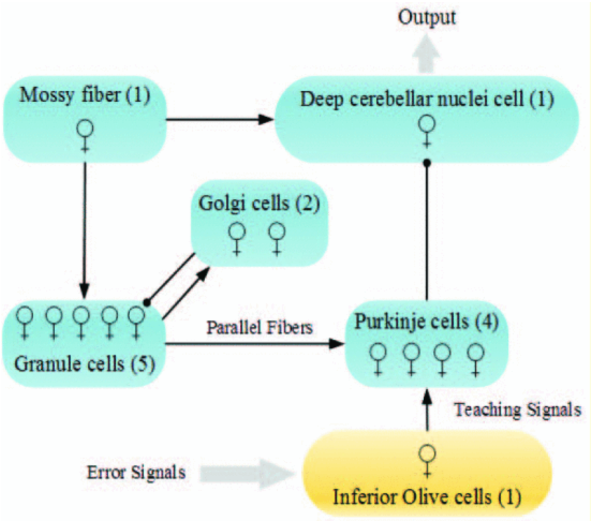

Figure 4.4: The structure of the neural network of the cerebellum model.

4.5 Background of the proposed work

The literature survey highlights the following key points:

- Realistic Cerebellar SNN Architecture: This work presents an FPGA-implemented cerebellar Spiking Neural Network architecture that is biologically accurate. Through online learning, modular microcircuitry, and synaptic plasticity, it can execute adaptive motor control in real-world scenarios, providing a solid basis for simulating intricate motor actions. It reaffirms that real-time SNN applications in embedded hardware systems are feasible.

- Connectome-Based CPG Controller: In this thesis, a connectome-informed SNN-based Central Pattern Generator (CPG) controller that can be adjusted for robot movement is presented. This work demonstrates the versatility and reusability of neuromorphic motor controllers by utilizing biologically inspired connection and translating it to hardware. It emphasizes the value of organized SNN topologies and modularity, which complement the project's cerebellar modeling methodology.

- Neuromorphic FPGA Infrastructure for Robotic Arms: The modular, neuromorphic control system on FPGA that combines SNNs and sensory feedback for real-time robotic arm control is presented in this study. In line with the project's focus on multi-layered spike processing and hardware efficiency, the study confirms that scalable, low-latency systems can allow closed-loop control and physiologically plausible behaviors.

- Real-Time Cerebellar Neural Network Implementation: Hao and associates investigate a neural network inspired by the cerebellum that can adapt and fix errors in real time. Their hardware approach successfully replicates cerebellar learning processes like motor refinement and timing. The objective of the current effort, which is to use cerebellar models for adaptive spike-driven motor articulation, is supported by this.

In combination, these findings highlight how important it is to use modular neuron layers, real-time learning mechanisms, and deliberate architectural design to balance biological plausibility with hardware limitations. Building on these discoveries, the proposed work uses AdEx and Izhikevich models to create a multi-layered cerebellum-inspired SNN, which is then simplified to a 50-250-5 neuron topology. In the end, this architecture will help close the gap between brain-inspired computing and useful robotics by supporting adaptive motor articulation control, scalable implementation, and effective spike processing.

Chapter 5

Proposed Methodology

5.1 System Overview

The proposed system simulates a Spiking Neural Network (SNN) with cerebellar inspiration that is intended for adaptive motor articulation control. Using models of biologically realistic spiking neurons, it simulates the layered architecture of the cerebellum and transfers it to a scalable, hardware-friendly framework.

- **Input Layer: Mossy Fibers (50 neurons):** Spike trains are created from sensory or pre-processed control signals. Working as afferent channels into the network, each mossy fiber transmits positional or temporal information. The dynamics of the input determine which spike encoding techniques, such as rate or latency coding, are used.

- **Processing Layer – Granular Cells (250 neurons):** These neurons encode the incoming spike patterns sparsely and dispersedly. Granular cells provide for deeper representation of temporal patterns by exploiting their divergent connections to increase the complexity of input signals. This layer's output simulates the dispersion of cerebellar signals by forming parallel fibers that protrude into the Purkinje layer.

- **Output Layer – Purkinje Cells (5 neurons):** The Purkinje cells provide spike outputs that power the matching actuators or motors by integrating signals from several parallel fibers. To enable precise motor articulation, each Purkinje neuron is linked to a particular robotic arm degree of freedom (DoF).

- Synaptic Weight Adaptation Module: The implementation of both supervised and unsupervised learning techniques allows for the dynamic adjustment of synaptic weights. The network can gradually learn and improve motor patterns thanks to reward-based feedback mechanisms for supervised learning and spike-timing dependent plasticity (STDP) for unsupervised adaptation.

- Simulation and Spike Monitoring Interface: The network is designed with hardware efficiency in mind, primarily for FPGA implementation, even if the current testing is simulation-based. In order to guarantee real-time performance, low latency, and minimal hardware resource consumption, low-complexity neuron models (such as AdEx and LIF) are selected.

- Hardware Implementation (Planned): The network is designed with hardware efficiency in mind, primarily for FPGA implementation, even if the current testing is simulation-based. In order to guarantee real-time performance, low latency, and minimal hardware resource consumption, low-complexity neuron models (such as

AdEx and LIF) are selected.

This cerebellum-inspired model achieves robust and adaptable motor control in real-time applications by utilizing technical scalability as well as biological plausibility. Future integration with actual robotic systems will be made easier by the layered structure, which permits modular testing and improvement.

5.2 AdEx Model

The Adaptive Exponential Integrate-and-Fire (AdEx) model offers high biological realism but is resource-intensive for hardware implementation. Membrane Potential Equation:

$$C\frac{dV}{dt} = -g_L(V - E_L) + g_L\Delta_T e^{\frac{V-V_T}{\Delta_T}} - w + I$$

Adaptation Current Equation:

$$\tau_w\frac{dw}{dt} = a(V - E_L) - w$$

After-spike resetting condition:

$$ifV \geq V_{th}, then : \{\, V \leftarrow V_{reset}, w \leftarrow w + b$$

- V: Membrane potential.
- C: Membrane capacitance.
- g_L: Leak conductance.
- E_L: Leak reversal potential (resting potential).

- Δ_T: Slope factor controlling the sharpness of the spike onset.

- V_T: Membrane potential threshold for the exponential term.

- w: Adaptation current.

- I: External input current.

- τ_w: Time constant of the adaptation current.

- a: Strength of the subthreshold adaptation coupling.

- V_{th}: Spike threshold potential.

- V_{reset}: Reset value for V after a spike.

- b: Spike-triggered increment in the adaptation current.

5.3 Methodology

5.3.1 Development of Single Neurons

- Verilog HDL was used to create a single neuron model based on the Adaptive Exponential Integrate-and-Fire (AdEx) model.

- To guarantee precise membrane potential dynamics and spike generation, the spiking behavior of the AdEx neuron was simulated and confirmed using various input stimuli using ModelSim.

5.3.2 Multi-Layer SNN Construction

A biologically inspired Spiking Neural Network (SNN) was developed using Verilog, based on the Adaptive Exponential Integrate-and-Fire (AdEx) neuron model. The architecture is influenced by cerebellar microcircuitry and is designed for pattern recognition and specialization through unsupervised learning mechanisms. The network mimics essential properties of real neural systems such as sparse connectivity, synaptic plasticity, and competitive firing behavior.

The architecture consists of three primary layers:

- **Input Layer (Mossy Fibers)**: Comprising 50 neurons, this layer receives two encoded spike patterns (A and B). Pattern A activates neurons 0–9, and Pattern B activates neurons 10–19. Pulsed current injections (I = 50) are applied every 20 clock cycles to simulate spiking behavior.

- **Intermediate Processing Layer (Granular Cells)**: Implemented using 250 AdEx neurons, each receiving inputs from 15 randomly selected neurons from the previous layer. This layer integrates weighted inputs and generates spikes upon reaching membrane thresholds. The sparse and random connections promote diverse and distributed spike patterns.

- **Output Layer (Purkinje-like Neurons)**: Consists of 5 neurons that receive inputs from 15 randomly selected granular cells. A Winner-Takes-All (WTA) mechanism ensures only the neuron with the strongest activation fires, simulating

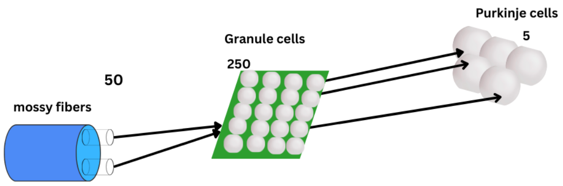

Figure 5.1: Structure of the network created

competitive learning. Spike-Timing Dependent Plasticity (STDP) is employed to strengthen synapses from active inputs and weaken others, promoting distinct specialization for patterns A and B.

The network connectivity reflects a hierarchical, feedforward structure with diverging and converging synapses, allowing efficient spike-based pattern discrimination. Over time, output neurons develop a selective preference for specific input patterns, demonstrating the network's potential in neuromorphic motor control and low-power, real-time inference systems.

5.3.3 Spike Behavior Monitoring

- To observe neuronal firing patterns, synaptic integration, and temporal synchronization, spike trains were captured and examined at every layer.

- To evaluate network dynamics, parameters like latency, frequency, and spike timing were looked at.

5.3.4 Learning Algorithm Implementation

- The network incorporates both supervised and unsupervised learning mechanisms to adapt synaptic weights dynamically during simulation. A key component is the biologically inspired Spike-Timing Dependent Plasticity (STDP), which governs how connections between neurons strengthen or weaken over time based on spike timing.

- STDP operates on the principle that the relative timing of spikes between a pre-synaptic and a post-synaptic neuron determines the direction and magnitude of weight updates. If the pre-synaptic neuron fires shortly before the post-synaptic neuron, the synapse is strengthened, a process known as Long-Term Potentiation (LTP). Conversely, if the post-synaptic neuron fires before the pre-synaptic neuron, the synapse is weakened, known as Long-Term Depression (LTD).

- In this architecture, STDP is implemented specifically between Hidden Layer 2 neurons (pre-synaptic) and Output Layer neurons (post-synaptic). Each output neuron monitors the activity of its 15 connected pre-synaptic neurons when it spikes.

  - If a pre-synaptic neuron fired recently (within the defined STDP time window), the synaptic weight is increased, reinforcing the connection.
  - If the pre-synaptic neuron either did not spike or fired outside the timing window, the weight is decreased, discouraging weak or irrelevant associations.

- This learning strategy enables output neurons to become selectively tuned to specific input patterns, promoting pattern specialization. Additionally, to prevent ambiguity in classification, output neurons are penalized if they respond equally to multiple input patterns over repeated trials.

- Through repeated training cycles, this adaptive mechanism allows the network to develop distinct, stable mappings between input spike patterns and output neuron responses, which is essential for motor articulation control and decision-making in robotic applications.

5.3.5 Pseudocode for Network Architecture

Listing 5.1: Pseudocode of the Multi-layer SNN Architecture

```
Initialize all neurons, weights, and counters
Reset the network

For each cycle from 0 to 9999:
    If current cycle belongs to Pattern A:
        Inject I = 50 to input neurons 0 9   every 20 cycles
    Else (Pattern B):
        Inject I = 50 to input neurons 10 19   every 20 cycles

    // Hidden Layer 1
    For each neuron in HL1:
        Compute weighted input from connected input neurons
        Integrate using AdEx equations
        If v     threshold:
            Generate spike

    // Hidden Layer 2
    For each neuron in HL2:
        Sum spikes from HL1 (weighted)
        Integrate using AdEx
        If v     threshold:
            Generate spike

    // Output Layer
    For each output neuron:
        Sum weighted spikes from HL2
        Compute input current

    // Winner-Takes-All (WTA)
    Identify neuron with highest current (if margin met)
    Only allow winner to receive current and potentially spike

    // STDP Update
    If winner spikes:
        For each synapse:
            If presynaptic spike present:
                Increase weight
            Else:
                Decrease weight
        If neuron has fired for both A & B multiple times:
            Penalize its weights

    Record motor output spikes and update logs

Display final spike counts per motor for Pattern A and Pattern B
```

5.3.6 Pseudocode for STDP Learning Rule

Listing 5.2: Pseudocode of STDP Weight Update Mechanism

```
On each clock cycle:

    // Track pre-synaptic spike times
    for each hidden neuron i:
        if hidden_spikes[i] == 1:
```

```
        pre_last_spike_time[i]     current_time

// For each output neuron k:
if output_neuron[k] spikes:
    post_last_spike_time[k]     current_time

    for each synapse j connected to output neuron k:
        pre_id     conn_index[k][j]
        delta_t     post_last_spike_time[k] -
           pre_last_spike_time[pre_id]

        if delta_t     0 and delta_t < STDP_WINDOW: //
           causal (pre before post)
            weights[k][j]     min(weights[k][j] + A_plus,
               MAX_WEIGHT)
        else if delta_t < 0 and |delta_t| < STDP_WINDOW: //
           anti-causal
            weights[k][j]     max(weights[k][j] - A_minus,
               0)
```

5.3.7 Pattern Recognition and Encoding

- The Purkinje cell output was used to assess the network's capacity to categorize and encode spatiotemporal spike patterns.

- The motor control signals for articulation were deduced from this spike output.

5.3.8 Simulation and Analysis

- To verify resilience, synaptic plasticity, and functional behavior, the complete multi-layer network was simulated.

- Learning convergence and spike propagation fidelity were used to gauge the network's performance.

5.4 Block Diagram of the System

The system architecture comprises the following key components:

- Input Layer: Encodes incoming sensory or spike input signals from the environment, representing mossy fiber activity.

- Central Processing Layer: Consists of a multi-layer Spiking Neural Network inspired by the cerebellum, including 50 mossy fibers, 250 granular cells, and 5 Purkinje cells, which process and transform spike patterns.

- Decoder Module: Translates the spike outputs from Purkinje cells into corresponding joint angle commands using scaling and geometric computations.

- Output Layer: Generates control signals to drive the motors of the robotic arm, enabling precise motor articulation based on decoded neural activity.

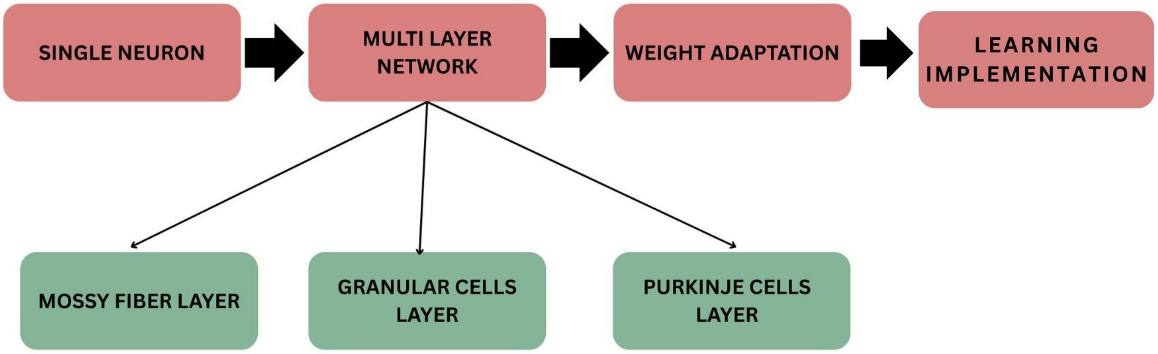

Figure 5.2: Block Diagram of the project

- Single Neuron: The foundation of the network begins with a single spiking neuron model, such as the Izhikevich or Adaptive Exponential (AdEx) neuron. This unit captures the core electrical dynamics of biological neurons and provides a scalable building block for larger networks.

- Multi-Layer Network: The network expands into a structured, cerebellar-inspired architecture comprising three primary layers:

  Mossy Fiber Layer

  Granular Cells Layer

  Purkinje Cells Layer

  This layered design mimics the cerebellum's ability to process and refine sensorimotor signals.

- Mossy Fiber Layer: This layer acts as the input interface, transmitting external or sensory information into the network. The mossy fibers relay excitatory signals to the granular cells.

- Granular Cells Layer: The granular cells receive inputs from the mossy fibers and perform input pattern expansion. This layer increases the network's dimensionality, enabling rich feature encoding and pattern separation.

- Purkinje Cells Layer: Serving as the decision-making or output layer, the Purkinje cells integrate signals from the granular layer and send inhibitory output. This layer is crucial for error correction and refined motor control in biological systems.

- Weight Adaptation: This stage involves the dynamic adjustment of synaptic weights based on neural activity and learning rules such as:

Spike-Timing Dependent Plasticity (STDP)

Weight adaptation enables the network to learn temporal and spatial patterns essential for intelligent behavior.

- Learning Implementation: The final block focuses on implementing the learned behavior or control logic using the trained network. Once weight adjustments converge, the network can be used in practical applications like:

  Pattern classification

  Motion control

  Cognitive decision-making

Chapter 6

Results and Discussions

6.1 Simulation Results

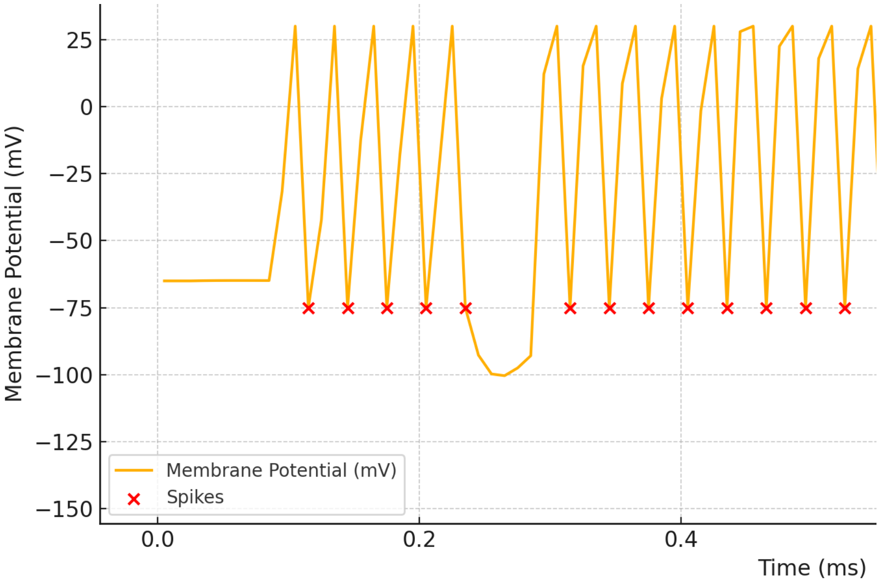

Figure 6.1: Spiking Pattern of the Granular cells of the network

As is typical of action potentials, the granular cells' membrane potential profile shows a sequence of abrupt depolarizations followed by repolarizations. The network's granular cells are highly sensitive and excitatory, as evidenced by the regular spikes that are produced in response to input stimuli. Accurate spike detection is confirmed by spike markers that line up with the membrane potential trace.

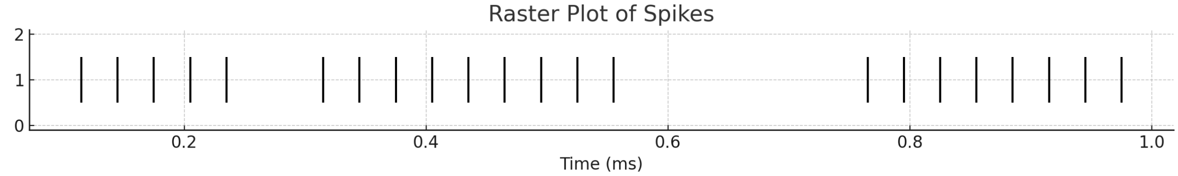

Figure 6.2: Raster plot of spikes of Granular cells in the network

The temporal distribution of spikes throughout the granular cell population is displayed in the raster graphic. The synchronous and repeating firing patterns across cells are highlighted by the vertical lines, each of which represents a spike event. These patterns show coordinated activity in response to shared input currents and support the constancy of spike creation.

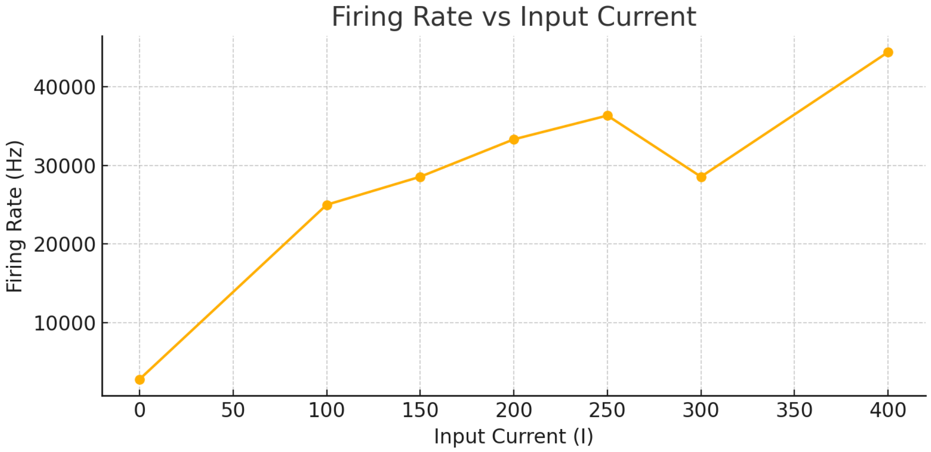

Figure 6.3: Firing rate vs Input current plot of Granular cells

The input current and the granular cell firing rate are shown to be positively correlated. The frequency encoding property of the AdEx model is demonstrated by the nonlinear increase in firing rate with increasing input current. This behavior demonstrates the granular cells' ability to transmit high-frequency impulses and their sensitivity to different stimulus intensities.

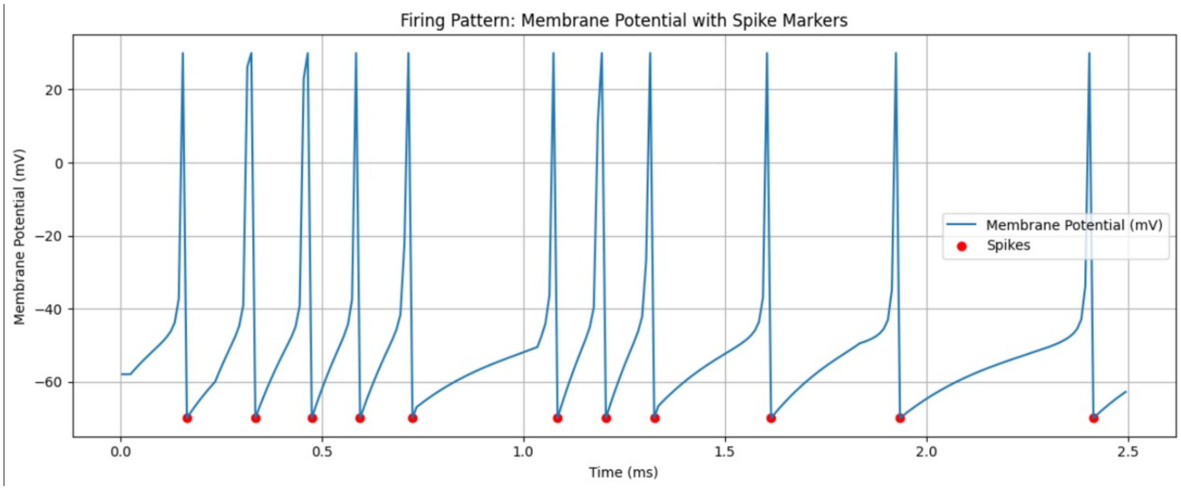

Figure 6.4: Firing Pattern of Purkinje Cells

Compared to granular cells, Purkinje cells exhibit larger, less frequent spikes in their membrane potential waveform. The fact that these spikes happen in reaction to accumulated synaptic inputs highlights Purkinje cells' function as network integrators. With distinct spike intervals, the waveform exhibits normal inhibitory neuron function.

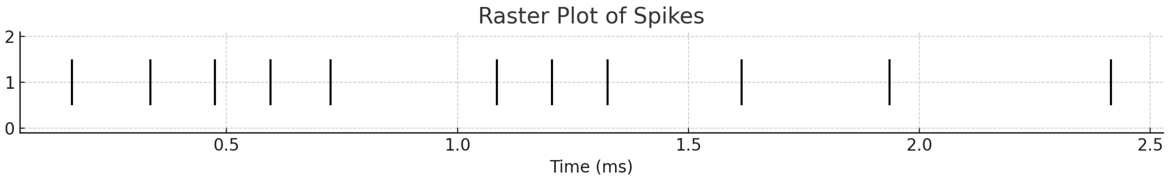

Figure 6.5: Raster plot of spikes of Purkinje Cells

Purkinje cells have a sparser distribution of spikes in their raster plot, which is in line with their inhibitory role. A regulating effect on downstream neuronal activity is suggested by the lower spike density in comparison to granular cells. Selective response to synaptic inputs is indicated by the temporal gap between pulses.

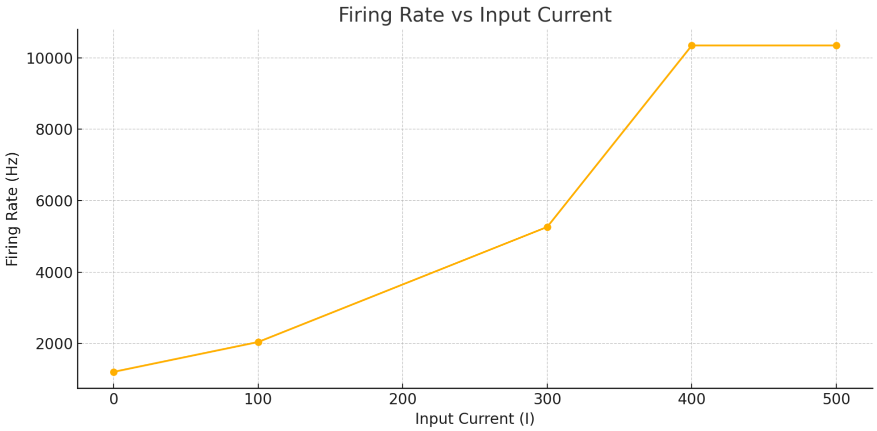

Figure 6.6: Firing rate vs input current plot of Purkinje Cells

With increased input current, Purkinje cells show an increasing trend in firing rate, albeit with a softer slope than granular cells. The idea that Purkinje cells selectively react to greater stimulation levels is supported by this tendency, which is consistent with their role as modulators of cerebellar motor control.

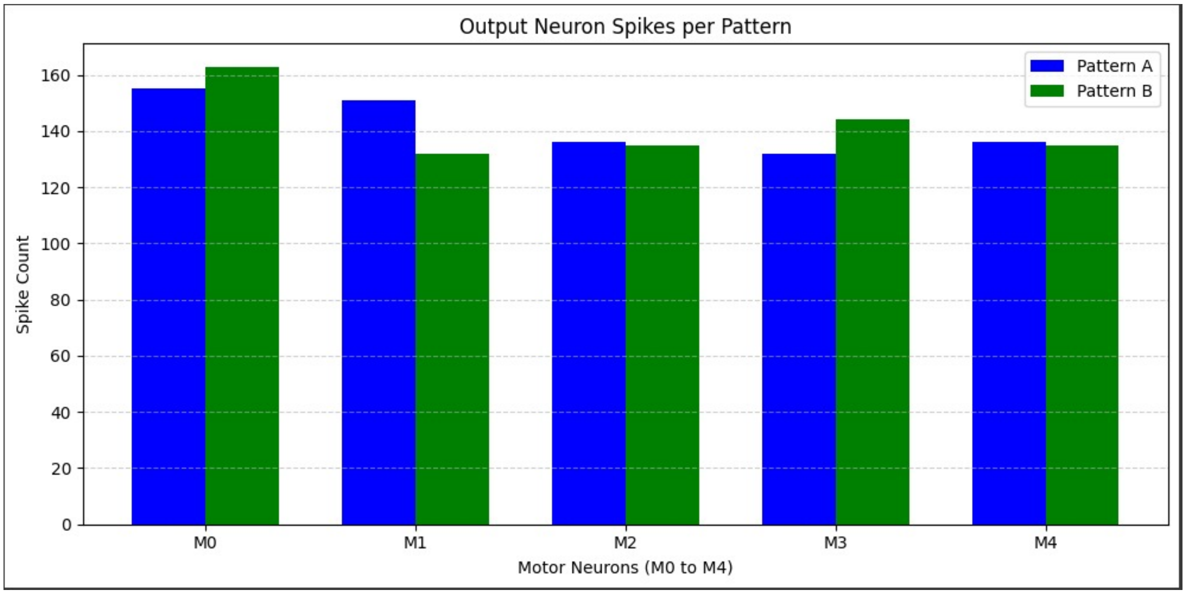

Figure 6.7: Network Output of Pre-Learning

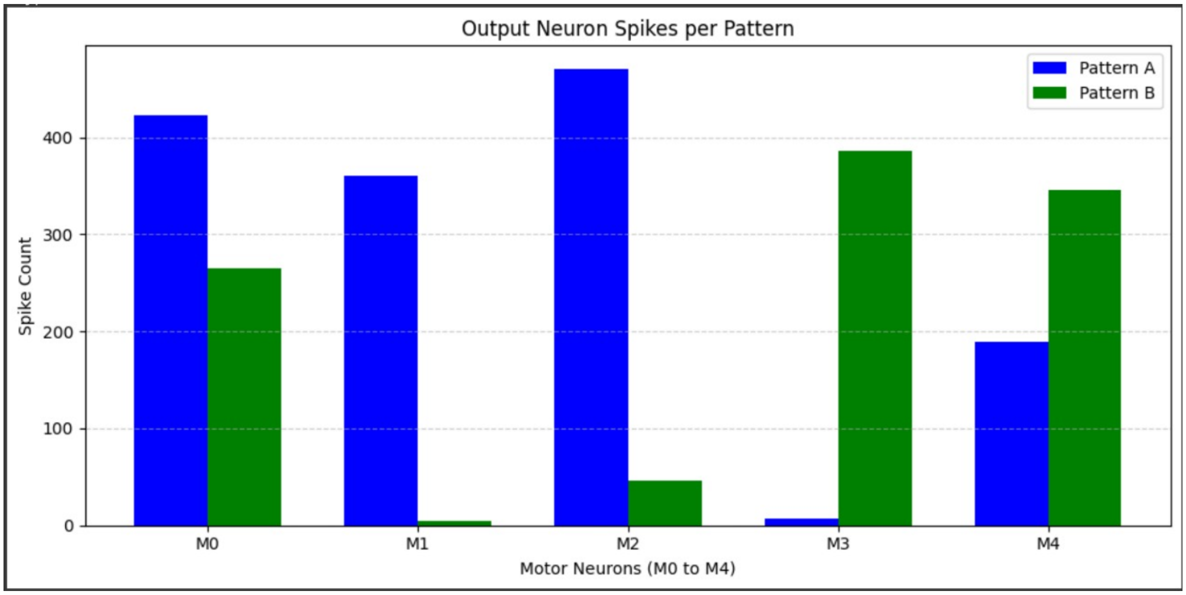

Figure 6.8: Network Output of Post-Learning

Figure 6.7: graph compares the total spike count of each motor neuron in response to two distinct input patterns — Pattern A (blue bars) and Pattern B (green bars). The x-axis represents the motor neurons (M0 to M4), and the y-axis shows the spike count over the same simulation period as the raster plot.

Observations: Minimal Differentiation Between Patterns:

For most neurons (M2, M3, M4), the spike counts for Pattern A and Pattern B are nearly identical.

Only neuron M1 shows a modest difference in spike count, with a decrease in spikes for Pattern B.

M0 shows a slight increase for Pattern B over A.

Lack of Specialization: There is no clear functional differentiation or specialization among neurons for recognizing one pattern over another.

Figure 6.8: bar chart shows the spike count of motor neurons (M0 to M4) for two different input patterns (Pattern A in blue and Pattern B in green) after learning has taken place in the spiking neural network.

Key Observations:

M0, M1, and M2 exhibit high spike activity for Pattern A, and low spike activity for Pattern B.

M3 and M4 exhibit high spike activity for Pattern B, and low spike activity for Pattern A.

The motor neurons have specialized in responding to specific input patterns, indicating successful learning and pattern discrimination by the network.

The separation in firing rates between patterns confirms that the motor layer has effectively learned to classify the input stimuli.

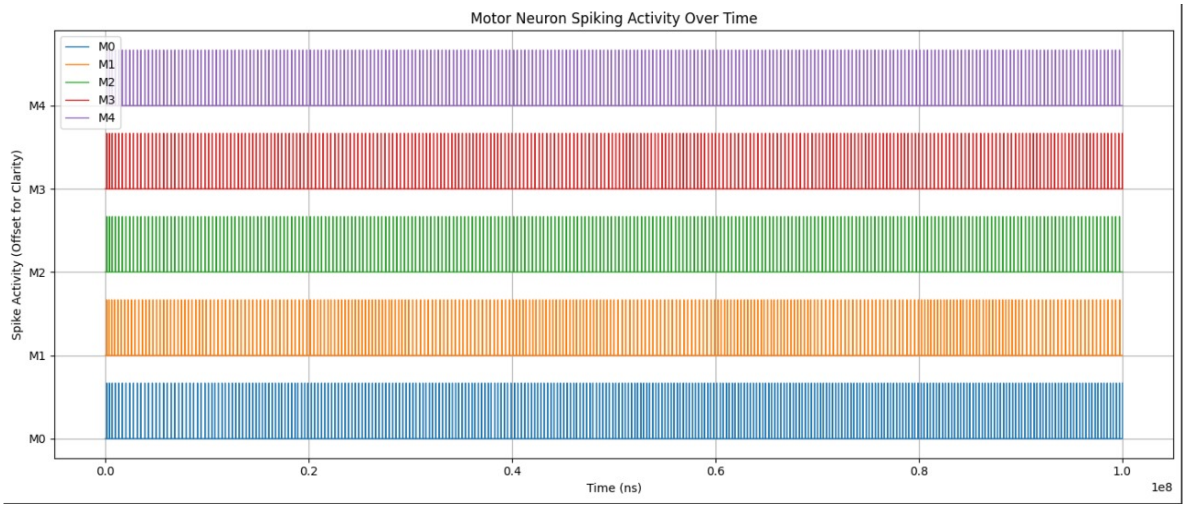

Figure 6.9: Rastor Plot of Pre-Learning Network

Figure 6.9: raster plot illustrates the firing times of five motor neurons (M0 to M4)

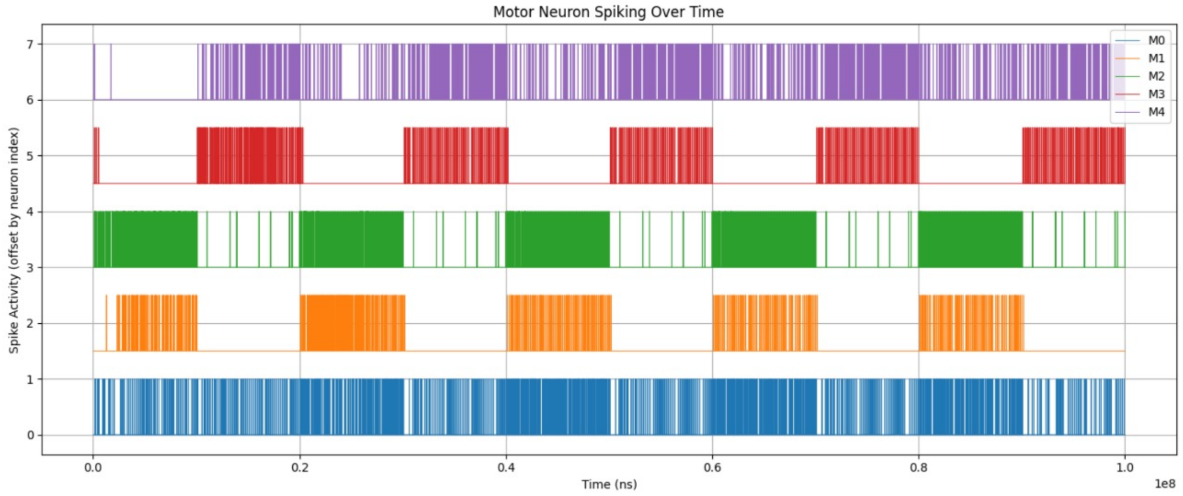

Figure 6.10: Rastor plot of Post-Learning of the Network

over a simulation duration of approximately 100 million nanoseconds. Each spike is marked as a vertical line, and neurons are displayed with an offset for clarity.

Observations: Uniform Firing Rate: Each motor neuron spikes with almost identical frequency and timing. The spike intervals are regular and consistent across all neurons.

Synchronous Behavior: There is no visible phase shift or temporal differentiation in the spike timings between different motor neurons.

No Selective Response: The neurons show no evidence of modulating their activity based on input characteristics. Instead, they exhibit a generic, rhythm-like firing that is independent of input pattern identity.

Figure 6.10: raster plot illustrates the temporal spiking behavior of the motor neurons (M0 to M4) over time after training.

Key Observations:

Each row corresponds to the spiking activity of one motor neuron.

Neurons M0 to M2 fire in synchronized bursts at regular intervals, reflecting a learned temporal response to Pattern A.

Neurons M3 and M4 show strong periodic spiking in alternate windows, corresponding to the learned response to Pattern B.

The activity is organized and consistent, demonstrating the temporal learning capability of the network.

Chapter 7

Conclusion and Future Work

In order to enable motor articulation control, the research demonstrated a cerebellum-like spiking neural network (SNN) implemented in a biologically inspired manner. Utilizing the Adaptive Exponential (AdEx) Integrate-and-Fire model, the network was able to replicate the actions of key cerebellar constituents such as Purkinje cells, granular cells, and mossy fibers. Through the use of firing rate measurements, raster plots, and spike pattern analysis, simulation findings verified that each neuronal layer responded correctly to changing input currents.

Purkinje cells demonstrated more selective spiking tendencies, functioning as modulators of motor output—replicating their biological inhibitory role—while granular cells displayed high-frequency and sparse encoding characteristics. Pattern recognition and adaptive motor encoding were facilitated by both supervised and unsupervised learning methods in the multi-layered network architecture, which showed coordinated spike-based processing.The efficacy of models inspired by the cerebellum for low-latency neural processing in neuromorphic systems is thus confirmed.

Overall, the findings support the feasibility of using SNNs to bioinspired control systems, paving the way for low-power, adaptive, real-time motor control systems that mimic the human cerebellum's functional structure.

7.1 Scalability and Future Applications

- Hardware Implementation on FPGA: Real-time operation, parallel processing, and energy-efficient computation will be made possible by switching from the current software-based approach to an FPGA platform. Deploying the network in robotic controllers and embedded devices requires this change.

- Scalability and Modular Network Expansion: To increase the complexity and realism of motor control, future iterations may incorporate other cerebellar structures like climbing fibers and basket cells. The network can also grow to handle increasingly complex motor tasks by increasing the number of granular and Purkinje cells.

- Advanced Learning processes: The network's flexibility to various motor learning settings can be further increased by incorporating homeostatic processes and biologically plausible plasticity principles like Spike-Timing Dependent Plasticity (STDP).

- Applications in Neuroprosthetics and Rehabilitation: The suggested model has great promise for the creation of exoskeletons, neuro-rehabilitation tools, and prosthetic limbs that are inspired by the brain. SNNs are ideal for individualized movement control in people with motor disabilities because to their adaptable nature.

References

[1] A. Cassidy, J. Georgiou, and A. G. Andreou, “Design of silicon brains in the nano-CMOS era: Spiking neurons, learning synapses and neural architecture optimization,” *Neural Networks*, vol. 45, pp. 4–26, 2013.

[2] G. Indiveri and S.-C. Liu, “Memory and information processing in neuromorphic systems,” *Proc. IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.

[3] G. Indiveri *et al.*, “Neuromorphic silicon neuron circuits,” *Front. Neurosci.*, vol. 5, p. 73, 2011.

[4] F. Akram, M. Shoaib, and H. I. Syed, “FPGA-based implementation of spiking neural networks for real-time processing,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 2, pp. 287–293, 2016.

[5] E. Stromatias, D. Neil, M. M. Khan, L. A. Plana, and S. B. Furber, “Live demonstration: Real-time object recognition using an event-based spiking neural network,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2015, pp. 2413–2413.

[6] P. U. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” *Front. Comput. Neurosci.*, vol. 9, p. 99, 2015.

[7] B. Linares-Barranco and T. Serrano-Gotarredona, "Memristance can explain spike-time-dependent-plasticity in neural synapses," *Nature Precedings*, pp. 1–10, 2009.

[8] S. K. Esser *et al.*, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proc. Natl. Acad. Sci. USA*, vol. 113, no. 41, pp. 11441–11446, 2016.

[9] A. Rathi, E. Sarmiento, A. van Schaik, and S. Mikaitis, "STDP-enabled learning on FPGA for pattern recognition," *Front. Neurosci.*, vol. 14, p. 305, 2020.

[10] J. Schemmel *et al.*, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 1947–1950.

[11] T. Stewart, X. Choo, and C. Eliasmith, "Symbolic reasoning in spiking neurons: A model of the cortex/basal ganglia/thalamus loop," in *Proc. 32nd Annu. Meet. Cogn. Sci. Soc.*, 2010.

[12] A. Vanarse, A. Osseiran, and A. Rassau, "A review of current neuromorphic approaches for vision, auditory, and olfactory sensors," *Front. Neurosci.*, vol. 10, p. 115, 2016.

[13] S. B. Furber *et al.*, "The SpiNNaker project," *Proc. IEEE*, vol. 102, no. 5, pp. 652–665, 2014.

[14] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[15] R. Legenstein, D. Brandner, and W. Maass, “A Hebbian learning rule gives rise to mirror neurons and links them to control theoretic inverse models,” *Front. Neural Circuits*, vol. 14, p. 42, 2020.

[16] C. Eliasmith and C. H. Anderson, *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems.* MIT Press, 2003.

[17] H. C. Tuckwell, *Introduction to Theoretical Neurobiology*, vol. 2. Cambridge Univ. Press, 1988.

[18] E. M. Izhikevich, “Simple model of spiking neurons,” *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, 2003.

[19] R. Brette and W. Gerstner, “Adaptive exponential integrate-and-fire model as an effective description of neuronal activity,” *J. Neurophysiol.*, vol. 94, no. 5, pp. 3637–3642, 2005.

[20] W. Maass, “Networks of spiking neurons: the third generation of neural network models,” *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, 1997.

[21] M. Pfeiffer and T. Pfeil, “Deep learning with spiking neurons: Opportunities and challenges,” *Front. Neurosci.*, vol. 12, p. 774, 2018.