Spiking Neural Network for Enhanced Mobile Robots' Navigation Control

Brwa Abdulrahman Abubaker
Dept Computer Science
College of Engineering and Science
Bayan University
Erbil, Iraq
Brwa.Abubaker@bnu.edu.iq

Saadaldeen Rashid Ahmed
Dept Computer Science
College of Engineering and Science
Bayan University
Erbil, Iraq
saadaldeen.aljanabi@bnu.edu.iq

Ari Taha Guron
Department of IT&Computer science,
college of IT& Computer Science ,
Catholic University in Erbil , Kurdistan region, Iraq.
ari.taha@cue.edu.krd

Mohammed Fadhil
Department of Medical Instrumentation Techniques Engineering, Technical Engineering College, Al-Kitab University,
Altun Kupri, Kirkuk, Iraq
mohammed.fadhil@uoalkitab.edu.iq

Sameer Algburi
College of Engineering Techniques
Al-Kitab University
sameer.algburi@uoalkitab.edu.iq

Bikhtiyar Friyad Abdulrahman
Department Civil Engineering
Erbil polytechnic university
Bikhtiyar.Abdulrahman@epu.edu.iq

2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS) | 979-8-3503-8306-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISAS60782.2023.10391395

***Abstract*—Contemporary robotics primarily emphasizes autonomous mobile robots, and Artificial Neural Networks (ANNs) have demonstrated their proficiency in managing intricate, nonlinear systems with illusive models. This study explores the progress made in third-generation neural networks, namely Spiking Neural Networks (SNNs), which has capabilities that beyond those of traditional NNs. We introduce a modular mobile robot navigation controller that utilizes SNNs to transmit both spatial and temporal information. The controller is constructed and evaluated within a simulated environment that accurately simulates real-life situations, utilizing promising Spiking Neural Networks (SNNs). This study seeks to improve the autonomous robot's collision avoidance and navigation capabilities by implementing a three-layered spiking neural network (SNN). Utilizing a customized variant of Spike-Timing-Dependent Plasticity (STDP) to train inhibitory synapses enhances the network's efficiency, resulting in a reduced number of required training iterations. A potential method for identifying synaptic connections in a concealed-layer SNN is introduced as an evolutionary methodology, which might be utilized to establish synaptic connectivity in the paper.**

Keywords— Spiking neural network, STDP, Evolutionary Algorithm.

I. INTRODUCTION

Navigation refers to the procedure via which mobile robots achieve a prearranged destination while circumventing specific dangers. This strategy is implemented based on sensor inputs and logical inferences, particularly in situations characterized by unknown variables, uncertainties, and a lack of clear organization. Autonomous navigation is an essential characteristic that mobile robots must possess in order to successfully carry out their duties [1]. Mobile robot navigation control has hitherto been limited to environments that possess clearly defined models. In this manner, traditional navigation controllers are generally restricted to overseeing uncomplicated, recurring assignments, such as guiding robots along pre-established routes. Constructing mathematical models under conditions of uncertainty and absence of organization is a challenging endeavor. Hence, the issue of creating navigation controllers for mobile robots in intricate situations is a formidable undertaking.

Due to their ability to represent and manage nonlinear systems, numerous controllers for mobile robots that utilize neural networks (NNs) have been successfully developed and deployed [2–15]. A notable current trend in the design of mobile robot controllers involves the merging of traditional processes with artificial neural network (ANN) methodology [12–15]. The incorporation of these methods signifies the beginning of a new age in enhancing the capabilities of navigation controllers, offering a more adaptable and reliable option for mobile robots operating in difficult conditions. Multiple studies have demonstrated that neurons in the brains of animals communicate with each other through spikes, which are brief electrical pulses. Spiking neural networks (SNNs) are a distinct type of neural network that originated from spike sequences and possess the ability to encode both spatial and temporal information. Spike coding in spiking neural networks (SNNs) allows for the transmission of spatiotemporal information, closely mimicking the activity of actual biological neurons. This sets SNNs apart from conventional models. Spike neurons in SNNs offer more precise representations of real neurons and enhance the speed of processing and communication.

According to several experts, artificial neural networks (ANNs) have progressed through three clearly defined generations. The second generation employed continuous activation functions to compute output signals, whereas the first generation utilized McCulloch-Pitts threshold neurons. The latest release is the third iteration, which incorporates spike neural networks (SNNs) [16]. Spike Neural Networks (SNNs) are more computationally potent than conventional neural networks (NNs) due to their utilization of specific spike timings to transmit data. SNNs have the ability to imitate any feedforward sigmoidal neural network and can approximate any continuous function [17]. Spike-based neural networks (SNNs) possess a significant edge over other forms of neural networks in terms of their ability to withstand and process noisy data. Moreover, the utilization of spike-based digital circuit modeling enables the implementation of spiking neural networks (SNNs) in a tangible form. SNNs possess significant processing capacity and have exceptional capabilities in pattern recognition and classification [18–29].

Natschläer and Ruf [18] propose a medically appropriate approach for identifying clusters in high-dimensional input fields using SNNs. Furthermore, this strategy is particularly useful in ever-changing situations. At a basic level, SNNs are accurate models that closely resemble biological systems, and they are also robust and flexible tools for various applications, especially in the fields of pattern recognition and hardware implementation. The captivating characteristics of Spiking Neural Networks (SNNs) have garnered considerable interest, resulting in ongoing investigation and the emergence of novel findings. The Spike Response Model (SRM), Probabilistic Spiking Neuron Model (PSNM), Leaky Integrated-and-Fire (LIF) Neuron Model, and Dynamic Firing Threshold Model are among the various models of spiking neurons that have been created. Although the LIF model is widely recognized and commonly employed for simulating SNNs, the PSNM model presents a unique spiking neuron model that distinguishes itself due to its exceptional resilience. Significant computational models have been introduced in the field, including the Liquid State Machine (LSM) and evolving support vector networks (SNNs), the Neurogenetic Brain Cube (NeuCube), and the Spike Pattern Association Neuron (SPAN) Architecture. NeuCube is a dynamic spiking model designed to mimic brain data [26], while SPAN excels at learning the relationships between input and output spike patterns and generating desired spike trains.

The SNN training algorithms can be broadly classified into two categories: supervised and unsupervised. Several unsupervised spike-based learning techniques encompass long-term potentiation (LTP) and long-term depression (LTD) [50], spike-based Hebbian learning, and spike-based deep learning (STDP). Supervised spike-based approaches encompass several methods such as statistical learning techniques, SpikeProp, evolutionary strategies, linear algebra techniques, ReSuMe, the SPAN method, and others. SNNs have proven to be effective, especially in robotics tasks including path planning, environment sensing, and robot behavior controllers [30]. Robots functioning in complex and unpredictable surroundings are more compatible with Spiking Neural Networks (SNNs) compared to conventional Artificial Neural Networks (ANNs) due to SNNs' natural capacity to transmit both spatial and temporal data. Prof. Floreano's laboratory has successfully enhanced the structure and parameters of Spiking Neural Network (SNN) in robot controllers by the utilization of genetic algorithms. This research has demonstrated the robustness and adaptability of SNNs.

The iSpike C++ library, developed by Gamez et al., enables efficient communication between the iCub humanoid robot and models of spiking neural networks. This achievement is a notable outcome of recent research. Thanks to a novel learning rule presented by Andre and colleagues, self-organizing neural networks (SNNs) can now function as controllers for robot simulations, resembling the functionality of the human brain. This learning rule is based on Spike-Timing-Dependent Plasticity (STDP). Luque et al. present a spiking neural network (SNN) that mimics the structure of the cerebellum. This SNN is designed to have improved resistance to noise and is capable of maintaining corrective models. It enables accurate control of a robot arm with non-stiff joints. Alnajjar et al. created a hierarchical adaptive controller based on SNN to assist a physical mobile robot in efficiently navigating dynamic settings. Paolo et al. demonstrate the versatility and use of Spiking Neural Networks (SNNs) in various robotic contexts by employing a three-layered SNN with Spike-Timing-Dependent Plasticity (STDP) learning rules as a controller for robot target-approaching.

This publication presents our contribution to the study: a target-approaching controller that employs behavior-based techniques and incorporates Spiking Neural Networks (SNNs). The three-sub-controller modular navigation controller offers a complete and adaptable solution for robot navigation in challenging scenarios. It relies on previous principles of wall-tracking and obstacle-evading controllers. The LIF model, or leaky integrate-and-fire neuron model, computes the cumulative value of postsynaptic potentials and initiates the generation of action potentials when the membrane potential surpasses a specific threshold. The decline of the cell membrane potential in the absence of input is represented as a leak process [31]. The LIF model effectively stores data by considering the numerical and temporal characteristics of brain action potentials, regardless of their specific waveform. This method streamlines the explanation of action potentials in biological neurons by breaking them down into distinct occurrences that happen at certain time intervals before delving into a comprehensive model of their structure. The relationship between the injected current, I(t), and the membrane potential, u(t), is governed by a linear differential equation, which is based on fundamental concepts from electrical theory. In order to mimic the structure of a cell membrane, we have connected capacitor C and resistor R in a parallel configuration. In the absence of external stimulation, the membrane potential returns to its disrupted baseline state. Hence, the linear differential equation governs the response of the membrane potential to the input current I(t).

In section 2 of the study provides an overview of fundamental concepts and approaches, including STDP, the Leaky Integrate-and-Fire Neuron Model, and Spiking Neural Networks. This section also includes information about the sensors used by the robot, as well as its kinematics and dynamics. Section 3 centers on the simulated robot and encompasses the examination procedures, instructional process, and use of spike-based neural networks. Section 4 presents a concise summary of the study's main discoveries and their consequences. This study not only establishes the foundation for future research, but also presents a thorough framework for examining and advancing the subject. It outlines potential methods for improving the current model and addresses its practical implications [34-40].

II. Theory and Techniques

The complex patterns found in actual neural networks have served as a source of inspiration for the creation of artificial neural networks (ANNs), which are a specific sort of computer system. Artificial neurons, which are interconnected computer units, serve as the fundamental components of these systems. They possess the ability to learn and execute certain tasks. Modifying the interconnections among artificial neural networks enables them to adjust and enhance their performance. The

Threshold Logic Unit, commonly referred to as a threshold gate, is a prevalent and fundamental paradigm of artificial neural networks. The architecture of these neural networks transforms continuous inputs into discrete outputs. It achieves this by aggregating all the weighted inputs and comparing the total to a pre-established threshold, as depicted in Figure 1. We propose a modification to the neuron model that enables the conversion of real-valued inputs to real-valued outputs. Instead of doing a comparison between the weighted sum of the inputs and a threshold, an activation function is employed. The sigmoid function is a frequently used activation function for this type of task.

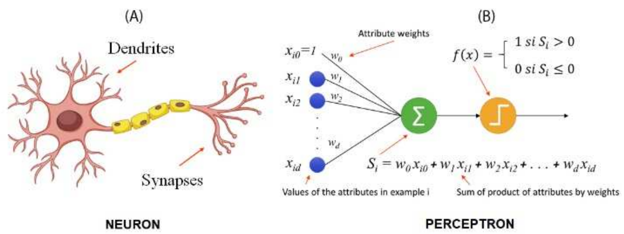

Figure 1: Threshold Logic Unit Model and Feedforward Neural Network Structure

In this analysis, we explore two key components of constructing neural networks, employing a comprehensive diagram. The Threshold Logic Unit neuron model, denoted as (A), compares the sum of the weighted inputs to the threshold value ✓. In accordance with the binary output principle, neurons in this model will transmit a signal of 1 if the weighted sum exceeds the threshold, and a signal of 0 otherwise. The diagram illustrates the standard structure of a feedforward neural network. This network architecture consists of three input neurons, two output neurons, and a hidden layer with five neurons. In a feedforward network, the input neurons are responsible for receiving data, which subsequently propagates through the hidden layer and ultimately reaches the output neurons. Collectively, these elements represent the fundamental principles of neural network architecture.

A. *Spiking Neural Networks*

From a biological perspective, spiking neural networks (SNNs) vary from networks that employ real-valued neurons in that they assume the output of the activation function represents the present firing rate of a biological neuron. Nevertheless, studies suggest that the possibility of data loss exists when solely taking into account firing rates, without addressing precise firing timings [32]. Research indicates that some tasks and computations can be completed within a timeframe of 20-30 milliseconds by organic brain networks, even when the neurons' firing frequency is below 100 Hz. More precisely, if only the firing rate is taken into account, the minimum sample time would be 20 to 30 milliseconds. The calculations themselves would require more time. The computational time needed by artificial neural networks would therefore exceed that of comparable biological neural networks.

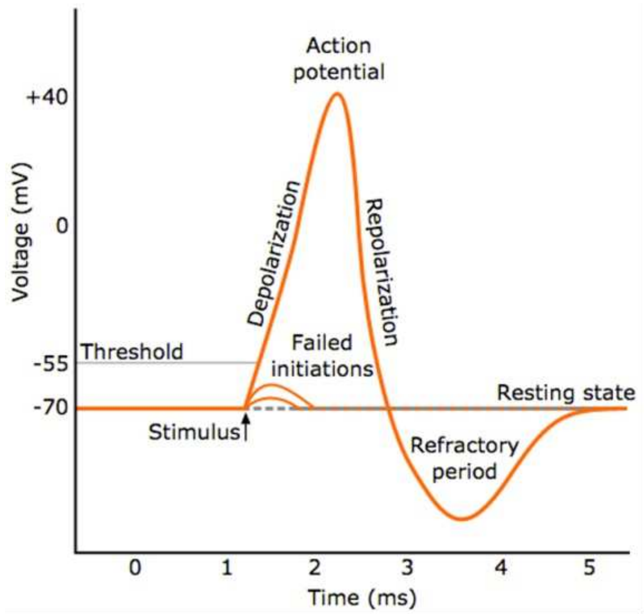

Figure 2: Izhikevich model-based spike creation [18].

B. *The Leaky Integrate-and-Fire Neuron Model*

The LIF neuron model serves as the foundation for comprehending the dynamics of neural networks. The membrane potential must attain a particular threshold for proper functioning, at which juncture the postsynaptic potentials are aggregated to initiate action potentials. The model additionally integrates a leak mechanism [7] to guarantee a progressive reduction in the cell membrane potential in the absence of input. The LIF model compresses neural action potentials into discrete temporal events, emphasizing the significance of timing and frequency as information carriers, instead of attempting to replicate the complex forms observed in real neurons. The model utilizes a linear differential equation to represent the variation of the membrane potential, which is symbolized as u(t). The cell membrane can be represented as a parallel combination of a resistor R and a capacitor C, using principles from electrical theory. This representation helps to understand the relationship between the injected current I(t) and the membrane potential. Without any input, the membrane potential remains at its resting value, denoted as (u)rest. This implies that the equation accurately represents the relationship between the membrane potential and the input current I(t).

The linear differential equation governing the Leaky Integrate-and-Fire (LIF) neuron model is expressed as follows:

$$\frac{du(t)}{dt} = 1 + + \frac{u(t) - urest}{RC} + \frac{I(t)}{C} \tag{1}$$

The Leaky Integrate-and-Fire (LIF) neuron model is characterized by a linear differential equation governing the evolution of its membrane potential over time, denoted as (u(t)). This mathematical expression incorporates essential components reflecting the physiological behavior of biological neurons. The equation includes terms that account for the leaky nature of the neuron, represented by (u(t)-urest)/RC, where (u_rest) signifies the resting potential and (RC) involves the resistance and capacitance of the cell membrane. The second term, (I(t))/C, encapsulates the impact of the injected current (I(t)) on the membrane potential. The LIF model simplifies the intricate dynamics of biological neurons, emphasizing the timing and frequency of action potentials rather than the exact shape, allowing for a more computationally efficient representation in neural network simulations.

C. Spike-Timing-Dependent Plasticity

Spike-timing-dependent plasticity (STDP) is a vital mechanism of neuroplasticity that enables neurons to establish synaptic connections by considering the precise timing of their firing sequences. The short synaptic distance protocol (STDP) is based on the premise that synaptic connections are weakened when the firing of the postsynaptic neuron occurs in the opposite direction of the presynaptic neuron. By employing this rule that is contingent on time, neurons are able to establish connections between inputs that occur within a precise temporal framework in a manner that is causally linked. STDP, despite being widely recognized as a crucial element of memory and learning, fails to explain the more complex learning events that involve rewards and punishments. Dopamine functions as an additional neurotransmitter in this context, regulating changes in synaptic weights through an error-based reinforcement signal. Dopamine is a neuromodulator that enhances long-term depression (LTD) when it is produced in response to inputs that are not predictive or have negative predictions. Conversely, it promotes long-term potentiation (LTP) when it is released in response to inputs indicating rewarding events through spike-timing-dependent plasticity (STDP). Dopamine and spike-timing dependent plasticity (STDP) collaborate to generate an asymmetrical STDP window, which improves reward prediction. This complex approach offers a thorough comprehension of synaptic plasticity in relation to learning and memory.

$$\dot{c} = {}^{-c}/_{\tau_c} + \mathrm{STDP}(\tau)\delta(t - t_{pre/post} \tag{2}$$

$$\dot{s} = c\dot{d} \quad \tau = t_{post} - t_{pre} \tag{3}$$

In the context of equations 2 and 3, several critical parameters come into play: 'd' represents the extracellular dopamine concentration, τ(c) serves as the time constant, and δ(t) denotes the Dirac delta function, responsible for incrementally adjusting the value of the variable 'c.' When the presynaptic neuron fires at time t-pre, followed by the postsynaptic neuron at time t-post, these occurrences instigate changes in the variable 'c.' The extent of this modification is determined by the magnitude of STDP(τ). The dynamic evolution of this change is visually represented in Figure 3, illustrating how 'c' responds to neuronal spiking. The parameter 'τ' signifies the temporal discrepancy between the presynaptic and postsynaptic neuron spikes.

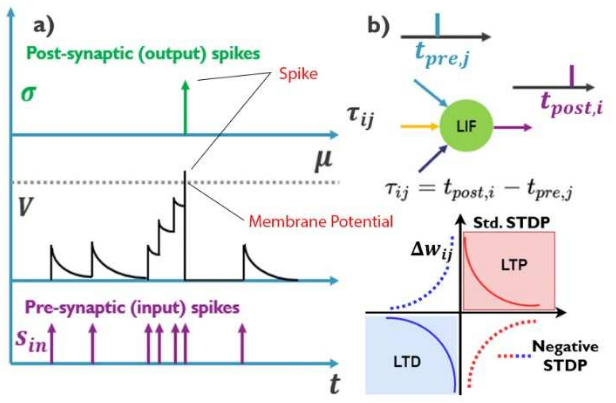

Figure 3: The cell membrane potential is unique at every presynaptic spike. It increases for excitatory synapses and decreases for inhibitory synapses. When a threshold is crossed, a spike is released.

D. Kinematics and dynamics

Assuming that the wheels of the robot roll without slipping, and considering the robot as a rigid body with wheels capable of moving only perpendicular to the wheel axis, the kinematics of the entire robot, including the variation of the direction of motion (ℑ), and the position coordinates (x and y) can be defined by the speeds vL and vR of the left and right wheels. Using w = ℑ to represent the rotation of the robot around the instantaneous center of rotation and v to denote the speed of the center-of-mass of the robot, the following relations are established:

$$v = \frac{Vl+Vr}{2} \tag{4}$$

$$v = \frac{Vl-Vr}{2R} \tag{5}$$

E. Sensors

1) The odometer sensor

An essential component for monitoring the distance traveled by a robot is an odometer sensor. Navigation, mapping, and localization are among the many applications for it. Typical motion tracking technologies employed by odometer sensors encompass inertial sensors and wheel encoders, enabling the sensor to accurately track the robot's movements. Ground-based robots often utilize wheel encoders, which employ sensors to track the rotations or revolutions of the wheels. The sensor can determine the distance covered by detecting and tallying the rotations, based on the established wheel circumference. This approach generates a dependable estimation of displacement, making it widely utilized in differential drive robotics and automotive applications.

At the initiation of each simulation, the odometer is aligned with the real position of the robot. As the robot traverses its path, the estimated position is iteratively adjusted, considering both the speed and angular speed of the robot. This adjustment incorporates noise from a normal distribution N(0, p2), where o = 0.0003. Estimating the present position is not solely dependent on the current movement parameters but is also influenced by the estimated position in the preceding step, p=0.04.

2) Ultrasonic sensors

Ultrasonic sensors are commonly used in mobile robot navigation systems to identify impediments and measure distances. The sensors operate by tracking the duration it takes for high-frequency sound waves to return to the environment after reflecting off nearby objects. An ultrasonic sensor records signals, which are then encoded and transmitted by support vector neurons (SNNs) to the sensory neurons of the network. The sensory neurons transmit signals to the motor neurons, which govern the locomotion of the robot. Multiple studies have demonstrated that ultrasonic sensors enhance the precision of mobile robot navigation by identifying and evading obstacles [33]. Efficient in controlling robotic systems. Robots has the capacity to identify objects at different distances, enhancing their ability to navigate and evade hazards. The field of vision of the sensor is determined by the angle at which it is positioned, resulting in data collection within either a small or wide range. Precision control, facilitated by accurate distance measurements, enables the ability to plan paths and manipulate objects. The ability to control in real-time enhances responsiveness and agility by facilitating rapid reaction times. Ultrasonic sensors possess the ability to detect impediments even when they are not directly visible, hence minimizing the probability of collisions. The availability of mounting and integration options facilitates the incorporation of these devices into robot systems. Robots possess the capability to travel, comprehend their environment, and accomplish tasks when specific characteristics are taken into account.

III. Simulated Robot

A. *Application of neural networks with spikes*

This study's spiking neural networks (SNNs) utilize leaky integrate-and-fire (LIF) neurons, as detailed in Section 2.3, with parameters specified in Table 1. The membrane resistance and time constant are incorporated into the synaptic weights (wij) for computational simplification of the membrane potential. In the LIF model, neurons emit action potentials in the form of Dirac $\Im$ functions at firing times t. Consequently, the term RI(t)/Tm in Equation 2.1 can be substituted with $\sum_{i=1}^{n} wij\, \Im(t - ti)$ in the equation governing the membrane potential for neuron j. Here, i denotes incoming neurons ranging from 1 to n, and w^{ij} represents the connecting weights. This modification streamlines the membrane potential equation.

$$\frac{duj}{dt} = \frac{[uj(t) - Urest]}{Tm} a_0 + \sum_{i=1}^{n} (ij\, \Im(t - ti)) \qquad (6)$$

The firing times T_i represents the instances when neuron I emits action potentials. Upon reaching the threshold, the membrane potential triggers the emission of an action potential, and the potential is subsequently reset to the reset potential u_r. Synapses are represented as weights within the range w=[1, 1], where negative weights denote inhibitory synapses, and positive weights correspond to excitatory synapses.

Table 1: Values for the parameters in the LIF-neuron model

Parameter	Value
Membrane time constant, Tm	10 ms
Rest potential, $_{rest}$	0 mV
Reset potential, u_r	10 mV
Threshold, v	1 V
Refractory period, T_r	2 ms

the spiking neural network model is described as using the pair-based STDP model to update the excitatory synapses. Table 2 provides the precise specifications of the STDP. Neurons are updated at a rate of 1000 Hz, whereas synaptic weights are modified at a rate of 50 Hz, synchronized with the robot's frequency. This is quite intriguing. The synchronization is essential since the usefulness of the modulatory success signal relies on the state of the robot. Hence, the success signal is contingent upon the state. The neurons' spiking activity during the period between synaptic updates is accurately explained by the eligibility trace.

TABLE 2: VALUES FOR THE PAIRWISE STDP MODEL PARAMETERS.

Parameter	Value
Presynaptic temporal window, T_{pre}	20 ms
Presynaptic temporal window, T_{post}	20 mV
Presynaptic weight update, A+	0.021
Postsynaptic weight update, A	-0.022
Reward temporal window, T_{pre}	201 ms

B. *Training process*

Neural networks are utilized to instruct a robot on how to navigate around barriers and successfully accomplish a pre-established objective. The training scenarios closely resemble the test cases, although they are not identical. The stage consists of an arena measuring 6 by 6 meters, with intermittent square barriers measuring 1 by 1 meter. Initially, the robot is oriented in any desired direction. The destination place is selected at random from a variety of locales indicated by green stars. The goal is for the robot to navigate towards the target. The training process ends if the robot successfully achieves the target within a predetermined amount of time steps, or if it becomes immobile for a prolonged duration due to colliding with a wall or obstacle, indicating that it is stuck.

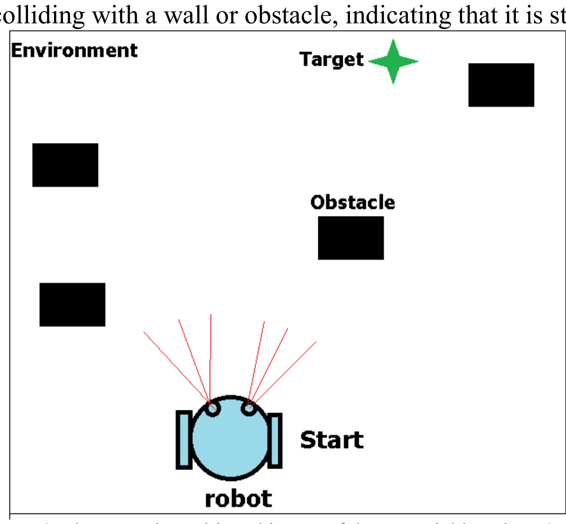

Figure 4: The target is positioned in one of the potential locations (green star) during training.

The training process involves updating the synaptic weights of the spiking neural networks using STDP, detailed in Section 2.4. All neurons receive the same global modulatory success signal throughout training, denoted as M. The value of M is contingent on the network's performance, assessed by criteria including distance to the target, angle to the target, and proximity to obstacles. Successful navigation to the target involves minimizing distance and angle while avoiding collisions with obstacles. Consequently, the modulatory success signal M incorporates reward parameters: R_d (distance to the target), R_a (angle to the target), and R_o (distance to the nearest obstacle). The specific values of these parameters are tailored to the desired behavior and objectives of the robot. Through the adjustment of synaptic weights based on the modulatory success signal M, the spiking neural networks learn to optimize the robot's movements, ensuring efficient target-reaching while preventing collisions with obstacles.

$$Rd = f(x) = \begin{cases} 1 - \frac{d}{dv} \; if \; dv > d \\ 0, \; \text{otherwise} \end{cases} \tag{6}$$

C. Testing

Each training cycle concludes with an assessment to ascertain the proficiency of the networks. Throughout this assessment, four predetermined test scenarios are employed to determine the suitability of the networks. The test cases encompass scenarios in which the destination location, initial robot position, and direction may vary, thereby ensuring a thorough assessment of the networks' performance. The trial outcomes demonstrate the versatility and efficacy of the trained networks across various environments.

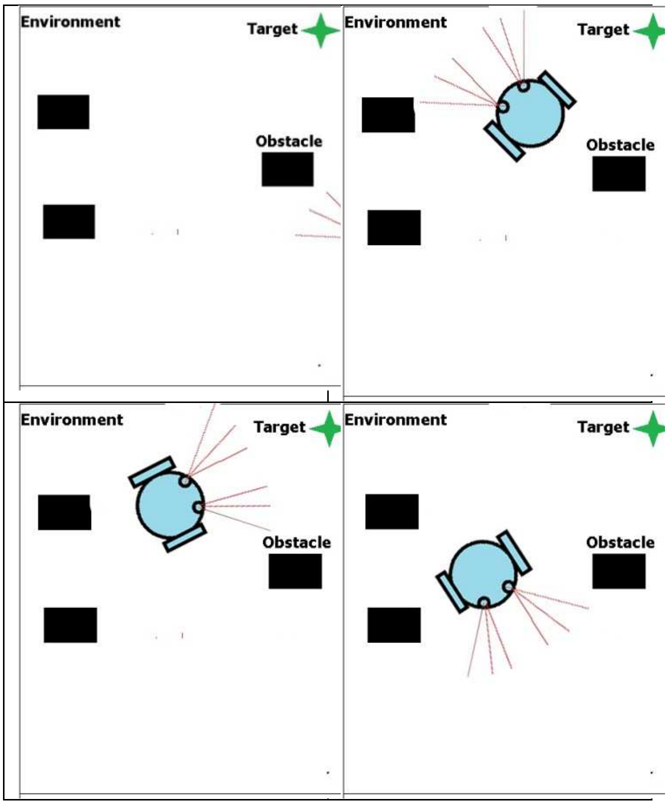

Figure 5: The layouts of the four test cases that determine the network's fitness.

As a component of its training regimen, the robot undergoes eleven tests that are generated randomly following each training cycle. We continue this process until we have trained the robot in 150 unique scenarios. This operation is repeated nine times using the random number generator, each time with a different seed. Figure 6, located on the right side, displays the mean number of arrivals for various seeds of the random number generator. This image effectively demonstrates the high level of proficiency and adaptability of the trained robot in various settings.

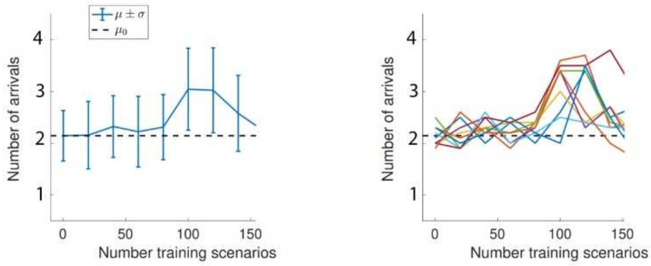

Figure 6. Depicts the average number of arrivals (μ) with standard deviation, plotted against the number of training rounds. The graph also refers to the average number of arrivals when the network undergoes no training, denoted as μ0. This visual representation offers insights into the progressive improvement in the robot's performance over successive training rounds, providing a quantitative measure of its learning and adaptation capabilities.

As anticipated, the exact performance has a degree of variability due to the diverse training scenarios. However, a consistent observation is that, in most cases, the network tends to achieve peak performance after approximately 90-110 training scenarios. Figure 7 visually represents the network's performance on the four test cases following training as outlined in Section (training) based on 150 different scenarios. Notably, the robot demonstrates successful arrival at the target in all four test cases, highlighting the effectiveness of the training process in enhancing the robot's navigation capabilities.

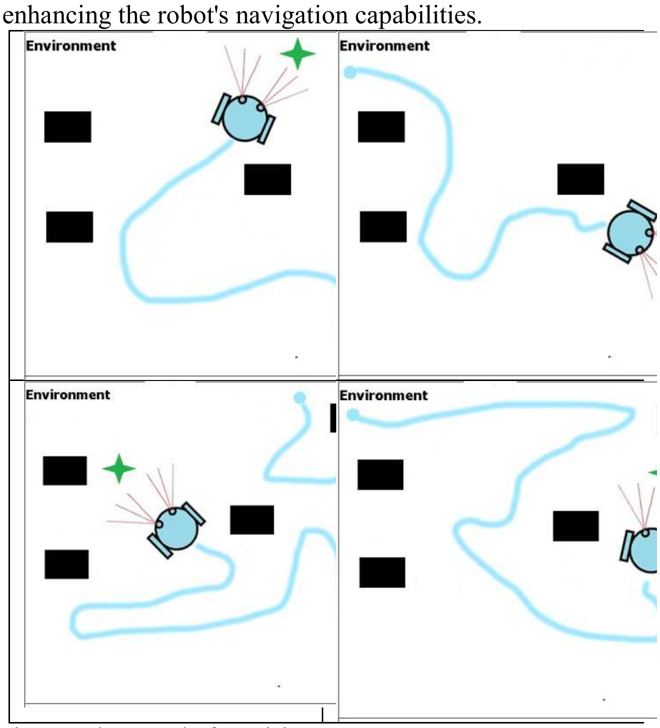

Figure 7. The network after training

According to the data, it seems that training increases the mean number of arrivals until it reaches a plateau. The lack of noticeable change in the standard deviation is very remarkable. Despite undergoing training, the outcomes of recurrent network testing exhibit inconsistency across different iterations. The network's sensitivity to performance changes indicates that it is still influenced by inaccurate location and direction predictions, implying that uncertainty remains a persistent issue even after training.

IV. Conclusion & Future Work

Our experiment confirmed that a three-layer spiking neural network trained with STDP exhibits effective behavior in both collision avoidance and target navigation. The network attains its maximum performance after being trained on over 100 instances, demonstrating its ability to acquire knowledge and improve movement optimization. It is crucial to remember that the network is prone to errors in its projected placements and orientations, which might provide challenges in real-world applications. Despite encountering collisions in 18% of the arena configurations, the spiking neural network successfully reached the objective in 82 instances. Further investigation is required to improve the network's ability to withstand collision events, and this outcome offers valuable insights into the network's unpredictable behavior.

A. Future Work

The current research represents a significant advancement in the application of spiking neural networks for target navigation and collision avoidance in autonomous robots. Numerous captivating paths must be investigated to advance this subject. In order for the network to be effectively implemented in real-world scenarios, it is imperative that it possesses the capability to endure inaccuracies in both position and direction forecasts. The primary emphasis will be on implementing strategies that offer adaptability in uncertain situations and minimize the consequences of mistakes. Furthermore, the functionality of the network may be more comprehended by subjecting it to examinations in intricate real-life scenarios. Hybrid designs have the potential to enhance performance by integrating spiking neural networks with other models. Exploring more sophisticated learning mechanisms, such as reinforcement learning, could improve training procedures. Future study should focus on prioritizing energy efficiency and conducting performance evaluations in dynamic settings that entail human-robot interaction. Through an exploration of these characteristics and the improvement of current models, the study aims to accelerate the progress of self-governing robotic systems

References

[1] Y. Wang, M. Cartmell, Q. Tao, H. Liu, A generalized real-time obstacle avoidance method without the Cspace calculation, J. Comput. Sci. Technol. 62005 (20) (2005) 774–787.

[2] D.A. Pomerleau, Neural Network Perception for Mobile Robot Guidance, Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[3] A.G. Barto, R.S. Sutton, C.W. Anderson, Neuronlike adaptive elements that can solve difficult learning control problems, IEEE Trans. Syst. Man Cybern. 13 (5) (1983) 834–846.

[4] K.H. Low, W.K. Leow, M.H. Ang, Integrated planning and control of mobile robot with self-organizing neural network, in: Proceedings of IEEE ICRA, IEEE Press, New York, 2002, pp. 3869–3875.

[5] Y.K. Na, S.Y. Oh, Hybrid control for autonomous mobile robot navigation using neural network based behavior modules and environment classification, Auton. Robots 15 (2) (2003) 193–206.

[6] L. Cheng, Z.-G. Hou, M. Tan, Y. Lin, W. Zhang, Neural-network-based adaptive leader-following control for multi-agent systems with uncertainties, IEEE Trans. Neural Netw. 21 (8) (2010) 1351–1358.

[7] Z.-G. Hou, L. Cheng, M. Tan, Multi-criteria optimization for coordination of redundant robots using a dual neural network, IEEE Trans. Syst. Man Cybern. B Cybern. 40 (4) (2010) 1075–1087. [8] L. Cheng, Z.-G. Hou, M. Tan, Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model, Automatica 45 (10) (2009) 2312–2318.

[9] J. Li, J. Yi, D. Zhao, G. Xi, A reinforcement learning based radial-basis function network control system, In: Proceedings of ISNN 2005, Lecture Notes in Computer Science, vol. 3496, Springer, Berlin, 2005, pp. 640–645.

[10] J. Li, J. Yi, D. Zhao, G. Xi, Cooperation learning for behavior-based neural-fuzzy controller in robot navigation, in: Proceedings of IFAC World Congress 2005, vol. 16, Part 1, IFAC, Czech Republic, pp. 1270–1275.

[11] J. Li, Soft computing method for the autonomy navigation for the mobile robots (Ph.D. thesis), Institute of Automation, The Chinese Academy of Sciences, China, 2005.

[12] J. Ye, Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot, Neurocomputing 71 (2008) 1561–1565.

[13] J. Ye, Tracking control for nonholonomic mobile robots: integrating the analog neural network into the backstepping technique, Neurocomputing 71 (2008) 3373–3378.

[14] F.G. Rossomando, C. Soria, R. Carelli, Autonomous mobile robots navigation using RBF neural compensator, Control Eng. Pract. 19 (2011) 215–222.

[15] O. Mohareri, R. Dhaouadi, A.B. Rad, Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks, Neurocomputing 88 (2012) 54–66.

[16] J. Vreeken, Spiking Neural Networks, An Introduction, Technical Report UUCS-2003-008, Institute for Information and Computing Sciences, Utrecht University, 2002. 〈http://ailab.cs.uu.nl/pubs/SNNVreeken Introduction.pdf〉.

[17] W. Maass, C.M. Bishop, Pulsed Neural Networks, MIT-Press, Cambridge, MA, 1999.

[18] T. Natschläer, B. Ruf, Spatial and temporal pattern analysis via spiking neurons, Network: Comput. Neural Syst. 9 (3) (1998) 319–332.

[19] S.M. Bohte, H.L. Poutre, J.N. Kok, Unsupervised clustering with spiking neurons by sparse temporal coding and multiplayer spike neural network, IEEE Trans. Neural Netw. 13 (2) (2002) 426–435.

[20] B. Ruf, Computing and learning with spiking neurons theory and simulations (Ph.D. thesis), Institute for Theoretical Computer Science Technische University at Graz, Austria, May 1998. [21] F. Ponulak, A. Kasinski, Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting, Neural Comput. 22 (2) (2010) 467–510.

[22] S. Schliebs, M. Defoin-Platel, S. Worner, N. Kasabov, Integrated feature and parameter optimization for evolving Spiking neural networks: exploring heterogeneous probabilistic models, Neural Netw. 22 (5–6) (2009) 623–632.

[23] S. Wysoski, L. Benuskova, N. Kasabov, Fast and adaptive network of Spiking neurons for multi-view visual pattern recognition, Neurocomputing 71 (2008) 2563–2575.

[24] N. Kasabov, Integrative connectionist learning systems inspired by nature: current models, future trends and challenges, Nat. Comput. 8 (2) (2009) 199–218.

[25] N. Kasabov, To spike or not to spike: a probabilistic spiking neuron model, Neural Netw. 23 (1) (2010) 16–19.

[26] N. Kasabov, Evolving Spiking neural networks and neurogenetic systems for spatio- and spectro-temporal data modelling and pattern recognition, in: Proceedings of IEEE WCCI 2012, Lecture Notes in Computer Science, vol. 7311, Springer, Berlin, 2012, pp. 234–260.

[27] N. Kasabov, NeuCube evoSpike architecture for spatio-temporal modelling and pattern recognition of brain signals, in: Proceedings of ANNPR 2012, Lecture Notes in Artificial Intelligence, vol. 7477, 2012, pp. 225–243.

[28] X.Q. Wang, Z.-G. Hou, M. Tan, Y. Wang, Corridor-scene classification for mobile robot using Spiking neurons, in: Proceedings of IEEE ICNC 2008, vol. 4, IEEE Press, New York, 2008, pp. 125–129.

[29] X.Q. Wang, Z.-G. Hou, M. Tan, Improved mobile robot's corridor-scene classifier based on probabilistic Spiking neuron model, in: Proceedings of IEEE ICCICC, IEEE Press, New York, 2011, pp. 348–355.

[30] R.S.A. Alamdari, Unknown environment representation for mobile robot using spiking neural networks, Trans. Eng. Comput. Technol. 6 (2005) 49–52.

[31] W. Gerstner et al. Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition. Cambridge University Press, 2014. Chap. 1.3 Integrate-And-Fire Models, 19.1 Hebb rule and experiments. doi:

10.1017/CBO9781107447615. url: https://neuronaldynamics.epfl.ch/online/index.html.

[32] W. Maass. Networks of spiking neurons: The third generation of neural network models. Neural Networks 10.9 (1997), 1659–1671. issn: 0893-6080. D

[33] Y. Zheng, B. Yan, C. Ma, X. Wang, and H. Xue, “Research on obstacle detection and path planning based on visual navigation for mobile robot,” in Journal of Physics: Conference Series, 2020. doi: 10.1088/1742-6596/1601/6/062044..

[34]Ahmed, S. R., Ahmed, A. K., & Jwmaa, S. J. (2023, June). Analyzing The Employee Turnover by Using Decision Tree Algorithm. In 2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-4). IEEE.

[35]N. Z. Mahmood, S. R. Ahmed, A. F. Al-Hayaly, S. Algburi and J. Rasheed, "The Evolution of Administrative Information Systems: Assessing the Revolutionary Impact of Artificial Intelligence," 2023 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkiye, 2023, pp. 1-7.

[36] M. A. Shihab, S. A. Aswad, R. N. Othman and S. R. Ahmed, "Advancements and Challenges in Networking Technologies: A Comprehensive Survey," 2023 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkiye, 2023, pp. 1-5.

[38]Abbood, Z. A., Yasen, B. T., Ahmed, M. R., & Duru, A. D. (2022). Speaker identification model based on deep neural networks. Iraqi Journal For Computer Science and Mathematics, 3(1), 108-114.

[39] YASEEN, Baraa Taha; KURNAZ, Sefer; AHMED, Saadaldeen Rashid. Detecting and Classifying Drug Interaction using Data mining Techniques. In: 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE, 2022. p. 952-956.

[40] Abdulateef, O. G., Abdullah, A. I., Ahmed, S. R., & Mahdi, M. S. (2022, October). Vehicle License Plate Detection Using Deep Learning. In 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 288-292). IEEE.