MEMRISTOR BASED NEURAL NETWORKS: FEASIBILITY, THEORIES AND APPROACHES

a thesis submitted to
The University of Kent
in the subject of computer science
for the degree
of doctor of philosophy.

By
Xiao Yang

December 2014

Abstract

Memristor-based neural networks refer to the utilisation of memristors, the newly emerged nanoscale devices, in building neural networks.

The memristor was first postulated by Leon Chua in 1971 as the fourth fundamental passive circuit element and experimentally validated by one of HP labs in 2008. Memristors, short for memory-resistor, have a peculiar memory effect which distinguishes them from resistors. By applying a bias voltage across it, the resistance of a memristor, namely memristance, is changed. In addition, the memristance is retained when the power supply is removed which demonstrates the non-volatility of the memristor.

Memristor-based neural networks are currently being researched in order to replace complementary metal-oxide-semiconductor (CMOS) devices in neuromorphic circuits with memristors and to investigate their potential applications. Current research primarily focuses on the utilisation of memristors as synaptic connections between neurons, however in any application it may be possible to allow memristors to perform computation in a natural way which attempts to avoid additional CMOS devices. Examples of such methods utilised in neural networks are presented in this thesis, such as memristor-based cellular neural network (CNN) structures, the memristive spiking-time dependent plasticity (STDP) model and the exploration of their potential applications.

This thesis presents manifold studies in the topic of memristor-based neural networks from theories and feasibility to approaches to implementations. Studies are divided into two parts which are the utilisation of memristors in non-spiking neural networks and spiking neural networks (SNNs). At the beginning of the thesis, fundamentals of neural networks and memristors are explored with the analysis of the physical properties and $v - i$ behaviour of memristors. In the studies of memristor-based non-spiking neural networks, a staircase memristor model is presented based on memristors which have multi-level resistive states and the delayed-switching effect. This model is adapted to CNNs and echo state networks (ESNs) as applications that benefit from memristive implementations. In the studies of memristor-based SNNs, a trace-based memristive STDP model is proposed and discussed to overcome the incompatibility issues of the previous model with all-to-all spike interaction. The work also presents applications of the trace-based memristive model in associative learning with retention loss and supervised learning.

The computational results of experiments with different applications have shown that memristor-based neural networks will be advantageous in building synchronous or asynchronous parallel neuromorphic systems. The work presents several new findings on memristor modelling, memristor-based neural network structures and memristor-based associative learning. These studies address unexplored research areas in the context of memristor-based neural networks to the best of our knowledge, and therefore form original contributions.

Acknowledgements

First of all I would like to thank my supervisor Professor Frank Wang for the opportunity of undertaking this work with his supervision. His guidance and support throughout my PhD leading to this thesis has been invaluable. I am grateful to Professor Richard Jones, Professor Sally Fincher and Doctor Fred Barnes for their invaluable help and support to my studies in the School of Computing. I would like to thank Professor Howard Bowman for the discussions we had regarding neural networks about my research which were very helpful.

Furthermore, I would like to thank Professor Leon Chua for his discussions and insightful comments on my work of memristor-based neural networks during his visit to the United Kingdom. I am also very grateful to Doctor Zdeněk Biolek for his patient replies in discussing memristor modelling.

I wish to express my gratitude to my many friends who make my time in Kent very enjoyable. I would like to thank Mian-Guan Lim, Bin Ye, Adrien Witon and Gbola Akanmu for the great atmosphere in our work place and for our group discussions on research, and especially Wanlong Chen for the joint research on memristors. I wish to thank Doctor Sining Wu for his guidance at the beginning of my research work.

Finally, I owe a big thank you to my family. A special thanks to my parents for the opportunity and the courage they have given me to my live abroad. I am also very grateful to my wife for all her love, support and patience.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Brain-like computing

Distinct from traditional computing, brain-like computing is inspired by biological mechanisms. It is composed of a large number of interconnected neurons working in parallel to solve specific problems. The networks are not programmed since the connections between neurons are "weighted" according to the correlations of data they have already learned. This idea, named "synaptic plasticity", is also inspired by the studies of biological neural networks. Brain-like computing is a complement to conventional computing rather than a replacement of it because of the distinctions between them. Since conventional computing uses algorithmic approaches, it is competent at fast arithmetic or with unambiguous tasks but lacks abilities such as massive parallelism and fault tolerance. An example is that cognition, perception, and learning on biological spatial and temporal scales can be easily achieved by mammalian brains, however such tasks remain out of reach for modern computers. As a complement, brain-like computing which benefits from similar mechanisms of biological neural networks can do these jobs that conventional computing cannot do well. Artificial neural networks that are foundation stones for brain-like computing have become a thriving research field nowadays. Since the first artificial neural network was proposed, research was split into two distinct approaches which separately focus on biological processes in the brain and the applications of neural networks to artificial intelligence.

Artificial neural networks are developed based on the understandings of the distinctive features of the brain, such as its ability to learn and remember. Since

Donald Hebb proposed the famous Hebb's postulate in his book "The organization of Behaviour" in 1949, it is widely believed that learning and memory depend on the changes in the efficacy of synaptic connections. Hebb brought together many previous studies and findings on plasticity, learning and memory and suggested an experimental testable prediction of synaptic plasticity [45]. Hebb's postulate summarised that if a presynaptic cell repeatedly takes part in firing a postsynaptic cell, then the synaptic connection between them should be enhanced. The inverse to Hebbian learning was hypothesised by Stent [95] to explain the loss of synaptic connections since Hebb's postulate lacks a mechanism to weaken the synaptic connection. Although, from the view of current research, Hebbian learning is based on an oversimplified view of neuronal morphology [90], it is quantitatively implemented in artificial neural networks through the changes of synaptic efficacy between neuron units. Another mechanism called spiking-time dependent plasticity (STDP) was originally proposed by Gerstner, Ritz and Van Hemmen [40] as an advanced learning algorithm focusing on the temporal correlations of spikes. There is a large and growing interest in spike-timing dependent plasticity, and especially its applications in neuromorphic engineering, meanwhile experimental evidence has been reported by neuroscience groups [6, 73] which shows that STDP occurs in the brain.

Currently, there are many active projects that focus on brain-like computing under the neuromorphic concept that was developed by Mead and Ismail [76] in the late 1980s. The term neuromorphic [67] is used to describe very-large-scale integration (VLSI) systems containing electronic analog circuits that mimic neurobiological architectures present in the nervous system. Nowadays, not only analog but also digital or mixed-mode analog/digital VLSI systems, which implement neural system models and software algorithms, are described by the term neuromorphic. Neuromorphic engineering is an interdisciplinary study that takes inspiration from biology, physics, mathematics, computer science and engineering. It demonstrated promising performance on artificial neural system, such as vision systems, auditory processors and autonomous robots, whose physical architecture and design principles are based on biological nervous systems. As the term neuromorphic combines "neuro" and "morphic", the key to neuromorphic design is understanding the morphology of individual neurons, circuits, and overall architectures and how this creates the desired computations [67]. Furthermore, it is important to understand how the morphology influences the representation of information, robustness, learning, development and evolutionary change of biological nervous systems. Such technologies have been adapted for several sophisticated projects like Blue Brain, SpiNNaker, and MoNETA to simulate part of the brain's functions.

Blue Brain

The goal of the Blue Brain project is described by Markram [72] as

"to simulate the brains of mammalians with a high level of biological accuracy and, ultimately, to study the emergence of biological intelligence."

It provides a computational platform to simulate the mammalian brain down to the molecular level, which has succeeded in simulating a rat cortical column containing 10,000 neurons. Building on Blue Gene/L supercomputers, the Blue Brain project can provide a better understanding of the biological neural networks, and the obtained data could even help to study neurological diseases.

SpiNNaker

SpiNNaker, which stands for Spiking Neural Network Architecture which was funded by EPSRC, is a hardware-based real-time simulator following an event-driven computational approach [39]. Dedicated hardware and software are developed to simulate the features of biological neural networks in the following ways:

1. Parallel computation
2. Spiking communications
3. Event-driven behaviour
4. Distributed memory
5. Reconfigurability

It mainly consists of the SpiNNaker chips which are the cores of the simulator and the full-custom routers which are responsible for the communications between cores and the input/output links. It is a successful project which contributes to the follow-up project of Biologically Inspired Massively Parallel Architectures (BIMPA) in order to build a larger scale machine with SpiNNaker.

MoNETA

The recent project MoNETA, which stands for Modular Neural Exploring Travelling Agent which was funded by the U.S. Defense Advanced Research Projects Agency (DARPA), represents a family of neural modelling projects in the neuromorphics lab at Boston University [98]. The MoNETA project has been designed in Cog Ex Machina that was jointly developed by HP and Boston University. In the future, it will be based on hardware architecture with memristive devices as the analog synaptic memories [92]. STDP-based learning is implemented in the project, which aims to build neuromorphic chips with memristors. The first version has been completed, and it was able to negotiate the Morris water maze task [2] which is a behavioural procedure used in the study of spatial learning and memory.

1.2 The fourth element: memristor

The memristor, short for memory-resistor, is a newly emerged device which was postulated by Chua in 1971 as the fourth circuit element [25]. Following his concept, there are four fundamental circuit elements, and each element represents a two-variable relationship between the four basic circuit variables, namely, charge q, current i, flux φ and voltage v. Including the two constitutive relationships defined by

$$q(t) = \int_{-\infty}^{t} i(t)dt \tag{1}$$

and

$$\varphi(t) = \int_{-\infty}^{t} v(t)dt \tag{2}$$

there are six two-variable combinations of the four circuit variables in total as shown in Table 1. In addition to the known links which are the resistor, capacitor and inductor, the postulate of the memristor reveals the missing link between flux φ and charge q. In general, the memristor is a two-terminal passive element whose resistance varies according to the past history of current i because it integrates

Circuit Element	Relationship	Representation
Resistor	$v - i$	$dv = Rdi$
Capacitor	$q - v$	$dq = Cdv$
Inductor	$\varphi - i$	$d\varphi = Ldi$
Memristor	$\varphi - q$	$d\varphi = M(q)dq$
N/A	$q - i$	$q(t) = \int_{-\infty}^{t} i(\tau)d\tau$
N/A	$\varphi - v$	$\varphi(t) = \int_{-\infty}^{t} v(\tau)d\tau$

Table 1: Four elements and the two-variable combinations of the four basic circuit variables

the current i over time.

The memristor has been found in nanoscale by several groups utilising different materials such as the titanium-dioxide based memristor [97], the ferroelectric memristor [19], the tungsten-oxide based memristor [17, 53] and diamond-like carbon based memristor [24]. During the development of the memristor, research into its applications covers several scientific fields, in which applying the memristor in artificial neural networks is intensively studied because of its intrinsic features. This will be covered by this thesis, and the problem and objectives will be discussed in the following section. In addition, the memristor has been applied to content-addressable memory (CAM) [22, 107] and the ternary CAM (TCAM) [57] to reduce their size and power consumption. However, these are just a few examples of possible applications of memristors. In following chapters, more applications will be introduced for the sake of completeness.

1.3 Statement of the problem and objectives

Since the learning capability of artificial neural networks is desired in many applications, researchers from engineering fields are interested in implementing the framework of artificial neural networks by adapting hardware elements. Neuromorphic engineering is a derivative of this research. In neuromorphic engineering, the neurons and synapses are largely built by transistors which are based on CMOS technology. However, current CMOS technology is approaching a bottleneck in terms of size. The fundamental barrier, the size of its constituting atoms, may result in a cease in Moore's Law. On the contrary, the neural networks implemented by neuromorphic engineering have continued to expand in last decade. Since it is estimated that a human brain has 10^{10} neurons and 10^{14} synapses (on average, each neuron is connected to other neurons through approximately 10,000 synapses), any realistic implementation of a synapse should ideally be at least four orders of magnitude smaller than that required to build a neuron. Therefore, there are several challenges in current implementations which are based on CMOS technology: the size of the neuron and the size of the synapse. The new device, namely the memristor, is a promising candidate to replace the CMOS-based transistor in order to reduce the size of traditional neural network circuits.

Since both neural networks and memristors are promising technologies, they can be combined together to improve current neuromorphic systems. However, as newly emerging devices and compared to neural networks, memristors are not well studied from theories to applications. To date, memristors are still at an early development stage, and various physical implementations of memristors are emerging. Depending on physical materials, memristors demonstrate different physical properties and behaviour. It is yet unclear, to what extent, various memristors are beneficial to different kinds of neural networks and feasible applications. In addition, there are some fundamental open problems in the implementations of memristor-based neural networks. For example, to what extent can memristors replace more conventional devices in neural networks, and exact improvement can be achieved in terms of size and power consumption. Furthermore, certain modifications and specific learning rules are required to apply memristors to current systems, which implies there is no general implementation for all tasks. Therefore, we have to consider proper approaches to the utilisation of memristors in specific neural network applications in order to achieve satisfactory results.

The aim of this research is to investigate memristor-based neural networks from theories to feasible approaches. At the beginning of this thesis, fundamental theories are reviewed to show the advantages of memristors in different applications. In the main part, it provides improved models and possible implementations for the memristor-based neural networks both in non-spiking neural networks and spiking neural networks. In non-spiking neural networks, it focuses on the applications of piecewise linear memristors which have a significant delayed-switching effect and cellular neural network based structures where memristors are used as local connections. The main idea is to take advantages of the structures and memristors in order to enable large neighbourhood connections yet simple implementations. In spiking neural networks, the research focuses on the applications of non-linear memristors and STDP based learning. This aims to provide a memristive STDP model which is compatible with both the nearest-neighbour and all-to-all interactions. In addition, by introducing a rate-based term and supervised learning implementation, this model provides an ability of retention loss and supervised learning respectively which are unexplored research areas in this field. Thus, the network which is built on the proposed models could provide more flexibility and options for current and future research.

1.4 Contributions

In this thesis, the main contributions are presented next:

- **A staircase memristor model:** Based on the delayed switching effect and experimental results, a staircase memristor model is proposed to take the advantage of the delayed-switching effect, however, in other cases the delayed switching effect should be avoided. It not only has the functionality of the basic memristor model but also provides extra stability as a programmable memristor.

- **Staircase memristor based CNN structure and its applications:** A staircase memristor based CNN structure is proposed and applied to an echo state network (ESN). It is a new learning algorithm with promising performance in demonstrated applications such as pattern recognition and data prediction.

- **Memristive STDP model:** While current memristive STDP models do not cope with all-to-all interaction, a trace-based model to cope with both nearest-neighbour and all-to-all interactions is presented, taking the advantages of memristors and thus being suitable for memristor-based spiking neural networks.

- **Applications of the proposed model to associative learning with retention loss and supervised learning:** The proposed memristive STDP model for spiking neural networks is evaluated through associative learning, retention loss and supervised learning, demonstrating its ability to adapt to different tasks.

1.5 Structure of the thesis

The introductory chapter is followed by one background chapter, providing additional information about neural networks, memristors and the research areas related to the thesis.

Chapter 2

The second chapter presents the background of neural networks and memristors. Biological mechanisms behind neural networks are briefly introduced and discussed with artificial neural networks. Memristor fundamentals focus on the delayed-switching effect and memristor models which are widely utilised in current research fields and their applications. Numerous mathematical models and simulators are proposed to describe the behaviour of the memristor but there is a focus in this chapter on the HP memristor model, the piecewise linear memristor model and the cubic polynomial memristor model.

Following the background chapter, the research of the thesis is presented in Chapters 3 to 6.

Chapter 3

In this chapter, the staircase memristor model is investigated for specific tasks where a stable programmable memristor is required such as the cellular neural network. Then the memristive cellular neural network is implemented based on the staircase memristor. This structure is adapted to the paradigm of reservoir computing, which makes the implementation of echo state network applicable with CNN circuits.

Chapter 4

Chapter 4 presents the simulation results of the proposed memristive CNN in image processing with selected template settings. Besides, the echo state network which uses a similar structure is evaluated through a benchmark test which uses the Mackey-Glass data.

Chapter 5

This chapter describes the proposed memristive STDP model for STDP learning in spiking neural networks. Its compatibility with both nearest-neighbour and all-to-all interactions is investigated with the memristor model. Furthermore, a possible implementation of the proposed model in a neuromorphic system is discussed.

Chapter 6

Chapter 6 presents the practical applications and evaluations of the proposed memristive STDP model with both all-to-all and nearest-neighbour interactions. Associative learning and retention loss are simulated by slightly modifying the parameters of the model, which shows the proposed model can be adapted to different applications. Furthermore, a supervised learning is presented using the proposed model under a proper learning rule.

The thesis is concluded in Chapter 7. Whilst, Appendix A provides additional information about the full publication list and research activities involved.

Chapter 7

This chapter draws conclusions, providing a summary of the contributions and the analysis of the results obtained. In addition, the possible future research directions are discussed.

Appendix A

All the publications and acronyms pertinent to this thesis are listed in this appendix and show the course of my research.

1.6 Publication list

The following list of publications has directly resulted from the the research presented in this thesis.

Peer-Reviewed Conference Papers:

- Xiao Yang, Wanlong Chen, Frank Z. Wang. "A supervised spiking time dependent plasticity network based on memristors." Computational Intelligence and Informatics (CINTI), 2013 IEEE 14th International Symposium on, pp. 447-451.

- Xiao Yang, Wanlong Chen, Frank Z. Wang. "The memristor-based associative learning network with retention loss." Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on, pp. 249-253.

- Xiao Yang, Wanlong Chen, Frank Z. Wang. "The staircase memristor and its applications." Electronics Circuits and Systems (ICECS), 2014 IEEE 21st International Conference on, pp. 259-262.

Journal Papers:

- Frank Z. Wang, Na Helian, Sining Wu, Xiao Yang, Yike Guo, Guan Lim, and Md Mamunur Rashid. "Delayed switching applied to memristor neural networks." Journal of Applied Physics, 111, pp. 07E317-07E317, 2012

- Frank Z. Wang, Leon O. Chua, Xiao Yang, Na Helian, Ronald Tetzlaff, Torsten Schmidt, Caroline Li, Jose Manuel Garcia Carrasco, Wanlong Chen, and Dominique Chu. "Adaptive Neuromorphic Architecture (ANA)." Neural Networks, 45, pp. 111-116, 2013.

Chapter 2

Memristor-based Neural Networks: Fundamentals

Mathematical tools have been supporting development in various scientific fields, such as offering a scope to understand the phenomena of the neural networks and electronic systems. In 1971, a new device, namely the memristor, was postulated by Chua [25] from the view of symmetry and completeness in mathematical links among the four basic circuit variables, which was also proved on the grounds of electromagnetics. However, the postulation was not of wide concern to researchers until almost 40 years later when the titanium-dioxide based memristor was realised by one of the HP labs in 2008 [97]. Hence, research on the memristor was revived, and many studies have been dedicated to investigating the characteristics of the memristor and its applications across many fields. In this chapter, we first explore neural networks from the biological mechanisms behind them and artificial neural networks which abstract and extract the biological mechanisms. After this brief introduction, advantages of memristors in neural networks are listed and discussed from different aspects. Following this, we trace back to the original concept of the memristor and the broadened concept of the memristive system. Then the delayed-switching effect is studied from its impacts on power consumption and usage. Afterwards, several memristor models which focus on different characteristics will be introduced. For the sake of completeness, other applications of memristors are introduced at the end of this chapter.

2.1 Biological mechanisms behind neural networks

To establish new understanding of the human brain and the mechanisms behind it, over the past few decades, a large number of experimental results on synaptic plasticity have been accumulated. Synaptic plasticity is where synapses, the connection sites between neurons, have a plastic property which allows them to mutate according to different rules. It is widely assumed to be the mechanism behind memory and learning since it is believed that the learning and memory of human beings is carried out by changing synapse strength [74]. In current studies, such as that of artificial intelligence and neural networks, synaptic plasticity is represented by "weight" which influences the amplitude of postsynaptic response to an incoming potential. In this section, a brief introduction on biological mechanisms of synaptic plasticity and the relevance with memristors will be given.

Since there are a number of distinct types of neurons, in this introduction, we are concerned with the structural features and operations of typical neurons. An example of two interconnected neurons is illustrated in Fig. 1. For the sake of simplicity, some detailed components are neglected but we fundamentally understand that a neuron primarily consists of four parts: dendrite, cell body, axon and axon terminal. Dendrites are organised in many of the dendritic trees where signals are received and carried towards the cell body. A cell body contains the nucleus of the cell and connects to the axon. Axon branches, at the end of the axon, conclude at axon terminals where synapses are formed. The most common synapses are chemical synapses as shown in Fig. 1 which convert electrical signals into chemical signals and then transmit the signals to the postsynaptic neuron. When an action potential is caused in the presynaptic cell body and arrives at synapses along the axon, it triggers the migration of synaptic vesicles, which contain neuron transmitters, towards the postsynaptic membrane. Vesicles will release neurotransmitters into the synaptic cleft, which is the tiny gap between two nerve cells, when they have fused with the presynaptic membrane. Neurotransmitters travel across the synaptic cleft, bind with receptors of the postsynaptic neuron and influence the excitability of the postsynaptic neuron. Eventually, the signals of the presynaptic neuron arrive at the postsynaptic neuron through synapses. All the signals

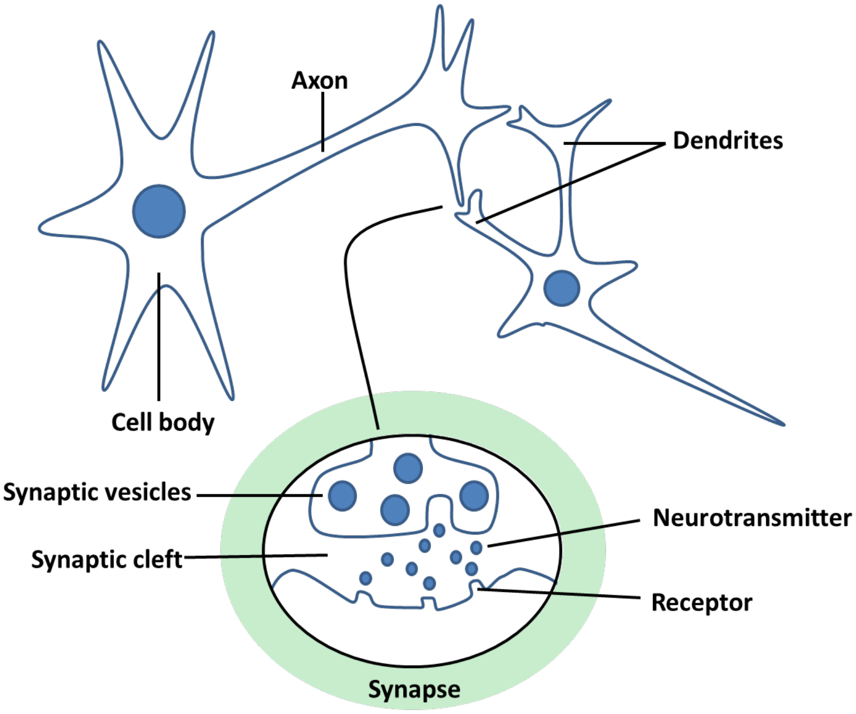

Figure 1: Two neurons are connected through the connection sites - synapses. The cell body collects signals from presynaptic neurons through dendrites. If an action potential is caused and arrives at synapses along an axon, it will be converted into chemical signals in synapses and then transmitted to the postsynaptic neuron.

from dendrites will be collected and added to cause an action potential in the postsynaptic neuron if the total summation is large enough.

The example shown in Fig. 1 is a simplified case since the interconnections of natural neurons are enormous and much more complex. It is estimated that there are approximately 10 billion neurons in the human cortex, and each of which is connected, on average, to 10 thousand other neurons. That makes the brain astonishingly complex with highly interconnected networks of neurons and trillions of synapses or synaptic connections [88].

2.1.1 Plasticity in biological neural networks

Hebbian learning

Hebbian learning is a biological process which describes the basic mechanism of synaptic plasticity. Hebb's postulate of learning is one of the oldest and most famous learning rules. It was introduced by Donald Hebb [45] in 1949, and the main theory is summarised later as “cells that fire together, wire together”.

Hebbian learning can be developed and expanded into two parts: “cells that fire together, the strength between them is enhanced” and “cells that fire asynchronously, the strength between them is weakened”. However, in Hebb’s original postulate, the mechanism of weakening synaptic strength has not been proposed. A few years later, Stent [95] proposed the inverse to Hebbian learning to explain the loss of synaptic connections. Obviously, Hebbian learning is a time dependent mechanism since the strength changes depend on the exact time of the occurrence of the cell’s signals.

As shown in Fig. 1, synapse is the connection site of the presynaptic neuron and the postsynaptic neuron. According to Hebb’s postulate, if the presynaptic cell repeatedly takes part in firing the postsynaptic cell, then the synaptic strength should be enhanced. Otherwise, if the presynaptic cell does not take part in firing the postsynaptic cell, the synaptic strength should be depressed.

Long-term potentiation and long-term depression

Long-term potentiation (LTP) and long-term depression (LTD) are important mechanisms in biological neural networks. Both of them have been found in the hippocampus but represent different phenomenons. They are possible evidence to support both Hebb’s postulate and the inverse to Hebbian learning which suggest that synaptic plasticity is achieved through changing the synaptic strength.

It has been found that excitatory postsynaptic potentials (EPSPs) evoked by an electrical stimulation in a rabbit hippocampus increased after a repeated high-frequency stimulation [11]. This phenomenon that induces a long-lasting increase in the plasticity of a connection is called LTP. It is a possible mechanism for information storage since the hippocampus is a region known to be crucial for long-term memory formation. The repeated high-frequency stimulation presumably leads to LTP that implies the connected cells are correlated, and therefore the result of synaptic strengthening is consistent with Hebbian learning.

Conversely, LTD is the functional inverse of Hebbian learning, which induces a long-lasting decrease in the synaptic response of neurons to a stimulation. There are two types of LTD, which are heterosynaptic LTD and homosynaptic LTD.

For heterosynaptic LTD, the depressing synapse is not active during LTD induction. Its electrophysiological evidence has been found in the hippocampus. When attempts have been made to induce LTP of one pathway by a repeated high-frequency stimulation, LTD occurs in the inactive pathways. Homosynaptic LTD is input specific and restricted to the individual synapse. It was discovered in the hippocampus as well as the heterosynaptic LTD. Typically the homosynaptic LTD is induced by a low-frequency stimulation.

Spiking-time-dependent plasticity

Based on Hebbian learning, spiking-time dependent plasticity (STDP) was proposed as an advanced learning algorithm focusing on the temporal correlations of spikes [40], and experimental evidence has been reported by neuroscience groups [6, 73]. Markram et al. [73] controlled pre- and postsynaptic spike timing and discovered that the sign and magnitude of LTP and LTD indeed depend on the order and timing of the pre- and postsynaptic spikes as predicted by Gerstner, Ritz and Van Hemmen [40]. This had been investigated in detail and gave a typical STDP window by Bi and Poo [7]. It has been observed that, typically, LTP is produced if the presynaptic cell fires a few milliseconds before the postsynaptic cell; the opposite temporal order leads to LTD. In addition, longer delays only have a trivial impact on synaptic plasticity. The observed result shows, under STDP, that long-term plasticity critically depends on the millisecond timing of presynaptic and postsynaptic spikes.

In the STDP paradigm, the synaptic connection strength is modified according to the exact timings of the spikes of presynaptic and postsynaptic neurons. An illustration of a typical STDP learning window is shown in Fig. 2.

Synaptic strength between two connected neurons is expressed as a function of the time difference ΔT between the postsynaptic spike T_{post} and the presynaptic spike T_{pre}. Such correlation is described as

$$\Delta w = f(\Delta T)$$

$$\Delta T = T_{post} - T_{pre}$$

For a positive time difference ΔT, the change of synaptic strength Δw is positive, and it reduces as $|\Delta T|$ increases. That means the postsynaptic spike is

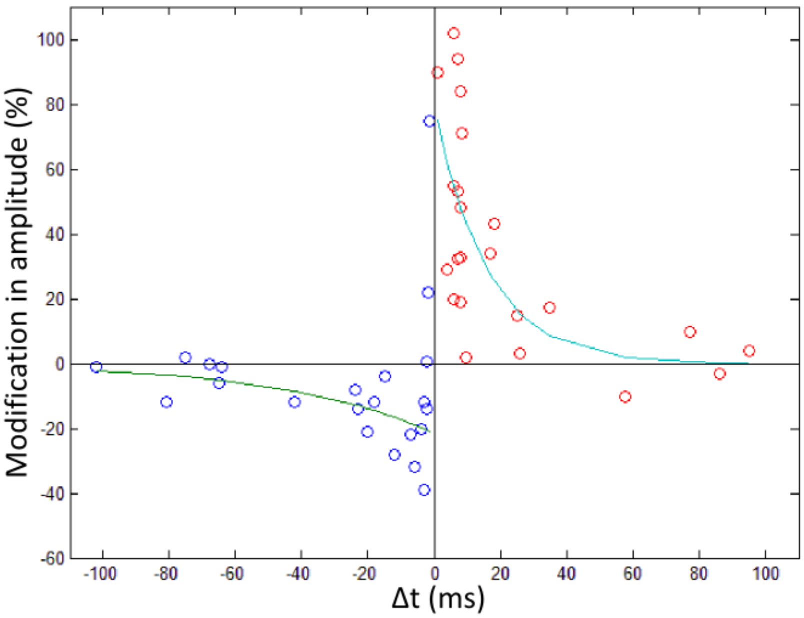

Figure 2: A typical STDP learning window. Experimental data from [6] show the impacts of different temporal orders of presynaptic spikes and postsynaptic spikes on biological synapses. The modifications in excitatory postsynaptic current (EPSC) amplitude are measured in pA and normalised to percentage. The negative delays ($\Delta T < 0$) and positive delays ($\Delta T > 0$) are separately shown on the left side and right side.

fired after the presynaptic spike reaches the postsynaptic cell, and therefore the postsynaptic spike is very likely induced by the pre-synaptic spike. The synaptic strength w between these two neurons is enhanced. For a negative time difference ΔT, there will be a negative synaptic weight change, and it reduces as $|\Delta T|$ increases. Consequently, the strength will be weakened, and w is depressed.

2.2 Artificial neural networks

2.2.1 Non-spiking neural networks

Non-spiking neural networks are considered as the 2nd generation of artificial neural networks [69], which utilise activation neuron models such as the sigmoid model, and the activation is representative of the "mean firing rate" of a neuron. Such networks have been successfully adapted to many engineering applications and have proven effective at modelling some cognitive processes, which normally involve a two-part computation. For the first part, inputs weighted by synapses are added, whilst in the second stage, the sum is applied to the neurons. Normally, a sigmoidal neuron applies a sigmoid transfer function which outputs real-valued numbers in the range $(0, 1)$. The neural networks are sets of connected artificial neurons, for example, the multi-layer perceptron. The learning rules for such neural networks determine how to adjust the weights to improve the performance of the task.

2.2.2 Spiking neural networks

The nervous system requires a continuous energy supply to maintain the ability to perceive, remember and react to the outside world, however the overall power consumption is remarkably low, which inspires researchers to study spiking neural networks [86]. In artificial intelligence, spiking neural networks (SNNs) are built using biologically plausible spiking neuron models, and are considered as the 3rd generation of artificial neural networks [69]. The distinct difference between SNNs and the neural networks mentioned previously is the method by which neurons communicate. In SNNs, spikes that are very short signals are utilised to deliver information from one neuron to other neurons. By utilising spikes, the concept of time is introduced into neural networks, and in addition, it resembles the biological neural networks which also use spikes (electric pulses) for communications. Thus, a significant advantage of SNNs is that they are innately embedded in time. For example, spiking latencies, refractory periods and network oscillations all give rise to an intrinsic ability to process time-varying data naturally compared to non-spiking neural networks. In the mathematical model of SNNs, it mainly involves computations of summing all the weighted inputs, integrating the sum over time and comparing the result with a threshold. If the threshold is reached, a spike is emitted by the neuron.

2.2.3 Learning in artificial neural networks

The most attractive capability of artificial neural networks is its ability to learn. To some degree, it mimics the biological neural networks' ability of adapting to the surrounding environment to achieve specific tasks. As introduced in Section

2.1, learning in biological neural networks is accomplished partially through the concept of synaptic plasticity which modifies the connection strength in the networks. This concept is absorbed by artificial neural networks, and several learning paradigms are developed based on it, such as supervised learning, unsupervised learning and reinforced learning.

Supervised learning

In supervised training, a given set is split into input data and desired output data. The input data refers to some sequence of sensory data such as a sound waveform and image pixels. The desired output data refers to some sequence of goals that should be achieved by the network and corresponding input data. During the training phase, the pair of input and desired output data is passed through the network, which alters the weights of the network. After all the epochs are complete, the trained network is tested by test data to determine the performance of the training. Supervised learning process normally involves minimising the error between desired output and actual output by utilising various techniques such as gradient descent.

Unsupervised learning

Distinct from supervised learning, unsupervised learning receives only input data whilst the desired output data are not provided. By putting input data into the network, the system learns to gain experience from the input and tries to find patterns. Once the network has adapted to the input data, internal representations of the features of input data are built, thus utilising it for decision making and predicting future inputs. The previously mentioned STDP is a kind of unsupervised learning, which will be introduced and utilised in Chapter 5.

Reinforced learning

Reinforced learning involves actions and rewards or punishments, which somewhat resemble the learning of animals in nature. It is closely related to the field of control theory in engineering and widely applied to engineering applications such as robotics. The actions result in rewards if the desired outcome is produced, conversely, a punishment is received if the desired outcome is not achieved. By introducing the rewards and punishments, the behaviours are corrected correspondingly and learned to produce desired outcomes.

2.3 Memristor-based neural networks

Using memristors, especially as synapses in neural networks, has been proposed and discussed in relation to various kinds of neural networks. Snider [94] proposed a synchronous network based on digital circuits and memristors to implement STDP learning. Asynchronous networks based on CMOS neurons and memristors were proposed by [1, 67] using spike signals. A simplified memristor-based neural network was demonstrated in [16, 80, 100], which shows that such a network is capable of associative learning, and it reproduced the Pavlov's dog experiment. Researchers are interested in applying memristor in artificial neural networks because the memristor has promising features which are primarily:

- memory effect
- nanoscale size
- non-volatility
- passivity

Memory effect As mentioned previously, the memristor's resistance, namely memristance $M(q)$, varies according to the past history of current i, which implies the memristor has a memory effect. This effect is, to some extent, very similar to the learning process of animals and synaptic plasticity because the learning outcomes depend on the past history. Moreover, similar behaviours to LTP and LTD have been implemented by memristors in [54], in which positive 3.2 V pulses were used to induce LTP and negative -2.8 V pulses induced LTD as shown in Fig 3. This shows the potential of memristors to mimic synaptic plasticity by applying a stimulus for a certain time. Memristors are also related to synaptic plasticity by STDP learning and the concept of back-propagation spikes which are used in combination with a presynaptic spike to vary the conductance of memristors. Linares-Barranco and Serrano-Gotarredona [66] used this concept to demonstrate

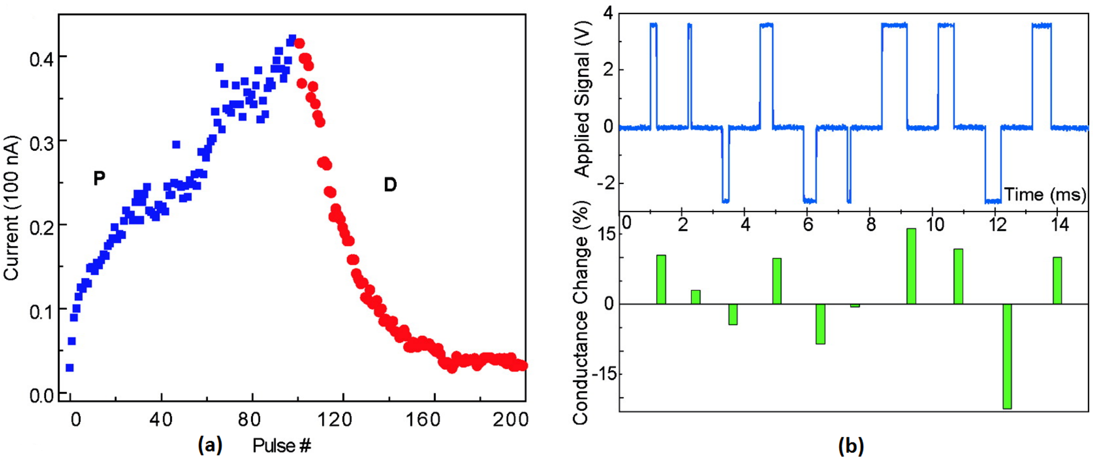

Figure 3: Similar behaviours to LTP and LTD are induced by applying consecutive pulses to the memristor (From [54]). (a) The conductance of the memristor is increased or decreased by applying positive or negative pulses. (b) The variation of the memristor's conductance according to the mixed potentiating and depressing pulses.

that LTP and LTD can be induced when memristors and STDP rule are used with a specific spike shape. Even retention loss could be mimicked by a memristor with a forgetting effect which occurs in the tungsten-oxide based memristor [17, 20]. A diffusion term is introduced to capture the forgetting effect and explain the short-term memory behaviour of the memristor. This aforementioned research explored the potential of a memristor's memory effect and its applications in synaptic plasticity, which only utilises one memristor but demonstrates promising capabilities of synapses based on different physical materials or learning rules. On the contrary, one CMOS transistor is not capable of producing these features alone. Therefore, the memory effect of the memristor is the most advantageous and significant feature that attracts researchers.

Nanoscale size Since the titanium-dioxide based memristor was fabricated by one of the HP labs, more and more memristive devices have been developed by nanoscale technology. The titanium-dioxide memristors are fabricated with 50nm half-pitch in present research, however memristor based nano-crossbar circuit could be fabricated with half-pitch down to 17nm by using nanoimprint lithography in the future [56, 103]. At that time, memristor-based memory could provide the very promising capacity of as much as 100 $Gbit/cm^2$, in contrast to CMOS-based Dynamic Random Access Memory (DRAM) which could provide approximately 46 $Gbit/cm^2$ [50, 59]. Moreover, the $LaAlO_3/SrTiO_3$ junction presents the identical pinched hysteresis loop of the memristor, which shows the potential of the memristor to be scaled down to half a nanometer [37]. Although, in current research, the size of the memristor varies depending on different physical materials, it has the potential to be reduced in the future considering that memristor is still a newly emerged device.

Non-volatility The memristor is also a non-volatile device which is able to retain stored information even when not powered. Although volatile memristors (tungsten-oxide based materials) exist and demonstrate a forgetting effect over time, most memristors developed in the past 7 years are non-volatile or have a good non-volatility, such as the HP memristor, the ferroelectric memristor and the ZnO thin film memristor [48]. Non-volatility is a very promising feature for the implementation of artificial neural networks, which could store synaptic weights for a long time even without a power supply. Therefore, the last states of synapses will not be lost and can be recalled instantly when the network is utilised again. This would eliminate a period for setting up the previous states of synapses, meanwhile, it saves the memory space required to store the synaptic weights.

Passivity Passivity was discussed by Chua [25] through investigating the constitutive relation of the memristor, which gives the instantaneous power which is dissipated by the memristor:

$$p(t) = M(q(t))i^2(t) \tag{3}$$

Hence, if the memristance $M(q(t)) \geq 0$, the power dissipation is always a positive term which shows that the memristor does not store any energy. Consequently, the memristor is a passive device which only dissipates power. On the contrary, the CMOS devices, for example transistors, are active devices which require an external power supply for their operations, which implies these devices consume more power than the passive devices in real applications. This feature is important for large-scale implementation of neuromorphic systems in terms of green computing and power saving.

Non-linearity In contrast to a linear resistor, the memristor has a dynamic relationship between current i and voltage v as defined in [25]:

$$M(q) = d\varphi(q)/dq \tag{4}$$

where the memristance $M(q)$ is a charge dependent term and varies according to the integration of current over time. As a non-linear analog device, it is suitable for neural network applications since biological systems are non-linear systems in nature. It implies that a single memristor is able to mimic the non-linear feature of the synapse, however, a digital system requires several bits to store the same information.

2.4 Memristive system

In 1976, Chua and Kang [28] gave the original definition of the memristor and developed a broadened class of dynamical systems named the memristive system which is defined by

$$\dot{x} = f(x, u, t) \tag{5}$$

and

$$y = g(x, u, t)u \tag{6}$$

where u and y denote the input and output of the system respectively. x is the state of the system. In Eq. (6), the function g is a scalar function, which means the output y is zero whenever the input u is zero and regardless of the state x of the system. It recalls the passivity of the memristor introduced in Section 2.3 and implies no energy is stored in the memristive system.

By expanding the dimension of the state space of the dynamical system, an n^{th}-order current-controlled memristive system can be defined by

$$\begin{cases} \dot{w} = f(w, i, t) \\ v = R(w, i, t)i \end{cases} \tag{7}$$

where the term w denotes the n dimensional state variable of the system. v and i are the voltage and current respectively. Similarly, an n^{th}-order voltage-controlled memristive system can be defined by

$$\begin{cases} \dot{w} = f(w, v, t) \\ i = R(w, v, t)v \end{cases} \tag{8}$$

By introducing the n dimensional state variable, the memristive system is not necessarily limited to the original postulate because the memristor is just a special case of the memristive system which only depends on the charge or flux.

2.4.1 The memristor

The memristor is postulated through the constitutive relation between the flux φ and charge q, however, a physical relation between φ and q is not necessary [26]. In electronics, the flux and charge can be mathematically defined as the integral of voltage and current over time respectively:

$$\varphi(t) = \int_{-\infty}^{t} v(\tau) d\tau \tag{9}$$

and

$$q(t) = \int_{-\infty}^{t} i(\tau) d\tau \tag{10}$$

Since the memristor represents the relation between φ and q, it could also be expressed by the derivative of φ (voltage v) and the derivative of q (current i) according to Eq. (9) and (10). For a charge-controlled memristor, the voltage across it is given by

$$v(t) = M(q(t))i(t) \tag{11}$$

where

$$M(q) \equiv d\varphi(q)/dq \tag{12}$$

In this case, the $M(q)$ is called memristance and has the unit of Ohms, which is determined by the charge, or in other words the history of current i. The name of the memristor derives from memory-resistor which combines the memory effect and the resistor.

Similarly, a memristor can be controlled by the flux, which gives

$$i(t) = G(\varphi(t))v(t) \tag{13}$$

where

$$G(\varphi) \equiv dq(\varphi)/d\varphi \tag{14}$$

is called memductance and has the unit of Siemens. From the definition of $M(q)$ and $G(\varphi)$ in Eq. (12) and (14), as a special case, if φ and q have a linear relation, the memristance $M(q)$ and memductance $G(\varphi)$ are actually equivalent with resistance and conductance respectively. However, generally, the memristor is considered as a non-linear element.

2.4.2 The pinched hysteresis loop

The pinched hysteresis loop shown in the $v - i$ plane has been identified as the fingerprint of the memristor by applying a sine-wave like signal, which is illustrated in Fig. 4. The simulation results shown in Fig. 4 exhibit two important features: Hysteresis and Zero crossing.

Hysteresis The movement of the $v - i$ curve is indicated by the dashed arrows. In the first quadrant, the $v - i$ curve starts from a zero point and moves towards the top-right. At first, the change rate of current i lags voltage's until the $v - i$ curve reaches the turning point at the top-right. Then the current i begins to precede the voltage v until the $v - i$ curve returns to zero point. Conversely, in the third quadrant, the situation is reversed, and the current i precedes the voltage v at the beginning but lags the voltage v after the turning point at the bottom-left. The hysteresis implies that the maxima and minima of the applied sinusoidal input voltage v and the memristor current i do not occur at the the same time, and hence there is always a hysteresis between voltage and current.

Zero-crossing It is worth noting that both of the current i and voltage v have the same frequency, and they cross the zero point at the same time. This peculiar characteristic is called zero-crossing which highlights that the memristor is a passive (no energy storing) element. Moreover, as shown in Fig. 4, by increasing the frequency of the input voltage from ω to 10ω, these curves tend to be infinitely

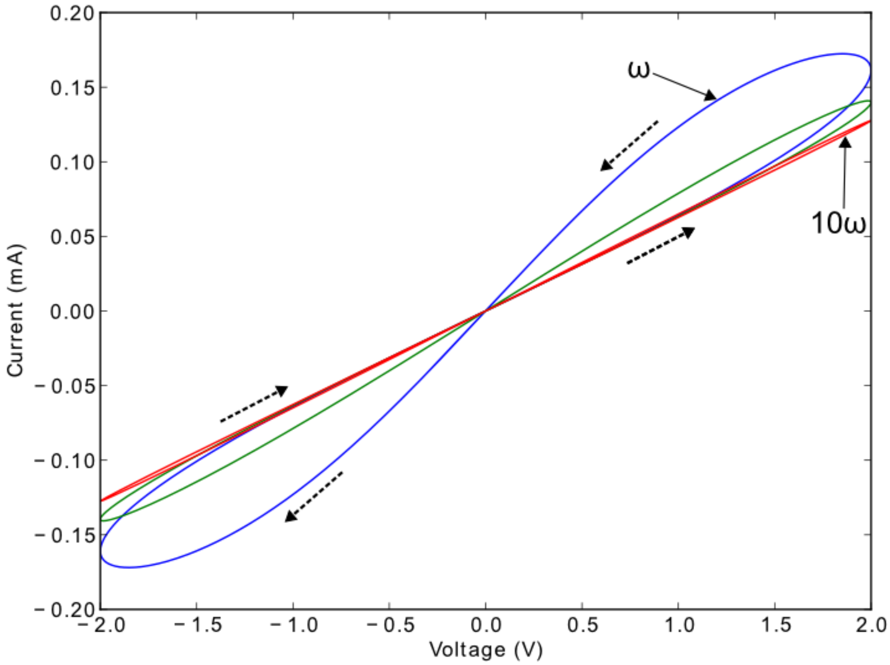

Figure 4: The pinched hysteresis loop of the memristor. When the frequency of the applied sinusoidal signal increases, the loop tends to be a straight line but the voltage and current always cross zero at the same time.

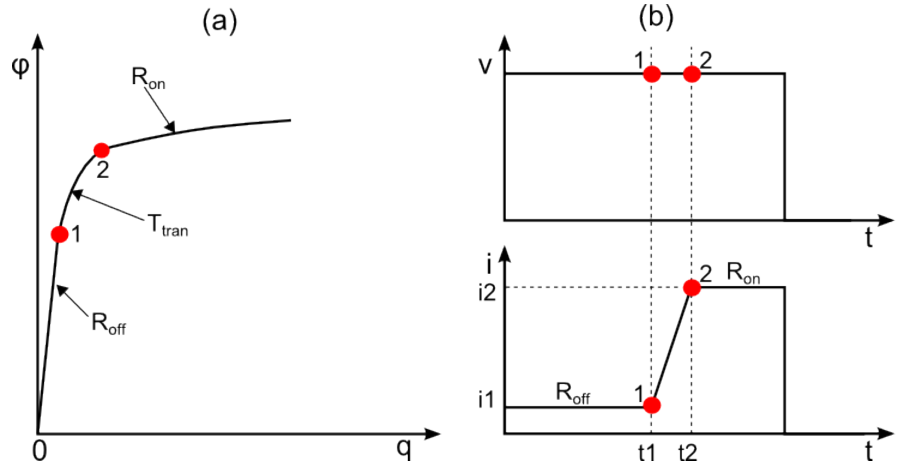

Figure 5: A demonstration of the delayed-switching effect. (a) A typical $q - \varphi$ curve of the memristor which can be divided into three parts according to its slope. (b) By applying a consecutive pulse, the time delay (as seen between $t1$ and $t2$) occurs when the memristor switches from R_{off} to R_{on}.

close to a straight line because, at a very high frequency, the charge q of the memristor has only changed a little during a very short time before it is swept away quickly. As a result, the memristance remains more or less unchanged.

2.5 Delayed-switching effect

It was found that switching in a memristor takes place with a time delay and this peculiar scenario is named the delayed-switching effect by Wang et al. [99].

From the definition of the memristance in Eq. (12), it is indicated that the memristance is actually the slope of the $q - \varphi$ curve. Hence, the $q - \varphi$ curve shown in Fig. 5(a) can be approximately divided into 3 parts: R_{off} (before point 1), T_{tran} (between points 1 and 2) and R_{on} (after point 2). Therefore, it gives a piecewise linear memristor which has a maximum memristance R_{off} and a minimum memristance R_{on}. It can be described in a general form

$$R_{mem} = \begin{cases} R_{off} & q < q1 \\ R_{on} & q > q2 \\ M(q) & q1 \leq q \leq q2 \end{cases} \tag{15}$$

where $q1$ and $q2$ are corresponding values of the charge at point 1 and 2. $M(q)$ varies according to the charge q during the transition time.

Between points 1 and 2 is the transition period where the memristance changes from R_{off} to R_{on} or from R_{on} to R_{off}. In Fig. 5(b), the transition period T_{tran} is observed by applying a consecutive voltage. The transition time between $t1$ and $t2$ indicates that the switching in the memristor between R_{on} and R_{off} takes place with a time delay because the memristor possesses certain inertia. However, in an ideal binary memristor, the transition period should not be shown in most real applications.

Further research revealed that the time delay actually depends on the voltage used which means a higher voltage produces a shorter delay because a higher voltage induces a higher current. For a charge-controlled memristor, since the memristance is a function of charge, a large current results in a sharp increment of memristance during the transition period, and the transition period shrinks in terms of time.

2.5.1 Impact of the delayed-switching effect

It has been considered that charge is the time integral of current, wherefore it could be represented by the area covered by waveforms of current shown in Fig. 6. It clearly shows that, in Fig 6(a), a consecutive pulse signal leads to the state change of the memristor and overcomes the time delay T_{tran}. The total charge $q(t)$ required to change the state from R_{off} to R_{on} is illustrated as the shaded area. Under the same circumstance, to achieve the state change, the signal shown in Fig.6(b) requires at least 4 pulses according to the shaded area. Hence, the total charge required to overcome the time delay can be defined mathematically by:

$$q(t) = \int_{-\infty}^{t} i(\tau) d\tau = \int_{-\infty}^{t} \frac{V}{R_{off}} d\tau = \frac{V}{R_{off}} T_{tran} \tag{16}$$

In order to calculate the time delay T_{delay} induced by the the pulse signal shown in Fig. 6(b), we define

$$T_{delay} = \frac{q(t)}{i(t)T_w} T = q(t)/(\frac{V}{R_{off}} D) = q(t)R_{off}/VD \tag{17}$$

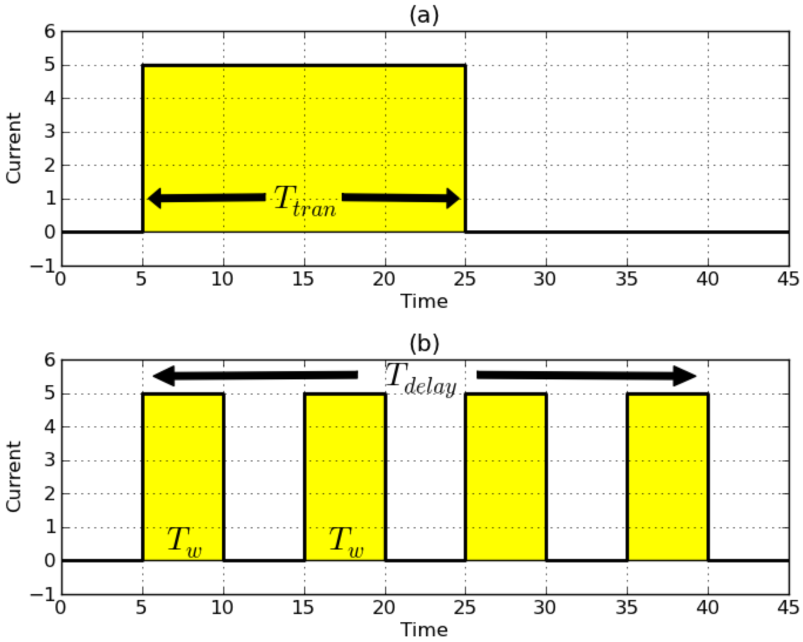

Figure 6: A demonstration of the delayed switching caused by different pulse signals. (a) The total time delay under a consecutive current pulse is measured by T_{tran}, and the total charge q required is given as the shaded area. (b) By applying a pulse signal with a period of $2T_w$, the time delay T_{delay} is obtained by calculating the number of pulses required to meet the total charge q given by (a).

where T_w is the duration of the active state of the pulse signal in a period T. D denotes the duty cycle which is defined by T_w/T. Consequently, if the duty cycle D is a constant, the Eq. (17) elaborates that the time delay T_{delay} is dependent on the voltage of the applied signal.

Obviously, the time delay will affect the state of the memristor in actual applications since if the input voltage is removed before the switching takes place, for example the applied time T_a is smaller than the time delay T_{delay}, the memristance remains unaltered. Therefore, in order to switch a memristor, T_a should be chosen in the way that $T_a > T$ according to Eq. (17). Otherwise, the time delay will lead to unexpected results in some applications, which should always be avoided.

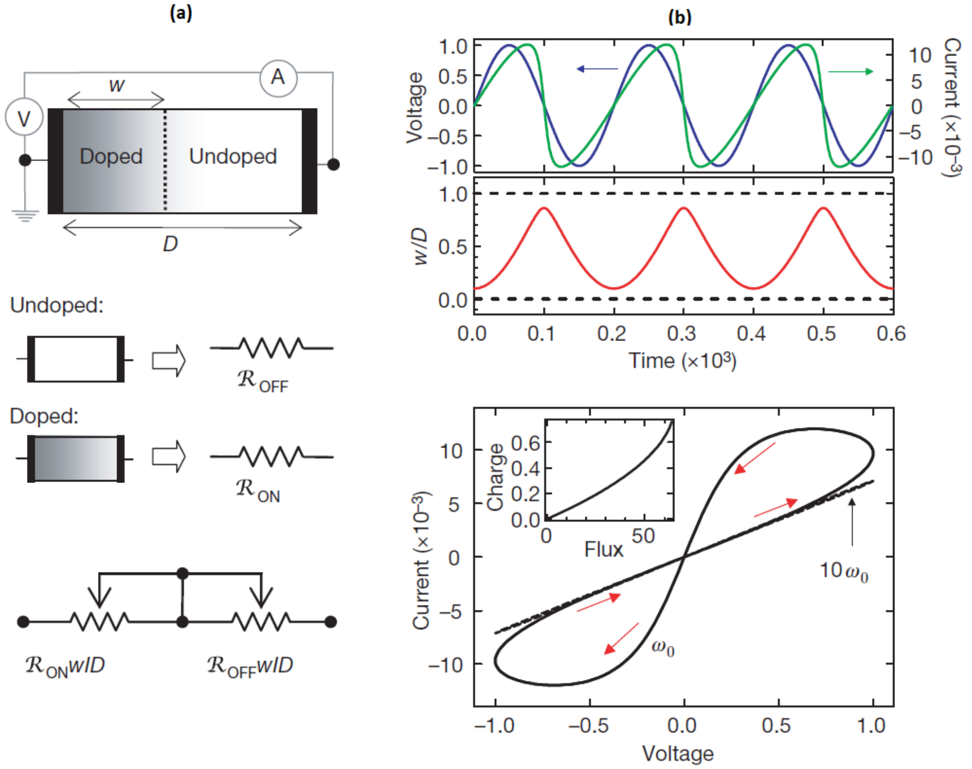

Figure 7: The titanium-dioxide based memristor proposed by HP (From [97]). (a) Once the semiconductor film is fully undoped or doped, the resistance of the memristor equals R_{off} or R_{on} respectively. (b) The pinched hysteresis loop is observed, and the hysteresis between applied voltage and current is shown at the top.

2.6 HP memristor model

The memristor found by one of the HP labs is based on titanium dioxide, which was an important discovery in the field because it revitalised the concept with the claim “The Missing Memristor Found” [97]. Before the claim, there were many elements which had similar behaviours to the memristor, for example as mentioned in [26, 84], however, none of them have been related to the memristor. Their research not only revives the studies of the memristor but also provides an applicable device for real applications.

The HP memristor is constructed from two Pt nodes and a thin semiconductor film sandwiched between them. Within the semiconductor film, it contains a layer of insulating TiO_2 and a layer of doped oxygen-poor TiO_{2-x} material. The effective transport mechanism in this device is through the drift of oxygen vacancies originating within the oxygen deficient TiO_{2-x} layer, which shifts the dividing line between the TiO_2 and TiO_{2-x} layers. The region with a high concentration of dopant has low resistance R_{on}, and conversely, the remainder with a low dopant concentration has much higher resistance R_{off} as shown in Fig. 7(a). By applying an external bias voltage, the boundary between the doped region and undoped region will move towards the undoped region, and hence the width w of the doped region will increase until it reaches the total width D of the semiconductor film and switches to low resistance R_{on}.

Based on observations and experiments, a simple model is proposed by Strukov et al. [97] to describe this device:

$$v(t) = \left(R_{on}\frac{w(t)}{D} + R_{off}\left(1 - \frac{w(t)}{D}\right)\right) i(t) \tag{18}$$

where $w(t)$ is the width of the doped region at time t. D is the full width or thickness of the semiconductor film and must be on nanometer scale. $i(t)$ and $v(t)$ imply the applied voltage and current which pass through the memristor.

Looking at the above equation (18), obviously it is very similar to the form of Ohm's Law but somewhat different. If the terms in the big parentheses are treated as one term, it can be interpreted as the resistance of memristor $M(q)$ gives

$$M(q) = R_{on}\frac{w(t)}{D} + R_{off}\left(1 - \frac{w(t)}{D}\right) \tag{19}$$

Normally $R_{off} \gg R_{on}$ is satisfied, and therefore the first term can be neglected in some cases.

Consequently, equation (18) can be rewritten as $v(t) = M(q)i(t)$ which applies to Ohm's Law. However, it is not sufficient to describe the behaviours of HP memristor since we do not know the term $w(t)$. Hence, another equation (20) was given by HP to define $w(t)$,

$$\frac{dw(t)}{dt} = \mu_v \frac{R_{on}}{D} i(t) \tag{20}$$

where μ_v is the average ion mobility. By recalling the concept of the memristive system, this model can be considered as a first-order memristive system since only

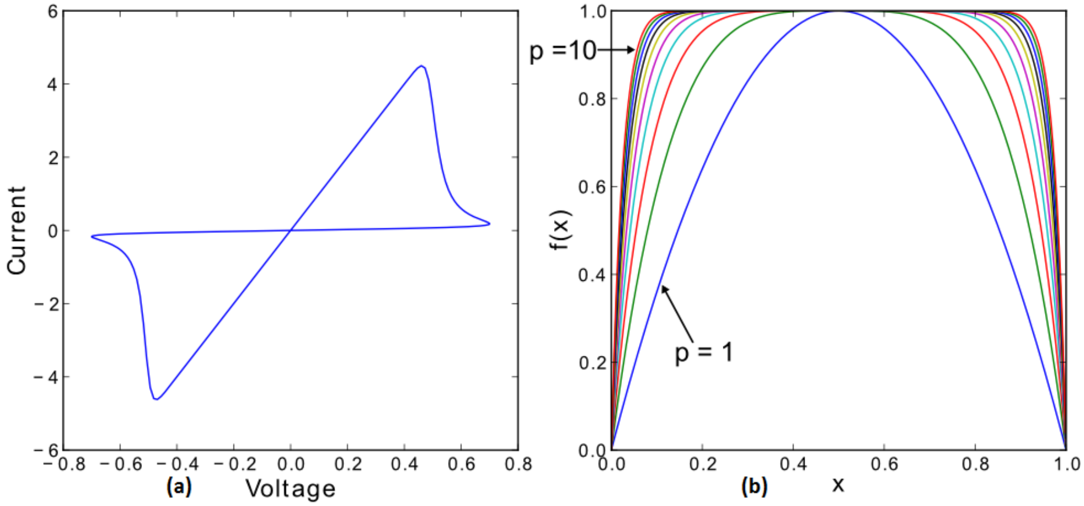

Figure 8: The boundary effect and the window function. (a) The $v - i$ curve with boundary effect reduces the change rate of w when it reaches either boundaries. (b) With an increasing value of p, the boundary effect disappears slowly since $f(x)$ is almost a constant for all the values of x.

one state variable w is defined. Moreover, as shown in Fig. 7, this model also exhibits the pinched hysteresis loop which demonstrates that it is a memristor.

2.6.1 Boundary effect and window function

The HP memristor model mentioned above is based on linear ionic drift, however, in nanoscale devices, a small voltage can produce significant non-linearities in ionic transport. In order to model the non-linear drift when w is approaching zero or D (boundaries of the device), a window function $F(w)$ is normally multiplied to the Eq. (20) which yields Eq. (21). Therefore, the ionic drift at the boundaries is reduced drastically as shown in Fig. 8(a) using the original window function proposed in [97].

$$\frac{dw(t)}{dt} = \mu_v \frac{R_{on}}{D} i(t) F(w(t)) \tag{21}$$

However, in order to model different and more sophisticated memristor dynamics, several models were proposed in the literature based on the HP memristor such as Joglekar's model [55], Biolek's model [10], the Boundary Condition Memristor (BCM) model [31] and the ThrEshold Adaptive Memristor (TEAM) model [62].

Explicit comparisons between these models have been investigated by Ascoli et al. [3] from theories to cases studies, and therefore we only briefly introduce selected models in this section.

Joglekar and Wolf [55] proposed a window function in the form of

$$f(x) = 1 - (2x - 1)^{2p} \tag{22}$$

where $x = w/D$ and $x \in (0, 1)$. The term p is a positive number and determines the moving speed of w between 0 and D. The advantage of this function is that, by selecting different values of p, it provides different non-linearities of the boundary effect as shown in Fig. 8(b). In the special case, when p is a very large number, as seen in Fig. 8 where $p = 10$, the non-linearity tends to disappear and turns into linear ionic drift. Moreover, it guarantees the moving speed of w reaches zero when it approaches the boundaries ($x = 0$ or $x = 1$). However, this function is independent of the current i.

Another model, namely Biolek's model, has been used in [9, 10] to build a SPICE model of the memristor with non-linear dopant drift. The current i is introduced into the this model as defined by

$$f(x, i) = 1 - (x - stp(-i))^{2p} \tag{23}$$

where $x = w/D$ and p is a positive integer which are similar to Jaglekar's model. $stp(-i)$ is defined by

$$stp(i) = \begin{cases} 1 & for \quad i \geq 0 \\ 0 & for \quad i < 0 \end{cases} \tag{24}$$

If the current i increases x, it is considered to be positive. Therefore, by reversing the sign of the current, a discontinuous transition will occur to ensure that the window function approaches 0 when the w/D approaches either boundary.

Corinto and Ascoli [31] offered a novel model which is based on boundary conditions. It provides the opportunity to suitably tune the memristor dynamics at boundaries to capture wide range of non-linear behaviour. By defining the boundary conditions C_n (where $n = 1, 2, 3$)

$$\begin{aligned} C_1 = \{&x \in (0,1) \text{ or } (x = 0 \text{ and } \eta v > v_{th0}) \\ &\qquad \text{or } (x = 1 \text{ and } \eta v < -v_{th1})\} \end{aligned} \tag{25}$$

$$C_2 = \{x = 0 \text{ and } \eta v \leq v_{th0}\} \tag{26}$$

$$C_3 = \{x = 1 \text{ and } \eta v \geq -v_{th1}\} \tag{27}$$

where $x = w/D$ and $x \in (0, 1)$. v_{th0} and v_{th1} are the threshold voltages which are non-negative and needed to determine the states or conditions. For the sake of simplicity, $\eta = 1$ in this case. Therefore, this model can be defined with conditions above as

$$f(x, \eta v) = \begin{cases} 1 & \text{if} \quad (25) \text{ holds} \\ 0 & \text{if} \quad (26) \text{ or } (27) \text{ holds} \end{cases} \tag{28}$$

By this means, Eq. (28) mathematically demonstrates a window function with vertical transitions which depend on the the threshold voltages. This model benefits from the explicit boundary thresholds which determine the evolution of the dynamics, and thus reflect the non-volatile feature of the memristor. Although BCM is a simple model compared to other models, it is capable of accurately capturing non-linear dynamics of various nano structures [3].

Different from above models, the TEAM model uses the undoped region as state variable x which is defined by

$$\frac{dx}{dt} = \begin{cases} k_{off} \left(\frac{i}{i_{off}} - 1\right)^{\alpha_{off}} f_{off}(x) & for\ i > i_{off} > 0 \\ 0 & for\ i_{off} > i > i_{on} \\ k_{on} \left(\frac{i}{i_{on}} - 1\right)^{\alpha_{on}} f_{on}(x) & for\ i < i_{on} < 0 \end{cases} \tag{29}$$

where k_{off} is a positive constant, while k_{on} is a negative constant. α_{off} and α_{on} are constants, while i_{off} and i_{on} are current thresholds. f_{off} and f_{on} are window functions that only allow the state variable varies within the range $[x_{on}, x_{off}]$.

Possible implementations of f_{off} and f_{on} were given as

$$\begin{cases} f_{off}(x) & = exp\left(-exp(\frac{x-a_{off}}{w_c})\right) \\ f_{on}(x) & = exp\left(-exp(-\frac{x-a_{on}}{w_c})\right) \end{cases} \tag{30}$$

where w_c is a fitting parameter. Reasonable values for bounds x_{off} and x_{on} are a_{off} and a_{on} respectively since a_{off} and a_{on} denote the points when both window functions equal zero. Thus, this model suggests that a memristor has a non-linear dependency on charge and state variable and exhibits current thresholds.

Another recently proposed window function [23], as defined in Eq. (31), uses a piecewise function to simulate the asymmetric ionic drift at the boundaries and provides the ability to mimic the symmetric boundary effect by selecting the appropriate parameter a in the range of $(0, 1)$.

$$f(w) = \begin{cases} 1-(w-a)^2/a^2 & 0 \leq w \leq a \\ 1-(w-a)^2/(1-a)^2 & a \leq w \leq 1 \end{cases} \tag{31}$$

It offers a flexible way to simulate the advanced non-linearity because the semiconducting film can be divided into several regions and each region can have a different moving speed of w according to the experiments.

2.7 Piecewise-linear memristor model

The piecewise-linear memristor model is widely used in the studies of neuromorphic applications and memory designs. This model often has two states or multiple states, and hence it can be divided in to two categories: binary memristor model and multi-state piecewise memristor model. In both of these models, one threshold or multiple thresholds exists to control the states.

2.7.1 Binary memristor model

A binary memristor model which only exhibits two states 0 and 1 represented by R_{off} and R_{on} respectively is derived from the digital memristor which is used in digital circuits. It is generally modelled by taking the similar form of Eq. (15)

$$R_{mem} = \begin{cases} state \quad 0 \quad (R_{off}) & v < V^{T-} \\ state \quad 1 \quad (R_{on}) & v > V^{T+} \\ unchanged & V^{T-} \leq v \leq V^{T+} \end{cases} \tag{32}$$

but the memristance is unchanged if the applied voltage is within the range between the thresholds. Therefore, the states between state 0 and state 1 are ignored. However, Eq. (15) shows a more general case that states between R_{off} and R_{on} can be observed. Obviously, this model is controlled by the voltage and the negative and positive thresholds voltages V^{T-} and V^{T+}.

Another binary memristor model is defined by Hu and Wang [47]

$$M(u(t)) = \begin{cases} M' & \dot{u}(t) > 0 \\ M'' & \dot{u}(t) < 0 \\ unchanged & \dot{u}(t) = 0 \end{cases} \tag{33}$$

where u is the applied voltage of the memristor and $\dot{u}(t)$ is the the derivative of u with respect to time t. By replacing M' with 0 and M'' with 1, it models a digital memristor which is controlled by the applied voltage u and specifically its derivative $\dot{u}(t)$. By increasing or decreasing the applied voltage $u(t)$, the memristor switches between the two states 0 and 1 without the negative and positive threshold voltages defined in Eq. (32). It provides another way to control the switching of the memristor using the voltage and is used in analysing some features of the memristor-based recurrent neural networks with time delays, such as the global uniform asymptotic stability and passivity analysis.

Form Eq. (32) and (33) above, it is clear that a delayed-switching effect exists with either case when the defined switching conditions are not satisfied. Since the binary models only have two states, the transition period is concealed and memristance remains unchanged to avoid violating the stability in digital circuits.

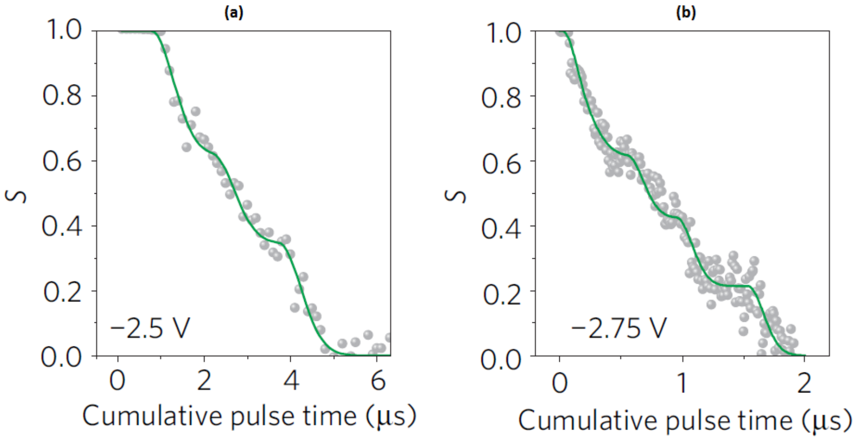

Figure 9: The switching dynamics of a ferroelectric memristor (From [19]). By applying consecutive pulses with different amplitudes, the switching dynamics of (a) and (b) are different, and a higher voltage induces much wavier variations of the state variable s.

2.7.2 Multi-state memristor model

In contrast to the binary memristor, a multi-state memristor has multi-level resistance states rather than just 0 and 1. The multi-state memristor was mentioned by Chua [25] when he postulated the memristor. Originally, it was proposed by dividing the $\varphi - q$ curve into several parts to achieve the piecewise linearity, and hence memristance has multiple values. However, with the development of the memristor, several multi-state memristors are found with intrinsic multi-level resistance rather than with the help of peripheral circuits. The multi-level memristive mechanism and characterisations of the FeO_x memristor are investigated by [18], which demonstrates that the state can be tuned and controlled by external electric conditions. Such behaviour can also be seen with Cu_xO and ZrO_2 based devices which provide multi-level resistance switching, such as five-level states [102], and can be used as the multi-level cell (MLC) in non-volatile memory. Similar behaviour also has been found in the ferroelectric memristor which is based on ferroelectric tunnel junctions (FTJs), although the FTJs so far have only been considered for binary storage [19]. FTJs are tunnel junctions compose of two metal electrodes which are separated by a thin ferroelectric layer. The experiment shows that, by applying a consecutive pulse, the resistance varies from OFF to

ON in a wavy way as shown in Fig. 9. It is argued that ferroelectric domain dynamics dominate the resistance variations in the ferroelectric memristor and thus the wavy variations signal the presence of several areas with different switching dynamics. In this case, the peculiar behaviour is caused by the nucleation effects in ferroelectric barriers because the nucleation centres need to be activated and therefore yielding delays. Though it aims to provide a virtually continuous range of resistance levels, it could also provide multi-level resistance by applying an appropriate duration of voltage. Furthermore, real applications are investigated based on the multi-state memristor by Kim et al. [60] who built a hybrid system for data storage and neuromorphic applications, which is integrated on the top of a CMOS chip. It demonstrates the promising performance of the multi-state memristors which can store images in high-density hybrid systems which could provide the ability to store data up to 10 different levels. However, modelling a multi-state memristor is very challenging because specific materials have different switching mechanisms. A general representation of the multi-state memristor can be defined using a piecewise function

$$R_{mem}(q) = \begin{cases} R_0 & q \leq q_0 \\ R_1 & q_0 < q \leq q_1 \\ R_2 & q_1 < q \leq q_2 \\ \vdots & \qquad \vdots \\ R_n & q_{n-1} < q \leq q_n \end{cases} \tag{34}$$

where the resistance level is determined by the charge q, and the constitutive relation of φ and q is implicit because it depends on the materials.

A SPICE model of the memristor with multi-level resistance states has been proposed by Fang Xu-Dong [36] recently, which is an ideal piecewise linear memristor model and follows the similar approach of Eq. (34). In the following Chapter 3, a more sophisticated multi-state memristor model, namely staircase memristor, is proposed to mimic the multi-level resistance based on the delayed-switching effect, and further studies of the multi-state memristor will be demonstrated.

2.8 Cubic polynomial memristor model

The pinched hysteresis loop is the fingerprint of the memristor. In order to analyse such behaviour and the responses to the sine-wave-like input voltage source or current source, the memristor with a cubic polynomial constitutive relationship is presented as [26]

$$\varphi = \frac{1}{3}q^3 + q \tag{35}$$

Obviously, this model describes the constitutive relation of $\varphi - q$ curve by a monotone-increasing function. Since the memristance is the slope of the $\varphi - q$ curve, the derivative of Eq. (35) defines the memristance R_{mem}

$$R_{mem}(q) = d\varphi(q)/dq = q^2 + 1 \tag{36}$$

In this case, the R_{mem} is always a positive term which has a minimum value 1 if $q = 0$.

Its various characterisations can be investigated by simply applying a sinusoidal current source $i(t)$ and $t \geq 0$

$$i(t) = Asin(\omega t) \tag{37}$$

In order to obtain the $v - i$ curve, it requires the calculation of charge q from the integration of current. Thus assuming the initial charge $q_0 = q(0) = 0$ we obtain

$$q(t) = \int_0^t Asin(\omega\tau)d\tau = \frac{A}{\omega}(1 - cos(\omega t)) \tag{38}$$

Therefore, by substituting Eq. (38) into (35), the corresponding flux of the current source is

$$\varphi(t) = \frac{A}{\omega}(1 - cos(\omega t))\left(1 + \frac{1}{3}(\frac{A^2}{\omega^2}(1 - cos(\omega t)^2))\right) \tag{39}$$

By differentiating the obtained flux φ, the voltage v across the memristor is

$$v(t) = A\left(1 + \frac{A^2}{\omega^2}(1 - cos(\omega t))^2\right)sin(\omega t) \tag{40}$$

So far, all the required variables are acquired through the Eq. (35) and (37), subsequently, the $v - i$ curve, $\varphi - q$ curve and the variation of memristance R_{mem}

can be illustrated for investigating the responses to the sinusoidal signals with different A and ω. This model also exhibits the pinched hysteresis loop, which proves that it is a memristor. The advantage of this model is, in contrast to the HP model, that it has an explicit constitutive relation of φ and q, and hence it is very convenient to theoretically analyse the basic properties of the memristor with different sine-wave like signals by some simple calculations. However, this model lacks the ability to mimic some physical details such as the boundary effect because the memristance $R_{mem} \to \infty$ if $q \to \infty$. By introducing a window function $F(x)$ which limits the charge q, the memristance can be constrained within the range of $[R_{on}, R_{off}]$, which will make it more applicable in simulations of memristor-based applications.

2.9 Voltage-controlled memristor models with threshold effect

Except the piecewise linear memristor model, the HP memristor model and the cubic polynomial memristor model solely depend on the charge q, which does not have explicit thresholds. In this case, even a very small current will eventually switch a memristor, however, in the Si based memristive device, there exists a threshold effect. Consequently, if the bias voltage is less than the threshold V_T, the memristance will not be changed [53]. In titanium-dioxide based memristors, a dynamical threshold effect also has been discovered, which requires a specific voltage to switch the memristor. The threshold effect will make the memristor more controllable since it is possible to perform read and write operations on the memristor by applying a bias voltage below or above V_T respectively. It will be beneficial to some practical applications because a read operation will not affect the state of the memristor, and therefore it is more reliable.

The voltage-controlled memristor normally follows the definition of the voltage-controlled memristive system shown in Eq. (8) which gives

$$\begin{cases} i = g(w, v)v \\ \frac{dw}{dt} = f(w, v) \end{cases} \tag{41}$$

where w is a variety of physical state variables; i is the current across the memristor; v is the voltage drop across the device. The function g denotes the memductance, and hence, in this case the memristance is not only controlled by the state variable but also the voltage applied to the memristor. The function f may describe the ionic drift under the electric fields in the case of the titanium-dioxide based memristor. In addition to the models introduced in Section 2.6.1, such as BCM, Linares-Barranco et al. [67] assumed a possible non-linear f which depends on the voltage

$$f(v) = \begin{cases} I_0 sign(v) \left(e^{|v|/v_0} - e^{v_{th}/v_0} \right) & \text{if} \quad |v| > v_{th} \\ 0 & otherwise \end{cases} \tag{42}$$

where I_0 and v_0 are parameters that may not depend on w; v_{th} denotes the threshold voltage.

Another similar function f has been proposed by Pershin and Di Ventra [78] using the following equation

$$\dot{x} = (\beta V_M + 0.5(\alpha - \beta)(|V_M + V_T| - |V_M - V_T|))\, \theta(x - R1)\theta(R2 - x) \tag{43}$$

where $\theta(\cdot)$ is a step function, and V_M is the voltage drop across the memristor. The parameters α and β characterise the change rate of memristance when $|V_M| \leq V_T$ or $|V_M| \geq V_T$ respectively. $R1$ and $R2$ are limiting values in the step functions to guarantee that the memristance varies only between $R1$ and $R2$, otherwise the change of x will be zero.

It is worth noting that although the voltage-controlled memristor model with threshold aims to mimic the memristive devices with threshold effect, it is also applicable to other memristors without threshold effect. In applications, the threshold could be a reference point which is often used in non-linear electronics. In memristive devices, the change rate of conductance g normally follows a representative sinh-like curve, which means only larger voltages can induce much greater changes in g [94]. Therefore, a reference point where the change rate starts to increase sharply can be selected to represent the threshold.

Model	Linear	Non-linear	Discrete	Threshold
HP model	+	+/−	−	−
Binary model	−	−	+	+
Multi-state model	−	−	+	+
Cubic polynomial model	−	+	−	−
Voltage-controlled model with threshold	+	+	−	+

Table 2: A general comparison of the mentioned memristor models.

2.10 Applications

Since the memristor is still at the development stage in the research labs, it is difficult to obtain a uniform physical memristor for test or real applications. Thus, the mentioned models are widely utilised in simulating memristor-based applications according to different properties. Table 2 shows a simple comparison of the basic properties of the memristor models, in which, a "+" sign denotes the model has the property, otherwise a "-" sign is utilised. It clearly shows that different models focus on different properties, for example the HP memristor model focuses on linear or non-linear dopant drift properties, the binary and multi-state memristor models focus on the discrete property and the voltage-controlled memristor model with threshold focuses on the threshold effect. Hence, they lend themselves to different applications.

Digital memory

Since current computers work on binary codes, a straightforward application of a binary memristor is the digital memory, such as resistive random access memory (RRAM) and content addressable memory (CAM). Since CMOS technology is reaching the bottleneck in terms of size, the emerging RRAM is expected as a potential replacement in the future. According to the international technology roadmap for semiconductors [51], the advantages of the RRAM are the significant short operation time and the ability to realise the three-dimensional

	Traditional Technologies			Emerging Technologies		
	DRAM	SRAM	Flash NAND	FeRAM	PCRAM	Memristive device
Knowledge level	mature	mature	mature	product	advanced	early stage
Half pitch (nm)	50	65	90	180	65	≤ 50
Read time (ns)	< 1	< 0.3	< 50	< 45	< 60	< 50
Write/Erase time(ns)	< 0.5	< 0.3	10^6	10	60	< 50
Write voltage (V)	2.5	1	15	$0.9 - 3.3$	3	< 3
Read voltage (V)	1.8	1	2	$0.9 - 3.3$	3	< 3
Write endurance	10^{16}	10^{16}	10^5	10^{14}	10^9	10^{10}
Write energy (fj/bit)	5	0.7	10	30	6×10^3	$< 10^3$
Retention $(years)$	*seconds*	N/A	> 10	> 10	> 10	> 10
Density $(Gbit/cm^2)$	6.67	0.17	2.47	0.14	1.48	100
Highly scalable	major technology barriers			poor	promising	promising

Table 3: Comparison of traditional and emerging memory technologies. Reproduced and modified from [35].

CMOL (standing for CMOS + MOLecular scale devices hybrids) structure. A typical comparison of the memory technologies is presented in Table 3 according to [35, 64, 105, 106]. Although currently the development of memristive devices is at a very early stage, it has highest density among these typical memory technologies. In terms of speed and endurance, its performance is worse than traditional technologies such as DRAM and static RAM (SRAM), however, it is superior to Flash memory. Since memristor is a kind of non-volatile memory, its retention ability definitely outperforms DRAM and SRAM. Even compared to emerging technologies such as ferroelectric memory (FeRAM) and phase-change memory (PCRAM), it also has smaller half-pitch and promising scalability and similar performances in terms of speed, endurance and retention. Considering that memristive devices are still at the very early stage of development compared to other technologies, in the future, performances can be further improved along with its development. Based on CMOL structures, the RRAM can achieve ultra-high density by stacking the memristive crossbar array at the top of a CMOS layer [96]. In this structure, each memristive device is a binary cell, meanwhile, the CMOS layer plays the role of coding, decoding and line driving. Hence, both the memristive device layer and the CMOS can keep a relatively simple structure and the much higher density can be achieved by RRAM compared to traditional technologies.

The binary memristor model is also beneficial to the CAM design to increase the packing density and provide non-volatility. Different from conventional memory, CAM focuses on very-high-speed searching applications by searching the content rather than the address, and therefore a basic CAM cell has two basic functions which are bit storage and bit comparison. Eshraghian et al. [35] investigated several variations of hybrid CAM cells using different architectures where two memristors are utilised as bit-storage in each cell, and the bit comparison is performed by the CMOS logic circuits. Based on their simulation results, a reduction of approximately 96% in average power consumption has been found since the root-mean-square (RMS) value of current is 47 μA less than the conventional SRAM based CAM cell. At the same time, for a single cell, a 46% area reduction is noted by the memristor-based CAM cell. In another CAM cell design [22], the bit comparison is also implemented by the memristor which compares and stores the comparison results, and therefore a further area reduction could be expected.

On the other hand, because of the nature of the multi-state memristor, it is possible to store more information in one memristor, which has been investigated by [60].

Neuromorphic engineering

As neuromorphic engineering aims to implement the biological architectures in the nervous system by utilising the very-large-scale integration (VLSI) system with analog elements, the memristor models with non-linear features are promising candidates to implement the memristor-based neuromorphic systems. In the following sections, the related applications are summarised in two categories.

Hodgkin-Huxley model

The Hodgkin-Huxley model derives from the experiment on the giant axon of the squid, which is a mathematical model to describe the ionic mechanisms involved in excitation and inhibition of the nerve cell [46]. In this model, the total current through the membrane is defined by summing the sodium ion current (I_{Na}), the potassium ion current (I_K) and the leaking current (I_l). The voltage-gated sodium and potassium ion channels are represented by two time varying conductance g_{Na} and g_K respectively. Recently, these two time varying elements have been replaced by two memristors, which demonstrates that g_{Na} is in fact a second-order memristor and g_K is a first-order memristor [27]. By this means, the Hodgkin-Huxley model can be implemented with memristors, since in basic circuit theory g_{Na} and g_K cannot be prescribed as a function of time.

Synapse

In addition to the applications introduced in Section 2.3, the analog memristor is also applied to other neuromorphic systems. The memristor-based adaptive, recurrent neural network is explored by Snider [93] for complex pattern recognition problems using the non-linear feature and threshold of the device. The communication between neurons is based on pulse signals by a positive pulse V and a negative pulse $-V$. Because of the threshold effect, voltage V can only cause a very small change on the conductance of the memristor, however, if the positive pulse V and the negative pulse $-V$ are overlapped, a $2V$ voltage will change the conductance by a much great amount. Hence, the memristor is controlled by the voltage drop across it, and the conductance change is proportional to the width of the overlap. Similar mechanisms have been used in [21] to implement unsupervised image learning, which used the model similar to Eq. (43).

Besides the threshold based model, it is possible to implement the instar and outstar learning with binary memristor models [91]. In this case, the analog signal in the range $[0, 1]$ is converted by the "thermometer code" to a digital format using an "M of N" bit encoding. Then the encoded value is weighted by the binary memristors and summed as the output to the postsynaptic neuron. According to the instar learning

$$\frac{dw}{dt} = \eta y(x - w) \tag{44}$$

or the outstar learning

$$\frac{dw}{dt} = \eta x(y - w) \tag{45}$$

the state of the binary memristor will be changed. x is the input signal; y is the state variable of the postsynaptic neuron, and w is the binary synaptic weight.

Chapter 3

Memristive Non-spiking Neural Networks

As introduced in Chapters 1 and 2, applications of memristors are widely studied in spiking neural networks, especially using a pair of forward and backward spikes. The learning process is implemented by applying a proper pulse voltage and pulse width according to the weight update rule, and therefore the conductance of the memristor can be adjusted by the rule itself rather than storing an explicit value in the memristor. However, as mentioned previously, non-spiking neural networks use real-valued numbers as the input and output. In particular, they require that the conductance of the memristor is adjusted to a specific value. An example is the cellular neural network (CNN) which normally uses the pre-defined, real-valued numbers as the connection weights between neurons. Because of the non-linearity of the analog memristor, setting its conductance to a specific value is more difficult than with the binary memristor. In this chapter, a staircase memristor model which has multi-level resistance is proposed based on the delayed-switching effect and the multi-state memristor model. By applying the proposed model to the CNN, a memristive CNN can be achieved with more convenient setup processes, through which it is possible to implement a large neighbourhood. Based on the structure of the memristive CNN, the application of memristors in an echo state network (ESN) is investigated.

3.1 Memristors as programmable weights

In non-spiking neural networks, synaptic weights primarily exist in three forms which are

1. constant weights: the connection strength between neurons are pre-defined during the network setup and are invariable
2. random weights: the synaptic weights are randomly generated during the network setup and are invariable
3. plastic weights: the synaptic weights are generated in some form and varied during the training phase

Theoretically, analog memristors can be utilised as these kinds of synaptic weights, however, because of their non-linearity, precisely programming a memristor to a specific resistance is difficult, and especially as the synaptic weight has to be changed quickly and frequently. In [26], from the current-voltage characteristics of several memristor examples, it is shown that the variation of memristance is not as smooth as with the theoretical memristor model but in fact quite fluctuated. This implies, in realistic applications, that it is difficult to achieve a specific and precise value. A general solution to programming a memristor to a specific value is the use of a lookup table which contains the corresponding voltage and duration required. Thus, a memristor has poor performance as a constant weight or plastic weight in terms of speed and efficiency, which leads to research in memristors with multi-level resistance.

3.2 Staircase memristor model

3.2.1 The concept of staircase memristor

The concept of the staircase memristor derives from the "delayed-switching effect" of piecewise linear memristors [99]. As mentioned previously, this effect indicates that switching in a memristor takes place with a time delay because the memristor possesses certain inertia [100]. A staircase memristor is considered to have a delayed-switching effect between several discrete resistance levels, and hence the variation of memristance is like a staircase. In practice, memristors with multi-level resistance are observed in [18, 19, 102] and a theoretical SPICE model was proposed in [36]. In particular, in ferroelectric memristors, significant delays of resistive states are observed. In some applications, for example in neuromorphic engineering, these delays should be avoided because they resist the change of the conductance of memristors which will consume more time and power on training. However, it is beneficial to some applications, such as programming a memristor to a specific value, and therefore an alternative way to use delays rather than to avoid them is presented.

3.2.2 Modelling a staircase memristor

A staircase memristor is obtained by dividing the $q - \varphi$ curve into several linear segments, which implements a piecewise linear memristor introduced in Chapter 2. Since the slope of the $q - \varphi$ curve denotes the memristance of the memristor, the same number of stairs on the memristance can be observed as shown in Fig. 12(a).

Non-linear drift model

To model the staircase memristor, we consider a charge-controlled piecewise memristor first. Based on the experimental results of the existing HP memristor model with non-linear drift [97], by applying the window function $w(1-w)/D^2$ to the right side of Eq. (21), it will result in the following equation

$$\frac{dw}{w(1-w)} = \mu_v \frac{R_{on}}{D} i \frac{1}{D^2} dt \tag{46}$$

Then, both sides could be integrated which yields

$$-\ln\left|\frac{-w+1}{w}\right| + C = \mu_v \frac{R_{on}}{D^3} q \tag{47}$$

Since D is normalised to 1 and $0 < w < 1$, the absolute symbol can be removed. If we assume that the initial value of charge q is $q_0 = q(0) = 0$, we have

$$\frac{-w+1}{w} = e^{-q\mu_v \frac{R_{on}}{D^3} + C} \tag{48}$$

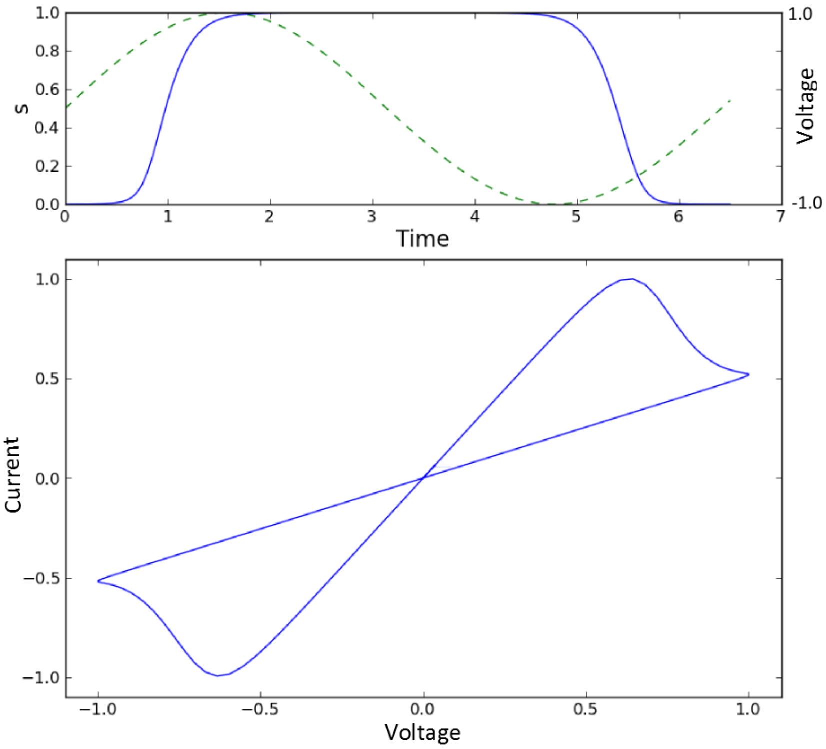

Figure 10: A non-linear drift model which mimics the boundary effect of the HP memristor model. The dash line denotes the voltage signal. A pinched hysteresis loop is observed by applying a sinusoidal voltage signal.

Finally, by simplifying Eq. (48), we can have the solution and a state variable s which has the following relation with charge q

$$s = w = \frac{1}{1 + e^{-qk_p + \rho}} \tag{49}$$

where $k_p = \mu_v \frac{R_{on}}{D^3}$ and ρ is the constant C. In this case, the state variable s models the boundary effect of the memristor and will be used in the proposed staircase memristor model.

The derivative of (49) gives the change rate of state variable s

$$\frac{\mathrm{d}s}{\mathrm{d}q} = \frac{k_p e^{-qk_p+\rho}}{(1 + e^{-qk_p+\rho})^2} \tag{50}$$

where k_p denotes the propagation speed of charge in memristor. ρ/k_p is a constant term which determines the middle point of the transition period of the memristor

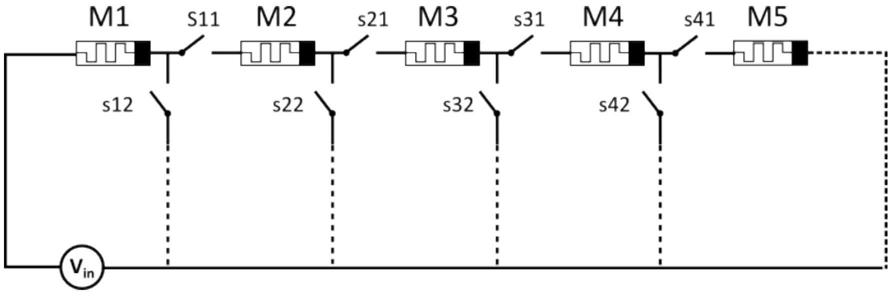

Figure 11: A schematic digram of a conceptual staircase memristor.

between ON and OFF. By substituting Eq. (49) into

$$R(q) = sR_{on} + (1 - s)R_{off} \tag{51}$$

and

$$v(t) = R(q)i(t) \tag{52}$$

we model the similar behaviour of an HP memristor model with non-linear drift which is controllable by varying the parameters k_p and ρ. If k_p is a large number, it requires less charges to change the state of the memristor. By varying ρ, a virtual threshold (where the state begins to change much more) is controlled, and therefore a larger ρ results in that more charges are required to switch the memristor. In Fig. 10, the new model exhibits similar behaviour as shown in Fig. 8(a). In this case, a large amount of charge q is required to switch the memristor when the state variable approaches the boundaries, and therefore the memristance at ON and OFF states can be held longer and yield a time delay which is essential for building a staircase memristor.

The staircase model

In the case of the staircase memristor model, we assume that it consists of several non-linear drift models which are somehow connected in series with switches as illustrated in the conceptual schematic diagram of Fig. 11. It demonstrates a staircase memristor model with 6 resistance levels, and each non-linear drift model is activated one by one. At the first, switch $s12$ is turned on and $s11$ is turned off, and hence only the memristor M1 is connected to the circuit. When M1 is switched from ON to OFF, switch $s12$ is turned off. Then the switches $s11$ and

$s22$ are turned on, which only connect M2 to the circuit besides M1. Until all the memristors are connected and switched to the OFF state, a staircase memristor model with 6 resistance levels is achieved. In a word, the conceptual circuit works somewhat like a digital potentiometer which is built by memristors.

Based on the conceptual circuit, each memristor in the circuit is considered as a region which delays the variation of the resistance level. The delays are caused by the activation processes of the regions, which make the staircase memristor stay at one resistance level until a virtual threshold of the corresponding region is reached. The proposed staircase memristor model has a similar mechanism to that has been found in the ferroelectric memristor [19] which contains several dynamic regions. Because of the nucleation effect in the ferroelectric memristor, such dynamic regions need to be activated, which explains the delay observed in the evolution of its state. Interestingly, these regions are dynamical in ferroelectric memristors and depend on the applied voltage, meanwhile the switching dynamics in these regions are different. However, in a staircase memristor model, for reasons of stability and controllability, each region is considered to be constant rather than varying dynamically.

If we assume a staircase memristor model has N regions that have the same characteristics, for example the same k_p, ρ and the fraction w_i which is normalised by the total width D, the conceptual circuit can be summarised by the following equation:

$$\begin{aligned} s &= \sum_{i=0}^{N-1} w_p + s(i) \\ &= \sum_{i=0}^{N-1} w_p + \frac{w_i}{1 + e^{-qk_p + \rho + Q^i_{min}}} \end{aligned} \tag{53}$$

where w_p is defined by

$$w_p = w_0 + w_1 + w_2 + \cdots + w_{i-1} \tag{54}$$

which is the total width of the previous regions. s denotes the state variable of the staircase memristor and varies between “0” and “1”. Theoretically, more than one region is prohibited from activating at the same time as the implementation of the conceptual circuit.

Therefore, a Heaviside function $H(i)$ is multiplied to (53) which gives

$$s = \sum_{i=0}^{N-1} H(i)(w_p + \frac{w_i}{1 + e^{-qk_p^i + \rho^i + Q_{min}^i}}) \tag{55}$$

where $H(i)$ is defined by

$$H(i) = H(| - k_p^i q| - Q_{min}^i) \tag{56}$$

Q_{min}^i denotes the minimum quantity of charge required to enter current region i with values $i = 0, 1, 2, 3, \cdots$. If the total charge passed the staircase memristor exceeds Q_{min}^i, current region i is activated. If all the regions have the same width w_i, w_i then equals S_{max}/N, and therefore w_p equals $i \cdot w_i$. However, in the case that the regions have a different proportion of total thickness, the term w_i will be replaced by a varied number according to the proportion of the region i as well as w_p.

Then the memristance $R(q)$ of the staircase memristor is determined by recalling Eq. (51)

$$R(q) = sR_{on} + (1 - s)R_{off} \tag{57}$$

where R_{on} and R_{off} are the memristance of ON and OFF states respectively. Hence, all the equations required to model the proposed staircase memristor model are presented. Since the $\varphi - q$ relation and required thresholds have not been investigated in current research of memristors with staircase behaviour, experiment results of physical staircase memristors with such parameters are not available yet. Therefore, for illustrative purpose, we select possible parameters according to the down-to-up domain experiment in the ferroelectric memristor [19]. It assumes a staircase memristor with 4 regions and 5 resistance levels, which has been demonstrated in [19]. Simulation results of the proposed staircase model is illustrated in Fig. 12, where the fingerprint of the memristor, the pinched hysteresis loop of $V - I$ curve, has been observed by applying a periodic sinusoidal signal $5sin(\pi t)$. By varying the frequency of the applied signal, similar behaviour to Fig. 4 will be observed. Therefore, by increasing the frequency, the pinched loop tends to be a straight line and always crosses the zero point. The rise and decline of the state variable s along the sinusoidal signal is symmetric in Fig. 12(c), however, it could be asymmetric according to the observed behaviour of practical experiments by defining a separate equation for the decline of s. Because the rise and decline of s is symmetric, the rise curve and decline curve are overlapped in Fig. 12(d). Since the proposed staircase model is based on the HP memristor model which has been widely discussed and studied, it could be applied to the existing applications conveniently. Moreover, this model provides non-linearity to the variation of state variable s rather than changing s abruptly and a predictable resistance switching time for a given input.

3.3 Memristive cellular neural network

CNN is a biologically motivated neural network and important for applications in practice. The mainly uniform processing elements, called cells or artificial neurons, are placed on a regular geometric grid (which could be a square, hexagonal, or other patterns). The structure of CNN is defined "Any cell in a cellular neural network is connected only to its neighbour cells" [29]. It consists of an array of non-linear, locally coupled neurons. By limiting the connections of the neurons, the complexity of cellular neural networks is reduced significantly. However, it still has satisfactory capability to process large amounts of information in real time. Although all the cells are only locally connected, the cells which are not directly connected together can affect each other indirectly by the propagation effect of cellular neural networks.

3.3.1 CNN cell

Theoretically, a CNN can be defined by any dimension depending on the applications. If we define a cellular network with size $M \times N$ such that all the cells are arranged in m rows and n columns, the cell on the i_{th} row and j_{th} column is called $C(i, j)$. Therefore, the neighbour of $C(i, j)$ is defined by $C(k, l)$. According to the generic definition of cellular neural networks, the state of the cell $C(i, j)$ is

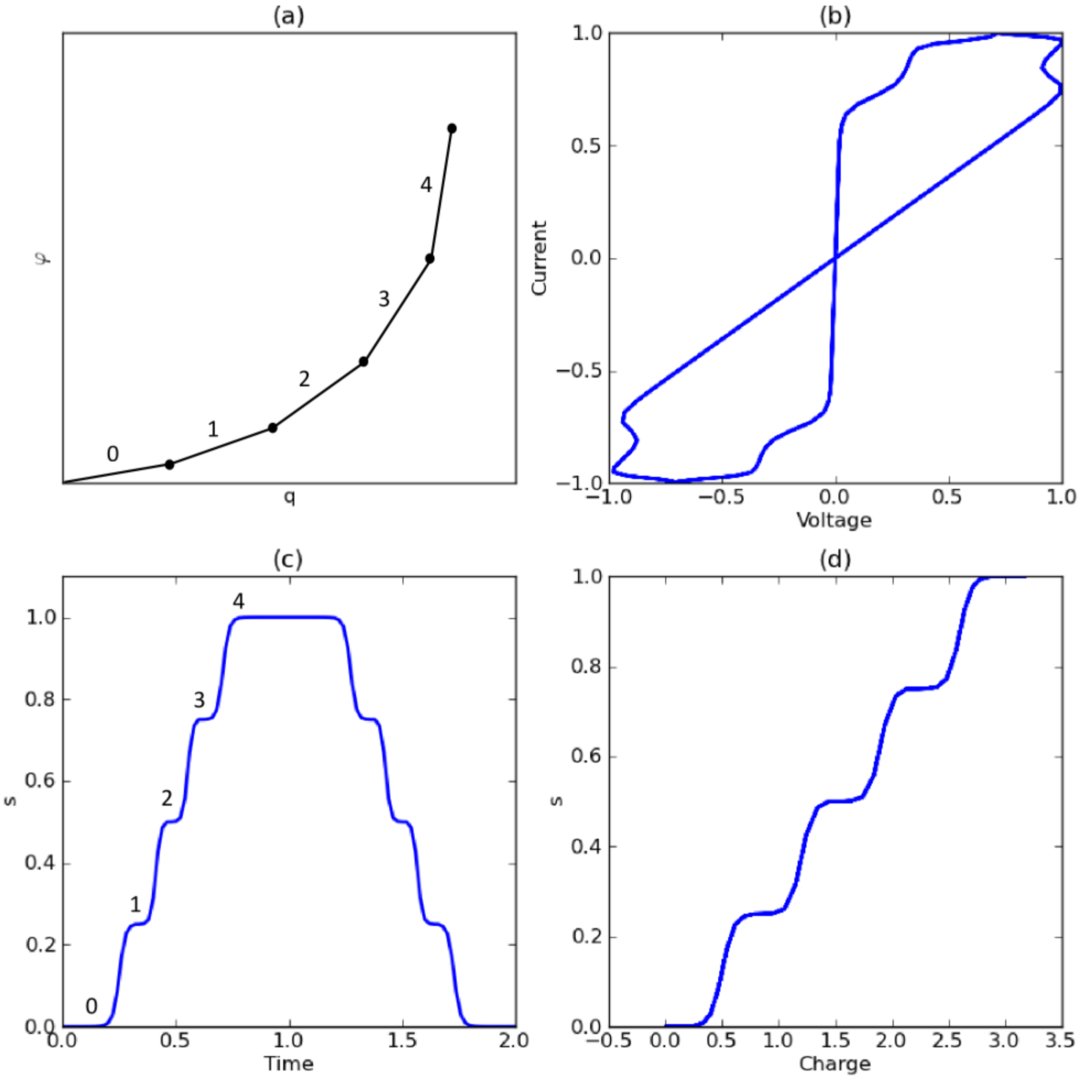

Figure 12: Simulation results of the staircase memristor model. (a) A $q-\varphi$ curve is divided into 5 segments which represent 5 resistance levels. It means this staircase memristor has 4 regions which are region 0:$\{0 \to 1\}$, region 1:$\{1 \to 2\}$, region 2:$\{2 \to 3\}$, region 3:$\{3 \to 4\}$. (b) A pinched hysteresis loop of current and voltage of staircase memristor. (c) By applying a periodic sinusoidal signal $5 \cdot sin(\pi t)$, the state variable s of staircase memristor varies like a staircase waveform. (d) The variation of s along the charge q. Parameters used here are: $N = 4, Q^0_{min} = 0C, Q^1_{min} = 0.7C, Q^2_{min} = 1.4C, Q^3_{min} = 2.1C, k_p = 20, \rho = 10, R_{off} = 40\Omega, R_{on} = 1\Omega$.

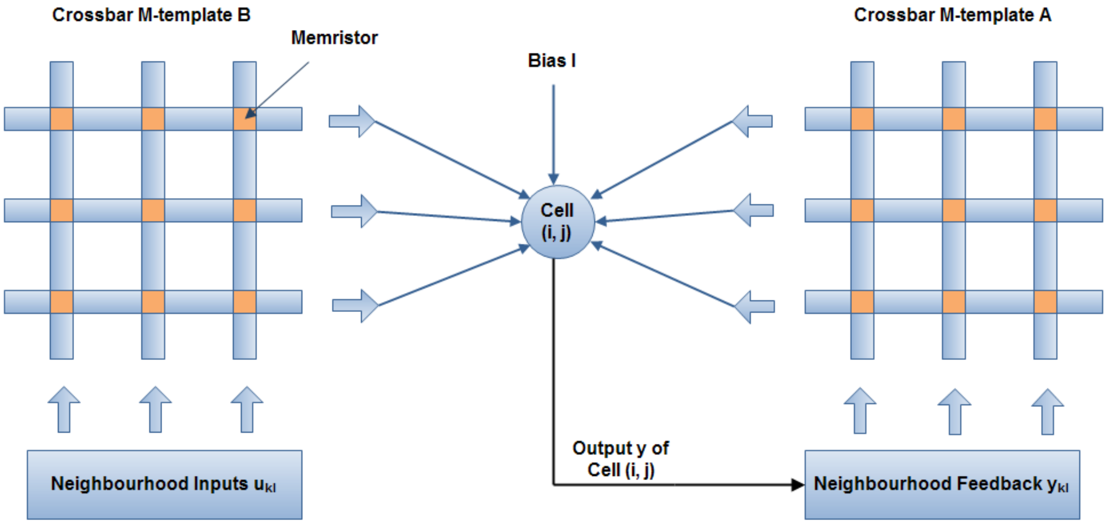

Figure 13: Memristor based CNN templates

given:

$$\frac{dx_{i,j}}{dt} = -x_{i,j} + I + \sum_{k=1}^{M}\sum_{l=1}^{N} B(i,j;k,l)u_{k,l} + \sum_{k=1}^{M}\sum_{l=1}^{N} A(i,j;k,l)y_{k,l} \tag{58}$$

and the output of $C(i,j)$ is determined by a piecewise linear function:

$$y_{i,j} = \frac{1}{2}(|x_{i,j}+1| - |x_{i,j}-1|) \tag{59}$$

where I is the bias. k and l denote the row number and column number of the neighbour. Both the feed-forward template $B(i,j;k,l)$ and feedback template $A(i,j;k,l)$ have the same size in the neighbourhood. The feed-forward interaction is a weighted sum of the input voltages $u_{k,l}$ of all the neighbours. Similarly, the feedback interaction is a weighted sum of the output voltages $y_{k,l}$ of all the neighbours.

3.3.2 Memristor-based template

Templates of CNN are very important as they determine the cell states and outputs of the CNN network. After the invention of the CNN paradigm, lots of templates were discovered. Currently, there exists a large template library for various applications. Due to this case, for some existing applications, it is not necessary to find new templates, and hence the main task is selecting appropriate CNN templates from the template library and setting them in real CNN circuits. Afterwards, the chosen templates will not be changed until another template is required.

The CNN neighbourhood of the cell $C(i,j)$ is defined by

$$\begin{aligned} N_r(i,j) = \{C(k,l)|max\{|k-i|,|l-j|\} \leq r, \\ 1 \leq k \leq M; 1 \leq l \leq N\} \end{aligned} \tag{60}$$

where r is a positive integer number, and k and l are the row number and column number of the neighbour. Usually, the $r = 1$, $r = 2$ and $r = 3$ neighbourhoods are called 3×3 neighbourhood, 5×5 neighbourhood and 7×7 neighbourhood respectively. Obviously, it shows that the neighbourhoods are symmetric in the sense that if $C(i,j)$ is a neighbour of $C(k,l)$, then $C(k,l)$ is a neighbour of $C(i,j)$ as well. By increasing the neighbourhood r, the number of connections of each CNN cell is significantly increased. Thus, the memristor is a promising candidate for neighbourhood connections, as it is possible to implement a large neighbourhood size r.

Since using general HP memristors in CNN has been discussed in [65], here we investigate memristor-based templates from the view of using staircase memristors and the crossbar structure which could be adapted to the CMOL structure. A schematic diagram is illustrated in Fig. 13 which consists of two templates A and B based on the crossbar structure. The inputs u_{kl} and feedback y_{kl} are weighted by corresponding memristors and summed together with a bias signal to change the state of the cell $C(i,j)$. The new architecture of CNN aims to take advantages of memristors, such as nano-scale size, non-volatile and programmable features. The implementation requires hybrid CMOS and memristor circuits with some additional control circuitry, for example programming and inverting modules. Such additional circuitry may take up more space or consume more energy

Resistance Expected	R = 100 Ω		R = 200 Ω		R = 300 Ω	
Frequency	Staircase	HP	Staircase	HP	Staircase	HP
(Hz)	(%)	(%)	(%)	(%)	(%)	(%)
$10f$	0.74	1.5	0.245	0.5	0.017	0.167
$9f$	0.74	6.5	0.245	1.0	0.007	1.33
$8f$	0.74	7.0	0.245	2.25	0.003	0.67
$7f$	0.74	4.5	0.245	3.0	0.1	2.33
$6f$	0.74	4.0	0.245	4.75	0.11	1.5
$5f$	0.74	1.0	0.245	0.495	0.1	0.0
$4f$	0.74	14.5	0.245	2.25	0.11	3.33
$3f$	0.72	13.5	0.245	4.5	0.1	1.33
$2f$	0.71	14.5	0.245	10.0	0.11	8.5
f	0.72	13.5	0.185	25.5	0.1	8.33

Table 4: Staircase memristor model vs general HP memristor model. It measures differences between expected resistance values and actual resistance values of both staircase and general HP memristor models in percentage. The base frequency f of the applied pulse signal is $\frac{1}{2\pi}$ with a duty cycle $D_c = 0.5$.

than the original CNN circuit in some cases. However, current investigations on memristor-CMOS hybrid integrated circuits suggest a strong likelihood that these hybrid circuits reducing the size or power consumption. Especially, when a large number of CMOS devices can be replaced by memristive devices, the circuit is more likely to take the advantage of memristive devices [35, 60, 104] since more space is saved for additional circuitry. Otherwise, saved space is insufficient for additional circuitry which will take up extra space. Therefore, considering CNN circuits with a large number of connections, the memristor is still a promising candidate to implement CNN connections in the future. Furthermore, using staircase memristors in CNN circuit is worth investigating because, in some specific applications, it allows a simpler way to programme the CNN template compared to general HP memristors.

A simple comparison between a staircase memristor model and a general HP memristor model is shown in Table 4, which describes the errors between expected resistance levels and the actual resistance levels obtained by applying the same pulse signal with an amplitude $A = 5\,V$ and a duty cycle $D_c = 0.5$. The HP memristor model proposed in [97] and described by Eq. (18) and (20) is used for comparison with parameters $R_{on} = 1\,\Omega$, $R_{off} = 400\,\Omega$, $\mu_v = 5 \times 10^{-2}\,m^2 s^{-1} V^{-1}$ and the width D is normalised to 1. In contrast to general HP memristor model, a staircase memristor model proposed in this thesis and defined by Eq. (55) with 5 resistance levels is used. It models a staircase memristor containing five resistance levels according to the ferroelectric memristor and other multi-level memristive devices which exhibit five-level resistance states [19, 102]. The errors ϵ is measured by

$$\epsilon = \frac{|R_a - R_e|}{R_e} \times 100 \tag{61}$$

where R_a is the actual resistance obtained and R_e is the expected resistance.

Both the staircase memristor model and the general HP memristor model are programmed towards the expected resistance levels $R = 100\,\Omega$, $R = 200\,\Omega$ and $R = 300\,\Omega$. By decreasing the frequency of the pulse signal from $10f$ to f, the error of the general HP memristor grows significantly. It shows that the general HP memristor model is very sensitive to the frequency, however, the staircase memristor model's resistance is very reliable with a much smaller fluctuation around the expected resistance level. By varying the expected resistance level, the staircase memristor model still have a reliable performance. In contrast, the errors of the general HP memristor model fluctuate a lot because of the non-linearity. As mentioned previously, the change rate of memristance is a non-linear function with respect to charge q, and therefore if the expected resistance (in this case $R = 100\,\Omega$) is close to boundary R_{on}, the large current will lead to a larger charge. Thus, the resistance may have a significant change in a short time. In the cases of $R = 200\,\Omega$ or $R = 300\,\Omega$ which is close to the boundary of R_{off}, a small increment in charge q will result in a large increment in actual resistance, and thus greater errors are observed. The comparison result implies that the staircase memristor is more reliable than the general HP memristor if a specific resistance level is required.

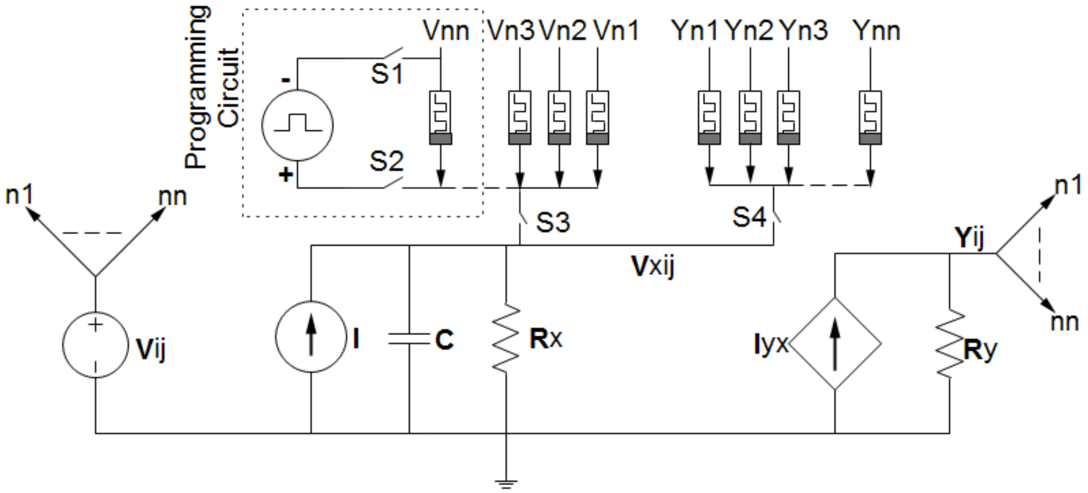

Figure 14: A possible circuit implementation of a CNN which is based on staircase memristors. Each staircase memristor is a connection between the cell (i, j) and one of its neighbours. Cell (i, j) receives all the weighted and summed inputs and outputs from it neighbours and propagates its input V_{ij} and output Y_{ij} to all its neighbours. When programming a staircase memristor, $S3$ and $S4$ will be turned off to isolate staircase memristors to avoid influence from CNN circuits. $S1$ and $S2$ will be turned on to connect staircase memristors to programming circuits.

3.3.3 Implementation design

According to the schematic diagram Fig. 13, a possible circuit implementation of CNN connections based on staircase memristors is demonstrated in Fig. 14. All the inputs and outputs of neighbour cells are weighted by staircase memristors and then summed separately. The weighted and summed inputs and outputs of neighbour cells will contribute to the state of the cell C(i, j). In order to programme the staircase memristor according to different applications, the switch S3 and S4 will be disconnected to isolate staircase memristors, which avoids the influence from the CNN circuit. S1 and S2 will be turned on to connect the required staircase memristors and the provided pulse signal, and hence the memristance of a staircase memristor can be varied by controlling the duration of the pulse. After all, the CNN cell interacts with its neighbours via the programmed memristive templates until the template has to be changed again.

3.4 Memristive echo state network

Reservoir computing is an exciting approach that aims to overcome the training problem that exists in traditional recurrent neural networks (RNNs). It is well-known that training RNNs is inherently difficult even with the important yet powerful error back-propagation (BP) algorithm. It is a time-consuming and computationally expensive job to train RNNs, however there is still a possibility that the training may fail to converge. In the paradigm of reservoir computing, a “reservoir” is a collection of states of neuron activations between the input and output layers. It is generated with random connection weights and used to extract features from the input signals. Distinct from other neural networks, only the readout weights between the “reservoir” and the output layer are trained. The term “reservoir computing” comes mainly from the echo state network (ESN) [52] and the liquid state machine (LSM) [70] which share the concept of a “reservoir”. In principle, a “reservoir” is an excitable, dynamical medium and plays an important role in reservoir computing networks. Theoretically, any dynamical systems with rich dynamics are capable of building a reservoir. Since a memristive system is also a non-linear dynamical system, using memristors as reservoir components has been investigated by Kulkarni and Teuscher [61]. They used the graph-based approach to represent the reservoir network implemented by memristors. However, we propose an echo state network that is based on the memristive CNN structure where memristors are used as the local connections between nodes.

3.4.1 Memristor-based reservoir

In the original ESN, the given training input signal and target output signal are defined by $\mathbf{u}(n) \in \mathbb{R}^{N_u}$ and $\mathbf{y}^{target}(n) \in \mathbb{R}^{N_y}$ respectively. n is the discrete time in the dataset with values $n = 1, 2, 3, 4, \cdots$. N_u and N_y are the number of inputs and outputs in the network. The components of the reservoir are RNN type units with leaky-integrated discrete-time continuous values. The typical update equations are

$$\tilde{\mathbf{x}}(n) = tanh(\mathbf{W}^{in}[1; \mathbf{u}(n)] + \mathbf{W}\mathbf{x}(n-1)) \tag{62}$$

where $\tilde{\mathbf{x}}$ denotes the update of reservoir components, which collects both the inputs and the states of other units. $[a\,; b\,]$ denotes the vertical vector concatenation of vectors a and b.

The new states of the units are defined by

$$\mathbf{x}(n) = (1-\alpha)\mathbf{x}(n-1) + \alpha\tilde{\mathbf{x}}(n) \tag{63}$$

where $\mathbf{x}(n) \in \mathbb{R}^{N_x}$ is a vector of reservoir neuron activations at time step n. α is the leaking rate of the neuron, which is normally within the range $(0, 1]$. $\mathbf{W}^{in}$ is the input weight matrix containing the connection weights between inputs and the reservoir neurons, thus it has the size of $N_x \times (1 + N_u)$. $\mathbf{W}$ is the recurrent weight matrix which consists of connection weights between the reservoir neurons and has the size of $N_x \times N_x$, which implies that the reservoir neurons are fully connected.

The output $\mathbf{y}_n$ is defined by

$$\mathbf{y}_n = \mathbf{W}^{out}[1; \mathbf{u}(n); \mathbf{x}(n)] \tag{64}$$

Thus, the output weight matrix $\mathbf{W}^{out}$ has a size of $N_y \times (1 + N_u + N_x)$. So far, the work-flow of original ESN is defined, and there are 3 main differences compared to the CNN:

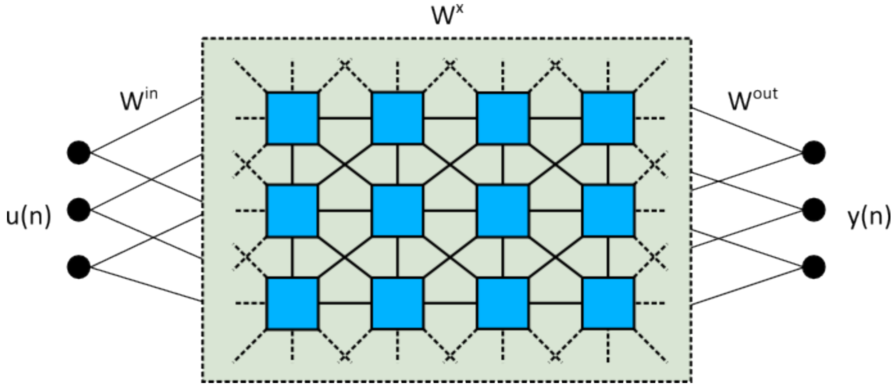

Figure 15: A reservoir with local connections which are implemented by memristors.

1. the network is randomly connected instead of locally connected

2. the network weights are randomly generated instead of a space-invariant template

3. the output is a linear function instead of a piecewise linear function

Since the memristor-based CNN structure is used as the reservoir, only the reservoir network is adjusted to adopt the proposed structure.

From the definitions of the states of the units in Eq. (63) and the its update in Eq. (62), the state vector $\mathbf{x}(n)$ is determined by its previous state $\mathbf{x}(n-1)$, the input $\mathbf{u}(n)$ and states of other units. Thus, according to the definition of CNN, the reservoir network is redefined to have a regular geometric grid and local connections by

$$\sum_{k=1}^{M}\sum_{l=1}^{N}\mathbf{W}^{x}(i,j;k,l)\mathbf{x}(n-1) \tag{65}$$

where we assume that the reservoir cell (i,j) has a neighbourhood size of $M \times N$ neighbours and then the update equation (62) can be rewritten as

$$\tilde{\mathbf{x}}(n) = tanh(\mathbf{W}^{in}[1;\mathbf{u}(n)] + \sum_{k=1}^{M}\sum_{l=1}^{N}\mathbf{W}^{x}(i,j;k,l)\mathbf{x}(n-1)) \tag{66}$$

where $\mathbf{W}^{x}$ is the matrix that denotes the local connections and is implemented by the memristors. This structure is slightly different from the traditional CNN which has a feedback loop containing the outputs of the CNN neurons. The feedback loop in traditional CNN is taken out because the reservoir size is independent of the input size or output size which may not have neighbours. Based on the proposed approach, the basic network is illustrated in Fig. 15 where the reservoir is implemented using local connections. If a reservoir has 100 neurons, the original ESN has 100×100 connections, however, this approach only has 100×8 connections. Therefore, the required connections are significantly reduced.

Chapter 4

Computational Results of Memristive Non-spiking Neural Networks

In this chapter, the proposed staircase memristor model, described in Chapter 3, is further investigated to mimic the behaviours of the ferroelectric memristor. Then, based on the staircase memristor, the proposed CNN structure is simulated by software to implement some frequently-used applications in image processing. Afterwards, the benchmark task, the Mackey-Glass signal dataset, is applied to the echo state network using memristor-based local connections. The aim of the experiments is firstly to assess the performance of the staircase memristor in mimicking the ferroelectric memristor with respect to the constitutive relation of $q - \varphi$; secondly, to assess how memristors perform as local connections in CNN and ESN; and moreover, to demonstrate that such a structure is capable of real applications.

4.1 Asymmetric behaviour of the staircase memristor

In Chapter 2, some switching dynamics of the ferroelectric memristor were demonstrated in Fig. 9 by applying consecutive pulses with different amplitudes. However, this only shows the variation of the state variable s from 1 to 0. In fact, in ferroelectric memristors, the switching dynamics from 0 to 1 is different from the switching dynamics from 1 to 0, which shows that the switching is an asymmetric process. Therefore, it is distinct from the symmetric staircase memristor demonstrated in Chapter 3, and the results should be different as well. For the purpose of demonstrating the ability of the staircase model in mimicking asymmetric behaviour, the similar behaviour of the ferroelectric memristor is simulated by the staircase model.

4.1.1 Simulation results

Because of the asymmetry of switching dynamics in different switching directions, two different paths exist which define an increment path (s varies from 0 to 1) and a decrement path (s varies from 1 to 0). In the case of the ferroelectric memristor, in the decrement path, s evolves towards 0 in a very wavy way. On the contrary, s evolves sharply in the increment path and does not present an obvious wavy dependence. Thus, we assume that the increment path has only one region which instantly evolves from 0.9 to 1. Moreover, the decrement path evolves from 1 to 0 and contains 4 regions. However, current research has only investigated staircase memristors by the strength of the electric field. Thus, the parameters can not be adapted to the proposed directly. In this simulation, we choose possible parameters to fit the experiment results obtained in [19]. The $R_{off}/R_{on} = 300$ was chosen as stated in [19]. By observing these phenomenon, an asymmetric model is built by setting proper thresholds for each region. The simulation result will show that parameters are reasonably chosen to reproduce the experiment results of the ferroelectric memristor.

The similar behaviours illustrated in [19] are shown in Fig. 16 which varies its resistance asymmetrically by applying a sinusoidal current source of $5sin(\pi t)$. The fingerprint of the memristor is the pinched hysteresis loop of the $I-V$ curve which has been demonstrated in Fig. 16(a) for this asymmetric case. It is worth noting that the peculiar $I-V$ curve in Fig. 16(a) is somewhat similar to the hard-switching case of the HP memristor, in which a small negative bias will switch the HP memristor from the on state to the off state. However, in this case, a small positive current is enough to switch s from 0 to 1. Thus, its resistance changes significantly back to R_{min} which leads to a very small voltage drop across the asymmetric memristor model. By investigating the relationship between charge q and the state variable s as shown in Fig. 16(b), it clearly shows that the rise and descent of s have two different paths as we defined. Consequently, a loop is observed in this case, and it shows that s increases rapidly but decreases much more slowly and behaves like a staircase. By comparing Fig. 16(c) and (d), the state variable s in the increment path varies quite sharply in contrast to the decrement path, and hence the increment path does not have a significant delayed-switching effect.

4.2 Memristor-based cellular neural network

4.2.1 Experiment setup

Based on the CNN circuit in Fig. 14, software based simulation has been designed to test the proposed memristor-based CNN. Since CNN is widely utilized in image processing, two tasks are selected to demonstrate its actual performance. Firstly, for the noise removal task, an image containing three objects and noise is selected. Then, for the edge detection task, the processed image in the noise removal task is used.

Flux-controlled memristor

Since the images are normally represented by voltage values in the CNN circuit, during image processing tasks, the input is actually the voltage. The dynamics of the CNN circuit can be described as:

$$dx_{i,j}/dt = -x_{i,j} + I + i_{output} + i_{input} \tag{67}$$

where

$$i_{input} = \sum_{k=1}^{M} \sum_{l=1}^{N} B_{i,j;k,l} u_{k,l} \tag{68}$$

and

$$i_{output} = \sum_{k=1}^{M} \sum_{l=1}^{N} A_{i,j;k,l} y_{k,l} \tag{69}$$

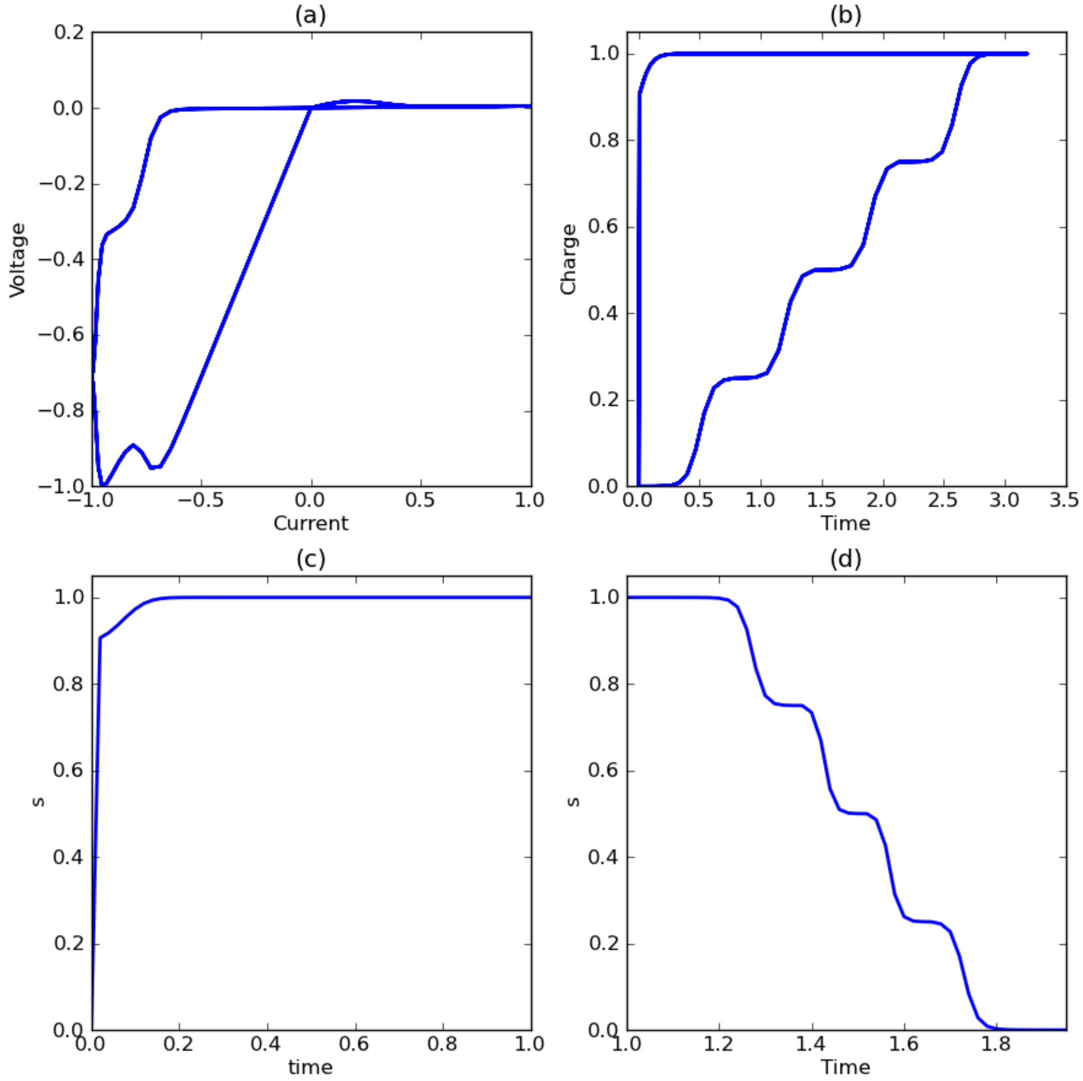

Figure 16: Simulation results of asymmetric behaviours of the staircase memristor model. (a) The $I-V$ curve for the staircase model with asymmetric behaviours. (b) The different switching dynamics of the increment path and decrement path are demonstrated. (c) In the increment path, by applying a small current, the state variable s varies from 0 to 1 in a very short time. (d) In the decrement path, the staircase is observed by applying a negative current. Increment path $\{N = 4, Q^0_{min} = 0C, Q^1_{min} = 14C, Q^2_{min} = 28C, Q^4_{min} = 42C, k_p = 20, \rho = 10\}$. Decrement path $\{N = 1, Q^0_{min} = 0C, k_p = 20, \rho = 0\}$. $R_{off}/R_{on} = 300$.

assuming that a CNN has size of $m \times n$ cells. Thus, the templates A and B which are the local connections are actually represented by the conductance. As a result, a flux-controlled memristor is most suitable for this task. As we know, the conductance is the inverse of the resistance, thus R_{max} induces a low conductance G_{min}, and R_{min} yields a high conductance G_{max}. Most behaviours are very similar to the charge-controlled memristor, and therefore the proposed model in Eq. (55) is also applicable to flux-controlled memristors. Based on this assumption, the Eq. (55) is adapted directly by introducing the flux term φ

$$s = \sum_{i=0}^{N-1} H(i)(w_p + \frac{w_i}{1 + e^{-\varphi k_p^i + \rho^i + \phi_{min}^i}}) \tag{70}$$

For the following experiments, a flux-controlled memristor is modelled with 5 regions using Eq. (70), which can represent 6 conductance levels. The main parameters are illustrated in Table 5. By dividing the whole width of an assumed staircase memristor into 5 regions which have their own width w_i, the model will have 6 memductance levels represented by G_i. These levels are reasonably selected and possible for physical memristors since so far current research shows that memristors can have up to 10 different state levels [60]. Although the threshold ϕ_i of each region has not been studied in current research of physical memristors, it could be tuned conveniently in the model when the real data on thresholds are available. Tuning k_p and ϕ_i jointly, the required time to activate different regions can be changed according to actual data obtained in the future.

Black and white edge detection

In the experiment, we assume a basic machine vision task of edge detection, and a black and white image containing three objects and Gaussian noise is captured. Because noise is anywhere in digital and analog devices and unavoidable, noise removal is normally the first step in image processing and related applications which are sensitive to noise. Then the processed image can be used for the edge detection task.

For the purpose of removing noises from the captured black and white image, noise removal templates shown in Table 6 were programmed to staircase memristor models by applying a pulse signal which has amplitude $+0.1V$ and period $2\pi/6$.

Main parameters

Number of regions: $N = 5$	Propagation speed: $k_p = 20$	Constant: $\rho = 10$

Region width w_i (%)

$w_0 = 0.25$	$w_1 = 0.25$	$w_2 = 0.25$	$w_3 = 0.125$	$w_4 = 0.125$	

Threshold ϕ^i_{min} (C)

$\phi^0_{min} = 0$	$\phi^1_{min} = 14$	$\phi^2_{min} = 28$	$\phi^3_{min} = 42$	$\phi^4_{min} = 56$	

Conductance level G (S)

$G_0 = 2.0$	$G_1 = 1.5$	$G_2 = 1.0$	$G_3 = 0.5$	$G_4 = 0.25$	$G_5 = 0.001$

Table 5: Parameters of the flux-controlled staircase memristor model used in CNN circuit simulation.

The bias current is set to $I = 0$.

Template A			Template B		
0	1	0	0	0	0
1	1	1	0	0	0
0	1	0	0	0	0

Table 6: Desired templates in the noise removal application

During the simulation time, for each iteration, the templates were applied to all CNN cells. Therefore, cells' states evolve along with each iteration in the simulation time. After some simulation time, several iterations were finished and by observing cells' states we can find that image noise is removed by the network. Then, the memristors are programmed again to implement the black and white edge detection templates as shown in Table 7 with bias current $I = -1.5$. However, since the memristor does not have negative resistance or conductance, extra components are required to implement the same function. Possible approaches are utilising negative resistance or inverting the corresponding input voltages before sending them to the local connections.

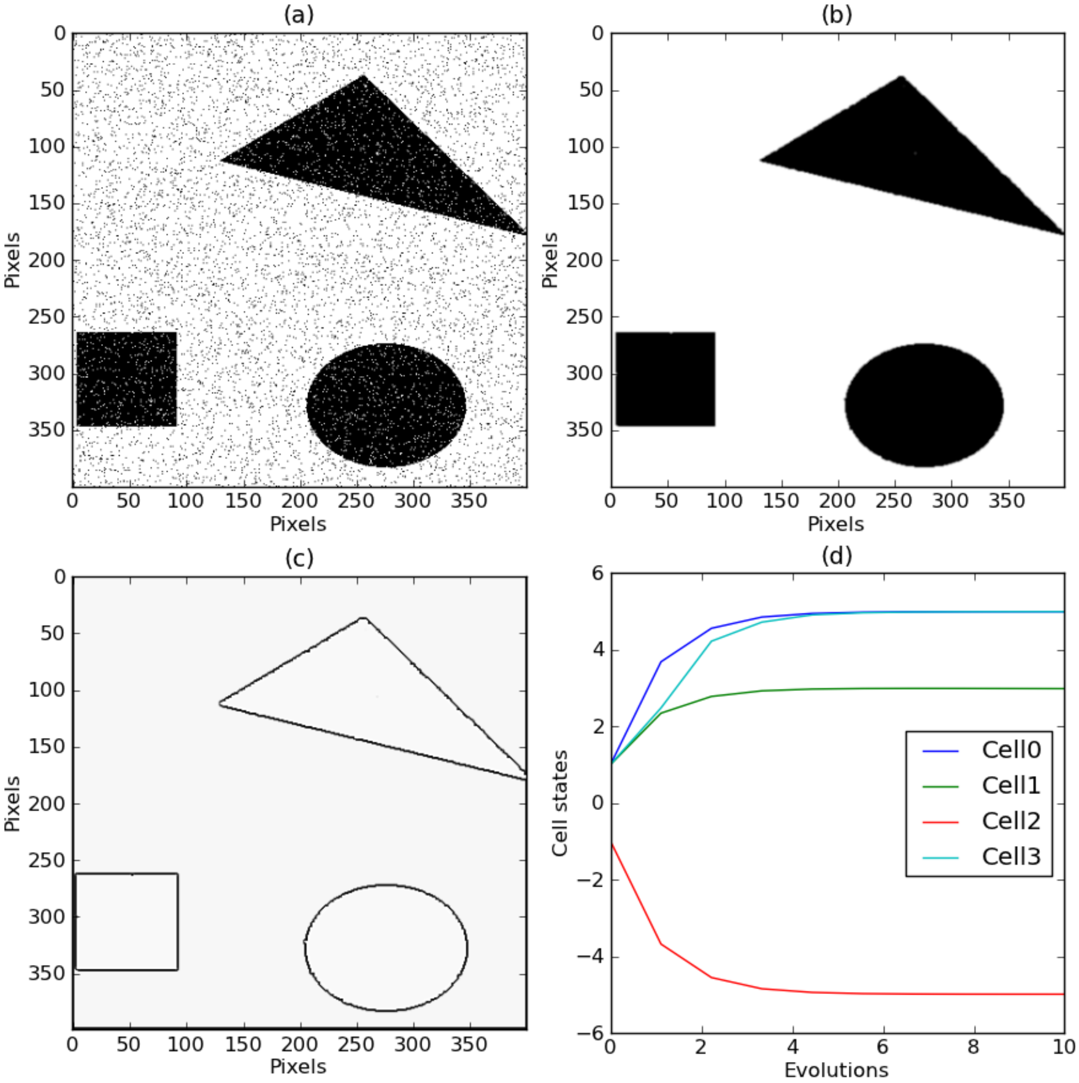

Figure 17: Simulation results of an edge detection task. (a) The original image with Gaussian noise. Image size is 400 × 400 pixels. (b) The result obtained after noise removal task. (c) The edges of the three objects are detected. (d) The evolution of states of selected cells are illustrated which demonstrates that all the cells converge from the initial points to stable points after certain evolutions.

Template A			Template B		
0	0	0	-0.25	-0.25	-0.25
0	2	0	-0.25	2	-0.25
0	0	0	-0.25	-0.25	-0.25

Table 7: Desired templates in the edge detection application

4.2.2 Results

From the results shown in Fig. 17, the whole process from removing the noise in captured images to detecting the edges of the three objects are undertaken by the memristor-based connections. It demonstrates that by re-programming the memristor-based connections to the desired templates, it is possible to undertake several image processing tasks in one circuit. For the case that more complex images are to be processed, the proposed memristive CNN is expected to have promising results as the experiment demonstrated here. The reason is CNN templates dominate the results of a task. If a correct template could be provided, satisfied results could be obtained as expected. One advantage of staircase memristors is its reliability on providing steady states and therefore correct and stable templates can be achieved, which is critical for image processing task using CNN. Although such tasks could be done by software, implementing them in hardware is desirable in engineering applications such as machine vision. In this experiment, the potential of staircase memristors in CNN is demonstrated. However, currently discovered memristors with multi-level resistance can only represent 5 to 10 levels, which limits their possibility for more general applications. For specific tasks, staircase memristors and their models are still very promising in terms of stability and size compared with traditional approaches.

4.3 Memristor-based reservoir

4.3.1 Benchmark task

In order to test the performance of the proposed memristor-based reservoir, the Mackey-Glass time series dataset is used in this task. The dataset is generated from the Mackey-Glass equation which is a non-linear time delay differential equation defined by

$$dm/dt = \beta \frac{m_\tau}{1 + m_\tau^n} - \gamma m \tag{71}$$

where m_τ is the value of m at the time $(t - \tau)$ and τ denotes the delay of the Mackey-Glass system. This equation is used by Mackey and Glass [71] to describe the physiological control system where m denotes the concentration of circulating blood cells. However, we only focus on the data itself rather than the physiological representations. The parameters selected to generate the required dataset are $\beta = 0.2$, $\gamma = 0.1$, $n = 10$ and $\tau = 17$ which gives mild chaos. In this task, the network aims to learn the generated dataset and predict the future values after training.

4.3.2 Experiment setup

Before the experiment, the dataset is divided into two separate parts which are the training set and test set. Each set contains 2000 values but only the values in the training set are used for training the network. The test set is used for comparing the actual results, thus evaluating the performance of the network in this prediction task. The whole process of the experiment of memristive ESN is:

1. generating a reservoir with the size of 32×32
2. programming the memristive connections to random conductance in the range of $[-0.5, 0.5)$
3. running the training set and collecting the activation states of reservoir neurons using Eq. (66) and (63) with a leakage rate $\alpha = 0.4$
4. training the readout weights using ridge regression with a regularization coefficient 1.0×10^8
5. running the test set and evaluating the performance using mean-squared error (MSE)

Since the Mackey-Glass equation only generates a time-series dataset, the network only has one input and one output. For the purpose of comparison, the original ESN is generated using the Python code developed by Mantas [68].

4.3.3 Results

Test No.	Memristive ESN	Original ESN
1	2.04×10^{-2}	1.57×10^{-2}
2	1.11×10^{-2}	1.94×10^{-2}
3	1.69×10^{-2}	1.16×10^{-2}
4	2.15×10^{-2}	1.60×10^{-2}
5	3.61×10^{-2}	9.22×10^{-3}
6	1.93×10^{-2}	1.29×10^{-2}
7	2.12×10^{-2}	3.02×10^{-2}
8	2.34×10^{-2}	2.74×10^{-2}
9	3.06×10^{-2}	2.16×10^{-2}
10	1.93×10^{-2}	1.67×10^{-2}
Average	2.20×10^{-2}	1.81×10^{-2}

Table 8: Simulation results of both ESN networks in 10 trials. In both cases, connection weights are randomly generated in the range of $[-0.5, 0.5)$. Results are measured by mean-squared error (MSE) following steps listed in Section 4.3.2.

In order to evaluate the performance of the memristor-based ESN, 10 running results are obtained for both the original ESN and the proposed ESN as shown in Table 8 using Python 2.7, Oger toolbox 1.1.3 and script developed by Mantas [68]. The results are measured using mean-squared error (MSE) as shown in Eq. (72) which computes the differences between the predicted results and the test set of Mackey-Glass dataset.

$$MSE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (\hat{Y_i} - Y_i)^2 \tag{72}$$

It is noticed that, in Fig. 18(a), the predicted signal is somewhat shifted from the target signal and thus a relatively large error is expected in this experiment. For both cases, the average MSE is around 2.0×10^{-2} in Table 8. Therefore, the average root-mean-squared error (RMSE) is around 1.4×10^{-1}. In order to measure the average RMSE to the scale of the target signal, the normalised RMSE can be obtained by

$$\frac{RMSE}{|m_{max} - m_{min}|} \tag{73}$$

where m_{max} and m_{min} are the maximum and minimum values of the target signal respectively. It gives that the normalised RMSE is approximately 17.5% which confirms the observed error in Fig. 18(a).

By setting the same reservoir size with 1024 neurons and the same leaking rate $\alpha = 0.4$, it shows that the original ESN has a better performance. This might be caused by the simplified CNN structure of the proposed reservoir. Since the proposed ESN is conceptually simple and computationally inexpensive, successfully applying ESN is sometimes empirical. Therefore, the further simplified ESN with a memristor-based CNN structure may lead to more stability problems than the original ESN and yield a slightly worse performance. In this comparison, the weight matrix of the reservoir in original ESN is optimised through normalising and setting its spectral radius. However, in memristive ESN, the weight matrix of the reservoir is not optimised thus there are opportunities to improve its performance by a proper optimisation. By investigating the example results shown in Fig. 18(c), the readout weights $\mathbf{W}^{out}$ are distributed in the range $(-6, 6)$ which is larger than the original ESN's range $(-2, 2)$. According to the practical guide [68], large output weights $\mathbf{W}^{out}$ may imply that the solution is sensitive and unstable because a tiny difference will be amplified by the output weights and lead to large deviations from the expected values. Therefore, the proposed ESN is more sensitive than the original ESN. For the purpose of improving the performance, a practical approach is selecting the parameters carefully and tuning the parameters manually or automatically through grid search which exhaustively searches for proper parameters by comparing the performance metric. Considering the very simplified memristor-based CNN structure, the proposed ESN structure is promising for some specific tasks which require smaller size and less computation, and it is worth further investigation.

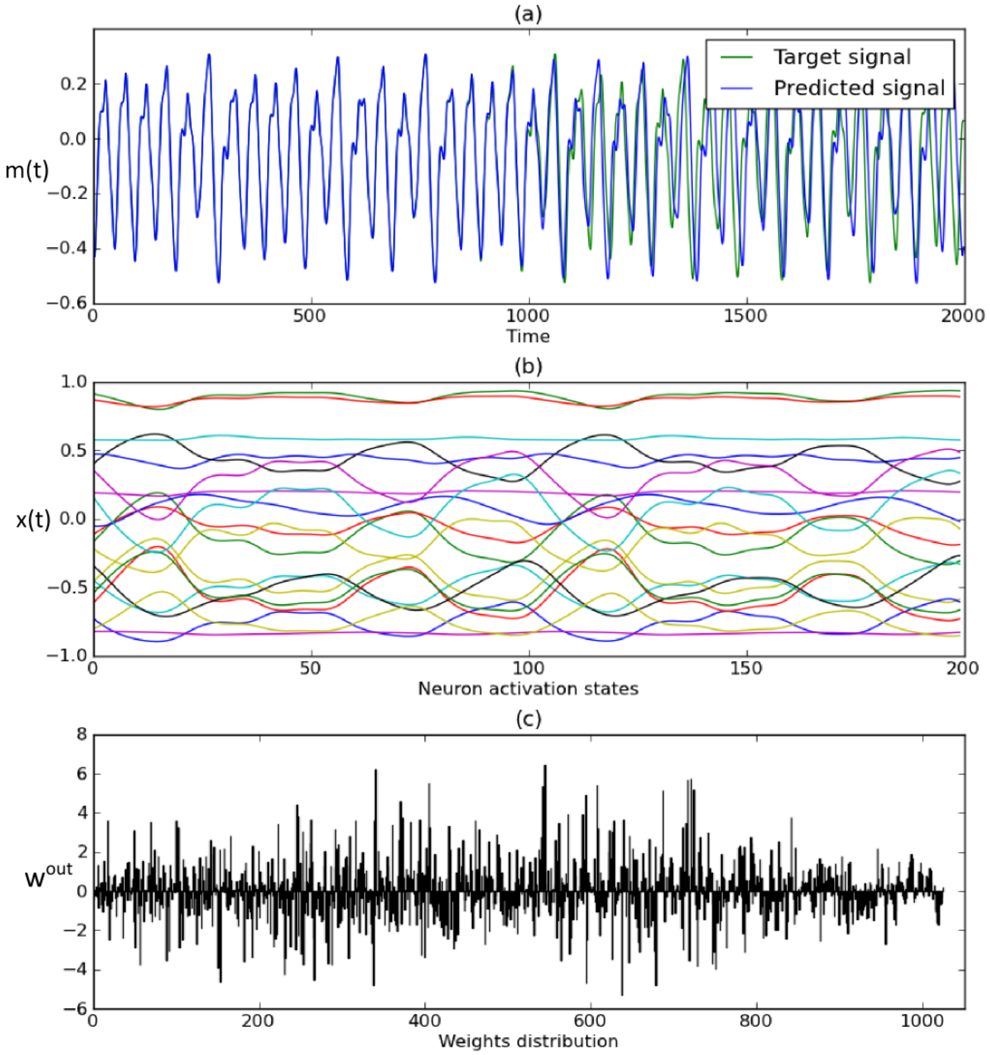

Figure 18: Simulation results of the ESN with memristor-base reservoir and local connections. (a) The predicted results (blue) of the trained network against target signal (green). (b) The activation states of selected reservoir neurons from the 32×32 reservoir. (c) The distribution of the readout weights $\mathbf{W}^{out}$.

Chapter 5

Memristive Spiking Neural Networks

Learning and memory are some of the most striking features of the brain. The studies of the brain and neural networks have become increasingly of interest to researchers and become interdisciplinary fields involving biology, psychology, mathematics, engineering and computer science. Beyond the conventional neural networks, spiking neural networks are usually used to study the behaviours of the brain, biological neurons and networks by passing spikes. The famous Hebb's postulate proposed by Donald Hebb is an important step in exploring the brain. Although, from the view of current research, Hebbian learning is based on an oversimplified view of neuronal morphology [90], it is quantitatively implemented in artificial neural networks through the changes of synaptic efficacy between neuron units. Another mechanism called spiking-time dependent plasticity (STDP) refines the traditional Hebbian learning. It was originally proposed by Gerstner, Ritz and Van Hemmen [40] as an advanced learning algorithm focusing on the temporal correlations of spikes. Nonetheless, experimental evidence has been reported subsequently by neuroscience groups [6, 73]. However, these are just some of the important discoveries in recent research and there are still numerous fields that have not been explored. These new biological discoveries have inspired engineers to realise intelligent systems and networks which, meanwhile, provide new insights in understanding the biological mechanism of neural networks.

Recently, an exciting overlap of neural networks and nanotechnology has created a pertinent research field which focuses on memristor-based spiking neural networks. This kind of network usually uses spiking neurons and memristor-based synapses. In particular, current research is interested in STDP which is an important synaptic modification rule as we introduced earlier in Chapter 2. Linares-Barranco and Serrano-Gotarredona [66] reveal that the memristor is intrinsically similar to a synapse, which is between a pair of communicating neurons. Through the changes of memristance, memristors can mimic LTD and LTP behaviours under the STDP rule. This notable discovery shows a possible way to realise a high-density and intelligent system based on the highly integrated nano devices — memristors. Several research groups have proposed their possible schemes which cover both synchronous and asynchronous implementations.

Jo et al. [54] and Snider [94] used similar methods of time-division multiplexing (TDM) and CMOS neurons to implement the STDP rule. TDM is used to create multiple logical communication channels between the presynaptic neurons and the postsynaptic neuron. These channels are responsible for carrying the computational communication from the presynaptic neuron to the postsynaptic neuron, such as the LTP and LTD timing information. According to the LTP and LTD timing, the square waveform based spike signals are subtracted and the parts beyond thresholds will induce the conductance change of the memristor. The advantage of this system is that it is compatible with digital systems which use a global clock to synchronise the communications. Nevertheless, the biological system is more or less like an asynchronous system without a global clock. Accordingly, possible asynchronous memristive STDP systems are proposed in [1, 63, 66]. Particular spike shapes are used to approximate the STDP learning window but there is a degree of resemblance between these systems. This approach uses the back-propagation concept to send the postsynaptic spike back through the input terminal of the postsynaptic neuron. By applying the voltage difference of the presynaptic spike and the back-propagation spike to the memristor between two neurons, the conductance between them is changed according to the voltage difference. This pair protocol which consists of a nearest pair of presynaptic and postsynaptic spikes (we call it nearest-neighbour spike interaction) works smoothly with these approaches. However, these memristive STDP models have some conflicts in explaining more complicated patterns such as the all-to-all spike interaction which is based on multiple spike interactions. A new memristive STDP model is required to be compatible with nearest-neighbour spike interaction and some complicated interaction schemes. In this chapter, memristors are applied to spiking neural networks using the STDP rule. A spike-trace based memristive STDP model is proposed for memristive synapses. In this model, each spike leaves a trace rather than a particular spike shape to implement the nearest-neighbour and all-to-all spike interactions. In the following sections, we will first investigate the pair-based memristive STDP models. Then we will discuss the incompatibility issues of current pair-based models with the all-to-all interaction scheme. With regards to any problems encountered, the incompatibility issues will be overcome by the proposed memristive STDP model. At last, using the new model, how to implement the nearest-neighbour and all-to-all interactions will be explained in detail.

5.1 Memristive STDP models

It has been revealed that the spike shape is critical to reproduce the STDP function $\xi(\Delta t)$ since different spike shapes will result in distinct STDP function[66]. In general, current asynchronous memristive STDP models assume that a spike generates a spike shape that has a positive part and a negative part and lasts for a set duration. For example, during the spike on-set, the spike shape has a positive amplitude but after that it decreases quickly to a negative amplitude until it reaches the end of a specific spike time. Currently, there are two typical pair-based memristive STDP models, and both of them use piecewise functions to generate a two-part spike. Since they are very similar, we will only explore the model based on the shape of an action potential in detail.

The model based on square and triangular waveforms

The memristive STDP model which is based on square and triangular waveforms has been demonstrated by Afifi, Ayatollahi and Raissi [1]. This model uses a negative pulse followed by a positive triangular pulse to form a spike pattern which is shown in Fig. 19. It is worth noting that the spike has a shorter negative part compared to the positive part. The temporal extents of the learning window are determined by the longer part of the spike. Because a negative pulse is followed by a positive pulse, it makes this model distinct from other models, whereby, a

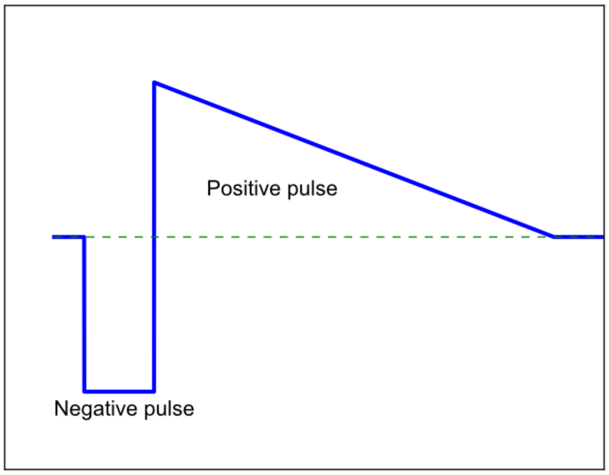

Figure 19: A memristive STDP model which has a spike shape consists of square and triangular waveforms.

positive pulse is followed by a negative pulse. However, such models can reproduce the STDP learning curve approximately.

The model based on the shape of an action potential

Another asynchronous memristive STDP model was proposed by Linares-Barranco and Serrano-Gotarredona [66]. This memristive STDP model places on the shape of the action potentials and the thresholds which determine the changes in the synaptic weight. The key to realise this model is the concept of back-propagation spike which is sent by a neuron to its presynaptic neurons when it fires. Thus, the time difference in STDP rule can be replaced by the voltage difference which is applied to the memristor in the circuit. In this way, this model assumes that a memristor with two thresholds can model the functionality of a synapse, and the synaptic strength can be represented by the memristance.

As shown in Fig. 20, this model has a spike shape that is very similar to action potentials which have a sharply and exponentially increased curve followed by a smoothly and exponentially decreased curve. The spike shape can be expressed

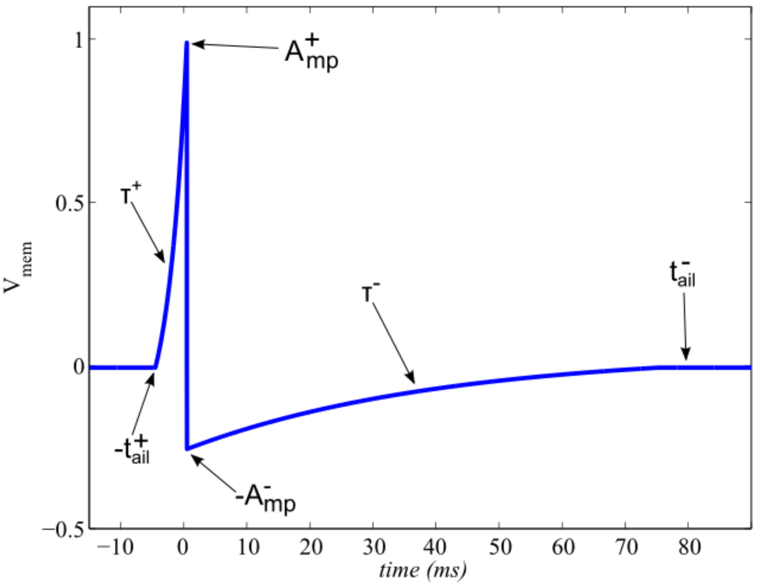

Figure 20: The spike shape produced by the piecewise function $spk(t)$.

mathematically in the form of a piecewise exponential function as follows

$$spk(t) = \begin{cases} A^{+}_{mp}\frac{e^{t/\tau^{+}}-e^{-t^{+}_{ail}/\tau^{+}}}{1-e^{-t^{+}_{ail}}} & \text{if } -t^{+}_{ail} < t < 0 \\ -A^{+}_{mp}\frac{e^{-t/\tau^{-}}-e^{-t^{-}_{ail}/\tau^{-}}}{1-e^{-t^{+}_{ail}}} & \text{if } 0 < t < t^{-}_{ail} \\ 0 & otherwise \end{cases} \tag{74}$$

The memristance modification is controlled by its physical state variable w and the voltage difference between the presynaptic and the postsynaptic spikes over the defined thresholds

$$\frac{dw}{dt} = f(v(\Delta t)) \tag{75}$$

where $v(\Delta t)$ denotes the voltage difference across the memristor. $\frac{dw}{dt}$ denotes the rate of the intrinsic physical state variable.

The function f could have several forms as mentioned previously in Chapter 2. However, concerning Eq. (75) in [67], it is a piecewise function which is only dependent on the voltage difference $v(\Delta t)$ and defined by

$$f(v(\Delta t)) = \begin{cases} I_0 sign(v(\Delta t))(e^{|v(\Delta t)|/v_0} - e^{v_{th}/v_0}) & \text{if } |v(\Delta t)| > v_{th} \\ 0 & otherwise \end{cases} \tag{76}$$

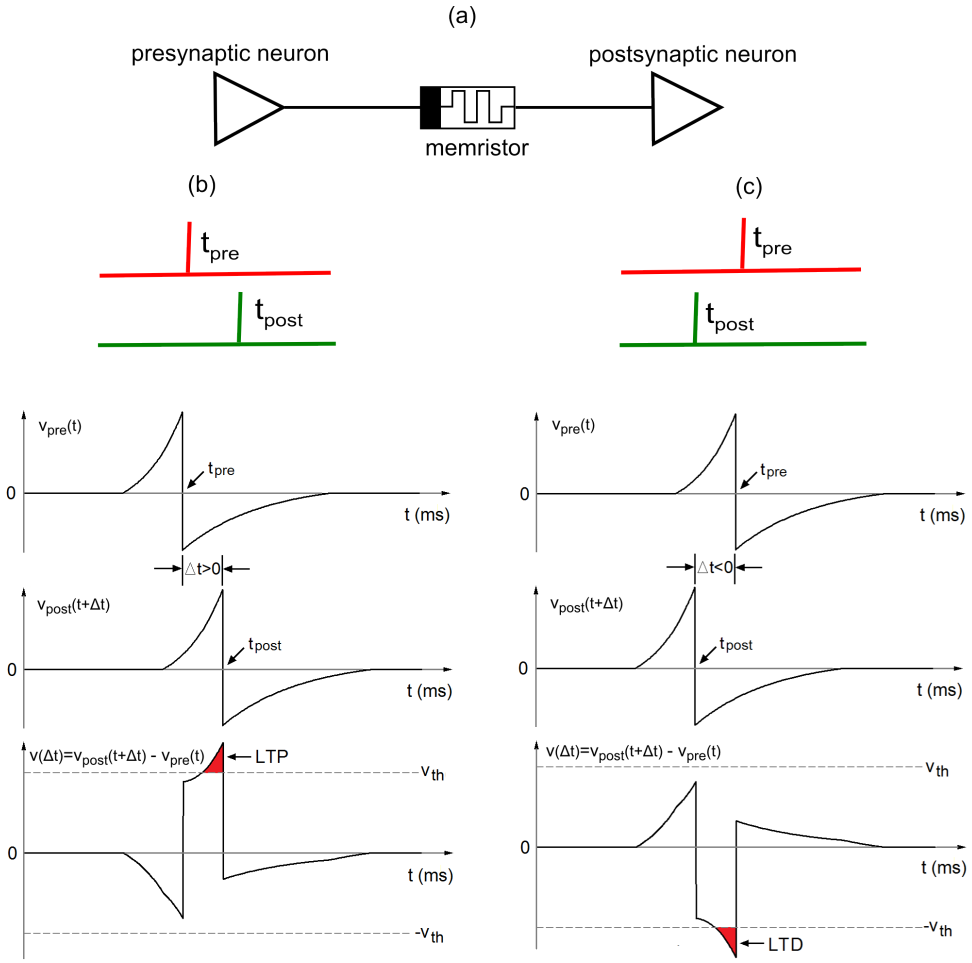

Figure 21: The memristive STDP model which has a spike shape of an action potential (Redrawn from [14]). The shaded areas beyond the thresholds are the changes in state variable of the memristor. (a) LTP is demonstrated when the presynaptic spike precedes the postsynaptic spike. In this case the time difference $\Delta t > 0$. (b) LTD is demonstrated when the postsynaptic spike leads the presynaptic spike. In this case, the time difference $\Delta t < 0$.

Thus, the memristance can only be changed if the voltage difference $|v(\Delta t)|$ is greater than the thresholds, and $v(\Delta t)$ is defined by

$$v(\Delta t) = v_{post}(t + \Delta t) - v_{pre}(t) \tag{77}$$

Both the postsynaptic spike signal v_{post} and the presynaptic spike signal v_{pre} have the shape defined in Eq. (74) but with different parameters in real applications. As an example, the voltage difference $v(\Delta t)$ is illustrated at the bottom of Fig. 21.

By integrating Eq. (75), the expected change in the state variable w of the memristor can be obtained by

$$\Delta w(\Delta t) = \int f(v(\Delta t))dt \tag{78}$$

According to Eq. (75), only the part over the thresholds is involved in changing the conductance, which is illustrated in Fig. 21. It is clear that the shaded area beyond the thresholds represents $\Delta w(\Delta t)$ in Eq. (78). As a result, by applying different Δt to these equations mentioned above, the STDP learning window can be predicted and shown in Fig. 22.

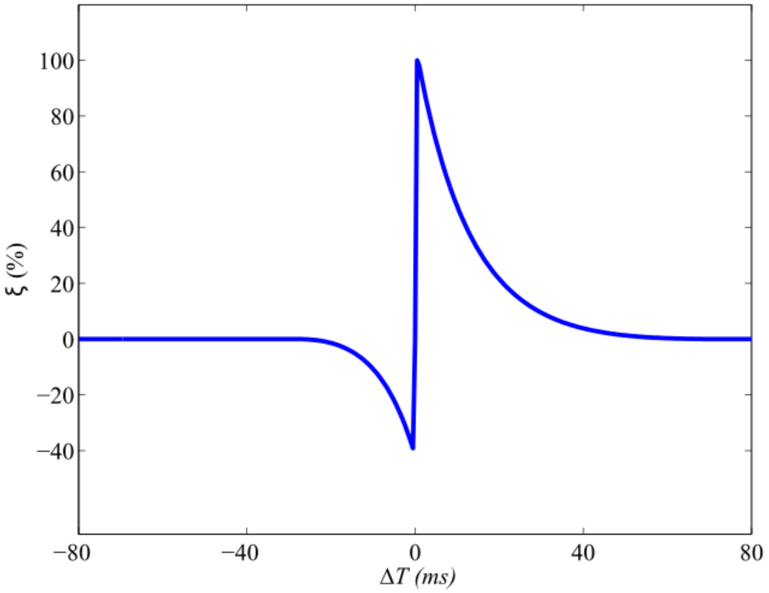

Figure 22: The STDP window function produced by the memristive STDP model in [67].

5.2 Incompatibility with all-to-all interaction

As we have discussed in Section 5.1, the current memristive STDP models can work successfully with the nearest-neighbour interaction, however, they lack the capabilities to implement the interaction schemes which are based on multiple spike interactions such as the all-to-all scheme. In the all-to-all interaction, each presynaptic spike interacts with all postsynaptic spikes and vice versa. It has been used in studying pair-based and triplet rules and discussed by Pfister and Gerstner [81, 82]. As described previously, in this scheme, in order to show that the presynaptic spikes contribute equivalently to LTP, all the preceding presynaptic spikes are required to be summed together or the spike amplitude should be increased according to the time differences between presynaptic spikes.

However, the memristive STDP models we introduced in Section 5.1 are not designed for the multiple spike interactions. Mainly, this is due to the following incompatibility issues:

- Since the positive part of the spike is significantly narrower than the negative part, the pre- or postsynaptic spikes are required to be very close in order to sum the positive parts.

- When the negative parts or positive parts of the pre- or postsynaptic spikes are summed together, the summation itself may be larger than the thresholds and induce the synaptic modification on the memristor.

By recalling Eq. (78), (77) and (76), we investigate the reasons behind the problems. From these equations, it is clear that the synaptic modification on the memristor is only based on the voltage difference $v(\Delta t)$ beyond the constant thresholds. However, in all-to-all interaction, the summation of the pre- or postsynaptic spikes may rise beyond the constant thresholds. This will lead to unwanted LTP or LTD on memristors, which explains the incompatibility issues. As shown in Fig. 23, two examples show that the summations of presynaptic spikes induce changes on memristors. In (a) and (b), three spikes are summed together, and therefore the negative threshold is reached which induces LTD on memristor. However, there is no summation effect on positive parts because these spikes are not close enough. In (c) and (d), the second presynaptic spike follows the first spike closely, and both the positive and negative thresholds are reached. These

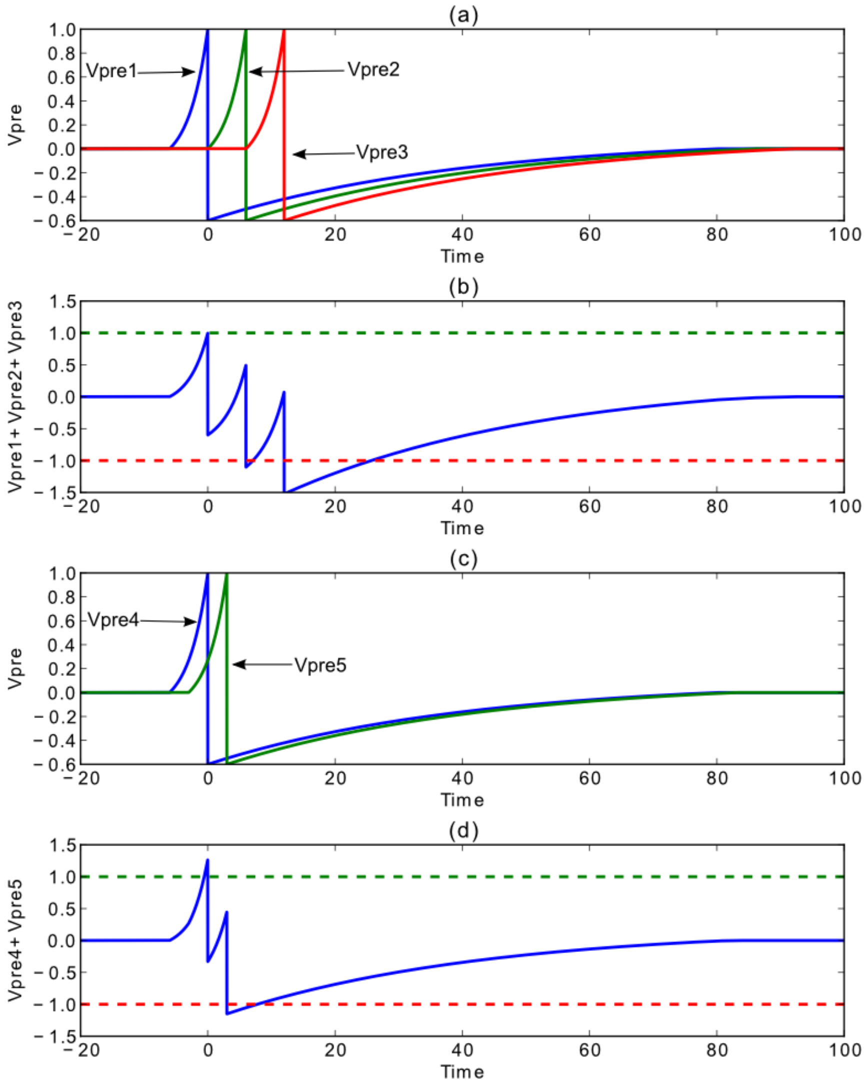

Figure 23: The demonstration of the incompatibility of the action potential shape based memristive STDP model. (a) Three presynaptic spikes are generated sequentially. (b) Three presynaptic spikes are summed to implement all-to-all interaction. The summation of presynaptic spikes is beyond the negative threshold. (c) Two presynaptic spikes are generated with small time intervals. (d) The summation of two presynaptic spikes is beyond both the positive and negative thresholds.

two examples show that this kind of memristive STDP model is not suitable for all-to-all interaction by its definition which is only dependent on the positive and negative thresholds. Hence, a new memristive STDP model is required to overcome these problems.

5.3 Trace-based memristive STDP model

Inspired by the work of Sjöström, Gerstner and Pfister [40, 81, 89] in SNNs and STDP learning, we present a new memristive STDP model for memristive neural networks based on spiking times of pre- and postsynaptic neurons and spike traces. To derive this learning rule, we begin with the concepts of STDP learning. It has been shown that postsynaptic spiking within a few milliseconds after presynaptic activation results in synaptic enhancement, whereas postsynaptic spiking within a few milliseconds before presynaptic activation results in synaptic depression. This has been illustrated and shown in Fig. 24 which demonstrates how the timing of spikes influences the synaptic modification from the experimental data obtained by Bi and Poo [7].

It is clear that, in Fig. 24, there exists a critical window of spike timing for synaptic modification. It is measured by defining how much time the postsynaptic spike precedes or lags behind the presynaptic spike, and therefore we have a time difference $\Delta t = t_{post} - t_{pre}$ which consists of positive and negative intervals. The fitting curves of both positive and negative intervals portray an identical tendency of which a smaller time difference $|\Delta t|$ has a distinct possibility of inducing a larger synaptic modification. However, multiple forms of STDP exist with similar scenarios but with different spiking time windows and synaptic modifications. For instance, in some systems, the synaptic modification according to the timing window is the exact opposite compared with the typical STDP which is illustrated in Fig. 24 [5, 44].

To define the relation of time difference Δt and synaptic modification, a popular choice to model this STDP window $f(\Delta t)$ [41] is

$$f(\Delta t) = \begin{cases} A^{+} e^{-\Delta t/\tau^{+}} & \text{if } \Delta t > 0 \\ -A^{-} e^{\Delta t/\tau^{-}} & \text{if } \Delta t < 0 \end{cases} \tag{79}$$

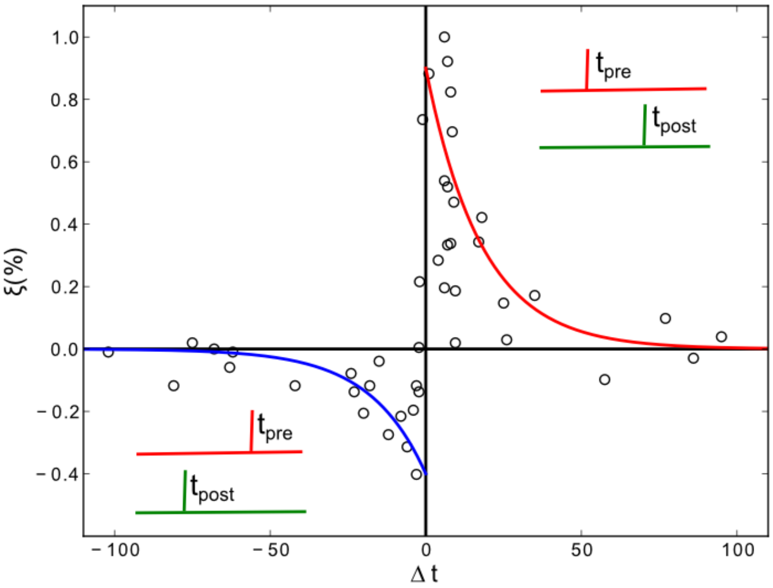

Figure 24: A typical window of spike timing in synaptic modification (redrawn from the biological experiment data in [7]). Data is fitted using Python, Numpy and Matplotlib.

where A^+ and A^- denote the maximum synaptic modification of the positive and negative time difference correspondingly. τ^+ and τ^- jointly determine the width of the whole STDP window.

5.3.1 Trace-based memristive STDP model

Since the incongruity issues of current memristive models are caused by the shape and the thresholds, the new memristive STDP model attempts to avoid the explicit thresholds and a particular shape like the action potential. By investigating Eq. (79), at different Δt, the areas under its curve can reproduce the same tendency of the STDP window. These areas can be represented by the charge q or flux φ which is obtained by integrating the current or voltage. By applying the voltage or current which has the shape of an exponential function to memristor, we found the potential to use the exponential waveform to build a memristive STDP model. Based on this, the new memristive STDP model is derived by supposing that each arrival of a presynaptic or postsynaptic spike leaves a trace which has an amplitude

A_{mp} at the moment of the spike arrival and decays exponentially in the absence of the spike. In the case where a postsynaptic spike lags behind a presynaptic spike, the trace left by the presynaptic spike will be read out at the moment of the arrival of the postsynaptic spike. By integrating the remaining trace after the postsynaptic activation, the synaptic modification is therefore changed.

In order to realise this process, firstly, two traces, which are generated by the pre- and postsynaptic spikes separately, have to be defined. The spike trace spk_i^+ left by a presynaptic spike at time t_i is defined by

$$spk_i^+(t) = \begin{cases} A_{mp}^+ \frac{e^{(-t+t_i)/\tau^+} - e^{-t_{ail}^+/\tau^+}}{1-e^{-t_{ail}^+/\tau^+}} & \text{if} \quad t_i < t < t_{ail}^+ + t_i \\ 0 & \text{if} \quad otherwise \end{cases} \tag{80}$$

and the spike trace spk_j^- left by a postsynaptic spike at time t_j is defined by

$$spk_j^-(t) = \begin{cases} -A_{mp}^- \frac{e^{(-t+t_j)/\tau^-} - e^{-t_{ail}^-/\tau^-}}{1-e^{-t_{ail}^-/\tau^-}} & \text{if} \quad t_j < t < t_{ail}^- + t_j \\ 0 & \text{if} \quad otherwise \end{cases} \tag{81}$$

The terms τ^+ and τ^- are constants that determine the rate of synaptic modification. The terms t_{ail}^+ and t_{ail}^- denote the time points when the traces reach the end. According to the experimental data obtained by Bi and Poo [7], the significant synaptic modification happens in a critical window of $[-80, 80]$, thus t_{ail}^+ and t_{ail}^- are normally equal to 80 and -80 correspondingly.

Consider the situation of LTP that a presynaptic spike spk_i^+ precedes a postsynaptic spike spk_j^-. The spike trace of the presynaptic spike will be read out from the moment t_j of the postsynaptic activation until the end of the trace $spk_i^+(t)$. Accordingly, the voltage applied to the memristor is

$$V_{mem}^+(t)|_{t_0=t_j} = A_{mp}^+ \frac{e^{(-t+t_i)/\tau^+} - e^{-t_{ail}^+/\tau^+}}{1 - e^{-t_{ail}^+/\tau^+}} \tag{82}$$

where the initial value of t is t_j. Since the modification of the memristance is governed by a function of its state variable w and the applied voltage V_{mem}:

$$\frac{dw}{dt} = f(w, V_{mem}) \tag{83}$$

the memristance can be changed by integrating Eq. (83). Therefore, we get the

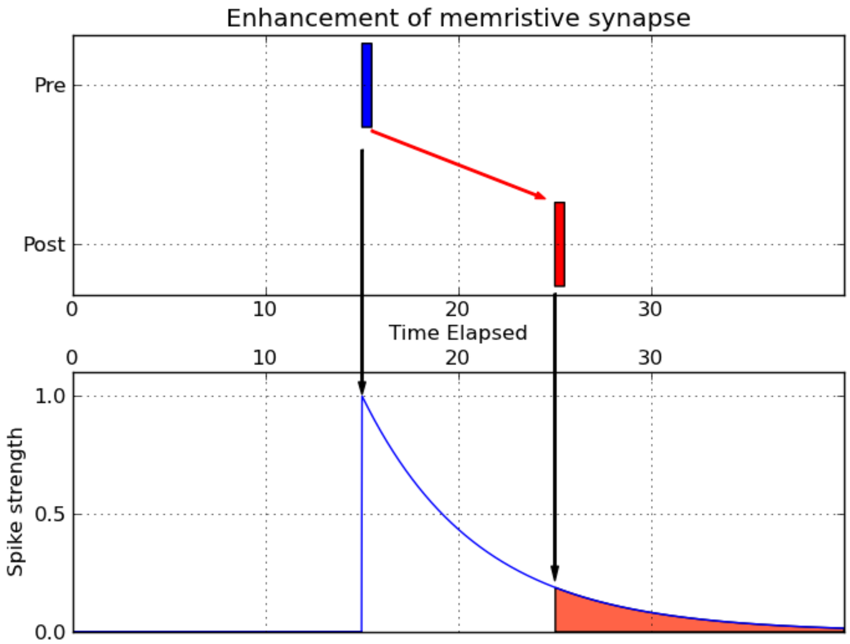

Figure 25: The enhancement of a memristive synapse. This graph sketches spike timings of pre- and postsynaptic neurons and the corresponding trace left by the spike of the presynaptic neuron. At the time of the postsynaptic spike, the trace after it is read out until the trace reaches the end we defined.

total change in the state variable of the memristor:

$$\Delta w(t) = \int_{t_0=t_j}^{t_{ail}^{+}+t_i} f(V_{mem}^{+}(t))dt \tag{84}$$

If we assume that f is a linear function and solely depends on voltage V_{mem}, the Eq. (84) can be re-written as

$$\Delta w(t) = k_m \int_{t_0=t_j}^{t_{ail}^{+}+t_i} V_{mem}^{+}(t) \tag{85}$$

This process is illustrated in Fig. 25, and LTP is induced at the time of the postsynaptic spike. The shaded area represents the expected change on the state variable w which is the integration of the trace V_{mem}^{+} between the time of the postsynaptic spike and the end of the trace.

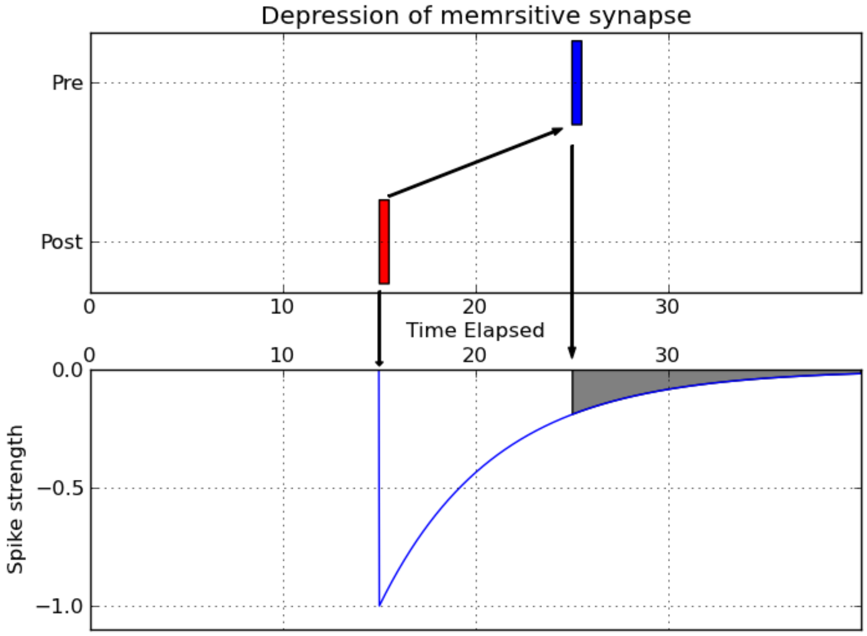

Figure 26: The depression of a memristive synapse. This graph sketches spike timings of pre- and postsynaptic neurons and the corresponding trace left by the spike of the postsynaptic neuron. At the time of presynaptic spike, the trace after it is read out until the trace reaches the end we defined.

Conversely, for the case that the postsynaptic spike spk_j^- precedes the presynaptic spike spk_i^+ which induces LTD, the total change on the state variable w of the memristor is

$$\Delta w(t) = k_m \int_{t_0=t_i}^{t_{ail}^- + t_j} V_{mem}^-(t) \tag{86}$$

where the voltage $V_{mem}^-(t)$ is the spike trace generated by the postsynaptic spike spk_j^-. This process is also illustrated in Fig. 26, and LTD is induced at the time of presynaptic spike. The shaded area represents the expected change on state variable w. Δw is therefore obtained by integrating the trace V_{mem}^- between the time of presynaptic spike and the end of the trace.

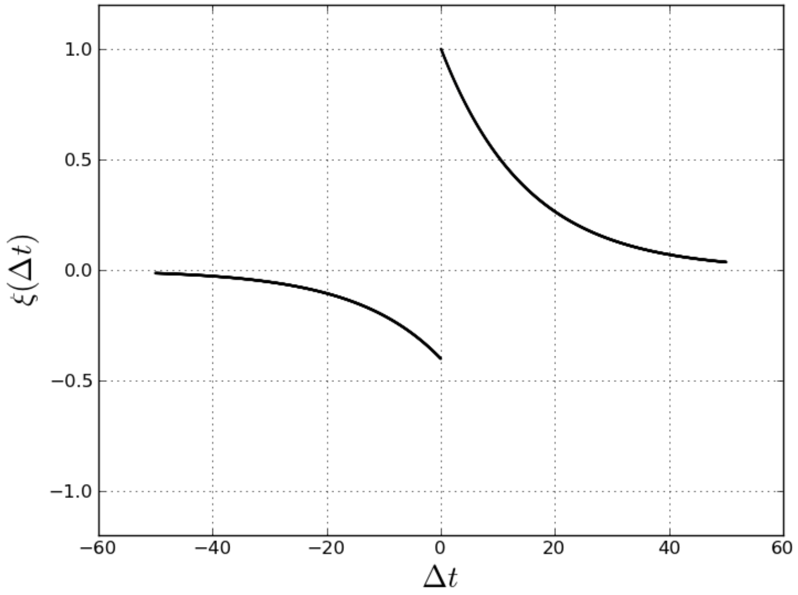

Figure 27: The synaptic modification of the trace-based memristive STDP model. The time difference Δt lies in the range $[-60, 60]$. The positive Δt induces LTP, and smaller $|\Delta t|$ leads to a larger synaptic modification on the memristor. The negative Δt induces LTD, and smaller $|\Delta t|$ leads to a larger synaptic modification on the memristor.

5.3.2 The relation with time difference

Since STDP learning window demonstrates the relationship of the time difference Δt and the synaptic modification ξ, a satisfactory memristive STDP model should be capable of showing that the change on state variable w depends on the time difference Δt. By investigating Eq. (84), the length of V^+_{mem} depends on the postsynaptic spike at time t_j, and hence, V^+_{mem} starts from t_j to $t^+_{ail} + t_i$. As we know that $\Delta t = t_j - t_i$, t_j and t_i can be substituted by $\Delta t + t_i$ and $t_j - \Delta t$ respectively.

Hence, if we substitute the t_i and t_j in Eq. (85) and (86), the total change on w for LTP is

$$\Delta w(t) = k_m \int_{\Delta t}^{t^+_{ail}} V^+_{mem}(t) \qquad \text{if} \quad t_i = 0 \tag{87}$$

and for LTD is

$$\Delta w(t) = k_m \int_{-\Delta t}^{t^-_{ail}} V^-_{mem}(t) \qquad \text{if} \quad t_j = 0 \tag{88}$$

Because t_{ail}^{+} and t_{ail}^{-} are constant terms, it is clear that Δw only depends on the time difference $|\Delta t|$. By varying Δt, the relation between Δt and the change of state variable Δw of the new memristive STDP model can be established. This relation is demonstrated in Fig. 27 which shows the synaptic modification of the new memristive STDP model is an exponential function of Δt in the positive and negative time intervals. It also resembles the shape of the standard STDP learning window which fits the biological experiment.

5.4 Memristive plasticity

It is notable that the synaptic modification we discussed in the new memristive STDP model is the total change on the state variable w depending only on Δt, rather than the actual change on the memristance R_{mem}. In this section, the modification on actual memristance R_{mem} is investigated by using the existing memristor model and the new memristive STDP model.

5.4.1 Change of memristance based on the new model

Herein we provide a theoretical analysis to prove that the synaptic modification of memristive synapse on time difference Δt obeys the standard STDP learning window. By combining the equations (87) and (88) together, we have the update of w by leaving out the constant term k_m

$$\Delta w = \begin{cases} -A_{+}\tau_{+}(e^{-t_{ail}^{+}/\tau_{+}} - e^{-\Delta t/\tau_{+}}) & \text{if } \Delta t > 0 \\ A_{-}\tau_{-}(e^{-t_{ail}^{-}/\tau_{-}} - e^{\Delta t/\tau_{-}}) & \text{if } \Delta t < 0 \end{cases} \tag{89}$$

Δw will be the increment or decrement of the state variable of the memristor. Since the HP memristor model is well-known and a popular choice in studying memristors, we first take the HP memristor model as an example.

Since the memristance in HP model is defined by

$$R_{mem} = R_{on}\frac{w}{D} + R_{off}\left(1 - \frac{w}{D}\right) \tag{90}$$

the change on memristance R_{mem} can be defined by

$$\Delta R_{mem} = R_{mem}(w_0 + \Delta w) - R_{mem}(w_0) \tag{91}$$

where w_0 denotes the initial value of state variable w of the memristor. By substituting Eq. (90) into (91), the total change of memristance ΔR_{mem} is

$$\Delta R_{mem} = (R_{on} - R_{off}) \frac{\Delta w}{D} \tag{92}$$

Apparently, the total change of memristance is a linear function of Δw. It will have the same shape with Δw as well as the standard STDP window function.

Influence of the boundary effect

In the case of the non-linear drift model with boundary effect, the term w/D is bounded into the range $[0, 1]$. However, this does not affect the new memristive STDP model between the boundaries. When the initial value of w/D is close to the boundaries, the shape of ΔR_{mem} will be slightly different with the standard STDP window. This is caused by the definition of the boundary effect that the w/D has to vary slowly when it is reaching the boundaries. Accordingly, the expected change on w/D will be smaller and, in extraordinary circumstances, it barely changes.

5.4.2 STDP variations

Besides the standard STDP learning, there are also some special cases of STDP learning. In fact, the size of the LTD timing window usually varies in different brain regions. Another case is the anti-STDP learning window, in which, the temporal requirements are the exact opposite compared with the standard STDP window. These variations exist in different brain regions and neuronal types, which are important for the synaptic plasticity. That the STDP varies dramatically from one neuronal type to another, which is hardly coincidental, but a reflection of the specific functionality of these neuronal types in the brain circuitry [90]. Therefore, we demonstrate the abilities of the new memristive STDP model in reproducing the STDP variations.

The anti-STDP learning

As to the case of the anti-STDP learning, there are two possible means to implementation in the new model. Since the memristor has polarity, implementation can be achieved by switching its polarity as shown in Fig. 28(a). This means it works like an inhibitory synapse between the pre- and postsynaptic neurons. It is worth noting that, for anti-STDP learning with a kind of HP memristor model, when the time difference $\Delta t > 0$, the presynaptic trace will be read out. However, the positive Δw leads to an increase in memristance (a decrease in synaptic strength) because the polarity of the memristor is switched. Conversely, when the time difference $\Delta t < 0$, the postsynaptic trace will be read out. The negative Δw leads to a decrease in memristance (an increase in synaptic strength). Another means to implement the anti-STDP learning is achieved by changing the signs of the amplitudes A_{mp} of the pre- and postsynaptic spikes. In this case, a positive Δt leads to a negative Δw which decreases the synaptic strength. However, the negative Δt leads to a positive Δw which enhances the synaptic strength. In a word, whether a memristor works like an excitatory or inhibitory synapse is determined by the integration of the trace read out by the model and the polarity of the memristor.

In order to implement STDP learnings with different timing windows or synaptic modifications, the parameters τ^+, τ^-, A^+_{mp} and A^-_{mp} can be tuned separately or jointly. As shown in Fig. 28(b), particular values of the mentioned parameters can be seen for illustrative purposes only.

5.5 Spike interactions with trace-based memristive STDP model

Since the new memristive STDP model aims to overcome the incompatibility problems of the current models with the all-to-all spike interaction, herein we focus on how the new model works with the all-to-all interaction as well as the nearest-neighbour interaction. Although there are several possible spike interaction schemes in standard STDP, the basic spike interactions are based on the form of one-to-one, one-to-multiple and multiple-to-one interaction. The all-to-all

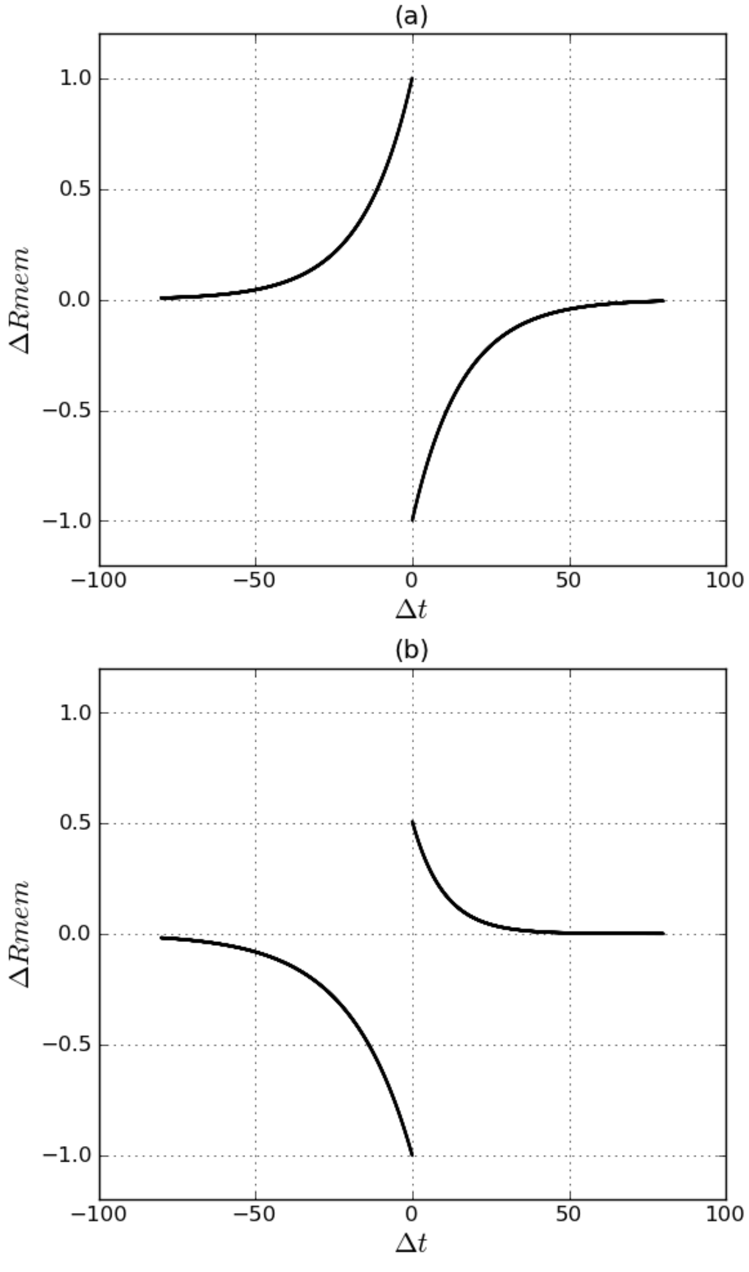

Figure 28: The STDP variations obtained through the actual changes in the memristance and the time difference Δt. (a) The anti-STDP learning that a positive Δt induces LTD in the memristive synapse. However, a negative Δt induces LTP in the memristive synapse. (b) A variation of the STDP learning is that a negative Δt is likely to produce a larger modification than a positive Δt.

interaction scheme is basically based on the form of one-to-multiple and multiple-to-one interaction. Meanwhile the nearest-neighbour interaction scheme is based on one-to-one interaction. Hence, by investigating these two interaction schemes, it is possible to implement other interaction schemes based on them and the new memristive STDP model.

5.5.1 All-to-all interaction

In all-to-all interaction, each presynaptic spike interacts with all postsynaptic spikes and vice versa [81, 82]. It means all the spike pairs contribute equally to inducing LTP and LTD. With regard to LTP, a postsynaptic spike spk_j^- at time t_j interacts with all the presynaptic spikes before it and induces LTP. Similarly, as to LTD, a presynaptic spike spk_i^+ at time t_i interacts with all the postsynaptic spikes before it.

All-to-all interaction is implemented by using two separate traces y_{pre} and y_{post} for both the pre- and postsynaptic spikes. Take the trace y_{pre} as an example; it is constructed by all the presynaptic spikes spk_i^+ with the following conditions:

1. each presynaptic spike $spk_i^+(t)$ ends at the moment of the arrival of the next spike $spk_{i+1}^+(t)$
2. the amplitude of a presynaptic spike $spk_{i+1}^+(t)$ depends on the last value $spk_i^+(t_{i+1})$ of the previous spike $spk_i^+(t)$

Hence, the new $spk_{i+1}^+(t)$ is

$$\hat{spk}_{i+1}^{+}(t) = (A_{mp}^{+} + spk_i^{+}(t_{i+1}))\frac{e^{(-t+t_{i+1})/\tau^{+}} - e^{-t_{ail}^{+}/\tau^{+}}}{1 - e^{-t_{ail}^{+}/\tau^{+}}} \tag{93}$$

It can be rewritten using Eq. (80)

$$\begin{aligned} \hat{spk}_{i+1}^{+}(t) &= spk_{i+1}^{+}(t) + spk_i^{+}(t_{i+1})\frac{e^{(-t+t_{i+1})/\tau^{+}} - e^{-t_{ail}^{+}/\tau^{+}}}{1 - e^{-t_{ail}^{+}/\tau^{+}}} \\ &= spk_{i+1}^{+}(t) + \hat{spk}_i^{+}(t) \end{aligned} \tag{94}$$

The second term in Eq. (94) is the part truncated from the spike $spk_i^+(t)$ by the condition 1. $\hat{spk}_i^{+}(t)$ denotes the new spike trace at time t_i. If $i = 1$, clearly the trace y_{pre} constructed by spike $spk_i^+(t)$ and $spk_{i+1}^+(t)$ in fact is the summation of them. In summary, the y_{pre} can be defined by

$$y_{pre} = \sum_{i=1}^{M} spk_i^+(t) \tag{95}$$

where i, which has values $i = 1, 2, 3, ...$, labels the number of the presynaptic spikes. This reveals that all the presynaptic spikes leave an accumulated trace y_{pre}. Consequently, when the postsynaptic spike arrives at time t_j, the accumulated trace y_{pre} will be read out from the moment of the postsynaptic spike by an amount left by y_{pre} until the arrival of the next presynaptic spike.

Similarly, for the trace of postsynaptic spikes, y_{post} is defined by

$$y_{post} = \sum_{j=1}^{N} spk_j^-(t) \tag{96}$$

where j, which has values $j = 1, 2, 3, ...$, labels the number of the postsynaptic spikes. The accumulated trace of postsynaptic spikes will be read out from the moment of the presynaptic spike.

As an example, LTP is demonstrated in Fig. 29(a) and (b) using the all-to-all interaction which shows the summed trace of the three presynaptic spikes interacts with a later postsynaptic spike. The first presynaptic spike fires at time 10 and leaves a trace which is summed by the second spike trace and later the third spike trace at time 20 and 30. At the arrival of the postsynaptic spike (at time 35), the presynaptic trace after postsynaptic spike is read out. Eventually, the shaded area contributes to the change on state variable Δw and leads to the enhancement of memristor synapse.

5.5.2 Nearest-neighbour interaction

If the interactions are restricted, and therefore only the nearest spike pairs interact, this instance is classed as a nearest-neighbour spike interaction. All the pre- or postsynaptic spikes leave separate spike traces and all the traces will not be accumulated to produce the trace y_{pre} and y_{post}. To implement the nearest-neighbour spike interaction, the summation effect in Eq. (95) and (96) has to

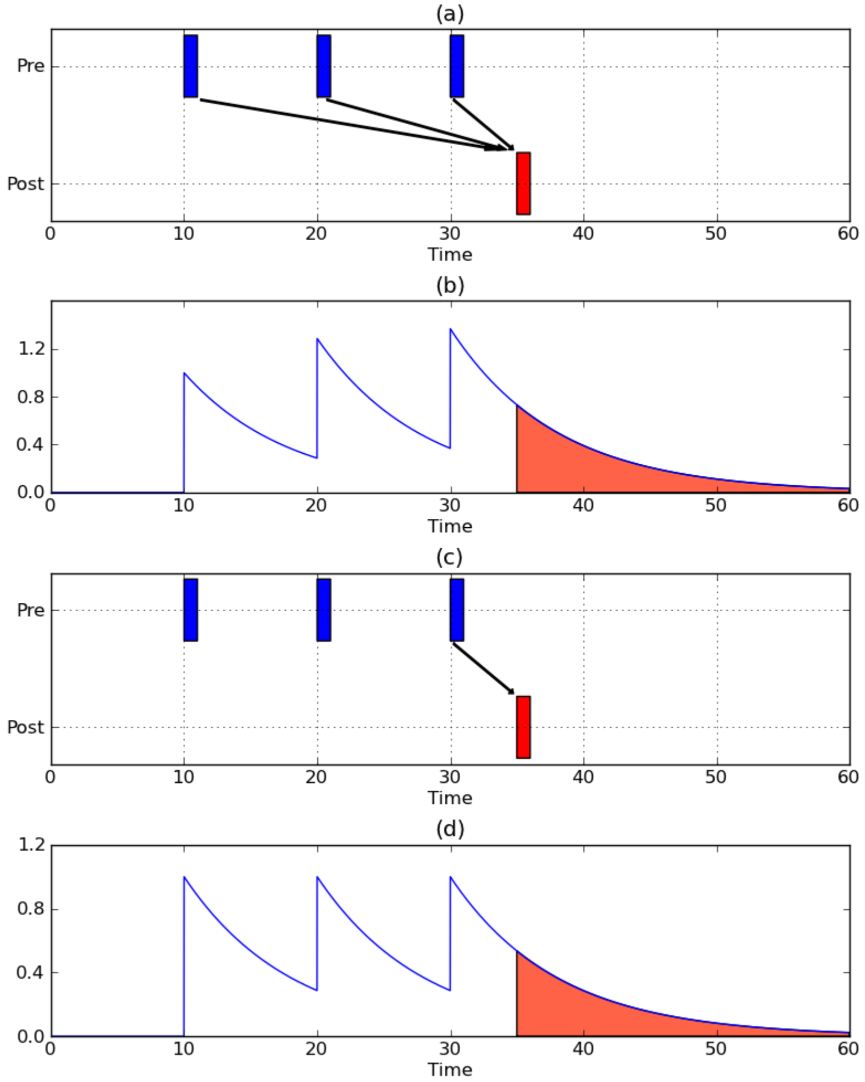

Figure 29: (a) All-to-all interaction. All the three presynaptic spikes (blue bars) at time 10, 20 and 30 interact with a postsynaptic spike (red bar) at time 35. (b) The presynaptic traces left by presynaptic spikes are accumulated and read out from the moment of the postsynaptic spike. The shaded area contributes to the change of state variable Δw and leads to a decrease in memristance (enhancement in synaptic strength). (c) Nearest-neighbour interaction. All the three presynaptic spikes (blue bars) leave a trace with a constant amplitude at time 10, 20, 30 without summation effect. (d) In this case, only the nearest presynaptic spike at 30 interacts with a postsynaptic spike (read bar) at 35 and its trace is read out from the moment of the postsynaptic spike. The shaded area contributes to LTP and a decrease in memristance (enhancement in synaptic strength).

be eliminated. Take the trace y_{pre} as an example; it is constructed by all the presynaptic spikes spk_i^+ with following conditions:

1. each presynaptic spike $spk_i^+(t)$ ends at the moment of the arrival of next spike $spk_{i+1}^+(t)$
2. the amplitude of presynaptic spike $spk_{i+1}^+(t)$ is constant and independent of the previous spike $spk_i^+(t)$

Hence, based on these conditions, the y_{pre} can be defined by

$$y_{pre} = \sum_{i=1}^{M} spk_i^+(t)H((t-t_i)(t_{i+1}-t)) \tag{97}$$

where $H((t-t_i)(t_{i+1}-t))$ is the Heaviside step function

$$H(n) = \begin{cases} 0 & \text{if} \quad n < 0 \\ 1 & \text{if} \quad n \geq 0 \end{cases} \tag{98}$$

As a result, each presynaptic spike has a constant amplitude and is only activated before the arrival of the next presynaptic spike. When the postsynaptic spike arrives at time t_j, it only interacts with the nearest presynaptic spike before it.

Similarly, the trace y_{post} of postsynaptic spikes is defined by

$$y_{post} = \sum_{j=1}^{N} spk_j^+(t)H\left((t-t_j)(t_{j+1}-t)\right) \tag{99}$$

When the presynaptic spike arrives at time t_i, it only interacts with the nearest postsynaptic spike before it and the trace y_{post} will be read out from the moment of the presynaptic spike until the next postsynaptic spike arrives. An example of nearest-neighbour interaction to produce LTP is demonstrated in Fig. 29(c) and (d). In this case, because there is no summation effect in producing the trace y_{pre}, only the nearest presynaptic spike at 30 actually interacts with postsynaptic spike at 35. Then the trace is read out from the moment of the postsynaptic spike and applied to the memristor. The shaded area is independent of other presynaptic spikes and contributes to the enhancement in synaptic strength.

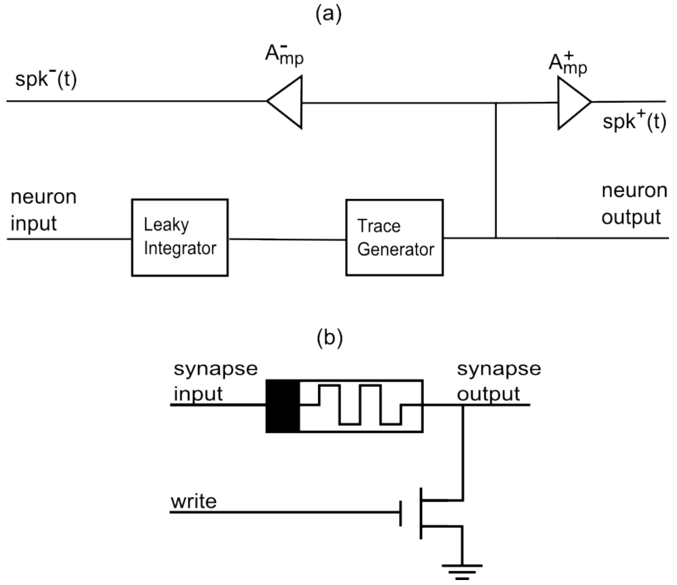

Figure 30: A schematic diagram of the basic units of the trace based memristive STDP model and the memristive synapse. (a) pre- and postsynaptic trace is delivered through the $spk^+(t)$ and $spk^-(t)$ terminals. (b) The memristive synapse is programmed when the *write* terminal is enabled.

5.6 The memristive neuromorphic system

We have introduced in previous sections a trace based memristive STDP model which is compatible with both the all-to-all and nearest-neighbour interactions. Neural networks implementations containing memristors are believed to be feasible with a comprehensive rule. Some neuromorphic structures have been proposed in [1, 63, 66] using memristive STDP models with similar neuron circuits. However, the proposed structures are not fully compatible with the trace based memristive STDP model because of the back-propagation concept in their models. Based on their studies, several changes are required to build a new neuromorphic structure which works with the trace based model. In this section, the possible means to apply the new memristive STDP model to the spiking neural network (SNN) are explored.

5.6.1 The neuron circuit

Since the SNN uses spiking neurons to produce spike signals which are more biologically plausible, firstly a neural block is needed to integrate the presynaptic spikes until the threshold is reached. At the moment of reaching the threshold, a postsynaptic spike should be fired and leaves a trace as proposed in the trace based model. By this means, it implements a simple leaky integrate-and-fire neuron. A possible structure for the basic unit is shown in Fig. 30(a). When the trace generator is triggered by the leaky integrator, a spike trace is generated and delivered to the neuron output. Meanwhile, the trace is multiplied by the factor A^{-}_{mp} and A^{+}_{mp}, subsequently, the pre- and postsynaptic traces are obtained.

This structure can be further arranged to be compatible with current neuromorphic systems. To this end, the original spike shape of the existing neuron circuit is preserved instead of using the proposed trace. The proposed model is purely a model used to monitor the spikes and the timing, generate pre- and postsynaptic traces and change the memristance according to the case of LTP or LTD. It is unnecessary to build a trace generator and have many modifications in existing neuron circuits.

5.6.2 The memristive synapse

The presynaptic spikes are delivered through the memristive synapse shown in Fig. 30(b). Only when LTP or LTD is induced, will the *write* terminal be enabled and the expected trace be applied to the memristor through the input. In such a circuit, it is important to ensure that the current from a synapse does not spread to its neighbours. This can be achieved by placing diodes on the right side of the memristor and before the postsynaptic neuron. To avoid the spike trace influencing the neuron circuit when updating the memristance, a diode can be placed before the synaptic input.

5.6.3 The architecture

For the purpose of building a network, the possible architecture can be arranged by using the memristor crossbar structure. This structure has been used by [53, 97] to fabricate the very high density memristor array because of the simple structure of the memristor. Based on the crossbar structure, the hybrid networks have been proposed in [13, 96, 104] which demonstrate the integrated circuits with memristors and CMOS components. The hybrid circuit uses the concept of the CMOL

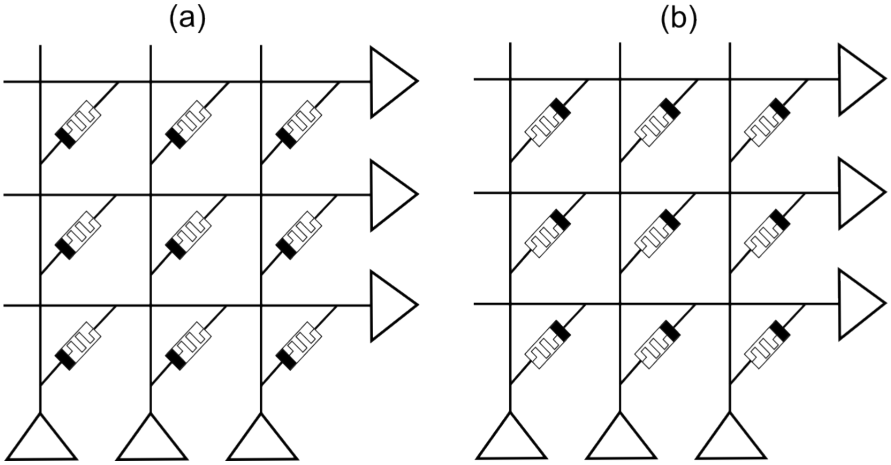

Figure 31: A schematic diagram of the architecture of the memristive neuromorphic system. (a) A neuromorphic structure with two layers of neurons implements the standard STDP. (b) A neuromorphic structure with two layers of neurons implements the anti-STDP.

(standing for CMOS + MOLecular scale devices hybrid) which builds a three-dimensional circuit. Typically, it consists of two layers which are the memristor crossbar layer and the CMOS layer. The memristor crossbar layer is arranged on the top of the CMOS layer, which could be further expanded from one to several layers. In this case, the schematic diagrams of two possible implementations of the standard STDP and the anti-STDP are illustrated in Fig. 31. For instance, in Fig. 31(a), a two-layer and 3 × 3 network is implemented based on the memristor crossbar structure. By reversing the polarity of memristors, shown in Fig. 31(b), the anti-STDP can be achieved as introduced in Section 5.4.2. Since the neurons are made by the conventional CMOS technology, they can be arranged underneath the memristor crossbar following the CMOL-like arrangement. The presynaptic neurons connect the positive terminals of memristors in the crossbar layer and then the negative terminals connect back to the postsynaptic neurons in the CMOS layer. Multiple-layer networks can be realised conveniently by repeating this connecting process.

Another possible scheme to implement a neuromorphic structure with the proposed memristive STDP model is the similar method used in [54, 94]. By this means, the presynaptic trace, the postsynaptic trace and the read-out trace will be modulated into logical signals. Based on the trace and the pre- and postsynaptic spikes timings, the modulated read-out trace will be applied to the memristors, and therefore implementing the STDP with the proposed model. However, these are only possible implementations giving a hint as to how to construct a neuromorphic system. There might be multiple different circuit implementations possible based on the above mentioned methods to realise the STDP with the proposed model but to explore this further is beyond the scope of this research.

Chapter 6

Computational Results for Memristive Spiking Neural Networks

In this chapter, the proposed trace based memristive STDP model, described in Chapter 5, is implemented and assessed empirically against the original memristive STDP model, namely the model based on the spike shape of the action potential [66, 67]. Both the nearest-neighbour spike interaction and the all-to-all spike interaction are applied to the proposed memristive STDP model by implementing small scale neural networks with spiking neurons and memristors.

The aim of the experiments is to firstly implement the all-to-all interaction and the nearest-neighbour interaction which are two important interaction schemes in standard STDP learning. Associative memory based on memristive neural networks is simulated with these two interaction schemes, given that the proposed memristive STDP model is able to cope with the basic multiple-to-one, one-to-multiple and one-to-one interactions. Secondly, to assess how the proposed model compares to the original model with respect to its all-to-all interaction. The original model has been studied from the theory to the circuit simulations which use the memristive FET-like devices to build a 4×4 feed forward memristive perceptron network [67]. It has become a significant work in the field of memristive STDP learning, and it has inspired the construction of variations to deal with various STDP learnings [15, 87]. Besides, the supervised STDP learning is discussed to demonstrate the benefit of the proposed model in supervised learning.

The remainder of this chapter is organised as follows. We explore associative memory which is based on memristive neural networks by implementing both the all-to-all and nearest-neighbour interactions. This experiment demonstrates that the proposed model can mimic the famous Pavlovian experiment on conditional reflex. The delayed-switching effect is discussed to show its role in such learning networks. The proposed model is later compared to the original model in terms of the all-to-all interaction which is based on the basic one-to-all and all-to-one interactions. Its compatibility is demonstrated through the comparison. Then, to conclude, we will implement the supervised learning in memristive spiking neural networks using the proposed model.

6.1 Associative memory with trace-based memristive STDP models

In Chapter 5, we demonstrated that the memristor indeed can implement a synapse with a proper learning rule. Its behaviour can easily be tuned to function as various STDP learning windows. For the purpose of this section, we have built a simplified memristive spiking neural network as shown in Fig. 32 based on the network used in [80, 99]. We then show that, with the proposed model, such networks are capable of simulating associative memory.

6.1.1 The network setup

As shown in Fig. 32, we consider a simplified neural network which comprises three neurons (two presynaptic neurons and one postsynaptic neuron) and the pre- and postsynaptic neurons are coupled by two memristive synapses (S1 and S2). As an example of the functionality that this network can provide, we can think about the Pavlovian experiment on the conditional reflex, in which the dog learns to associate a sound with food and eventually salivate without the intervention of vision. The first presynaptic neuron N1 (presumably located in the visual cortex) activates under various specific external stimuli namely "sight of food". The second presynaptic neuron N2 (presumably located in the auditory cortex) activates under certain external auditory events such as the "sound of a bell". The postsynaptic neuron N3 activates depending on previous training and

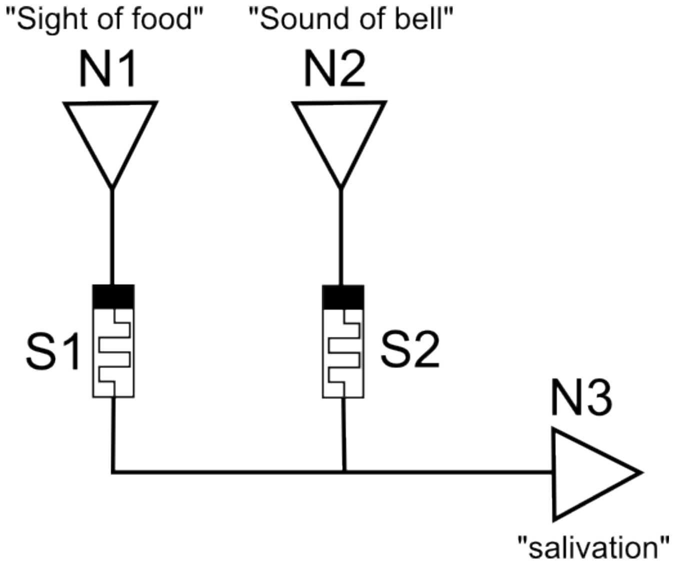

Figure 32: A spiking neural network using memristors as synapses.

leads to the “salivation” of the dog. However, in the beginning, only the “sight of food” can trigger “salivation” which means the “sight of food” and the “sound of a bell” are irrelevant events.

In order to associate the “sight of food” and the “sound of a bell”, certain conditions are fulfilled:

1. the “sight of food” and “sound of a bell” must be coexistent in time
2. the previous condition must be repeated several times
3. the “sound of a bell” should somewhat precede the “sight of food”

After a sufficient number of repeats of the above conditions simultaneously, the network should start establishing the association of the “sound of a bell” and the “sight of food” and eventually the “sound of a bell” can trigger “salivation” without the presence of food. This process of learning can be considered as a typical realisation of the famous Hebbian learning and can be further applied to standard STDP learning.

The spiking neuron

A classic and efficient spiking neuron is a leaky integrate-and-fire (LIF) neuron model which normally consists of a capacitor C in parallel with a resistor R. The

LIF neuron model is driven by the current I. This circuit can be described as

$$\tau_m \frac{du}{dt} = -u(t) + RI(t) \tag{100}$$

where the term τ_m denotes the time constant of the leaky integrator and is defined by

$$\tau_m = RC \tag{101}$$

The input current I is integrated by this RC circuit, and a spike is fired when the threshold voltage V_{th} is reached. Then the voltage u is reset to zero, and after an absolute refractory period t_{rp} the capacitor begins integrating the input current again. Usually, the voltage u refers to the membrane potential, and the time constant τ_m refers to the membrane time constant of the neuron. In this classic LIF neuron model, the form of an action potential is not described explicitly because it will be used to trigger the proposed trace based memristive model.

The memristive synapse

Memristors S1 and S2 in this experiment determine the connection strength between the presynaptic and postsynaptic neurons. Two flux-controlled memristor models with boundary (given in Eq. (102) and (103)) are applied to the network, which means the conductance of memristors are bounded between G_{off} and G_{on}.

$$G = (1 - s)G_{off} + sG_{on} \tag{102}$$

where the state variable s is defined by

$$s = \frac{1}{1 + e^{-\varphi k_p + \rho}} \tag{103}$$

In the beginning, the conductance of the memristive synapse S1 is inherently high (G_{on}), however, the memristive synapse S2 has a low conductance (G_{off}). During the experiment, the conductance of synapse S1 is constant since the strong connection is innate. Synapse S2 should be trained by the proposed model to build the connection between the "sound of a bell" and "salivation".

6.1.2 Spike interaction algorithms

Nearest-neighbour interaction algorithm

In order to discover the nearest spike pair, the timing of all the spikes are monitored and recorded. Algorithm 6.1 presents a high-level pseudocode of the nearest-neighbour interaction procedure employed in the network. The procedure starts with a list of the timings of all the presynaptic spikes and generates the presynaptic trace. At each iteration, if a spike arrives, the pre- or postsynaptic trace will be updated to a constant amplitude by $GeneratePreTrace()$ or $GeneratePostTrace()$. These two methods are responsible for generating the spike trace described in the proposed model.

If a presynaptic spike arrives just after a postsynaptic spike which is waiting for a presynaptic spike, the postsynaptic trace will be passed to the method *GetUpdateW*(). Then, in this method, the postsynaptic trace will be read out to change the conductance of the memristor. After the arrival of each presynaptic spike, the $wait\_for\_post$ is set to “1”, and a similar setting applies to the postsynaptic spike which sets $wait\_for\_pre$ to “1”.

All-to-all interaction algorithm

Similar to nearest-neighbour interaction, in order to implement the all-to-all interaction, the timing of all the spikes has to be monitored and recorded. Algorithm 6.2 presents a high-level pseudocode of the all-to-all interaction procedure employed in the network. The procedure starts with a list of the timings of all the presynaptic spikes and generates the presynaptic trace. Pre- and postsynaptic spikes are monitored at each iteration, and hence if a spike arrives, a spike trace will be added to the previous spike traces. From the moment of the presynaptic spike, the postsynaptic trace will be read out and passed to the method *GetUpdateW*(). Consequently, the conductance of the memristor will be changed accordingly.

6.1.3 The parameters

The same parameters are used for both the nearest-neighbour and all-to-all interaction experiments which consist of LIF neurons, the memristive synapses, and

Algorithm 6.1 High-level pseudocode of the nearest-neighbour spike interaction procedure employed in the simulation

Input: *the timing of all presynaptic spikes*

Output: *the update Δw on the memristor*

```
procedure NEAREST_ALG($pre\_timing, sim\_time$)
    Begin
        $t \leftarrow 0$
        $wait\_for\_pre \leftarrow 0$
        $wait\_for\_post \leftarrow 0$
        while $t < sim\_time$ do
            if $t == pre\_timing$ then
                $pre\_trace \leftarrow GeneratePreTrace()$
                if $wait\_for\_pre == 1$ then
                    $GetUpdateW(post\_trace)$
                    $wait\_for\_pre = 0$
                end if
                $wait\_for\_post = 1$
            end if
            $post\_spike \leftarrow UpdateResult()$
            if $post\_spike$ then
                $post\_trace \leftarrow GeneratePostTrace()$
                if $wait\_for\_post == 1$ then
                    $GetUpdateW(pre\_trace)$
                    $wait\_for\_post = 0$
                end if
                $wait\_for\_pre = 1$
            end if
        end while
    End
end procedure
```

Algorithm 6.2 High-level pseudocode of the all-to-all spike interaction procedure employed in the simulation

Input: *the timing of all presynaptic spikes*

Output: *the update Δw on the memristor*

```
procedure ALL2ALL_ALG($pre\_timing, sim\_time$)
    Begin
        $t \leftarrow 0$
        while $t < sim\_time$ do
            if $t == pre\_timing$ then
                $pre\_trace \leftarrow pre\_trace + GeneratePreTrace()$
                if $post\_trace! = 0$ then
                    $GetUpdateW(post\_trace)$
                end if
            end if
            $post\_spike \leftarrow UpdateResult()$
            if $post\_spike$ then
                $post\_trace \leftarrow post\_trace + GeneratePostTrace()$
                if $pre\_trace! = 0$ then
                    $GetUpdateW(pre\_trace)$
                end if
            end if
        end while
    End
end procedure
```

stimuli. Parameter details are shown in Table 9. Typical values of parameters were chosen with reference to [77] for the simulation components. The memristor synapse uses Eq. 102 and 103 which models a general HP memristor model with the original window function proposed in [97]. The chosen parameters can be tuned to memristors with different physical properties. However, the OFF/ON ratio should be sufficiently large. Thus, when the memristor is at low conductance, the presynaptic neuron is not able to fire the postsynaptic neuron. The high conductance should be large enough such that the presynaptic neuron is able to fire the postsynaptic neuron.

6.2 Simulation results

In this section, simulation results of both nearest-neighbour and all-to-all interactions are given with simulation procedures which are divided into three phases: Probing phase 1, Training phase and Probing phase 2. Since the experimental setups are the same for both nearest-neighbour and all-to-all interactions, the results will be discussed together.

6.2.1 Probing phase 1

In probing phase 1, the stimulus was firstly and only applied to the "sight of food" neuron N1 and then only applied to the "sound of a bell" neuron N2, which means firing periods of N1 and N2 are not overlapped. The purpose of this phase is to ensure that:

1. the "salivation" neuron N3 can be activated by the "sight of food" neuron N1 before training
2. neuron N3 cannot be activated by the "sound of a bell" neuron N2 before training
3. no association between "food" and "sound"

From the results shown in Fig. 33(c) and 34(c), before training, the "salivation" neuron fired only when the "sight of food" neuron was activated which

Simulation Summary

Simulation Time:	$T = 1000\ ms$
Time resolution:	$dt = 0.05\ ms$

Neural Model

Type description:	Leaky integrate-and-fire (LIF) neuron
Parameters:	Membrane time constant $\tau_m = 10\ ms$
	Membrane resistance $R = 1\ M\Omega$
	Spike threshold $v_{th} = 0.5\ V$
	Reset potential $v_0 = 0\ V$
	Refractory period $\tau_m = 4\ ms$

Synaptic Model

Type description:	Flux-controlled memristor
Parameters:	High conductance $G_{on} = 7\ \Omega^{-1}$
	Low conductance $G_{off} = 2\ \Omega^{-1}$
	Propagation factor $k_p = 1$
	Constant $\rho = 7$
	The state variable $s \in [0, 1]$

Inputs of N1 and N2

Type:	Constant stimulus $st = 5\ \mu A$

Table 9: Descriptions and parameters of the experimental setup

absolutely satisfies all the conditions mentioned above. Because the synapse S1 is always at the high conductance G_{on} but the synapse S2 is at the low conductance G_{off} initially which disconnects the neurons N2 and N3. Hence, before training, the connection between "sound of a bell" and "salivation" cannot be enhanced without postsynaptic spikes and LTP when only the neuron N2 is activated.

6.2.2 Training phase

During the training phase, stimuli were simultaneously applied to both of the neurons N1 and N2. Because neuron N3 was activated by neuron N1, the postsynaptic spikes were observed. This phase aims to:

1. stimulate the "sight of food" and "sound of a bell" neurons together
2. build the connection between the neurons N2 and N3 by applying the proposed model
3. train the network to learn the association between "food" and "sound"

In the proposed model, these aims can be achieved by overlapping the firing periods of the neurons N1 and N2. Since the long overlapping period implies that these two events are correlated, it is critical to build the correlation. As a result, the activation of neuron N3 led to the time difference Δt between the postsynaptic spikes and the presynaptic spikes from neuron N2. In the training phase, the postsynaptic spikes mostly arrived after the presynaptic spikes from neuron N2 with a positive time interval, thus LTP was induced and connection strength between the "sound of a bell" and "salivation" was enhanced. The conductance of the memristive synapse S2 was increased according to the STDP leaning and the proposed model. From the results shown in Fig. 33(d) and 34(d), in both spike interaction schemes, the enhancement was repeated many times which is sufficient to switch the memristor from G_{off} to G_{on}. Consequently, the connection between the neurons N2 and N3 became very strong as shown in the figures mentioned previously. At the end of the training, we can assume that the overlapped firing periods of the neurons N1 and N2 enabled the network to associate the "sight of food" and the "sound of a bell".

6.2.3 Validation and results

After the training, probing phase 2 was applied to the network in order to validate the assumption that was stated at the end of the training phase by:

1. stopping stimulating the “sight of food” neuron N1

2. stimulating the “sound of a bell” neuron N2

3. recording spikes from the “salivation” neuron N3

As shown in Fig. 33(c) and 34(c), in both nearest-neighbour and all-to-all interactions, neuron N3 was activated by neuron N2 without the help of neuron N1. In other words, a strong connection between the neurons N2 and N3 was established. Without the stimulation of “food”, the network presumed that “food” would be presented because of the stimulation of “sound” and began to produce the “salivation” spikes , and therefore we can confirm that the association between the “sight of food” and the “sound of a bell” was developed by the proposed model and proved by the results. The Pavlovian experiment was clearly reproduced by this simplified memristive spiking neural network.

It is worth noting that, according to the Fig. 33(d) and 34(d), the increment of the conductance G of the memristive synapse S2 is different although both of the experiments have the same simulation parameters. The increment of S2 in the all-to-all interaction is significantly sharper than the increment in the nearest-neighbour interaction since it rose to the highest conductance G_{on} rapidly before $400ms$. The difference on the increment of S2, to some extent, reveals the basic distinction between the two interaction schemes.

In the all-to-all interaction, since all the previous presynaptic spikes contribute to the postsynaptic spike directly or indirectly (and vice versa), the history of the neuron spikes play an important role in the interactions. By the means of summing all the traces of the previous presynaptic spikes, the timing-based plasticity (herein the STDP) is linked with the rate-based plasticity more tightly than the nearest-neighbour interaction. As a result, a higher frequency of spikes with a right timing pattern is more likely to induce LTP or LTD with a larger synaptic modification which can be reproduced by the proposed memristive STDP model. The tight collaboration is important in studying synaptic plasticity since rate-based plasticity is critical in explaining the learning of the brain [90] and was used to model the development of orientation tuning and ocular dominance in cat visual cortex by Bienenstock, Cooper and Munro [8]. In addition, from the previous research on synaptic activity [30, 58, 108], it is generally believed that long-term plasticity makes a fair contribution to the refinement of neural networks during their development. This implies that the development of neural networks relies on the participation of several synaptic plasticity rules of involved neurons and their history of activity. Therefore, to some extent, applying the proposed model with the all-to-all interaction scheme to the current neuromorphic system is important for applications which study the development of the neuronal circuits, and it is beneficial to current neuromorphic circuits through the application of nano-scale memristors with a simpler structure.

Delayed-switching effect in the synapse

In both experiments shown in Fig. 33(d) and 34(d), the conductance of the memristor evolves from the low conductance G_{off} towards G_{on} and the switching takes place with a time delay in the memristor. The time delay, in fact, implies that the memristor possesses certain inertia which delays the switching process. In particular, if we assume the memristor is a piecewise linear memristor with only two states OFF and ON, there will be a significant delayed-switching effect as mentioned previously in these two experiments. According to our discoveries in [99], a high voltage will induce a shorter time delay which, from another view, explains the experiment results that the increment of conductance in the all-to-all interaction is sharper than in the nearest-neighbour interaction. However, in the original postulate of the memristor [25] and some existing models [97, 19], the discoveries also imply that a small voltage may vary the status of the memristor if a sufficiently long duration is applied. Therefore, in this case, the time delay may be overtaken and the synapse S2 will eventually change its state. Clearly this scenario violates the Hebbian and STDP learning which needs the pre- and postsynaptic neurons to fire together. To avoid violating the learning rule, small voltage thresholds in memristors will help the circuits work properly by introducing a programming voltage above the threshold and a reading voltage below the threshold. Hence the programming and reading operations can be separated to prevent mis-operation. Furthermore, for the purpose of changing the state of the memristive synapse, a sufficiently long training period is critical

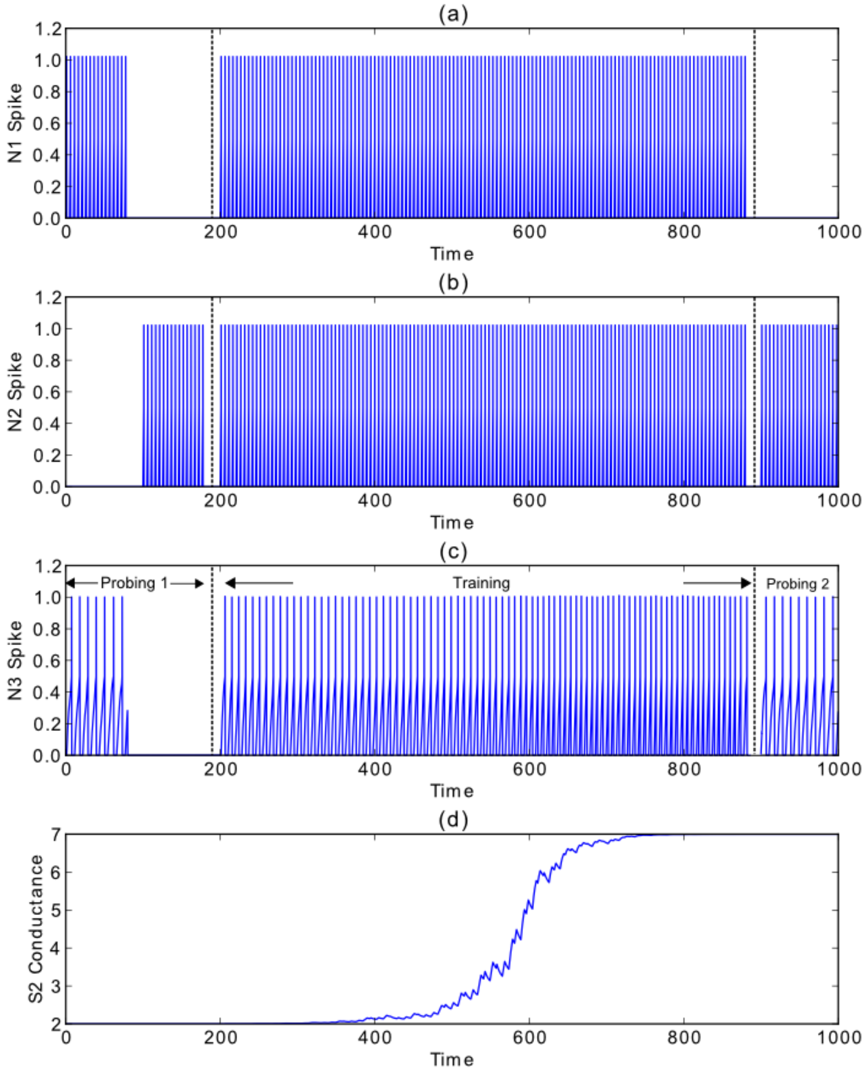

Figure 33: Simulation results of the proposed model with the nearest-neighbour interaction. (a) The output (spikes) of the “sight of food” neuron N1. It was immediately activated only at the beginning of the probing phase 1 and then activated again during the training phase. (b) The output of the “sound of a bell” neuron N2. It was activated in all the simulation phases for the purposes of probing and training. (c) The output of the “salivation” neuron N3. After the training, it cold be activated by the neuron N2 itself. (d) The modification of the conductance of the memristive synapse S2.

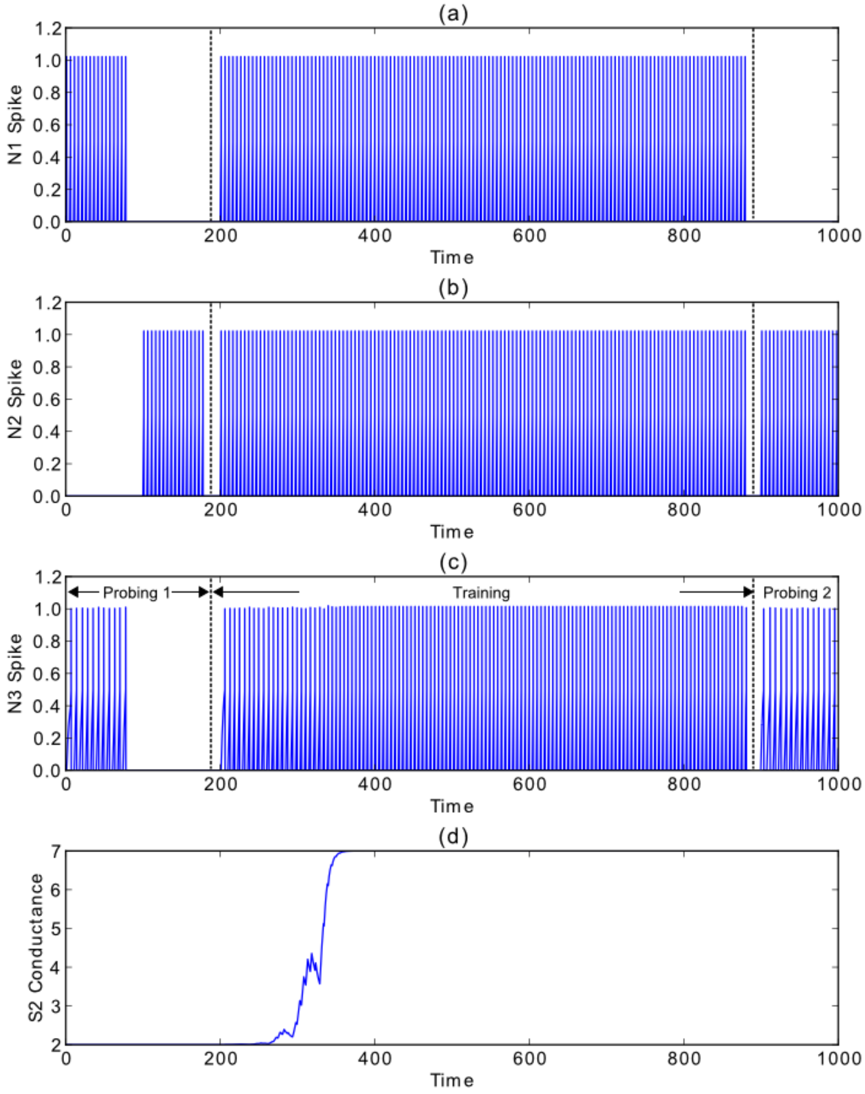

Figure 34: Simulation results of the proposed model with the all-to-all interaction. (a) The output (spikes) of the "sight of food" neuron N1. It was immediately activated only at the beginning of the probing phase 1 and then activated again during the training phase. (b) The output of the "sound of a bell" neuron N2. It was activated in all the simulation phases for the purposes of probing and training. (c) The output of the "salivation" neuron N3. After the training, it could be activated by neuron N2 without the help of N1. (d) The modification of the conductance of the memristive synapse S2 increased more sharply than the modification in the nearest-neighbour experiment.

otherwise the state will not be changed as expected, and it will lead to failed training. In conclusion, the delayed-switching effect should be carefully considered when training some memristive spiking neural networks such as the networks mentioned previously.

6.2.4 Retention loss

So far, the memristive spiking neural network with our proposed model has demonstrated associative learning using both nearest-neighbour and all-to-all interactions. Because of the non-volatile nature of the memristor, the training outcome will be retained for a long period of time even if the power is shut down. This means that once the association is built by the training, it will exist permanently, and therefore further training is unnecessary. Indeed, it is a desirable feature in neural networks which demand a long-term memory such as that of the brain. However, it does not demonstrate short-term memory or retention loss as seen in the networks of [80, 100], which is also critical for associative learning since the brain is unlikely to hold all information permanently. In Pavlov's classic experiment, another two conditions are required to be fulfilled:

1. reinforce the association periodically
2. lose the association after a certain time if only the "sound of a bell" is activated

The first condition implies that the association could be lost if it is not periodically reinforced, which indicates the exhibition of a somewhat short-term memory. The second condition suggests that, without activating the "sight of food" neuron N1, the stimulated "sound of a bell" neuron will lead to a penalty on the memristive synapse S2 therefore weakening the connection strength. After a sufficient time period, the connection will eventually be lost, and therefore the association between "food" and "sound" will be broken. Following these aforementioned conditions, it suggests that inhibition is induced if the "sight of food" neuron N1 is not participating in activating the "salivation" neuron N3. In order to achieve retention loss, we have to re-examine the results obtained in the previous simulations.

Since the joint activation of a neuron depends on multiple presynaptic neurons, the frequency of the postsynaptic spikes somewhat depend on the combination of presynaptic spikes with its presynaptic neurons. As shown in Fig. 33(c) and 34(c), when both of the presynaptic neurons N1 and N2 are fired together, the rate of the postsynaptic spike is higher than when only one presynaptic neuron is applied. Hence, we assume that the frequency of postsynaptic spikes can more or less indicate whether there is a joint contribution of neurons N1 and N2 in the given network. Thus, under the same strength of stimuli, the lower frequency has a larger possibility to inhibit the connection strength of S2 therefore weakening the association. Such an assumption recalls the discussion of the joint work of rate-based plasticity and timing-based plasticity. Therefore, a rate-based term is introduced to vary the amplitude of the postsynaptic trace

$$A_{mp}^{-} = e^{\frac{f_{post}-\rho}{-\tau_r}} \tag{104}$$

Hence the Eq. (81) of the postsynaptic spike trace could be altered as

$$spk_j^{-}(t) = \begin{cases} -e^{\frac{f_{post}-\rho}{-\tau_r}} \left(\frac{e^{(-t+t_j)/\tau^{-}} - e^{-t_{ail}^{-}/\tau^{-}}}{1-e^{-t_{ail}^{-}/\tau^{-}}} \right) & \text{if} \quad t_j < t < t_{ail}^{-} + t_j \\ 0 & \text{if} \quad otherwise \end{cases} \tag{105}$$

where ρ and τ_r are two constants which jointly determine the variation of the amplitude A_{mp}^{-} along the rate f_{post} of the postsynaptic spikes.

By introducing the rate-based term, the amplitude A_{mp}^{-} decreases when the rate f_{post} increases. Consequently, if only the stimulus of "sound of a bell" is applied, the postsynaptic spike trace amplitude A_{mp}^{-} will remain at a larger value. As a result, LTD has a larger impact on memristive synapse S2 than LTP.

Validation and results

A similar validation process to the previous experiments was used to evaluate the newly introduced term. It consists of three phases:

1. the probing phase
2. the training phase
3. the retention loss phase

As shown in Fig. 35, the modified nearest-neighbour interaction experiment, the probing phase and the training phase are the same as with the previous experiments which enhance the connection strength of S2. It shows that learning of the network was not affected by the rate-based term that was introduced and the network learned the association properly.

Retention loss phase

In the retention loss phase, during 900ms - 2000ms, the simulation time is longer than the training phase's since it takes more time to weaken the connection. When only the "sound of a bell" neuron N2 was fired, the frequency f_{post} of postsynaptic spikes was lower than in the training phase's. Although, at the beginning of the retention loss phase, neuron N3 still fired without the help of neuron N1, LTD began to take precedence and decreased the conductance of S2 gradually. At some point, the conductance began to decline significantly and eventually dropped to a value where the "salivation" neuron N3 stopped firing spikes. The results shown in Fig. 35(c) and (d) indicate that the proposed model with the newly introduced rate-based term is capable of reproducing all the conditions required by associative learning, in the form of both building the association and losing the association under certain circumstances.

Hence, all the outcomes of leaning and forgetting are preserved once the learning and forgetting processes terminate, which implies the outcomes are transformed into long-term memory which lasts for a much longer time period than the short-term memory. In order to implement short-term memory, we need a memristor that has a volatile feature to mimic retention loss with time. A possible and proper candidate is the nano-scale memristive device based on tungsten oxide which shows the typical memristor behaviours with a peculiar forgetting effect [17]. With this type of memristor, because the concentration gradient induced ion diffusion is opposite from the ion migration, it leads to a particular behaviour which resembles the forgetting curve in biological systems. The latest research [20] shows the conductivity of the memristor can constantly decline to different stable points from different initial conductivity, which resembles the transition from short-term memory to long-term memory because it may not lose all its conductivity. With these kinds of memristors which have forgetting effect, it is possible to refine our proposed memristive spiking neural network to mimic more advanced and desirable biological behaviours. Again, it demonstrates the variety and future potentials of memristive devices in neural networks from the aspect of their intrinsic characteristics which are similar to the biological behaviours.

6.3 Supervised learning with trace-based memristive STDP models

As mentioned previously, the STDP, in fact, is an unsupervised learning rule which is solely based on the timing of pre- and postsynaptic spikes, and it has been used to perform complex recognition tasks [43, 75]. In Section 6.1, it was shown that the STDP and proposed model enable associative learning using the memristive synapse. Moreover, the applications of unsupervised STDP learning with memristive devices are discussed in [54, 67, 87, 94], however, the supervised STDP learning with memristive devices has not been investigated. In fact, for specific task-oriented engineering applications that require an explicit goal definition, supervised learning might be more favourable than unsupervised learning. The supervised learning which normally involves an error back propagation process has been widely used in training conventional neural networks. Meanwhile, supervised learning in spiking neural networks has been developed using different approaches such as gradient descent [12], learning window [83] and Delta learning rule [77]. However, these rules are designed for training general spiking neural networks which are somewhat different from memristive spiking neural networks, especially in neuromorphic applications. It is required to find a proper supervised learning rule for the proposed memristive STDP model and its applications.

6.3.1 The network setup

In order to present supervised learning with the proposed memristive STDP model, a network will consist of 10 input neurons which produce input spike patterns and one LIF neuron which responds to all the inputs. The network is arranged in a feed-forward way which is similar to the simple perceptron network. Each input neuron is connected to the output LIF neuron through the memristive synapse thus all the input spikes are summed before sending them to the output neuron N_{out} as shown in Fig. 36. The task of the network is to learn

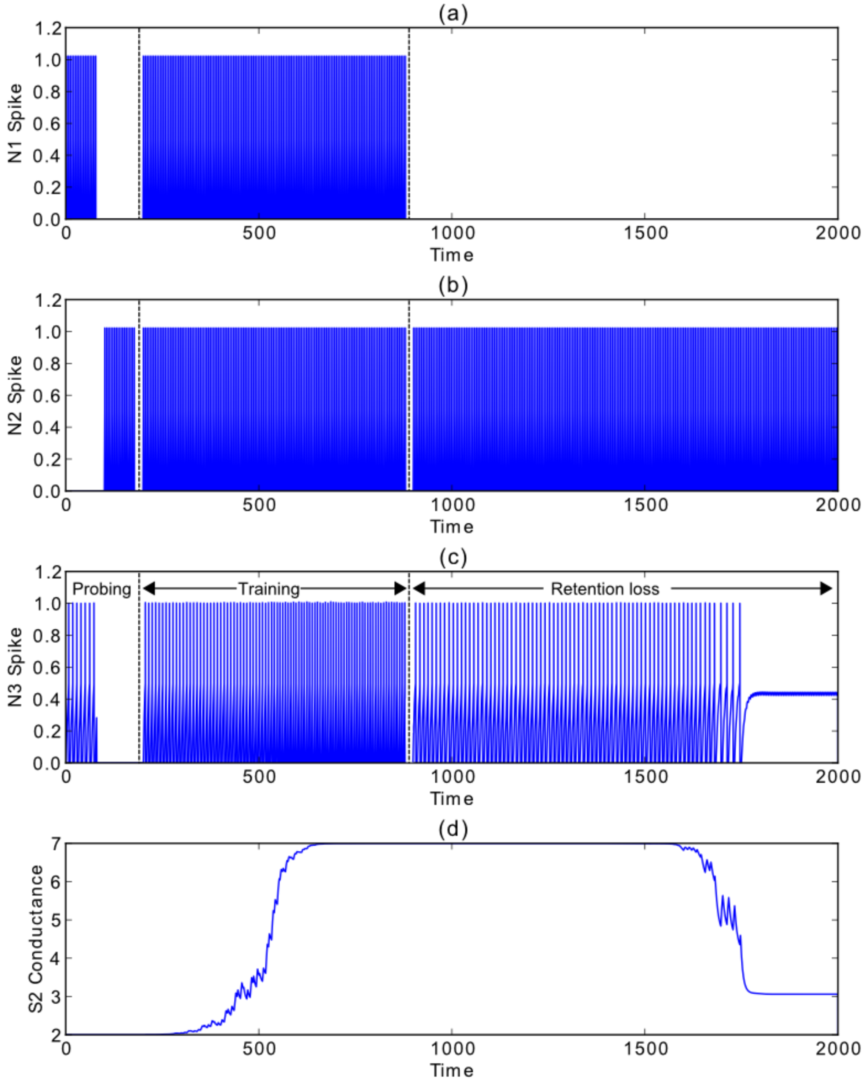

Figure 35: Simulation results of retention loss. (a) The output of the "sight of food" neuron N1. (b) The output of the "sound of a bell" neuron N2. (c) The output of the "salivation" neuron N3. After training, it could be activated by neuron N2 without the help of N1. (d) The strength of the memristive synapse S2 increases at the training phase but decreases at the retention loss phase. The terms used are: $\rho = 6.75$ and $\tau_r = 1.76$.

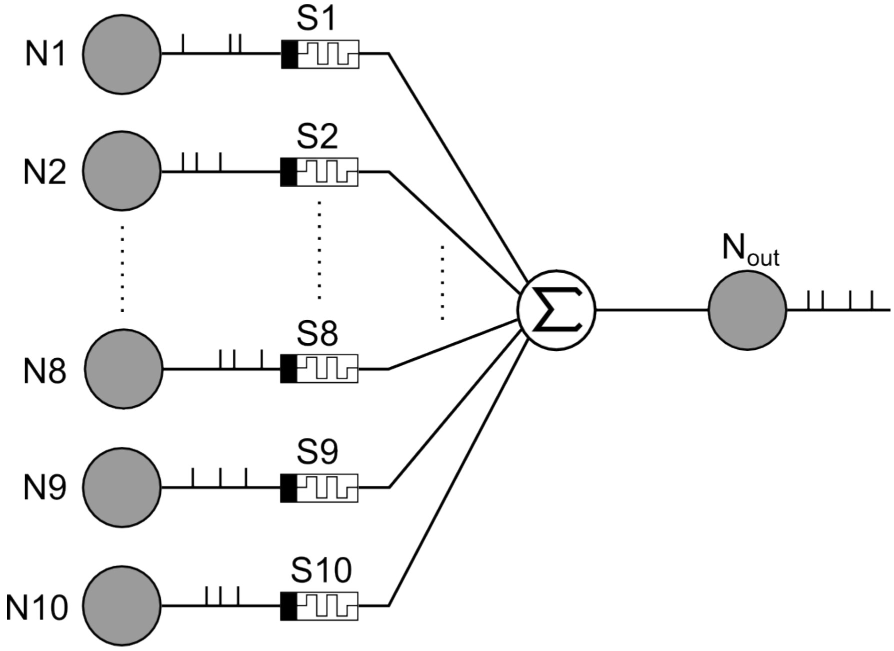

Figure 36: The memristive spiking neural network set for supervised learning.

the target spike pattern which is randomly generated. Initially, the connection strength, represented by the memristor's conductance $G_{mem}(w)$, is randomly generated within the range $G_{min} < G^{i}_{mem}(w) < G_{max}$ where w is the state variable which determines the update of the conductance G_{mem}.

6.3.2 The learning rule

Similar to other supervised learning rules, such as the spike pattern association neuron (SPAN), training a memristor-based spiking neural network requires the iteration of all the spike patterns. The connection strength is updated iteratively to approach the desired spike patterns by the predefined input spikes. Deploying this method, the same approach as with SPAN is followed, which adapts the Delta rule to train the memristor-based spiking neural network. The Delta rule is defined by

$$\Delta w = \alpha x_i (y_d - y_{out}) \tag{106}$$

where $(y_d - y_{out})$ can be represented by Δy which implies the distance between the desired and the actual outputs. x_i is the input of the presynaptic neuron i. It is designed for real-valued applications, and hence it is incompatible with the spike timings which usually are considered as the Dirac delta function.

As previously mentioned in Chapter 5, each spike will leave a trace with the shape of an exponential function at its arrival thus the spike trains which are implemented by the Dirac delta function are naturally transformed into spike traces. By defining some important terms the Delta rule used in SPAN can be adapted to train the memristor-based spiking neural network with the proposed model.

Error of the output

The error of the output is defined in Eq. (106), Δy, which is the difference between the target output and the actual output. In the trace-based memristive STDP model, the postsynaptic trace y_{post} is chosen to suit the Delta rule, and therefore the error of the output is defined by

$$E_{dr} = \Delta y = y^{d}_{post} - y^{r}_{post} \tag{107}$$

where d and r denote the target and real outputs respectively. It compares the real postsynaptic trace y^{r}_{post} to the desired postsynaptic trace y^{d}_{post} directly and can be used as one of the evaluation metrics to assess the performance of the supervised learning. A well trained and performed neural network normally produces a sufficiently small number.

Another metric of the output is the time difference between the desired postsynaptic spike δ_{post_d} and the real postsynaptic spike δ_{post_r}. However, it requires the sets δ_{post_d} and δ_{post_r} have the same size in terms of the number of postsynaptic spikes, otherwise it is inaccurate in comparing all the spikes in pair from the two sets. If the requirement of the size is satisfied, the error can be evaluated using root-mean-squared error (RMSE) which measures the average of the squares of the errors in the form of:

$$RMSE(\delta_{post}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\delta^{i}_{post\_d} - \delta^{i}_{post\_r})^2} \tag{108}$$

where the $\delta^{i}_{post\_d}$ and $\delta^{i}_{post\_r}$ denote the arrival time of the postsynaptic spike i in desired output and the real output respectively.

Error of the update of synaptic weight

The actual result of the supervised training should be exactly or approximately the same as the desired result, and hence the actual synaptic enhancement induced by LTP and the actual synaptic depression induced by LTD should be the same as the desired result in the best case. Thus, the difference between the actual and desired synaptic enhancement can be defined by

$$Ew_{LTP} = w^d_{LTP} - w^r_{LTP} \tag{109}$$

and similarly the error on synaptic depression is defined by

$$Ew_{LTD} = w^d_{LTD} - w^r_{LTD} \tag{110}$$

In fact, the Eq. (109) and (110) can be combined into one equation to represent the total error of the change of synaptic weight, however it is unnecessary in most cases because the error of the synaptic enhancement is capable of assessing the performance of the supervised learning. Firstly, the postsynaptic spike is induced by several presynaptic neurons which means there must be some neurons stimulating the postsynaptic neuron. Secondly, a direct result of the postsynaptic spike is the synaptic enhancement (LTP) if a presynaptic spike precedes it. Thirdly, the synaptic depression (LTD) is induced after the postsynaptic spike which implies generating precise postsynaptic spikes is more important.

6.3.3 Simulation and results

Simulation setup

The training and testing patterns are generated by the same network with constant connection weights. Each presynaptic neuron generates 10 presynaptic spikes at different times and the postsynaptic spikes are recorded and hence a pair of an input set and an output set is obtained. In an output set, each postsynaptic spike is considered as a class, and therefore the number of postsynaptic spikes in an output set denotes the number of the classes to be learned or verified. In this experiment, 8 desired output patterns are utilised and each pattern comprises at least 11 classes. Only one of them is used as the training pattern during training process, while the testing patterns are used to evaluate the generalisation ability of the trained network. For each training, it allows 50 epochs, and the training is repeated for 10 independent runs with different patterns.

Training

During the training, the memristive connections are programmed to random values and the postsynaptic neuron learns to fire at a specific time when certain patterns of the presynaptic spikes are generated by the presynaptic neurons. The desired postsynaptic trace y^d_{post} is compared to the actual postsynaptic trace y^r_{post}, and the result is represented by E_{dr}. According to the Delta rule, each presynaptic trace is multiplied by E_{dr} to get the required update trace V^i_{update}. Finally, the update trace V^i_{update} will be applied to the memristive synapse to change the connection strength. Hence the update of the memristor i is defined by

$$\Delta w^i = \alpha \int_0^T V^i_{update} = \alpha \int_0^T y^i_{pre} E_{dr} \tag{111}$$

where T is the simulation time of one iteration. By this way, the memristive synapse is updated after each iteration. Once a training set is finished, another training set will be utilised until all the training is finished. Therefore, the network is trained to minimise the output error E_{dr} by the first training set and then slightly adjusted by the following training sets. It has been found that this training approach provides an efficient way to converge the network, however, the first training patter may lead to a poor performance in some circumstances. If the training results are found to be well below expectation, changing the first training pattern can sometimes significantly improve the performance. Normally, this happens when the first training pattern is quite complex and dense, or the synaptic weights are not well distributed, which makes it complicated for the network to capture and learn in the beginning.

Evaluation of a single task

A single spike pattern was randomly generated for the purposes of training and testing in this task. The task is to learn mapping from the presynaptic spikes to the desired output set which contains 13 classes occurring at different times, for

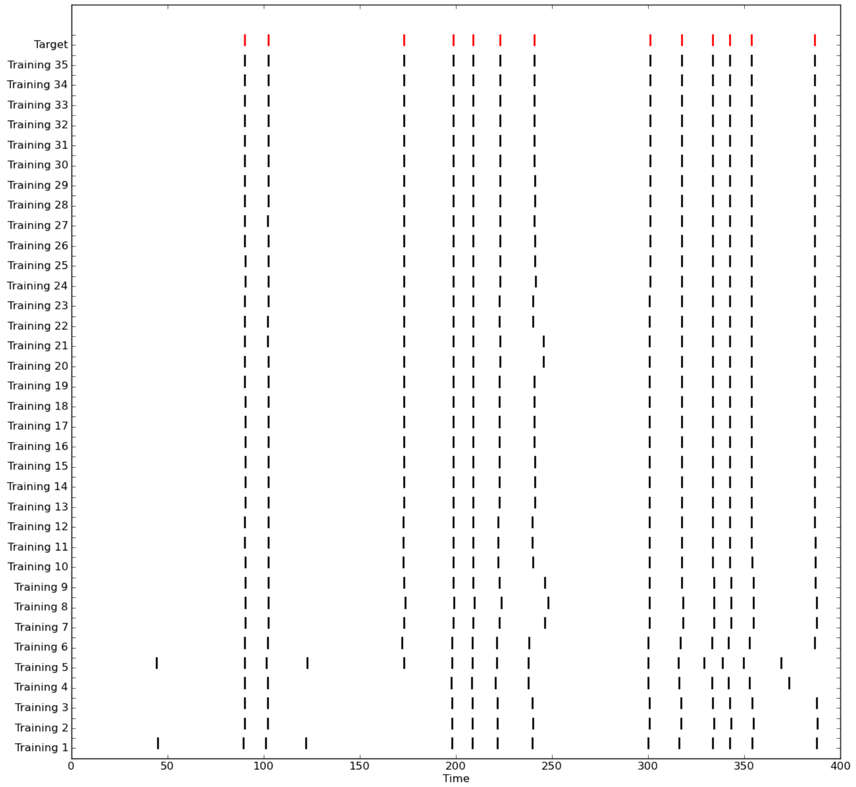

Figure 37: There are 35 epochs in the training and results of each epoch are compared with the target spikes (red bars at the top). It shows that actual postsynaptic spikes are trained to learn the target spikes.

example {90.0, 102.5, 173.0, 198.5, 209.0, 223.0, 240.5, 301.0, 317.5, 333.5, 342.5, 353.5, 386.5}ms. During training, the update of memristors are corrected by Eq. (111) using the differences between the desired output set and actual output spikes with a learning rate $\alpha = 0.3$. Fig. 37 demonstrates the learning process of the memristive spiking neural network towards desired output pattern through the learning rule in Eq. (111).

The desired postsynaptic spikes (red bars) at the top of Fig. 37 along with the postsynaptic spikes (black bars) are produced by the postsynaptic neuron. In the beginning, the timings of postsynaptic spikes are very different from the desired spike sequence, however, after several iterations the output spikes are approaching and moving towards the desired spike sequence. This has been validated by the evolution of the enhancement error Ew_{LTP} and the output error E_{dr} shown in Fig. 38(a) and (b) that clearly demonstrate the network learns and converges

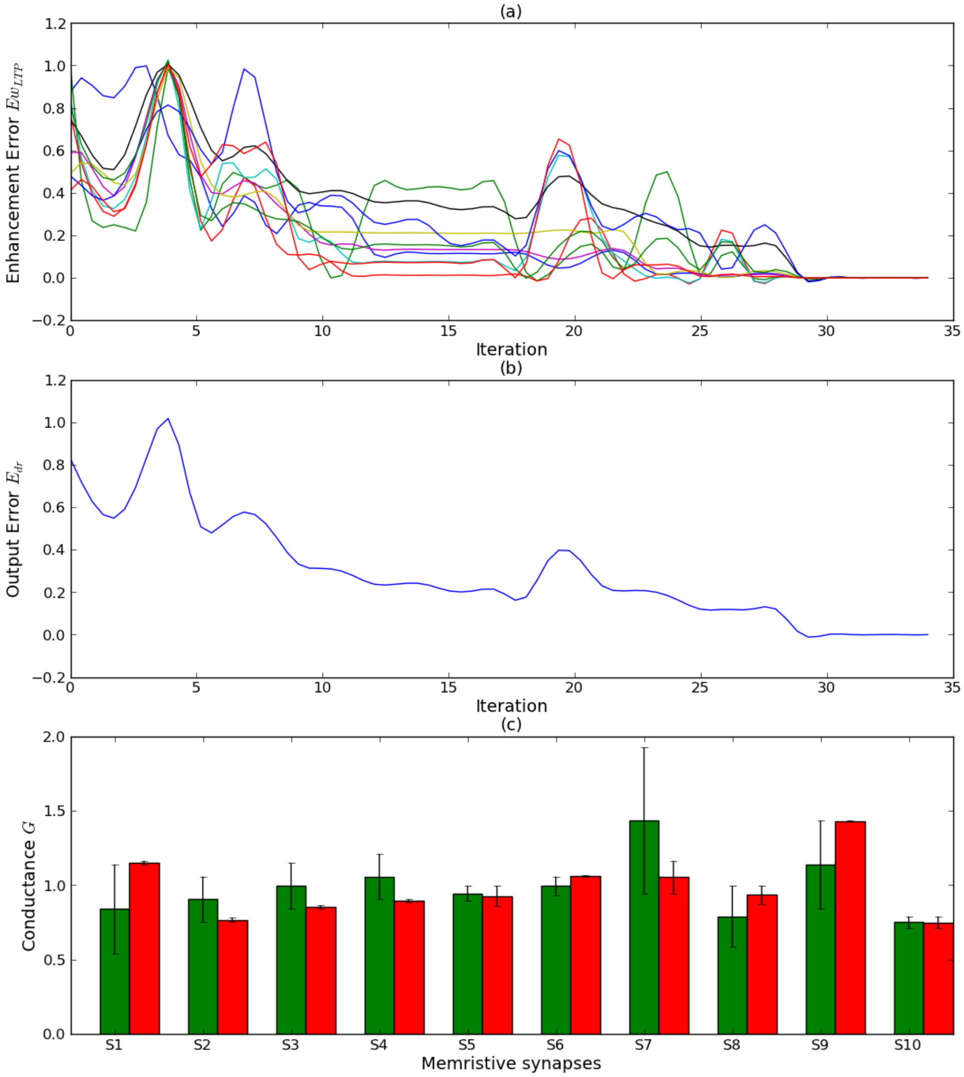

Figure 38: The evolution of the errors and synaptic weights. (a) The enhancement error of each connection is illustrated in different colours. It shows the tendency of the enhancement error along the training epochs. (b) The output error clearly converges to a small value, which means the network is well trained by the current training set. (c) The conductance of all the memristive synapses before training (green bars) and after training (red bars). The deviations are also illustrated to demonstrate the differences from the target synaptic weights.

to the desired output spike sequence. It is worth noting that the evolution of the enhancement errors Ew_{LTP} of the memristive synapses fluctuate a lot in contrast to the output error E_{dr}, which implies that the enhancement error Ew_{LTP} measures the influence of current postsynaptic spikes to the individual synaptic connection. Clearly, at a certain output error, the enhancement error of each synaptic connection is different because the enhancement error is very sensitive to each postsynaptic spike timing. As a result, a slight difference in the timings of some postsynaptic spikes may result in a large enhancement error, which makes the enhancement error very volatile during the learning. For a well trained network, most enhancement errors of synaptic connections should have only a small difference between them as shown in Fig. 38(a) after 25 iterations.

The conductance of memristive synapses before training (green bars) and after training (red bars) are illustrated in Fig. 38(c). After training, the deviations of the conductance of memristive synapses between the trained network and the desired network is significantly reduced in contrast to the network before training. Interestingly, Barbour et al. [4] studied synaptic plasticity from the view of synaptic weights distribution, and by analysing the measurements obtained from somatic recording, it shows that the analysis of weights distribution has the potential to become a powerful tool for studying the mechanisms of learning in the brain. Hence, the proposed model and learning rule have the optimistic potential to provide a platform to analyse the distribution of the conductance of memristors in real applications, which may also help biological studies as a complement.

Evaluation of multiple tasks

After the experiment of the single task, we investigate the abilities of the memristive spiking neural network to reproduce the desired spike patterns and timings. After the training process used in the single task, 7 testing patterns shown in Table 10 are utilised to probe the trained network, and the results are compared to the desired output set by the output error $RMSE(\delta_{post})$ and the class error $RMSE(\delta_{class})$ based on Eq. (108).

Since both of these two measurements adopt the approach of RMSE to determine the performance, in some worst-case scenarios, there is a problem that the number of spikes in the desired set and the actual set may be different. The best

Desired output patterns
Pattern 0:
{ 90.0, 102.5, 173.0, 198.5, 209.0, 223.0, 240.5, 301.0, 317.5, 333.5, 342.5, 353.5, 386.5 }
Pattern 1:
{ 43.5, 56.0, 69.5, 90.0, 107.0, 147.5, 163.0, 269.0, 290.0, 315.0, 342.5 }
Pattern 2:
{ 44.5, 63.5, 88.5, 105.0, 126.5, 146.0, 162.0, 271.0, 293.0, 312.5, 323.5, 352.0 }
Pattern 3:
{ 44.0, 60.5, 72.0, 90.5, 106.0, 146.0, 161.0, 271.5, 293.5, 316.0, 348.5 }
Pattern 4:
{ 44.0, 64.5, 89.0, 105.0, 131.5, 149.0, 164.0, 281.5, 298.5, 315.5, 343.5 }
Pattern 5:
{ 43.5, 56.5, 68.5, 89.0, 104.5, 129.5, 150.0, 164.0, 283.5, 301.5, 317.5, 350.5 }
Pattern 6:
{ 43.5, 56.0, 69.5, 90.0, 106.0, 129.5, 156.5, 283.0, 298.0, 315.5, 341.0 }
Pattern 7:
{ 44.0, 63.0, 89.0, 104.5, 145.5, 162.5, 265.5, 288.0, 313.0, 324.5, 350.5 }

Table 10: Desired output patterns used for evaluation of multiple tasks.

solution to overcome this issue is re-training the network by adjusting the learning rate α or epochs to achieve a satisfactory result. Another option is utilising the output error E_{dr} to measure the differences in outputs rather than utilising the difference in timings. When evaluating a single class, we avoid measuring the performance in terms of percentage accuracy since it is difficult to calculate the accuracy in temporal patterns. For example, in statistics, the percentage error is defined by

$$\delta = 100 \times \frac{|v_d - v_r|}{|v_d|} \tag{112}$$

It implies that the desired value v_d sometimes dominates the percentage especially in temporal patterns. If an output set {10, 30}ms is obtained from a trained network and the desire output set is {5, 25}ms, percentage errors are 100% and 20% respectively for the first spike and the second spike though the absolute errors of these two spikes are the same. To avoid the influence induced by the spike time, the absolute error is utilised in evaluating the trained network with the RMSE approach. If there are more classes in the actual output set than in the desired output set, the extra classes will be ignored and labelled as useless classes. Conversely, if there are less classes in the actual output set, the missed classes will be labelled as failed recognitions.

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Class RMSE
Class 1	0	-0.5	0	0	0	0	0	0.189
Class 2	-0.5	-1	1.5	0	0	0	-0.5	0.732
Class 3	-0.5	-0.5	1.5	0	0	0	0	0.627
Class 4	0	-1	0.5	0	-0.5	0	0	0.463
Class 5	0	-0.5	0.5	2	0	0	-1	0.886
Class 6	-0.5	0	-1	1	-0.5	-0.5	-1	0.732
Class 7	-0.5	0.5	-0.5	0.5	-2	-0.5	0	0.866
Class 8	-0.5	0.5	0.5	-1	-2	0	0	0.906
Class 9	-1	1.5	0.5	-0.5	-0.5	0	-0.5	0.779
Class 10	0	0.5	0	0	0.5	0	-1	0.463
Class 11	0.5	1	0	-5.5	5.5	-0.5	0	2.952
Class 12		0.5			1.5			1.118
Test RMSE	0.477	0.764	0.783	1.828	1.860	0.261	0.564	

Table 11: Results obtained by probing the trained network with testing spike patterns. Each actual class is evaluated in absolute error between the actual spike timing and target spike timings, and then the results of actual classes and tests are evaluated by RMSE in Section 6.3.2.

The analysis of the results shown in Table 11 suggest that the learning rule is capable of reproducing the desired testing output patterns within the average RMSE of 1ms. In each testing pattern, most classes are captured with errors within the range of $[-1, 1]$ms, and some classes are reproduced precisely with approximate error of zero. The largest errors appear in testing patterns 4 and 5 which both contain large errors in producing the class 11, however, other classes in these two testing patterns are well recognised. By comparing the performance in reproducing the classes, the worst cases were seen in class 11 because of the large errors caused in testing patterns 4 and 5. Except for the class 11, other classes are well reproduced, and the RMSE errors of the classes are also well distributed with a mean class error below 1ms. From results of testing patterns and the classes, the network is not well trained for testing patterns 4 and 5, which lead to large errors in test RMSE and class RMSE. If these two patterns are taken out from testing results, all the results are well distributed around the mean error. It implies that the network is not well trained by the training patterns, and hence some specific testing patterns and classes can not be correctly reproduced. However, it is capable of proving that, by training the network with more training patterns, the learning rule can lead the memristive spiking neural network to converge to an acceptable mean error both in spike patterns and individual classes.

6.3.4 Neuromorphic implementation

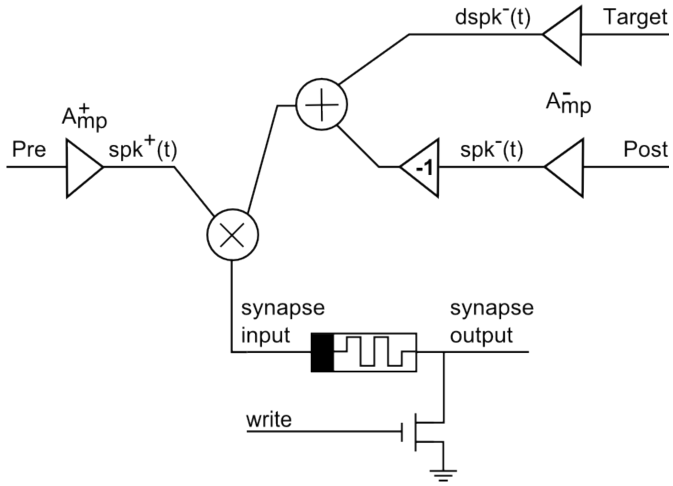

Figure 39: A schematic diagram of the possible neuromorphic structure of the supervised network.

A possible supervised learning rule was introduced previously, which can be adopted to the proposed memristive STDP model with a simpler neuromorphic structure compared with other learning rules. In this section, we show a possible scheme of its neuromorphic implementation. As shown in Fig. 39, additional elements are required to implement the supervised learning rule, however, still based on the spike trace model. During the training, the postsynaptic trace $spk^{-}(t)$ will be inverted and then summed with the target spike trace $dspk^{-}(t)$ by the summation circuits. The summation result will be multiplied by the presynaptic trace $spk^{+}(t)$ through the multiplication circuits. At last, the processed signal V^{i}_{update} will be applied to the memristor to update its conductance. Therefore, the connection strength will be updated, and the learning of the current connection will be complete.

As shown in Fig. 39 and the update processes, mainly three additional circuits (inverter circuit, addition circuit and multiplication circuit) are required to implement the supervised learning compared to the neuromorphic structure proposed in Chapter 5. Therefore, it is more applicable than implementing supervised learning based on the complex gradient approach which normally involves intensive computations in solving partial differential equations.

Chapter 7

Conclusions and Future Research

The research described in this thesis has presented studies of memristor-based neural networks of a wide range, from memristor-based models, structure to their potential applications. Several problems that have arisen with this topic have been addressed. The results presented in this work indicate that memristors are well-suited for neural networks and the neuromorphic systems. Memristor-based neural networks overcome limitations of purely CMOS-based neuromorphic systems in terms of size, complexity and power consumption. We first made a review on the fundamentals of neural networks and memristors. Memristors are studied from theories and physical properties to widely used models. The basic concepts, approaches and current research on memristor-based neural networks are introduced. The main context of this thesis has divided the studies into two separate topics: (1) memristor-based non-spiking neural networks which focus on unexplored research areas of staircase memristors and the memristor-based structures; (2) memristor-based spiking neural networks which focus on STDP learning with all-to-all spike interaction.

In the context of memristor-based non-spiking neural networks, this thesis has studied the staircase memristor which has a wavy variation on memristance. Given the comparison results of staircase memristors and analog memristors, staircase memristors are more stable in terms of errors between actual and expected memristance with same pulse signals. Furthermore, memristor-based CNNs and ESNs are explored by presenting possible implementations and potential applications in machine vision and prediction. The memristor-based spiking neural networks part is aimed at discovering the implementation of all-to-all spike interaction. The proposed approaches with trace-base spikes were evaluated by comparing them to previous models in terms of compatibility. Associative learning and supervised learning are applied to the proposed networks in order to show their feasibility and comparability.

The remainder of this chapter is organised as follows. Section 7.1 presents a summary of the contributions of this thesis, followed by a discussion of potential avenues for future research in Section 7.2.

7.1 Contributions

A summary of the contributions to the memristor-based neural networks research area in the context of feasibility, theories and approaches, discussed in Chapters 3 to 6, is presented in this section.

A staircase memristor model to describe and utilise the wavy behaviour of ferroelectric memristors is proposed, which assumes that there exists multiple regions in a memristor. The basic idea is to utilise the delayed-switching effect of memristors rather than avoiding them, thus staircase memristors could be applied to applications where a few stable states are required. An example of symmetric behaviour of staircase memristors was investigated by illustrating the pinched hysteresis loop and the evolution of states. The comparison to analog memristors was studied in Chapter 3, which demonstrated that analog memristors are more sensitive to frequency and memristance variations thus leading to larger errors. This model was outlined through simulations and by reproducing the asymmetric behaviour of ferroelectric memristors which show different dynamics for their rise and descent paths. It was also proposed that a digital potentiometer controlled by a microcontroller could be used to emulate the staircase memristor. Based on the delayed-switching effect, the number of pulses determines the level of the memristance and the sign of pulses determines the direction of memristance variation. By applying staircase memristors as local connections, a new CNN implementation was presented in conventional CNNs and ESNs. The motivation for applying memristors in CNN lies in that large local connections are available to CNN implementations, since current CNN circuits are not realised in large scale or local connections. It is expected that with this new approach it will be possible to reduce the size, power consumption and complexity of the conventional circuits, especially in an ESN where a reservoir normally consists of very large connections between nodes. On the other hand, it was proposed that ESNs could adapt memristor-based local connections within the reservoir by modifying the original algorithm, and therefore nodes are only locally connected. The results from the experiment which are presented in Chapter 4 support the theories and proposed approaches in the previously mentioned memristor-based neural networks. Focusing on machine vision tasks, including noise removal and edge detection, and the Mackey-Glass data set prediction, the new structures produced satisfactory performance. It is worth noting that the proposed ESN with memristive local connections had a slightly lower performance than the original ESN and lacked stability in terms of precision. This issue was caused by the simplified structure and could be improved by choosing proper parameters in empirical or grid search ways.

Focusing on STDP, memristor-based spiking neural networks were presented and discussed. Concerning the nearest-neighbour and all-to-all spike interactions, a trace-based STDP model was presented to overcome the compatibility problems. In existing memristive STDP models, STDP learning is adapted to memristor-based spiking neural networks using pre- and postsynaptic spikes which have the shape of a typical action potential. Through propagating postsynaptic spikes back to presynaptic neurons, memristors are controlled by the voltage difference between pre- and postsynaptic spike signals. We have studied existing memristive STDP models and therefore found that presynaptic or postsynaptic spikes can not be added which is the case of all-to-all spike interaction. Since the proposed trace-based memristive STDP model leaves exponentially decreased traces when spikes arrive, presynaptic or postsynaptic spikes can be added to demonstrate the effect of all previous spikes. Therefore, it could be used to cope with the all-to-all spike interaction whilst being compatible with the nearest-neighbour spike interaction. We have pointed out that existing research focuses on building associations on memristor-based neural networks, which lack the ability to forget established associations. It should be noted that, by introducing a rate-based term to make spikes' amplitudes adaptive, retention loss can be achieved by utilising the proposed model in a simplified neural networks. Therefore, after the association between two correlated events is established, the association would be weakened if only the learning neuron is stimulated. This scenario is similar to the famous

Pavlov's dog experiment when the bell rings without the presence of food. In such a way, the proposed model takes advantages of both spiking-time dependent plasticity and rate-based plasticity. In addition, delta rule was adapted to the proposed model in order to demonstrate the capability of supervised learning of memristor-based spiking neural networks. The experiments have shown that the trace-based memristive STDP model is compatible with both nearest-neighbour and all-to-all spike interactions. The memristor-based spiking neural networks presented in Chapter 6 learned to associate correlated events "sound of a bell" and "sight of food" when both events occurred. If only "sound of a bell" was given after the association established, retention loss occurred to forget the association. The simulation results of supervised learning indicate that memristor-based neural networks are capable of learning single or multiple tasks in terms of spike patterns.

The research described in this thesis has presented theories, novel ideas and feasible approaches to the utilisation of memristors in neural networks through new memristor models, new memristive STDP models and applications in CNNs, ESNs and spiking neural networks. These studies are unexplored research areas or improve existing systems and broaden the applications of memristor-based neural networks. The next decades will show the impact of memristors and memristor-based neural networks in both research and engineering fields. It is hoped that the explorations presented in this thesis will motivate more investigations and studies in this field and lead to sophisticated systems in the future.

7.2 Future Research

Following the ideas presented in this thesis, there are several potential directions for future research especially in the domain of memristor-based spiking neural networks.

The utilisation of memristors in neural networks is a thriving research field which has led to studies in multilayer neural networks, cellular non-linear networks and chaotic neural networks. However, memristor-based reservoir computing is a relatively new research area which has not been widely studied. Even the concept of reservoir computing itself is a new research branch in neural networks, and thus there are lots of interesting and note-worthy directions in the combination of memristors and reservoir computing. In this thesis, we have studied a memristor-based reservoir where memristors are the local connections of nodes, which is an initial attempt at investigating this area. In the following discussions, we attempt to extend the existing research and discover more potential directions.

In relation to adapting the proposed trace-based memristive STDP model to sophisticated spiking neural networks, a potential direction is to utilise the liquid state machine (LSM) which is a branch of reservoir computing and focusing on biological studies. LSM is proposed as a computational model for describing computations in biological neural networks. Similar to ESNs, nodes in its reservoir are randomly connected and are capable of real-time computation on continuous streams of data. Based on this network, spiking-time dependent plasticity and the proposed memristive model can be implemented to build a memristor-based LSM with spiking neurons and memristors. In this case, memristors become synaptic connections, and therefore we may explore whether the locally connected CNN structure could be adapted to such a network. Furthermore, memristor-based LSM could be evaluated by several benchmark tasks to demonstrate its capabilities in real applications and thus proving another avenue in memristor-based spiking neural networks.

Another interesting direction to explore is to evaluate different memristor-based dynamical systems in reservoir. As introduced previously, non-linear dynamical systems with rich dynamics are possible to implement the reservoir. Currently, non-linear dynamics of memristor-based circuits are widely studied such as the memristor oscillators [32, 49]. Thus, networks of memristor oscillators may be good candidates for the realisation of a reservoir. In fact, memristor models for building memristor oscillatory neurocomputers have been studied in [33] which gives novel insights into the non-linear dynamics of different memristor models. Following this study, we may build reservoirs with different memristor oscillator circuits to demonstrate different dynamics and performances in benchmark tasks, and thus we may investigate how different memristor models or oscillators outperform each other in various tasks. By this means, it provides in-depth studies in the utilisation of memristive dynamical systems in memristor-based reservoir computing. Meanwhile, this study may enable comparison to other dynamical systems such as Mackey-Glass oscillator used in reservoir computing and yield studies of reservoir computing from the view of memristive reservoirs.

Finally, as another direction for future research, it would be interesting to investigate memristor oscillators in adaptive learning. As we know in biological neural networks, firing is composed of oscillations of spikes. In fact, oscillations happen in neurons, especially correlated firing [34]. Thus, the neurons involved fire together is a simple kind of correlated firing. It has been found that correlated firing which occurs in olfactory systems and visual cortex has the form of oscillation of firing [38, 42]. Oscillations and rhythm also have been found in amoebae but utilised to adapt to their environment and to anticipate periodic events [85]. This research makes us think that neurons may not only learn from the association of stimuli but also the frequency or oscillation of stimuli. In addition, from long-term research, exogenous timing has been proposed in chronobiology which studies the timing mechanism in biological systems. Associated with the amoebae experiment, oscillation may explain why amoebae could predict periodic events so precisely. Such behaviour has been studied in [79, 101] which attempt to model amoebae learning using memristor-based oscillation circuits. Following these studies, we can find a way to design adaptive learning which utilises the memory effect of memristive devices to adapt stimuli and synchronise itself with stimuli. Therefore, this extended study may provide an interesting way to study chronobiology and correlated firing where the system itself correlates with exogenous stimuli.

Bibliography

[1] Afifi, A., Ayatollahi, A. and Raissi, F. (2009). STDP implementation using memristive nanodevice in CMOS-nano neuromorphic networks. *IEICE Electronics Express*, 6(3), pp. 148–153.

[2] Ames, H. et al. (2012). The Animat: New frontiers in whole brain modeling. *Pulse, IEEE*, 3(1), pp. 47–50.

[3] Ascoli, A. et al. (2013). Memristor model comparison. *Circuits and Systems Magazine, IEEE*, 13(2), pp. 89–105.

[4] Barbour, B. et al. (2007). What can we learn from synaptic weight distributions? *Trends in Neurosciences*, 30(12), pp. 622–629.

[5] Bell, C. C. et al. (1997). Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature*, 387(6630), pp. 278–281.

[6] Bi, G. and Poo, M. (2001). Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, 24(1), pp. 139–166.

[7] Bi, G.-q. and Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *The Journal of Neuroscience*, 18(24), pp. 10464–10472.

[8] Bienenstock, E., Cooper, L. and Munro, P. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *The Journal of Neuroscience*, 2(1), pp. 32–48, `http://www.jneurosci.org/content/2/1/32.full.pdf+html`.

[9] Biolek, D., Biolek, Z. and Biolkova, V. (2009). SPICE modeling of memristive, memcapacitative and meminductive systems. In *Circuit Theory and Design, ECCTD 2009 European Conference on*, pp. 249–252.

[10] Biolek, Z., Biolek, D. and V. Biolkova, V. (2009). SPICE model of memristor with nonlinear dopant drift. *Radioengineering*, 18(2), pp. 210–214.

[11] Bliss, T. V. and Lømo, T. (1973). Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of Physiology*, 232(2), pp. 331–356.

[12] Bohte, S., Kok, J. and La Poutré, H. (2000). Spike-prop: error-backprogation in multi-layer networks of spiking neurons. In M. Verleysen, ed., *Proceedings of the European Symposium on Artificial Neural Networks ESANN'2000*, D-Facto, pp. 419–425.

[13] Borghetti, J. et al. (2009). A hybrid nanomemristor/transistor logic circuit capable of self-programming. *Proceedings of the National Academy of Sciences*, 106(6), pp. 1699–1703, `http://www.pnas.org/content/106/6/1699.full.pdf+html`.

[14] Cai, W. (2011). *Nonlinear dynamics in memristive systems*. Ph.D. thesis, Technische Universität Dresden.

[15] Cai, W. and Tetzlaff, R. (2012). Advanced memristive model of synapses with adaptive thresholds. In *Cellular Nanoscale Networks and Their Applications (CNNA), 2012 13th International Workshop on*, pp. 1–6.

[16] Cantley, K. D. et al. (2012). Neural learning circuits utilizing nanocrystalline silicon transistors and memristors. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(4), pp. 565–573.

[17] Chang, T. et al. (2011). Synaptic behaviors and modeling of a metal oxide memristive device. *Applied Physics A*, 102(4), pp. 857–863.

[18] Chang, Y.-F., Chang, T.-C. and Chang, C.-Y. (2011). Investigation statistics of bipolar multilevel memristive mechanism and characterizations in

a thin FeO_x transition layer of $TiN/SiO_2/FeO_x/Fe$ structure. *Journal of Applied Physics*, 110(5), 053703.

[19] Chanthbouala, A. et al. (2012). A ferroelectric memristor. *Nature Materials*, 11(10), pp. 860–864.

[20] Chen, L. et al. (2013). A synapse memristor model with forgetting effect. *Physics Letters A*, 377(45-48), pp. 3260 – 3265.

[21] Chen, L. et al. (2014). Memristor crossbar-based unsupervised image learning. *Neural Computing and Applications*, 25(2), pp. 393–400.

[22] Chen, W., Yang, X. and Wang, F. Z. (2013). Delayed switching applied to memristor content addressable memory cell. In *Proceedings of the World Congress on Engineering*, vol. 1.

[23] Chen, W., Yang, X. and Wang, F. Z. (2014). A new memristor model for content addressable memory. In *Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on*, pp. 335–338.

[24] Chen, Y.-J. et al. (2014). Resistance switching induced by hydrogen and oxygen in diamond-like carbon memristor. *Electron Device Letters, IEEE*, PP(99), pp. 1–1.

[25] Chua, L. (1971). Memristor. the missing circuit element. *IEEE Trans Circuit Theory*, CT-18(5), pp. 507–519.

[26] Chua, L. (2011). Resistance switching memories are memristors. *Applied Physics A: Materials Science and Processing*, 102, pp. 765–783.

[27] Chua, L. (2012). The fourth element. *Proceedings of the IEEE*, 100(6), pp. 1920–1927.

[28] Chua, L. and Kang, S. M. (1976). Memristive devices and systems. *Proceedings of the IEEE*, 64(2), pp. 209–223.

[29] Chua, L. O. and Yang, L. (1988). Cellular neural networks: Applications. *Circuits and Systems, IEEE Transactions on*, 35(10), pp. 1273–1290.

[30] Cline, H. T. (1998). Topographic maps: Developing roles of synaptic plasticity. *Current Biology*, 8(23), pp. R836 – R839.

[31] Corinto, F. and Ascoli, A. (2012). A boundary condition-based approach to the modeling of memristor nanostructures. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 59(11), pp. 2713–2726.

[32] Corinto, F., Ascoli, A. and Gilli, M. (2011). Nonlinear dynamics of memristor oscillators. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 58(6), pp. 1323–1336.

[33] Corinto, F., Ascoli, A. and Gilli, M. (2012). Memristor models for pattern recognition systems. In R. Kozma, R. E. Pino and G. E. Pazienza, eds., *Advances in Neuromorphic Memristor Science and Applications*, *Springer Series in Cognitive and Neural Systems*, vol. 4, Springer Netherlands, pp. 245–267.

[34] Crick, F. (1995). *The astonishing hypothesis: The scientific search for the soul.* Scribner.

[35] Eshraghian, K. et al. (2011). Memristor MOS content addressable memory (MCAM): Hybrid architecture for future high performance search engines. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 19(8), pp. 1407–1417.

[36] Fang Xu-Dong, W. J.-J., Tang Yu-Hua (2012). SPICE modeling of memristors with multilevel resistance states. *Chinese Physics B*, 21(9), 98901.

[37] Fix, T., MacManus-Driscoll, J. L. and Blamire, M. G. (2009). Delta-doped $LaAlO_3/SrTiO_3$ interfaces. *Applied Physics Letters*, 94(17), 172101.

[38] Freeman, W. J. (1990). Nonlinear neural dynamics in olfaction as a model for cognition. In *Chaos in Brain Function*, Springer, pp. 63–73.

[39] Furber, S. et al. (2013). Overview of the SpiNNaker system architecture. *Computers, IEEE Transactions on*, 62(12), pp. 2454–2467.

[40] Gerstner, W., Ritz, R. and Van Hemmen, J. L. (1993). Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns. *Biological Cybernetics*, 69(5-6), pp. 503–515.

[41] Gerstner, W. et al. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383(6595), pp. 76–78.

[42] Gray, C. M. and Singer, W. (1989). Stimulus-specific neuronal oscillations in orientation columns of cat visual cortex. *Proceedings of the National Academy of Sciences*, 86(5), pp. 1698–1702.

[43] Gutig, R. and Sompolinsky, H. (2006). The tempotron: a neuron that learns spike timing-based decisions. *Nat Neurosci*, 9(3), pp. 420–428.

[44] Han, V. Z., Grant, K. and Bell, C. C. (2000). Reversible associative depression and nonassociative potentiation at a parallel fiber synapse. *Neuron*, 27(3), pp. 611–622.

[45] Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley.

[46] Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4), p. 500.

[47] Hu, J. and Wang, J. (2010). Global uniform asymptotic stability of memristor-based recurrent neural networks with time delays. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pp. 1–8.

[48] Huang, C.-H. et al. (2013). Manipulated transformation of filamentary and homogeneous resistive switching on ZnO thin film memristor with controllable multistate. *ACS Applied Materials & Interfaces*, 5(13), pp. 6017–6023, `http://pubs.acs.org/doi/pdf/10.1021/am4007287`.

[49] Itoh, M. and Chua, L. O. (2008). Memristor oscillators. *International Journal of Bifurcation and Chaos*, 18(11), pp. 3183–3206, `http://www.worldscientific.com/doi/pdf/10.1142/S0218127408022354`.

[50] ITRS (2007). International technology roadmap for semiconductors. Available online at `http://www.itrs.net`, accessed: 2011-09-01.

[51] ITRS (2010). International technology roadmap for semiconductors. Available online at `http://www.itrs.net`, accessed: 2014-03-20.

[52] Jaeger, H. (2001). The echo state approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148, p. 34.

[53] Jo, S. H., Kim, K.-H. and Lu, W. (2009). High-density crossbar arrays based on a Si memristive system. *Nano Letters*, 9(2), pp. 870–874, `http://pubs.acs.org/doi/pdf/10.1021/nl8037689`.

[54] Jo, S. H. et al. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Letters*, 10(4), pp. 1297–1301, pMID: 20192230, `http://pubs.acs.org/doi/pdf/10.1021/nl904092h`.

[55] Joglekar, Y. N. and Wolf, S. J. (2009). The elusive memristor: properties of basic electrical circuits. *European Journal of Physics*, 30(4), p. 661.

[56] Jung, G.-Y. et al. (2006). Circuit fabrication at 17 nm half-pitch by nanoimprint lithography. *Nano Letters*, 6(3), pp. 351–354, `http://pubs.acs.org/doi/pdf/10.1021/nl052110f`.

[57] Junsangsri, P. and Lombardi, F. (2012). A memristor-based TCAM (ternary content addressable memory) cell: Design and evaluation. In *Proceedings of the Great Lakes Symposium on VLSI*, New York, NY, USA: ACM, GLSVLSI '12, pp. 311–314.

[58] Katz, L. C. and Shatz, C. J. (1996). Synaptic activity and the construction of cortical circuits. *Science*, 274(5290), pp. 1133–1138, `http://www.sciencemag.org/content/274/5290/1133.full.pdf`.

[59] Kavehei, O. et al. (2010). The fourth element: characteristics, modelling and electromagnetic theory of the memristor. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 466(2120), pp. 2175–2202, `http://rspa.royalsocietypublishing.org/content/466/2120/2175.full.pdf+html`.

[60] Kim, K.-H. et al. (2012). A functional hybrid memristor crossbar-array/cmos system for data storage and neuromorphic applications. *Nano Letters*, 12(1), pp. 389–395, `http://pubs.acs.org/doi/pdf/10.1021/nl203687n`.

[61] Kulkarni, M. and Teuscher, C. (2012). Memristor-based reservoir computing. In *Nanoscale Architectures (NANOARCH), 2012 IEEE/ACM International Symposium on*, pp. 226–232.

[62] Kvatinsky, S. et al. (2013). TEAM: Threshold adaptive memristor model. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 60(1), pp. 211–221.

[63] Laiho, M. et al. (2010). Memristive synapses are becoming reality. *the newsletter of the Institute of Neuromorphic Engineering.*

[64] Lee, M.-J. et al. (2011). A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta_2O_{5-x}/TaO_{2-x} bilayer structures. *Nat Mater*, 10(8), pp. 625–630.

[65] Lehtonen, E. and Laiho, M. (2010). *CNN using memristors for neighborhood connections*, IEEE. p. 14.

[66] Linares-Barranco, B. and Serrano-Gotarredona, T. (2009). Memristance can explain spike-time dependent plasticity in neural synapses. *Nature Precedings.*

[67] Linares-Barranco, B. et al. (2011). On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. *Frontiers in Neuroscience*, 5, pp. 1–22.

[68] Lukoševičius, M. (2012). A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade*, Springer, pp. 659–686.

[69] Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9), pp. 1659 – 1671.

[70] Maass, W., Natschläger, T. and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11), pp. 2531–2560.

[71] Mackey, M. and Glass, L. (1977). Oscillation and chaos in physiological control systems. *Science*, 197(4300), pp. 287–289, `http://www.sciencemag.org/content/197/4300/287.full.pdf`.

[72] Markram, H. (2006). The Blue Brain project. *Nat Rev Neurosci*, 7(2), pp. 153–160.

[73] Markram, H. et al. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275(5297), pp. 213–215, `http://www.sciencemag.org/content/275/5297/213.full.pdf`.

[74] Martin, S., Grimwood, P. and Morris, R. (2000). Synaptic plasticity and memory: an evaluation of the hypothesis. *Annual Review of Neuroscience*, 23(1), pp. 649–711.

[75] Masquelier, T., Guyonneau, R. and Thorpe, S. J. (2008). Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. *PLoS ONE*, 3(1), p. e1377.

[76] Mead, C. and Ismail, M. (1989). *Analog VLSI Implementation of Neural Systems*. The Kluwer International Series in Engineering and Computer Science: VLSI, Computer Architecture and Digital Signal Processing, Springer.

[77] Mohemmed, A. et al. (2013). Training spiking neural networks to associate spatio-temporal input-output spike patterns. *Neurocomputing*, 107(0), pp. 3 – 10, Timely Neural Networks Applications in Engineering Selected Papers from the 12th EANN International Conference, 2011.

[78] Pershin, Y. and Di Ventra, M. (2010). Practical approach to programmable analog circuits with memristors. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 57(8), pp. 1857–1864.

[79] Pershin, Y. V., La Fontaine, S. and Di Ventra, M. (2009). Memristive model of amoeba learning. *Physical Review E*, 80(2), p. 021926.

[80] Pershin, Y. V. and Ventra, M. D. (2010). Experimental demonstration of associative memory with memristive neural networks. *Neural Networks*, 23(7), pp. 881–886.

[81] Pfister, J.-P. and Gerstner, W. (2005). Beyond pair-based STDP: A phenomenological rule for spike triplet and frequency effects. In *Advances in Neural Information Processing Systems*, pp. 1081–1088.

[82] Pfister, J.-P. and Gerstner, W. (2006). Triplets of spikes in a model of spike timing-dependent plasticity. *The Journal of Neuroscience*, 26(38), pp. 9673–9682.

[83] Ponulak, F. and Kasiński, A. (2010). Supervised learning in spiking neural networks with resume: Sequence learning, classification, and spike shifting. *Neural Comput*, 22(2), pp. 467–510.

[84] Prodromakis, T., Toumazou, C. and Chua, L. (2012). Two centuries of memristors. *Nat Mater*, 11(6), pp. 478–481.

[85] Saigusa, T. et al. (2008). Amoebae anticipate periodic events. *Phys Rev Lett*, 100(1), p. 018101.

[86] SenGupta, B. and Stemmler, M. (2014). Power consumption during neuronal computation. *Proceedings of the IEEE*, 102(5), pp. 738–750.

[87] Serrano-Gotarredona, T. et al. (2013). STDP and STDP variations with memristors for spiking neuromorphic learning systems. *Frontiers in Neuroscience*, 7(2).

[88] Shepherd, G. and Koch, C. (1990). Dendritic electrotonus and synaptic integration. *The Synaptic Organization of the Brain*, pp. 439–473.

[89] Sjöström, P. J. and Gerstner, W. (2011). Spike-timing dependent plasticity. *Frontiers in Synaptic Neuroscience*, pp. 35–44.

[90] Sjöström, P. J. et al. (2008). Dendritic excitability and synaptic plasticity. *Physiological Reviews*, 88(2), pp. 769–840.

[91] Snider, G. (2011). Instar and outstar learning with memristive nanodevices. *Nanotechnology*, 22(1), p. 015201.

[92] Snider, G. et al. (2011). From synapses to circuitry: Using memristive memory to explore the electronic brain. *Computer*, 44(2), pp. 21–28.

[93] Snider, G. S. (2007). Self-organized computation with unreliable, memristive nanodevices. *Nanotechnology*, 18(36), p. 365202.

[94] Snider, G. S. (2008). Spike-timing-dependent learning in memristive nanodevices. *Nanoscale Architectures, IEEE International Symposium on*, pp. 85–92.

[95] Stent, G. S. (1973). A physiological mechanism for Hebb's postulate of learning. *Proceedings of the National Academy of Sciences*, 70(4), pp. 997–1001.

[96] Strukov, D. et al. (2010). Hybrid CMOS/memristor circuits. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 1967–1970.

[97] Strukov, D. B. et al. (2008). The missing memristor found. *Nature*, 453(7191), pp. 80–83.

[98] Versace, M. and Chandler, B. (2010). The brain of a new machine. *IEEE Spectr*, 47, pp. 30–37.

[99] Wang, F. et al. (2010). Delayed switching in memristors and memristive systems. *IEEE Electron Device Letters*, 31(7), pp. 755–757.

[100] Wang, F. Z. et al. (2012). Delayed switching applied to memristor neural networks. *Journal of Applied Physics*, 111(7), pp. 07E317–07E317.

[101] Wang, F. Z. et al. (2013). Adaptive neuromorphic architecture (ANA). *Neural Networks*, 45, pp. 111–116.

[102] Wang, S.-Y. et al. (2010). Multilevel resistive switching in $Ti/Cu_xO/Pt$ memory devices. *Journal of Applied Physics*, 108(11), 114110.

[103] Williams, R. S. (2008). How we found the missing memristor. *IEEE Spectrum*, 45(12), pp. 28–35.

[104] Xia, Q. et al. (2009). Memristor-CMOS hybrid integrated circuits for reconfigurable logic. *Nano Letters*, 9(10), pp. 3640–3645.

[105] Yang, J. J. et al. (2008). Memristive switching mechanism for metal/oxide/metal nanodevices. *Nature Nanotechnology*, 3(7), pp. 429–433.

[106] Yang, J. J. et al. (2010). High switching endurance in TaO_x memristive devices. *Applied Physics Letters*, 97(23), 232102.

[107] Yang, X., Chen, W. and Wang, F. Z. (2013). A memristor-CAM (content addressable memory) cell: New design and evaluation. *International Conference on Computer Science and Information Technology*, pp. 1045–1048.

[108] Zhang, L. I. and Poo, M.-m. (2001). Electrical activity and development of neural circuits. *Nature Neuroscience*, 4(11), p. 1207.

Appendix A

Publications and Acronyms

All the publications with my contribution has been listed in chronological order:

1. Frank Z. Wang, Na Helian, Sining Wu, Xiao Yang, Yike Guo, Guan Lim, and Md Mamunur Rashid. “Delayed switching applied to memristor neural networks.” Journal of Applied Physics, 2012, 111, 07E317-07E317.

2. Frank Z. Wang, Leon O. Chua, Xiao Yang, Na Helian, Ronald Tetzlaff, Torsten Schmidt, Caroline Li, Jose Manuel Garcia Carrasco, Wanlong Chen, and Dominique Chu. “Adaptive neuromorphic architecture (ANA).” Neural Networks, 2013, 45, 111-116.

3. Xiao Yang, Wanlong Chen, and Frank Z. Wang. “A memristor-CAM (Content Addressable Memory) cell: new design and evaluation.” Computer Science and Information Technology, International Conference on, 2013, 1045-1048.

4. Wanlong Chen, Xiao Yang, and Frank Z. Wang. “Delayed switching applied to memristor content addressable memory cell.” In Proceedings of the World Congress on Engineering, 2013, 354-357.

5. Xiao Yang, Wanlong Chen, Frank Z. Wang. “A supervised spiking time dependent plasticity network based on memristors.” Computational Intelligence and Informatics (CINTI), 2013 IEEE 14th International Symposium on, 2013, 447-451.

6. Wanlong Chen, Xiao Yang, Frank Z. Wang. “Memristor content addressable memory.” Nanoscale Architectures (NANOARCH), 2014 IEEE/ACM International Symposium on, 2014, 83-87.

7. Wanlong Chen, Xiao Yang, and Frank Z. Wang. “HSMM: Hard switching memristor model.” In Proceedings of the World Congress on Engineering and Computer Science, 2014, 11-14.

8. Xiao Yang, Wanlong Chen, Frank Z. Wang. “The memristor-based associative learning network with retention loss.” Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on, 2014, 249-253.

9. Wanlong Chen, Xiao Yang, Frank Z. Wang. “A new memristor model for content addressable memory.” Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on, 2014, 335-338.

10. Xiao Yang, Wanlong Chen, Frank Z. Wang. “The staircase memristor and its applications.” Electronics Circuits and Systems (ICECS), 2014 IEEE 21st International Conference on, 2014, 259-262.

Acronyms

BCM Boundary Condition Memristor.

BIMPA Biologically Inspired Massively Parallel Architectures.

BP Back-Propagation.

CAM Content-Addressable Memory.

CMOL CMOS MOLecular.

CMOS Complementary Metal-Oxide-Semiconductor.

CNN Cellular Neural Network.

DRAM Dynamic Random Access Memory.

EPSC Excitatory Postsynaptic Current.

EPSP Excitatory Postsynaptic Potential.

ESN Echo State Network.

FeRAM Ferroelectric Random Access Memory.

FTJ Ferroelectric Tunnel Junction.

HP Hewlett-Packard.

LIF Leaky Integrate-and-Fire.

LSM Liquid State Machine.

LTD Long-Term Depression.

LTP Long-Term Potentiation.

MLC Multi-Level Cell.

MoNETA Modular Neural Exploring Travelling Agent.

MSE Mean-Squared Error.

PCRAM Phase-Change Random Access Memory.

RMSE Root-Mean-Squared Error.

RNN Recurrent Neural Network.

RRAM Resistive Random Access Memory.

SNN Spiking Neural Network.

SPAN Spike Pattern Association Neuron.

SPICE Simulation Program with Integrated Circuit Emphasis.

SpiNNaker Spiking Neural Network Architecture.

SRAM Static Random Access Memory.

STDP Spiking-Time Dependent Plasticity.

TCAM Ternary Content-Addressable Memory.

TDM Time-Division Multiplexing.

TEAM ThrEshold Adaptive Memristor.

VLSI Very-Large Scale Integration.