

**PROJECT REPORT**

ON

**STUDY OF DYNAMICS OF BASIC AND  
MEMRISTIVE EXACT IZHIKEVICH MODEL**

SUBMITTED BY

ANDRIYA JOMY

**REGISTER NO.: AM21PHY003**

In partial fulfillment of  
the requirements for award of the postgraduate degree in physics



**DEPARTMENT OF PHYSICS AND CENTRE FOR RESEARCH**

**ST. TERESA'S COLLEGE (AUTONOMOUS) ERNAKULAM**

**2021-2023**

**ST. TERESA'S COLLEGE (AUTONOMOUS)  
ERNAKULAM**



**M.Sc. PHYSICS PROJECT REPORT**

Name : ANDRIYA JOMY

Register Number : AM21PHY003

Year of Work : 2021-2023

This is to certify that the project "*Study of Dynamics of Basic and Memristive Exact Izhikevich Models*" is the work done by **ANDRIYA JOMY**.

  
Dr. PRIYA PARVATHI AMEENA

Head of the Department



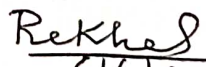
  
Dr. MARY VINAYA

Guide in Charge

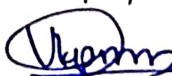
Submitted for the University Examination held in St. Teresa's College  
(Autonomous) Ernakulam

Examiners

1) Dr. REKHA .S

  
6/6/23

2) Dr. Louise Froebel P.G.



Date: 06/06/2023

DEPARTMENT OF PHYSICS  
ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM



CERTIFICATE

This is to certify that the project report entitled "STUDY OF DYNAMICS OF BASIC AND MEMRISTIVE EXACT IZHIKEVICH MODELS" is the bonafide work done by Ms. ANDRIYA JOMY (Reg. No:AM20PHY003) under the guidance of Dr. MARY VINAYA, Assistant Professor, Department of Physics and Centre for Research, St. Teresa's College (Autonomous), Ernakulam in partial fulfilment of the award of the Degree of Master of Science in Physics, St. Teresa's College (Autonomous), Ernakulam affiliated to Mahatma Gandhi University, Kottayam.

*Priya*

Dr. PRIYA PARVATHI AMEENA JOSE  
Head of the Department



*Mary*

Dr. MARY VINAYA  
Guide in Charge

## DECLARATION

I, hereby declare that the project work entitled “**STUDY OF DYNAMICS OF BASIC AND MEMRISTIVE EXACT IZHIKEVICH MODELS**” is a record of an original work done by me under the guidance of **Dr. MARY VINAYA**, Assistant Professor, Department of Physics and centre for Research, St Teresa’s College (Autonomous), Ernakulam in the partial fulfilment of the requirements for the award of the Degree of Master of Physics. I further declare that the data included in the project is collected from various sources and are true to the best of my knowledge.

Place : Ernakulam

Date : 06/06/2023



ANDRIYA JOMY



## **ACKNOWLEDGEMENT**

Every project big or small is successful largely due to the effort of a number of wonderful people who have always given their valuable or extensive cooperation.

First and foremost we would like to thank God Almighty for showering his blessings on us in this endeavor.

We express our sincere gratitude to Dr. Mary Vinaya and Dr. Sunsu Kurian, Assistant Professors, Department of Physics, St. Teresa's College (Autonomous), Ernakulam, under whose able guidance and supervision, we could complete our project. Her extensive cooperation and valuable guidance were the light in the dark for us in the course of this project.

We take this opportunity to thank Dr. Priya Parvathy Ameena Jose, Head of the Physics Department for the generous concern and support given to us. We express our sincere sense of gratitude to both teaching and non-teaching staffs of Department of Physics, St. Teresa's College (Autonomous), Ernakulam, for extending facilities required from time to time.

We would like to thank each and every one who has contributed to this work by their unfailing support throughout the preparation of the project.

**STUDY OF DYNAMICS OF BASIC  
AND MEMRISTIVE IZHIKEVICH  
MODELS**

# CONTENTS

<b>ABSTRACT</b>	1
-----------------	---

## **Chapter 1: NON-LINEAR DYNAMICS**

1.1 Characteristics of Non-Linear Systems	3
1.2 Bifurcation	5
1.3 Limit Cycles	6
1.4 Chaos	7
1.5 References	8

## **Chapter 2: NEURAL NETWORKS**

2.1 What is a Neuron?	9
2.2 Structure of a Neuron	10
2.2.1 Dendrites	10
2.2.2 Cell Body or Soma	10
2.2.3 Axon	11
2.3 Synapse	11
2.3.1 Electrical Synapse	12
2.3.2 Chemical Synapse	12
2.4 Action Potentials in Neurons	12
2.5 References	14

## **Chapter 3: NEURON MODELS**

3.1 Hodgkin-Huxley Neuron Model	15
3.2 Integrate and Fire Neuron Model	17
3.3 Izhikevich Neuron Model	20
3.4 References	21

## **Chapter 4: THE IZHEKEVICH NEURON MODEL**

4.1 Introduction	22
------------------	----



4.2 The Network System	23
4.3 Mean -Field Reduction	24
4.4 General Mean-Field Description	24
4.5 Density Function In Conditional Form	25
4.6 Lorentzian Ansatz	26
4.7 Heterogeneity with Lorentzian Distribution	26
4.8 Numerical Analysis	27
4.9 Extension to two-coupled population	30
4.10 Discussion	36
4.11 References	37

## **Chapter 5: DYNAMICS OF MEMRISTIVE IZHKEVICH MODEL**

5.1 Introduction	39
5.2 Memristor Properties	41
5.2.1 Flux-Charge Relation	41
5.2.2 Current-Voltage Relation	42
5.2.3 Resistance-Time Relation	43
5.2.4 Resistance -Voltage Relation	44
5.3 Extension Izhikevich Neuron Model	45
5.3.1 The Network System	45
5.3.2 General Mean-field Description	45
5.3.3 Density in Conditional Form	47
5.3.4 Lorentzian Ansatz	48
5.4 Result	49
5.5 References	50

<b>Chapter 6: DISCUSSION</b>	52
MATLAB Code for basic Izhikevich Model	54

## **ABSTRACT**

Neurons are the tools responsible for transmitting information all over the body via electrical and chemical signals that coordinate all the important functions in our life which makes it an incredibly complex communication system. The Izhikevich neuron model is a mathematical description of how neurons in the brain integrate and transmit information through electrical impulses. It is a simplified and computationally efficient model that captures the essential features of real neurons. The model consists of two differential equations that describe the dynamics of the membrane potential and the recovery variable. The dynamics of the model can be adjusted to replicate different types of neurons found in the brain, including regular spiking, fast spiking, and intrinsically bursting neurons.

In this project, to characterise the behaviour of neurons under electromagnetic stimulation and noise, the Izhikevich model is suggested and the characteristics were obtained. For further studies we extended the work by using a three-variable memristive Izhikevich model and derived the characteristics of neurons and the resulting equations were similar to that of the Izhikevich model when compared. The influence of internal and external magnetic fields on neurons can also be represented by this model.

# CHAPTER-1

## NON-LINEAR DYNAMICS

We can see lots of dynamical systems around us. A dynamical system is the one in which the state of its constituent particles varies with time. The way the system variables change can be expressed by virtue of differential equations. Some of the examples of dynamical systems are population growth, motion of celestial bodies etc.

Dynamical systems are expressed in two ways: discrete time steps or a continuous timeline.

A state space also called a phase space is a mode used within dynamical systems to capture the change in a system over time. A state space is a two or probably three-dimensional graph in which all possible states of system are represented.

We can model the change in a system state as either continuous or discrete. Firstly, continuous measuring the time intervals between our measurements are negligibly small making it appear as one long continuum and it is done through the language of calculus. On the other hand, we can measure time as discrete, meaning there is a discrete or definite time interval between each measurement and we use iterative maps to do this.

Continuous state systems can be further classified into:

- i. Linear
- ii. Non-linear

Continuous state systems can be used to describe the evolution of the state. In linear system, the evolution of state is governed by linear differential equations whereas in non-linear system the evolution is governed by non-linear equations.

Nonlinear dynamics is the study of systems that can exhibit complex and unpredictable behavior, often characterized by sensitivity to initial conditions, feedback loops, and non-linear relationships between different elements or variables. These systems can include physical systems like weather patterns, biological systems like ecosystems or neural networks, as well as social systems like stock markets or political movements.

Nonlinear dynamics is an interdisciplinary field that combines elements of mathematics, physics, biology, and other fields to better understand these complex systems and their behavior. This knowledge is used to develop models and simulations that can help us predict how these systems may behave under different conditions, and inform decision-making in a variety of contexts.

Nonlinear systems are a type of mathematical model that does not follow a linear relationship between the input and output variables. In other words, the output of a nonlinear system is not proportional to the input. Instead, nonlinear systems exhibit more complex behaviours and patterns, often involving feedback loops, oscillations, chaos and other interesting phenomena.

Nonlinear systems can be found in numerous fields such as physics, chemistry, biology, economics, engineering and many others. Because of their complexity, nonlinear systems can be challenging to analyze and predict. However, many techniques have been developed to study and simulate nonlinear behaviour, such as numerical methods, bifurcation analysis, chaos theory and adaptive control systems. These tools allow researchers to better understand the behaviour of complex systems and make more accurate predictions about their dynamics.

## **1.1 CHARACTERISTICS OF NON-LINEAR SYSTEMS**

Nonlinear systems can exhibit many characteristics that make them distinct from linear systems. Here are some of the key characteristics of nonlinear systems:

1. **Non-proportionality:** The relationship between input and output variables is not linearly proportional, meaning that changes in the input do not produce a corresponding change in the output.
2. **Sensitivity to Initial Conditions:** Small changes in the initial conditions of a nonlinear system can have a significant impact on the final outcome or behavior of the system.
3. **Emergent Behaviors:** Nonlinear systems can exhibit emergent behaviors or patterns that are different from the individual behavior of the components that make up the system.
4. **Feedback Loops:** Nonlinear systems often involve feedback loops, where the output of the system is fed back as input, leading to self-reinforcing or self-cancelling effects.

5. Chaotic Behavior: Certain nonlinear systems can exhibit chaotic behavior, meaning that they are highly sensitive to initial conditions and exhibit complex, seemingly random patterns of behavior.

6. Multiple Equilibria: Nonlinear systems may have multiple equilibria or steady states, meaning that the system can settle into multiple stable states depending on the initial conditions.

7. Non-locality: In some nonlinear systems, phenomena can arise that cannot be explained solely by local interactions between the components of the system, but instead require considering the system as a whole.

Understanding these characteristics is important for analyzing and predicting the behavior of nonlinear systems, and for developing computational and analytical tools to simulate and control them.

Nonlinear dynamics is the study of complex and often chaotic behavior in nonlinear systems. It involves the analysis of systems with multiple variables whose interaction is described by nonlinear equations. This makes it different from linear dynamic systems, where the behavior of the system is much more predictable.

Nonlinear dynamics plays a fundamental role in many fields, including physics, chemistry, biology, economics, and engineering. Some examples of systems that can be described using nonlinear dynamics principles include weather systems, population growth models, chemical reactions and chaotic systems.

Nonlinear dynamics often involves the use of advanced mathematical tools, such as differential equations, chaos theory, and bifurcation analysis, among others. These techniques allow us to understand and predict the behavior of complex and nonlinear systems, identify and analyze stable and unstable solutions, and recognize patterns of behavior.

Linear dynamics deals with systems that are additive, homogeneous, and have a proportional response to inputs. This means that double the input will likewise double the output. The behavior of the system is predictable and straightforward. Examples of linear systems include pendulums and springs.

Nonlinear dynamics deals with systems that are not additive, not homogeneous, and do not have a proportional response to inputs. Nonlinear systems are characterized by feedback loops and non-linear relationships between their inputs and outputs. These systems are often more complex and difficult to predict than linear systems. Examples of nonlinear systems include chaotic systems such as weather patterns, fluid dynamics, and biological systems.

In summary, the main difference between linear and nonlinear dynamics is that linear systems are predictable, while nonlinear systems are often more complex and difficult to predict.

One of the key features of nonlinear dynamics is that it can exhibit non-repeating patterns over long periods of time, even though those patterns are ultimately deterministic. This is the hallmark of chaos, which is one of the most fascinating aspects of nonlinear dynamics. Understanding nonlinear dynamics is important not only for understanding the behavior of complex systems, but also for developing practical applications, such as control algorithms for complex systems or predicting the outcomes of complex phenomena.

## **1.2 BIFURCATION**

A system experiences a bifurcation when a parameter change causes a qualitative shift in the output. A bifurcation can cause a cell membrane's voltage to shift under specific circumstances from being at rest to being oscillatory. Bifurcation analysis is frequently used to comprehend the oscillatory characteristics of tiny networks. Bifurcations occur in dynamical systems, which are mathematical equations.

The equation  $\frac{dx}{dt} = F(x, \mu)$  can be used to define a dynamical system in general, where  $x$  is an  $n$ -dimensional vector of unknowns,  $\mu$  is an  $m$ -dimensional vector of parameters, and  $F$  is a function that depends on the two.

Bifurcation analysis in neural networks is the study of the emergence and disappearance of stable attractors and their bifurcations as the network parameters are varied. Here, a bifurcation refers to a qualitative change in the behavior of the network as its parameters are adjusted.

In a neural network context, bifurcations typically involve the emergence or disappearance of fixed points (stable or unstable equilibrium points) or limit cycles. To better understand these phenomena, bifurcation analysis is performed by studying the dynamics of the network under

varying conditions, such as changes in input patterns or variations in the strengths of synaptic connections.

Bifurcation analysis can be used to identify regions of parameter space in which the network exhibits stable or unstable behavior, as well as to predict the onset of new patterns of activity or transitions between existing patterns. By providing insights into the behavior of complex systems like neural networks, bifurcation analysis can aid in the design and optimization of such systems for various applications.

### **1.3 LIMIT CYCLES**

In neural network analysis, limit cycles refer to the pattern of oscillations that occur in a network's activity over time. These cycles represent a stable and recurring pattern of neuron firing that can occur in certain networks.

The existence of limit cycles can have important implications for the behavior of the network. For example, the presence of limit cycles in a neural network can contribute to its ability to perform certain types of computations or to exhibit complex dynamic behaviors.

However, limit cycles can also be problematic in certain situations, particularly in the context of learning and training. If a network gets stuck in a limit cycle, it may be difficult to modify its behavior through training or other interventions.

Overall, the analysis of limit cycles is an important area of research in neural network analysis, as it can help us understand the behavior of neural networks in a wide range of contexts.

Limit cycles are observed in recurrent neural networks (RNNs) due to the presence of time-delayed feedback loops in the network. These feedback loops cause the activation of neurons to feedback on themselves, creating a periodic oscillation in the network's activity.

Examples of limit cycles in neural networks include:

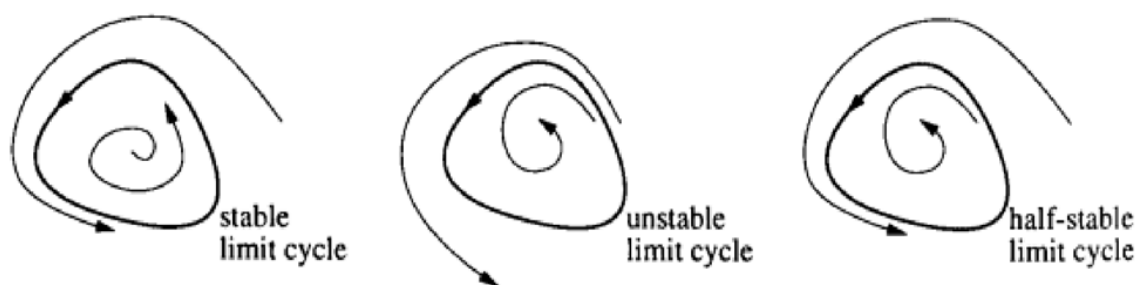
1. Hopfield Network - In this network, a set of binary threshold neurons are interconnected with symmetric weights. The network has a limit cycle where the binary states of its neurons oscillate among different stable states.

2. FitzHugh-Nagumo (FHN) Neuron Model - This network is a two-dimensional system of differential equations that models the behavior of a biological neuron. The network exhibits a limit cycle when the applied input is above a certain threshold.

3. Van der Pol Oscillator - This is a nonlinear oscillator that exhibits a limit cycle when the applied input is above a certain threshold. It is used to model several biological processes, including neural activity.

4. Coupled Oscillators - In a network of coupled oscillators, the oscillators are interconnected in such a way that they influence each other's dynamics. This results in a limit cycle that is synchronized across all the oscillators in the network.

Overall, limit cycles play an essential role in the dynamics of many neural networks, and they are important for modeling a wide range of systems in neuroscience and engineering.



*Figure 1: Different types of limit cycles [1]*

## 1.4 CHAOS

Chaos in neural network analysis refers to the phenomenon where small changes or perturbations in initial conditions can result in very large differences in the behavior of the neural network. This can make it difficult to predict or analyze the behavior of the neural network, as small changes in the input or in the network structure can result in vastly different outputs.

One way to deal with chaos in neural network analysis is to use statistical techniques such as sensitivity analysis or Monte Carlo simulations, which can help to quantify the effects of small changes or uncertainties in the initial conditions. Additionally, it may be helpful to use more robust or stable neural network architectures that are less sensitive to small perturbations. Overall chaos in neural networks can be challenging to deal with, but with careful analysis and



modelling techniques, it is possible to mitigate its effects and gain a better understanding of the behavior of these complex systems.

Here are a few examples of chaos in neural network analysis:

1. Sensitivity to initial conditions: Neural networks are highly sensitive to small changes in the initial conditions. Even a minor variation can significantly affect the output of the network, leading to chaotic behavior.
2. Chaotic attractors: Some neural networks exhibit chaotic attractors, which are regions of the network's state space where the system will eventually settle. These attractors are characterized by their sensitivity to initial conditions and the presence of complex, irregular oscillations.
3. Bifurcations: Bifurcations occur when the behavior of a neural network changes abruptly as a parameter is varied. In some cases, these changes can lead to chaotic behavior, as the network enters into a regime of complex, unpredictable dynamics.
4. Butterfly effect: The butterfly effect is a term used to describe the sensitivity of chaotic systems to small changes. In neural network analysis, it refers to the fact that a small change in one part of the network can have a significant impact on the behavior of the network as a whole.

Overall, the presence of chaos in neural network analysis underscores the complexity of these systems and the challenges involved in understanding and predicting their behavior.

## 1.5 REFERENCES

[1][https://www.google.com/imgres?imgurl=https://media.springernature.com/original/springer-static/image/prt%253A978-1-4419-9863-7%252F12/MediaObjects/978-1-4419-9863-7\\_12\\_Part\\_Fig1533\\_HTML.gif&tbnid=7lVyfIxcGO8JmM&vet=1&imgrefurl=https://lecturenotes.in/discussion-forum/question/what-are-limit-cycles-sko09t8mf&docid=tEuAscFkz\\_A7bM&w=498&h=138&source=sh/x/im/1](https://www.google.com/imgres?imgurl=https://media.springernature.com/original/springer-static/image/prt%253A978-1-4419-9863-7%252F12/MediaObjects/978-1-4419-9863-7_12_Part_Fig1533_HTML.gif&tbnid=7lVyfIxcGO8JmM&vet=1&imgrefurl=https://lecturenotes.in/discussion-forum/question/what-are-limit-cycles-sko09t8mf&docid=tEuAscFkz_A7bM&w=498&h=138&source=sh/x/im/1)

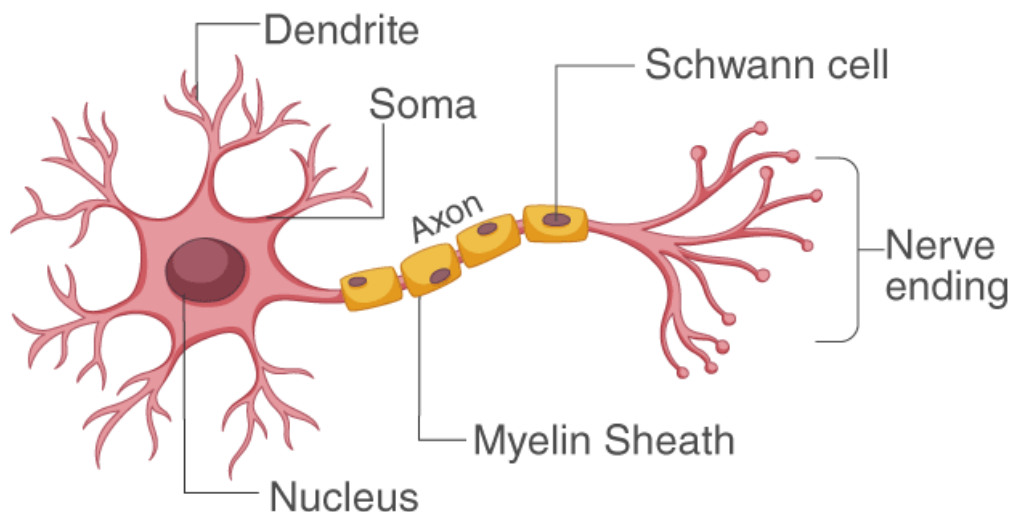
## CHAPTER-2

### THE NEURAL NETWORK

#### 2.1 WHAT IS A NEURON?

The basic unit of neuron system is known as neuron. The 10% of brain is comprised of neurons. There are approximately 86 billion neurons in the body. The nervous system decides all the basic functions of the body also helps in necessary functions such as how we act, helps us remember things and even alter the state of internal organs. For all this the nervous system relies on – the neurons. Group of these neurons are called the neural network. The main function of a neuron is to carry the information throughout the body. This is using chemical signals and electrical impulses. These impulses are sent to different parts of the brain i.e simply the excitation from one neuron to the other.

Depending on its purpose and location, a neuron can have different sizes and shapes. Dendrites, cell bodies, and axons are the three distinct components of all neurons.



*Figure 2: Structure of a Neuron [1]*

We can compare a neuron with a tree for simplicity. Dendrites which resemble branches are tiny filaments and are involved in the passage of messages from other neurons to the cell body. The axon which compared to root of the tree is also called the nerve fibre is neurons output

structure. The axon is where neurons may communicate with one another by exchanging electrical impulses or messages. It is a long projection that carries information from the soma to nearby cells. When to communicate with another, it sends an electrical signal called an action potential along the whole axon. The soma or the cell body resembles the tree trunk which is the powerhouse of the cell nerve cell and contains the nucleus. It transports DNA and proteins along the axon and dendrites. A cell nucleus, which produces genetic data and activates protein synthesis, is part of the soma. These proteins are necessary for the normal functioning of other neuronal components. Dendrites pick up the signal from an adjacent neuron's synapse. These dendrites send the impulse to the soma, or nucleus, of the nerve cells. This area processes the electrical impulse before sending it on to the axon. The impulse is transported from the soma to the synapse by the axon, the longest branch among the dendrites. The second neuron's dendrites get the impulse from the synapse after that. As a result, the human brain develops into a complicated network of neurons.

## **2.2 STRUCTURE OF A NEURON**

A neuron mainly comprises of three main parts each of which is discussed below:

### **2.2.1. DENDRITES:**

Dendrites are finger-like cells found on the terminal of a neuron. Dendrites are two millimeters long on average. These are tiny, branching fibres that protrude from the nerve cell's cell body. This fibre expands the region that may be used to receive incoming data. Several cytoskeletal components, the Golgi apparatus, ribosomes, and smooth endoplasmic reticulum are found in dendrites and are involved in the activity of protein synthesis in the dendrites during signal transmission. Dendrites get information from other neurons or the body's nerves. These signals are sent through the nerve systems to the brain, which relays the information to various body components so that a response can occur.

### **2.2.2. CELL BODY OR SOMA:**

The nucleus and other organelles which play important role in the neuronal function are located in the soma and it is spherical in shape. As the soma contains the nucleus the soma it carries the genetic information. The nucleolus, which creates the ribosomes required for protein synthesis, is also a part of the nucleus. The endoplasmic reticulum and the Golgi apparatus are also found in the cell body in addition to ribosomes. Together, these organelles help synthesize,

package, and sort proteins for delivery to other cell components. The primary function of neurons is to take information from the cells via dendrites and transmit it to other cells. The cell body of bipolar neurons is in the center, while one axon and one dendrite protrude from each end. Instead of both attaching directly to the cell body, unipolar neurons have a projection that connects it to both the axon and the dendrite. The cell body of multipolar neurons is connected to a lengthy axon and several dendrites.

### **2.2.3. AXON:**

Axons, which are extremely thin nerve fibers, are responsible for transporting nerve impulses from one neuron (nerve cell) to another neuron. A human hair is often thicker than an axon. A neuron is in charge of processing and relaying the electrical signals involved in absorbing sensory input, providing motor instructions to your muscles, and controlling movement. One axon connects each neuron to other neurons, muscle cells, or glandular cells. The axon can carry messages more swiftly the larger its diameter. Axons that are generally found inside a myelin sheath can be found in the deepest region of the neuron.

### **2.3 SYNAPSE**

The synapse is the point at which two neurons connect. A synapse is the point at which two neurons come together, or between a neuron and a target or effector cell, such as a muscle cell. Signals that are electrical or chemical can be sent via it. Between presynaptic and postsynaptic neurons, the synapse is created. The connection between a neuron and a muscle is called the neuromuscular junction [2]. By a synapse, nerve impulses are sent from the axon terminal of one neuron to the dendrites of the next neuron. The biological mechanism through which a neuron interacts with a target cell across a synapse is known as synaptic transmission. The synapse is divided into two types based on the way impulses are transmitted - It could be chemical synapse or electrical synapse. During chemical synaptic transmission, a neurotransmitter is produced from the pre-synaptic neuron and attaches to certain post-synaptic receptors. In electrical synapse direct and passive current is allowed to flow from one neuron to another. Electrical synapses are specialised ion channels that link the pre- and postsynaptic cells and which therefore become a pathway which has low resistance so that there is an easy current flow between any two cells. On the other hand, in a chemical synapse, a presynaptic neuron's action potential causes the release of neurotransmitters, which then causes a current to flow through the postsynaptic cell.

### **2.3.1 ELECTRICAL SYNAPSE:**

Electrical synapses happen faster than chemical synapses the transmission of an electric signal across the electrical synapse the same to the conduction of an impulse in an axon. When presynaptic and postsynaptic neurons are near to one another, gap junctions occur. At the gap junction, proteins channels provide a structural connection between pre- and postsynaptic neurons. Since electrical synapses can't transition from excitatory to inhibitory impulses, they are less flexible than chemical synapses [2]. It is found in several lower invertebrates. In humans, it can be found in the spaces between glial cells.

### **2.3.2 CHEMICAL SYNAPSE:**

More synapses occur chemically. Neurotransmitters mediate the passage of nerve impulses through chemical synapses. The synaptic cleft is a fluid-filled void between the two neurons. One neuron cannot be followed by another by a nerve impulse. Axon terminals have a knob-like structure that houses synaptic vesicles [2]. Synaptic vesicles from the terminal of the presynaptic neuron release neurotransmitters at the synaptic cleft as soon as the action potential reaches the terminals. The postsynaptic membrane's receptors allow neurotransmitters to attach to them. This causes ions to flow and voltage-gated channels to open. The postsynaptic membrane's polarity is altered as a result, and the electric signal is sent across the synapse. Neurotransmitters come in both excitatory and inhibitory kinds. Multiple cell types can respond differently to the same neurotransmitter. The neurotransmitter is excitatory and produces the action potential if there is a net influx of positively charged ions inside the cell which is called as EPSP, or excitatory postsynaptic potential [3]. The membrane hyperpolarizes and acts as an inhibitory neurotransmitter when the membrane potential becomes more negative. IPSP, or inhibitory postsynaptic potential, is produced by them [3]. When neurotransmitters bind to the receptor, either enzymes act on them or they are removed and recycled to stop the signal after it has been transmitted forward.

### **2.4 ACTION POTENTIALS IN NEURONS:**

Neurons are electrically excitable, meaning that they respond to input by producing electrical impulses that propagate as action potentials throughout the cell and its axon. Changes in the cationic gradient (primarily sodium and potassium) across their plasma membranes generate and propagate these action potentials [4]. These action potentials eventually reach the axonal terminal and depolarize neighboring cells via synapses. This action is how these cells interact with one another, namely at synapses through synaptic transmission. Normally, the interior of

the cell is negative in comparison to its exterior. This is the resting membrane potential of approximately  $-60\text{mV}$ . When the negative inside potential reaches the threshold (less negative), a neuronal action potential is generated. This change in membrane potential opens a voltage-gated cationic channel (sodium channel), resulting in depolarization and the generation of the neuronal action potential. Neuronal action potentials are essential for impulse propagation along any nerve fiber, even over long distances. They are also necessary for neuronal communication via synapses. Disruption of this mechanism can have severe consequences, resulting in a lack of impulse generation and conduction, as demonstrated by various neurotoxins and demyelinating disorders.

The equilibrium potential for sodium is  $+55\text{ mV}$  at normal body temperature and  $-103\text{ mV}$  for potassium. The action potential is generated in three stages: (1) depolarization, a change in membrane potential from  $-60\text{ mV}$  to  $+40\text{ mV}$  caused primarily by sodium influx; (2) repolarization, a return to the membrane's resting potential caused primarily by potassium efflux; and (3) after-hyperpolarization, a recovery from a slight overshoot of the repolarization.

The first stage is directed by increased membrane permeability to salt. As a result, removing extracellular sodium or inactivating sodium channels stops action potentials from being generated. The neuron cannot create another action potential immediately after an action potential is generated; this is known as the absolute refractory time. The sodium channels are currently inactivated and remain closed, whilst the potassium channels remain open. The relative refractory period follows, during which the neuron can only generate an action potential with a significantly higher threshold. This opens when some sodium channels are ready to open and many are still inactive, while some potassium channels are also open.

The length of the refractory periods governs how quickly an action potential can be generated and propagated. The action potential propagates until it is terminated at a synapse, where it can either produce the release of neurotransmitters or the conduction of ionic currents. The latter takes place at electrical synapses, where presynaptic and postsynaptic cells interact without using neurotransmitters. Neurotransmitters, on the other hand, are the norm and are released at chemical synapses and neuromuscular junctions.

If the local currents generated by depolarization along a part of the neuronal membrane are sufficiently strong, they can depolarize neighboring segments of the membrane to the threshold, spreading the action threshold down the membrane and along the neuron's axon. The extent to which the initial local currents expand before causing more depolarizations is the

decisive element in the speed of this propagation. The electrical resistance of the membrane and the internal contents of the axon are two factors that influence this speed. Internal resistance is lower in wider axons, and having more voltage-gated sodium channels in the membrane reduces membrane resistance as well.

Slower action potential propagation is caused by higher internal resistance and lower membrane resistance. Because the body does not have adequate space, the nervous system uses glial cells, notably oligodendrocytes and Schwann cells, to wrap themselves around axons, generating myelin sheaths, to maximize propagation velocity. These sheaths help to increase membrane resistance by filling up gaps where channels might otherwise leak.

## **2.5 REFERENCES:**

[1] Olivia Guy Evans, An Easy Guide to Neuron Anatomy with Diagrams, <https://www.simplypsych>

[2] <https://byjus.com/neet/difference-between-electrical-and-chemical-synapse/>

[3] <https://collegedunia.com/exams/synapse-biology-articleid-1427>

[4] Chen I, Lui F. Neuroanatomy, Neuron Action Potential. [Updated 2022 Aug 8]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2023 Jan-. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK546639/>

## **CHAPTER -3**

### **NEURON MODELS**

#### **3.1 HODGKIN-HUXLEY NEURON MODEL**

In order to explain the ionic mechanisms underlying the initiation and spread of action potentials in the squid giant axon, Alan Hodgkin and Andrew Huxley proposed the model in 1952. For this work, they were awarded the 1963 Nobel Prize in Physiology or Medicine.

The Hodgkin-Huxley model, also known as the conductance-based model, is a mathematical model of the initiation and spread of action potentials in neurons. An imprecise representation of the electrical properties of excitable cells, such as neurons and muscle cells, can be found by a set of nonlinear differential equations. The Hodgkin-Huxley (HH) spiking neuron model mimics the action potential, ionic channels, and spiking behaviours of the neuron to replicate its dynamic properties [3]. It is useful for examining the sensitivity of neurons. In the neuromimetic chips, the Hodgkin-Huxley Model with automatic parameter estimation is used [4].

The original model's concept was that there are three different types of ion channels present in the nerve membrane, more specifically the membrane of the squid giant axon. The first, also referred to as the leakage channels, has a constant, relatively low conductance. Even though they have a low overall conductance, potassium (K) ions have a higher conductance than sodium (Na) ions do. The resting membrane potential is primarily caused by leakage channels. The action potential is produced by the other two types of ion channels, both of which are voltage-dependent, meaning that the voltage across the membrane affects how much current flows through them. There are two sets of voltage-dependent channels, one of which is permeable to Na ions alone and the other to K ions only. Each voltage-dependent channel can be visualised as a tunnel with a few gates placed inside of it sequentially. All the gates in a channel must be open simultaneously for that channel to be open and allow ions to flow through. The channel is closed if even one gate is closed. The individual gates randomly open and close at a high rate, but the chance that a gate will be open known as the open probability depends on the voltage applied across the membrane. The gates are believed to function as charge-carrying particles at the molecular level, and as a result, their position within the



membrane, which determines whether they are open or closed, is influenced by the electrical potential across the membrane. Channel gates comprises of two classes: activation gates, whose open probability rises with depolarization, and inactivation gates, whose open probability falls with depolarization. The activation variable for a gate is the possibility of that gate being open at any given moment. The proportion of gates in the total population of that class that are open is defined by the activation variable, which also determines the probability that a single gate of that class will be open. Gate classes vary not only in how their activation variables change in response to voltage changes but also in how quickly those changes occur.

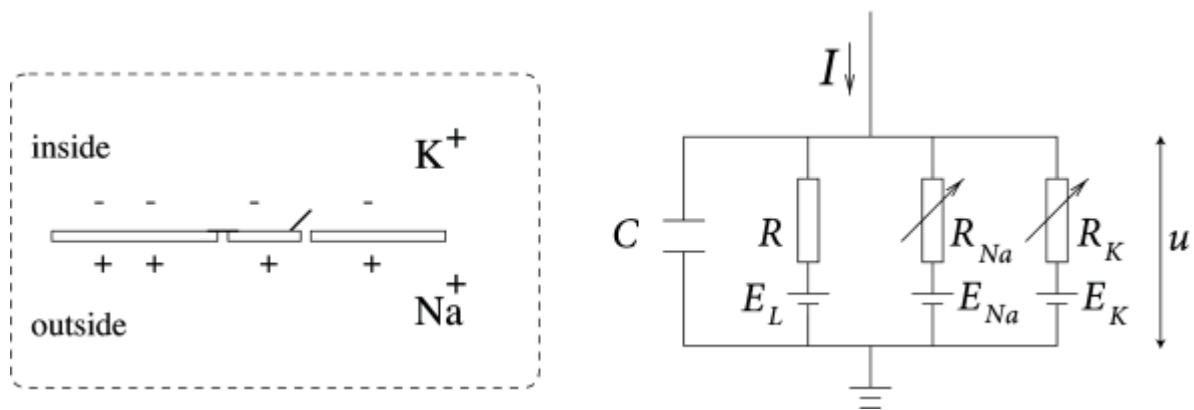


Figure1: Schematic representation of Hodgkin Huxley model [2]

HH Model can be explained with the help of the circuit given above. The semipermeable cell membrane serves as a capacitor and a barrier separating the interior of the cell from the extracellular fluid. A cell's input current  $I(t)$  may add to the capacitor's charge or leak out through the membrane's channels if it is introduced. A resistor represents each type of channel. The unspecific channel has a leak resistance of  $R$ , a resistance of  $R_{Na}$  for sodium channels, and a resistance of  $R_K$  for potassium channels. A battery is used to represent the Nernst potential caused by the difference in ion concentration. Each ion type has a different Nernst potential, and there are different batteries with the battery voltages  $E_{Na}$ ,  $E_K$ , and  $E_L$  for sodium, potassium, and the random third channel, respectively. According to the law of conservation of electric charge, an applied current  $I(t)$  may be divided into a capacitive current  $I_C$  that charges the capacitor  $C$  and additional components  $I_k$  that pass through the ion channels. Thus

$$I(t) = I_C(t) + \sum_k I_k(t)$$

$$C \frac{du}{dt} = - \sum_k I_k(t) + I(t)$$

Where  $u$  - voltage across the membrane

$\sum_k I_k$  - sum of the ionic currents which pass through the cell membrane

In short we can describe the HH model by the equation given below

$$\sum_k I_k = g_{Na} m^3 h (u - E_{Na}) + g_K n^4 (u - E_K) + g_L (u - E_L)$$

Where  $g_{Na}, g_K, g_L$  - Voltage dependent conductance

$E_{Na}, E_K, E_L$  - Reversal potentials

$m, n, h$  - Gating variables

The above-mentioned equations needed the numerical values for the unknown parameters to be given in before they could be used. It was necessary to identify the macro properties of the channel types such as ionic specificity, maximal conductances, and equilibrium potentials also we should determine how many activation and inactivation gates were present in each type of channel.

Amazingly, the HH model has been able to predict and describe a wide range of neuronal features. Numerous voltage-dependent channel types in addition to the basic HH pair have been included in this model's extensions, which have been utilised extensively in research throughout the globe.

### 3.2. INTEGRATE AND FIRE NEURON MODEL

Neuronal dynamics can be thought of as a summation process also known as a "integration" process in conjunction with a mechanism that initiates action potentials when a certain voltage is reached. Indeed, in studies, the moment the membrane potential reaches a certain threshold value from below is frequently characterised as the firing time. By using a formal threshold  $\vartheta$ , we define the critical voltage for spike initiation. We say that neuron  $i$  fires a spike if the voltage  $u_i(t)$ , which contains the sum of the effects of all inputs, reaches  $\vartheta$  from below. The firing time  $t_i^f$  is determined by the moment the threshold is exceeded. 'Integrate-and-Fire' models are a type of neuron model where action potentials are characterised as events. It is not attempted to define the appearance of an action potential. An equation that explains the evolution of the membrane potential  $u_i(t)$  and a mechanism to produce spikes are the two distinct parts of integrate-and-fire models that are both required to determine their dynamics. One of the most

extensively used models for evaluating the behaviour of neural systems is the integrate-and-fire neuron model. It explains a neuron's membrane potential in terms of synaptic inputs and the injected current that it receives. When the membrane potential hits a threshold, an action potential (spike) is created, but the actual changes associated with the membrane voltage and conductances generating the action potential are not included in the model. The synaptic inputs to the neuron are considered to be stochastic and are described as a temporally homogeneous Poisson process. Methods and results for both current synapses and conductance synapses are examined in the diffusion approximation, where the individual contributions to the postsynaptic potential are small. The focus of this review is upon the mathematical techniques that give the time distribution of output spikes, namely stochastic differential equations and the Fokker-Planck equation [1]. Because it can be evaluated analytically as well as capable of capturing many of the important properties of brain processing, the integrate-and-fire neuron model has become established as a classic model for the explanation of spiking neurons. The model's core is to separate the neuron's voltage fluctuations into two parts:

- 1) Below threshold, it is believed that the membrane operates passively (i.e. lacks voltage-dependent ion channels) and serves as a leaky capacitor whose voltage decays (or "leaks") to a resting level  $E_L$  (short for "E Leak ").
- 2) When the voltage attains the action potential threshold (due to injected currents charging up the membrane), the model assumes that the voltage spikes immediately to a hyperpolarized level  $V_{spike}$  and is then immediately reset [1].

The two components in the class of integrate-and-fire models are (i) a threshold for spike firing; (ii) a linear differential equation to describe the evolution of the membrane potential. Such a model is referred to as the Leaky Integrate -and- Fire model [2]. The fundamental electrical circuit for a leaky integrate-and-fire model is a capacitor C parallel to a resistor R, which is regulated by a current I(t) [5]. The variable  $u_i$  expresses the instantaneous value of membrane potential of neuron i. The potential is at its resting value in the absence of any input. The potential  $u_i$  will deviate from its resting value if an experimenter injects a current I(t) into the neuron or if the neuron gets synaptic signals from other neurons [2].

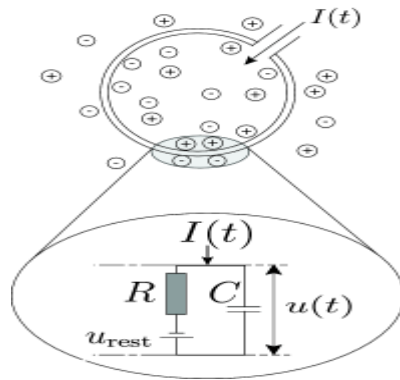


Figure2: A neuron receives a input current  $I(t)$  which inturn increases the electric charge inside the cell and the membrane of the cell acts like a capacitor parallel to resistor [2]

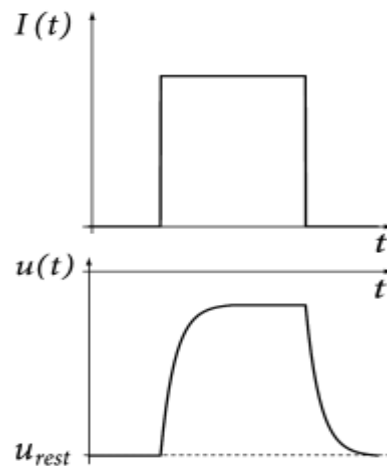


Figure3: The cell membrane reacts to a step current (top) with a smooth voltage trace (bottom)[2].

When the driving current  $I(t)$  disappears, the battery voltage  $u_{reset}$  determines the voltage across the capacitor. We divide the driving current into two components and utilise the rule of current conservation to analyse the circuit [2].

$$I(t) = I_R + I_C$$

$I_R$  – Resistive current through the resistor ( $I_R = \frac{u_R}{R}$ )

$u_R$  – Voltage across the resistor ( $u_R = u - u_{reset}$ )

The second term  $I_C$  changes the capacitor C

$$I_C = C \frac{du}{dt}$$

Thus, we can write from the above equations

$$I(t) = \frac{u(t) - u_{reset}}{R} + C \frac{du}{dt}$$

To this equation by multiplying R to each term we get a time constant  $\tau_m = RC$  thus we obtain a standard form

$$\tau_m \frac{du}{dt} = [u - u_{reset}] + R I(t)$$

The above equation is called the equation of a passive membrane [2].

The leaky Integrate and fire neuron model is quite simplified, and numerous components of neural dynamics are ignored. In particular, input, which may come from current injection or presynaptic neurons, is linearly integrated regardless of the state of the postsynaptic neuron. Also, it does not store the memory of the previous spike which is also considered as a limitation of the leaky integrate and fire neuron [2]. The integrate and fire neuron model is also computationally inefficient also generates the spikes accurately but it helps in the study of a large number of neurons which is one advantage of this model. Adaptations are not taken into account by the typical leaky integrate-and-fire paradigm. However, if mechanisms of adaptation improve the voltage dynamics of the leaky integrate-and-fire model, it can be an effective tool for precisely predicting the spike times of cortical neurons [2].

### **3.3. IZHKEVICH NEURON MODEL:**

The Izhikevich neuron model is the model of our interest. This model is developed by combining only the effective parts of the HH model and the integrate and fire neuron model making this model more biologically plausible as well as computationally effective. It is a two-dimensional continuous neuron model that is computationally simpler in that it can model tens of thousands of neuron networks in a very short amount of time biologically similar to the Hodgkin-Huxley model, and much simpler. The Izhikevich neuron model is effective at capturing the dynamics of neurons and exhibits a variety of fascinating spiking and bursting characteristics. While the cortically inhibitory interneurons are divided into (a) fast spiking (b) low threshold (c) spiking, the cortically excitatory neurons are divided into (a) regular spiking (b) inherently bursting (c)

chattering. As the system's parameters are changed, the IZH neuron model exhibits all of the mentioned spiking and bursting characteristics. The model also explains how thalamo-cortical neurons, one of the primary sources of input to the cortex, behave. The model can also display the kinetics of resonator neurons. The more details about the Izhikevich neuron model and its fundamental equations are described in the next chapter.

### 3.4 REFERENCES

[1] Burkitt AN. A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input. *Biol Cybern.* 2006 Jul;95(1):1-19. doi: 10.1007/s00422-006-0068-6. Epub 2006 Apr 19. PMID: 16622699.

[2] Wulfram Gerstner, Werner M Kistler, Richard Naud and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition.* 2014 July. <https://neurondynamics.epfl.ch/online/>

[3] Husbands, Philip, Owen Holland, and Michael Wheeler (eds), 'An Interview with Horace Barlow', in Phil Husbands, Owen Holland, and Michael Wheeler (eds), *The Mechanical Mind in History* (Cambridge, MA, 2008; online edn, MIT Press Scholarship Online, 22 Aug. 2013), <https://doi.org/10.7551/mitpress/9780262083775.003.0018>, accessed 1 June 2023.

[4] Hodgkin-Huxley model, Wikipedia- The free encyclopedia:

[https://en.wikipedia.org/wiki/Hodgkin%E2%80%93Huxley\\_model](https://en.wikipedia.org/wiki/Hodgkin%E2%80%93Huxley_model)

[5] Alan Guth, The Early Universe, Assignment 3 - MIT OpenCourseWare PROBLEM SET 2 September 24, 2013 [https://ocw.mit.edu/courses/8-286-the-early-universe-fall-2013/c4d7d64191100b82f4e4845f2af9c719/MIT8\\_286F13\\_ps3.pdf](https://ocw.mit.edu/courses/8-286-the-early-universe-fall-2013/c4d7d64191100b82f4e4845f2af9c719/MIT8_286F13_ps3.pdf)

## CHAPTER 4

### IZHIKEVICH NEURON MODEL

#### **4.1 INTRODUCTION**

A central topic of theoretical neuroscience research is to obtain computationally and/or analytically tractable models for understanding the neural dynamics that underpin brain disorders, such as epilepsy or Parkinson's disease etc. The dynamics associated with such functions or disorders are mainly resulted from the coordinated activity of large populations of interconnected neurons [1]. Neural mass models rooted in the mean field theory aims to describe the collective activity of a neural network in terms of mean-field variables such as the population firing rate and mean membrane potential. Spiking neural models involve variables closely related to biological measurements and realizations. The QIF model [2] is a popular model for large network simulations as it only has one equation for each neuron. Further, it can be considered a canonical model as any Class I excitable system close enough to the onset of oscillations can be transformed into this form. While the QIF model is useful, there are many behaviours of spiking neurons it cannot reproduce. To address this, several authors have developed two-dimensional integrate-and-fire model neurons. Examples include the Izhikevich neuron and the adaptive exponential (AdEx) neuron. These models display spike frequency adaptation (SFA) through a recovery variable and are capable of generating a variety of spiking dynamics reported in real neuron. The SFA mechanism can improve neural coding and computation at a lower metabolic cost and has also been demonstrated to be important in the emergence of network bursting and synchronization. Thus, derivation of mean-field descriptions for networks of neurons with SFA would be extremely valuable. Since the Izhikevich neuron is the most closely linked to the QIF neuron, it is the ideal candidate to explore emergent neurodynamics of neural networks through mean-field descriptions. The Izhikevich neuron model consists of a fast subsystem based on the QIF model and a slow subsystem modelling the adaptation mechanism. Thus, the Izhikevich network can be considered as a QIF network extended by SFA. In this article, we show it is feasible to extend the Lorentzian ansatz for the phase model, to the derivation of the exact mean-field models for a network of Izhikevich neurons. To achieve a closed set of mean-field equations, we turn to the population density approach for spiking neurons. The quasi-steady approximation for the continuity equation will be replaced with the Lorentzian ansatz to drop the assumption of

separation of time scales. The moment closure approach will be deployed to release the dependence of the adaptation variable on the membrane potential to help close the mean-field system.

## 4.2 THE NETWORK SYSTEM

The Hodgkin-Huxley type neuron model, which is biophysically accurate, was reduced by bifurcation analysis [3] to produce the Izhikevich model. With an adaptable quadratic mechanism, the central characteristics of neuronal activity are still present. The following discontinuous ordinary differential equations eq (1) describe the network model for a population of Izhikevich neurons (ODEs).

$$v'_k = v_k(v_k - \alpha) - \omega_k + \eta_k + I_{ext} + I_{syn,k} \quad (1a)$$

$$\omega'_k = a(bv_k - \omega_k) \quad (1b)$$

$$\text{If } v_k \geq v_{peak}, \text{ then } v_k \leftarrow v_{reset} \text{ and } \omega_k \leftarrow \omega_k + \omega_{jump} \quad (1c)$$

for  $k = 1, 2, \dots, N$ . Here,  $v_k(t)$  is the membrane potential of  $k^{\text{th}}$  neuron and  $\omega_k$  is the recovery current, which serves as an adaptation variable. The parameter  $\eta_k$  is the intrinsic current while  $I_{ext}$  is the external common current. We will assume that  $\eta_k$  are heterogeneous and drawn from a distribution  $\mathcal{L}(\eta)$  defined on  $(-\infty, \infty)$ . The term  $I_{syn}$  represents the total synaptic current due to the other neurons in the network. When the voltage reaches a cut off value  $v_{peak}$ , considered to be the peak of a spike, it is reset to the value  $v_{reset}$ . At the same time, the adaptation variable jumps by an amount  $\omega_{jump}$ , which affects the after-spike behavior. Neurons in the network are connected by synapses modelled by

$$I_{syn,k} = g_{syn} s (e_r - v_k) \quad (2)$$

where  $e_r$  the reversal potential and  $g$  is the maximum synaptic conductance. The synaptic gating variable  $s$  lies between 0 to 1, and represents the proportion of ion channels open in the postsynaptic neuron as the result of the firing in presynaptic neurons. The mechanism of synaptic transmission can be formally described by a linear system of ODEs with a sum of delta pulses corresponding to the times a neuron fires a spike. For example, the single exponential synapse is modeled by

$$s' = -\frac{s}{T_s} + \frac{s_{jump}}{N} \sum_{k=1}^N \sum_{t_k^j < t} \delta(t - t_k^j) \quad (3)$$



where  $\delta(t)$  is the Dirac delta function, and represents the time of the  $j^{th}$  spike of the  $k^{th}$  neuron

### 4.3 MEAN-FIELD REDUCTION

The network model described in the previous section is too complicated to perform tractable analysis especially when the number of neurons is large. In this section, we will develop a low-dimensional mean-field model to approximate the behaviour of the full network described by (1)-(3) within the thermodynamic limit, i.e., when  $N \rightarrow \infty$ . The mean-field approximation is essentially a technique that borrows concepts and methods from statistical physics, e.g., the population density approach [4], the continuity equation (or the Fokker-Planck equation when the system is subject to noise) [1, 5]. We will show how to describe some vital macroscopic variables such as the population firing rate and how to derive the reduced macroscopic dynamics, cast as ODEs, through step-by-step assumptions.

### 4.4 GENERAL MEAN-FIELD DESCRIPTION

We define the population density function  $\rho(t, v, \omega, \eta)$  [5,6] as the density of neurons at a point  $(v, \omega)$  in phase space and parameter  $\eta$  at time  $t$ . In the limit as  $N \rightarrow \infty$ , the principle of conservation mass leads to the following evolution equation for the density function, that is, the continuity equation,

$$\frac{\partial}{\partial t} \rho(t, v, \omega, \eta) + \nabla \cdot \mathcal{J}(t, v, \omega, s, \eta) = 0 \quad (4)$$

The probability flux  $\rho$  is intuitively the mass flow rate along a specific direction in phase space [6].

Next, we describe several macroscopic observables in terms of mean-field description, which are extremely useful in understanding neural activities underlying brain function. The population firing rate is the flux through the threshold  $v_{peak}$  over the entire range of  $w$  in phase space and  $\eta$  in parameter space, defining

$$r(t) = \int_{\eta} \int_{\omega} J^v(t, v_{peak}, \omega, s, \eta) d\omega d\eta \quad (5)$$

The mean membrane potential is defined as

$$\langle v(t) \rangle = \int \int \int_{\eta \ \omega \ v} v \rho(t, v, \omega, \eta) dv d\omega d\eta \quad (6)$$

Additionally, we define the mean adaptation current over the population as

$$\langle \omega(t) \rangle = \int \int \int_{\eta \ v \ \omega} \omega \rho(t, v, \omega, \eta) d\omega dv d\eta \quad (7)$$

Then, we approximate its derivative with respect to t, yielding

$$\langle \omega \rangle' \approx \langle G^\omega(v, \omega) \rangle + \int \int_{\eta \ \omega} \omega_{jump} J^v(t, v_{peak}, \omega, s, \eta) d\omega d\eta \quad (8)$$

Further, considering the linearity of G function with respect to v and w, and the description of the population firing rate in terms of flux, we finally derive ODE describing the evolution of the mean adaptation variable and the synaptic dynamics of the firing rate.

#### 4.5 DENSITY FUNCTION IN CONDITIONAL FORM

In this section, we take advantage of the population density approach and the moment closure assumption to reduce the dependence between the macroscopic variables [7]. We begin by taking the population density function in the conditional form and then the population firing rate in the general expression (7) is described by the conditional probability as

$$r(t) = \lim_{v \rightarrow v_{peak}} \int_{\eta} L(\eta) \rho^v(t, v|\eta) G^v(v, \langle \omega|v, \eta \rangle, s, \eta) d\eta \quad (9)$$

Next, we assume

$$\langle \omega|v, \eta \rangle = \langle \omega|\eta \rangle \quad (10)$$

which corresponds to a first order moment closure assumption. Then, we have

$$r(t) = \lim_{v \rightarrow v_{peak}} \int_{\eta} L(\eta) \rho^v(t, v|\eta) G^v(v, \langle \omega|\eta \rangle, s, \eta) d\eta \quad (11)$$

Similarly, the mean membrane potential is rewritten as

$$\langle v(t) \rangle = \int_{\eta} L(\eta) \int_{v} v \rho^v(t, v|\eta) dv d\eta \quad (12)$$

where we use the normalization condition on the marginal density of  $\omega$ . Furthermore, we integrate the general continuity equation (4) with respect to  $\omega$  and use (13), yielding

$$\frac{\partial}{\partial t} \rho^v(t, v|\eta) + \frac{\partial}{\partial v} [G^v(v, \langle \omega|v, \eta \rangle, s, \eta) \rho^v(t, v|\eta)] = 0 \quad (13)$$

To obtain this expression, we used the normalization condition on the marginal density of  $\omega$  and the fact that the flux vanishes on the boundary  $\partial\omega$ . Finally, using the moment closure assumption (15), we obtain the resulting modified continuity equation

$$\frac{\partial}{\partial t} \rho^v(t, v|\eta) + \frac{\partial}{\partial v} [G^v(v, \langle \omega|\eta \rangle, s, \eta) \rho^v(t, v|\eta)] = 0 \quad (14)$$

#### 4.6 LORENTZIAN ANSATZ

In this section, we will further simplify the expressions of the macroscopic variables  $r(t)$  and  $\langle v(t) \rangle$  and derive the mean-field approximation for the Izhikevich network by employing the Lorentzian ansatz [2]. To begin, we assume that the conditional probability  $\rho^v(t, v|\eta)$  and hence can be written in the form of Lorentzian distribution as follows,

$$\rho^v(t, v|\eta) = \frac{1}{\pi} \frac{x(t, \eta)}{[v - y(t, \eta)]^2 + x^2(t, \eta)} \quad (15)$$

where  $x(t, \eta)$  and  $y(t, \eta)$  are two time-dependent parameters defining half-width at half-maximum and location of the center, respectively. Moreover,  $y(t, \eta)$  is defined via the Cauchy principal value since the Lorentz distribution only has a mean in principal value sense.

#### 4.7 HETEROGENEITY WITH LORENTZIAN DISTRIBUTION

Further derivation of the mean-field description in terms of the macroscopic observables depends on the distribution of the heterogeneous parameter,  $\eta$  [8]. Specifically, we choose the heterogeneous current  $\eta$  to have a Lorentzian distribution with center  $\eta$  and half-width at half-maximum  $\Delta\eta$ . Then, we apply the residue theorem to compute the integrals in to obtain

$$r(t) = \frac{1}{\pi} x(t, \bar{\eta} - i\Delta_\eta) \quad (16)$$

$$\langle v(t) \rangle = y(t, \bar{\eta} - i\Delta_\eta) \quad (17)$$

If  $\mathcal{L}(\eta)$  has  $n$  poles in the lower half  $\eta$ -plane, one can readily obtain  $n$  sets of complex-valued mean-field ODEs. Recalling that we already obtained the dynamical system for the mean adaptation current and synapses, we obtain the reduction of the network of Izhikevich neurons (1)-(3) to the following the mean-field system of ODE

$$r' = \frac{\Delta\eta}{\pi} + 2r\langle v \rangle - (\alpha + g_{syn}s)r \quad (18)$$

$$\langle v \rangle' = \langle v \rangle^2 - \alpha\langle v \rangle - \langle \omega \rangle + \bar{\eta} + I_{ext} + g_{syn}s(e_r - \langle v \rangle) - \pi^2 r^2 \quad (19)$$

$$\langle \omega \rangle' = a(b\langle v \rangle - \langle \omega \rangle) + \omega_{jump}r \quad (20)$$

$$s' = -s/\tau_s + s_{jump}r \quad (21)$$

#### 4.8 NUMERICAL ANALYSIS

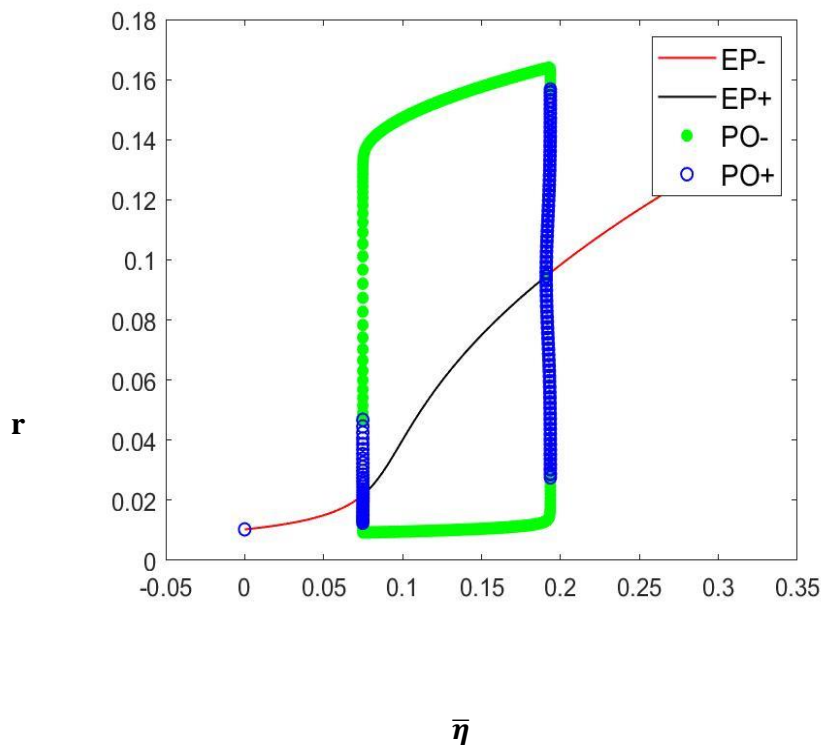
We now numerically examine the dynamics of the mean-field model and demonstrate its validity in terms of reproducing the macroscopic dynamics of the network of Izhikevich neurons. We consider an all-to-all coupled network with synapses governed by the single exponential model [4,7,9].

The parameter values used in all simulations can be found in Table 1

PARAMETER	VALUE	PARAMETER	VALUE
$\alpha$	0.6215	$\tau_s$	2.6
$g_{syn}$	1.2308	$e_r$	1
a	0.0077	b	-0.0062
$s_{jump}$	1.2308	$\omega_{jump}$	0.0189
$v_{peak}$	200	$v_{reset}$	-200

Table 1: Dimensionless parameters for Izhikevich neurons

We begin with the bifurcation analysis of the mean-field model (18-21). Fig. 1 shows how the population firing rate  $r$  qualitatively changes as the mean intrinsic current  $\bar{\eta}$  is varied. There are two subcritical Andronov-Hopf bifurcations (HP) at  $\bar{\eta} = \bar{\eta}_{HP} \approx 0.191$  and  $0.07$  respectively. The system displays two small ranges of bistability between the Hopf and SNLC bifurcations. The stable limit cycles (green dots) correspond to bursting solutions in the full network and stable equilibrium points (red lines) correspond to the tonic firing. This is clearly reflected in the time series of macroscopic in Fig. 2 and Fig. 3. The mean-field equations (18-21) exactly predict the behaviour of the full network, including the damped oscillations and the frequency of stable oscillation [4,7,9]. Prior work has shown that population bursting in the networks of Izhikevich neurons is due to a balance between the inputs (intrinsic and external applied currents and synaptic inputs), which cause the neurons to spike, and the slow adaptation current, which can terminate spiking [10]. For a given level of adaptation there must be sufficient input, but not too much. Hence the bursting in Fig. 1 occurs when the mean intrinsic current  $\bar{\eta}$  is not too small and not too large.



*Figure 1 Bifurcation diagram of the mean-field model with respect to the mean intrinsic current  $\bar{\eta}$ . The red (black) lines correspond to stable (unstable) equilibrium points and green (blue) dots corresponds to stable (unstable) limit cycles.*

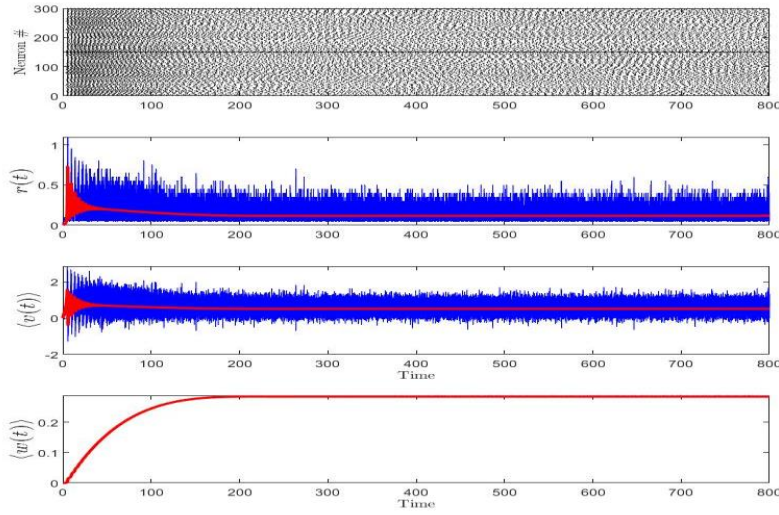


Figure 2

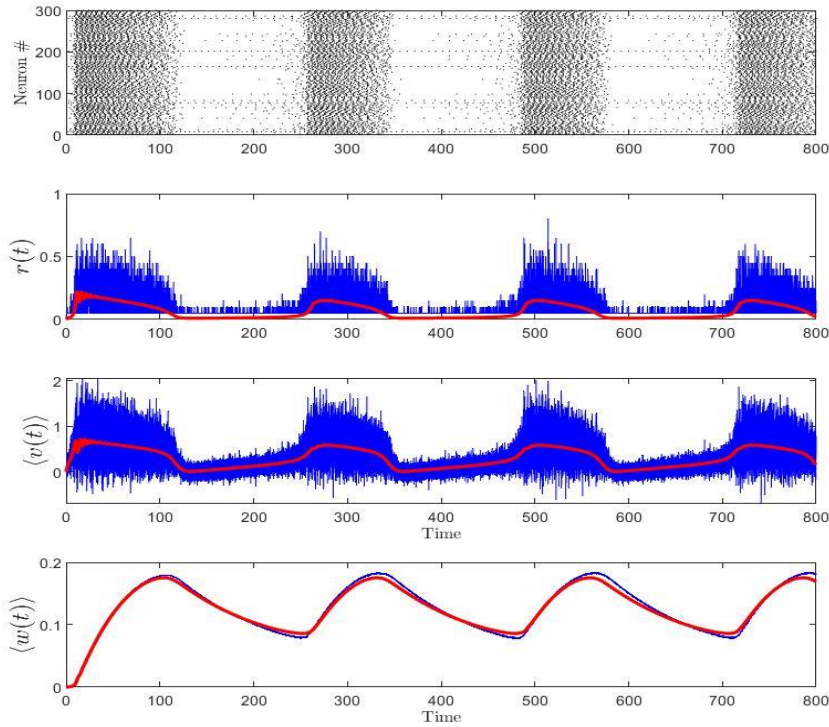


Figure 3

Fig 2 & Fig. 3: Comparison of the temporal behaviour of the network of Izhikevich neurons and the corresponding mean-field model when  $\bar{\eta}=0.25$  and  $\bar{\eta}=0.12$ . First row of panels are the raster plots of 300 randomly selected Izhikevich neurons from the  $N=10^4$  in the population. Other rows of panels show, respectively, the population firing rate  $r(t)$ , mean membrane potential  $\langle v(t) \rangle$  and mean adaptation variable  $\langle \omega(t) \rangle$  obtained from simulations of the full network (blue) and the mean-field model (red). Parameters:  $\Delta\eta=0.02$  and  $I_{ext}=0$

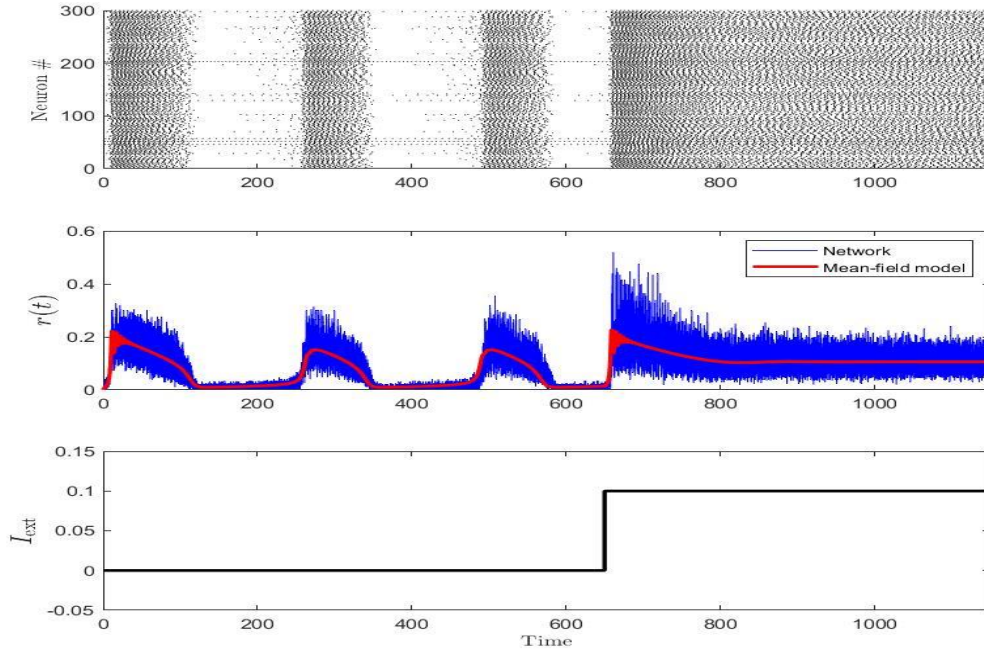


Fig. 4 shows the dynamics of the network of Izhikevich neurons compared with the corresponding mean-field model when  $\bar{\eta} = 0.12$  and  $\Delta\eta = 0.02$ . First panel is the raster plot of the spiking activity. The instantaneous population firing rate of the network and the mean-field model are depicted in blue and red respectively. Stimulus  $(t)$  is shown in the last panel. At time  $t = 650$  a current  $= 0.1$  is applied to all neurons

#### 4.9 EXTENSION TO TWO-COUPLED POPULATIONS

A large-scale neural network can also be thought of as numerous connected groups when different properties of the network's cells are taken into account. For example, neurons are classified as excitatory or inhibitory depending on the type of synapses they create, as in [11,12], or as strongly or weakly adapting based on the amount of spike frequency adaptation they exhibit, as in [9,13]. In this section, we analyse a network of all-to-all coupled strongly adapting neurons (population p) and weakly adapting neurons (population q). Izhikevich model characterizes each neuron as

$$v'_{m,k} = v_{m,k}(v_{m,k} - \alpha_m) - \omega_{m,k} + \eta_{m,k} + I_m^{ext} + I_{m,k}^{syn} \quad (22)$$

$$\omega'_{m,k} = a_m(b_m v_{m,k} - \omega_{m,k}) \quad (23)$$

$$v_{m,k} \geq v_m^{peak}, v_{m,k} \leftarrow v_m^{reset}, \omega_{m,k} + \omega_m^{jump} \quad (24)$$

where  $m = p, q$  represents the two populations with  $N_p$  and  $N_q$  cells, respectively. The subscript  $\{m, k\}$  denotes the  $k$ th neuron in population  $m$ . The subscript with only  $\{m\}$  represents the corresponding parameter is homogeneous within the population  $m$ , but heterogeneous across the two populations. We require two maximal synaptic conductances,  $g_{p,p}^{syn}$  and  $g_{q,q}^{syn}$  within the populations and two,  $g_{p,q}^{syn}$  and  $g_{q,p}^{syn}$  between the populations. Then, we have

$$I_{p,k}^{syn}(t) = [kg_{p,p}^{syn}s_p + (1-k)g_{p,q}^{syn}s_q](e_p^r - v_{p,k}) \equiv G_p S_p (e_p^r - v_{p,k}) \quad (25)$$

$$I_{q,k}^{syn}(t) = [kg_{q,p}^{syn}s_p + (1-k)g_{q,q}^{syn}s_q](e_q^r - v_{q,k}) \equiv G_q S_q (e_q^r - v_{q,k}) \quad (26)$$

where  $k = \frac{N_p}{N_p + N_q}$  is the proportion of strongly adapting neurons in the network and  $s_p$  (respectively,  $s_q$ ) represents the proportion of open synapses due to neurons in the strongly (respectively, weakly) adapting population. These gating variables are governed by the single exponential synapse model

$$s'_m = -\frac{s_m}{\tau_m} + \frac{s_m^{jump}}{N_m} \sum_{k=1}^{N_m} \sum_{t_{m,k}^j < t} \delta(t - t_{m,k}^j), m = p, q. \quad (27)$$

We apply Lorentzian ansatz for two populations like that and of the previous and assuming the Lorentzian distribution of heterogeneous currents for two populations as

$$L(\eta_m) = \frac{1}{\pi} \frac{\Delta_m^\eta}{(\eta_m - \eta_m)^2 + (\Delta_m^\eta)^2}, m = p, q. \quad (28)$$

In case of two population, we obtain eight differential equations. Three differential equations each for both population and two for the synaptic conductances.

Three differential equations for the population  $p$  with strong adaptation is given by

$$r'_p = \Delta_p^\eta / \pi + 2r_p \langle v \rangle_p - r_p [G_p S_p + \alpha_p] \quad (29)$$

$$\langle v \rangle'_p = \langle v \rangle_p^2 - \alpha_p \langle v \rangle_p - \langle \omega \rangle_p + \bar{\eta}_p + I_p^{ext} + G_p S_p [e_p^r - \langle v \rangle_p] - \pi^2 r_p^2 \quad (30)$$

$$\langle \omega \rangle'_p = a_p [b_p \langle v \rangle_p - \langle \omega \rangle_p] + \omega_p^{jump} r_p \quad (31)$$



And for population q with weak adaptation is given by

$$r'_q = \Delta_q^n / \pi + 2r_q \langle v \rangle_q - r_q [G_q S_q + \alpha_q] \quad (32)$$

$$\langle v \rangle'_p = \langle v \rangle_p^2 - \alpha_p \langle v \rangle_p - \langle \omega \rangle_p + \overline{\eta}_p + I_p^{ext} + G_p S_p [e_p^r - \langle v \rangle_p] - \pi^2 r_p^2 \quad (33)$$

$$\langle \omega \rangle'_p = a_p [b_p \langle v \rangle_p - \langle \omega \rangle_p] + \omega_p^{jump} r_p \quad (34)$$

Now they couple through synaptic currents and is given by

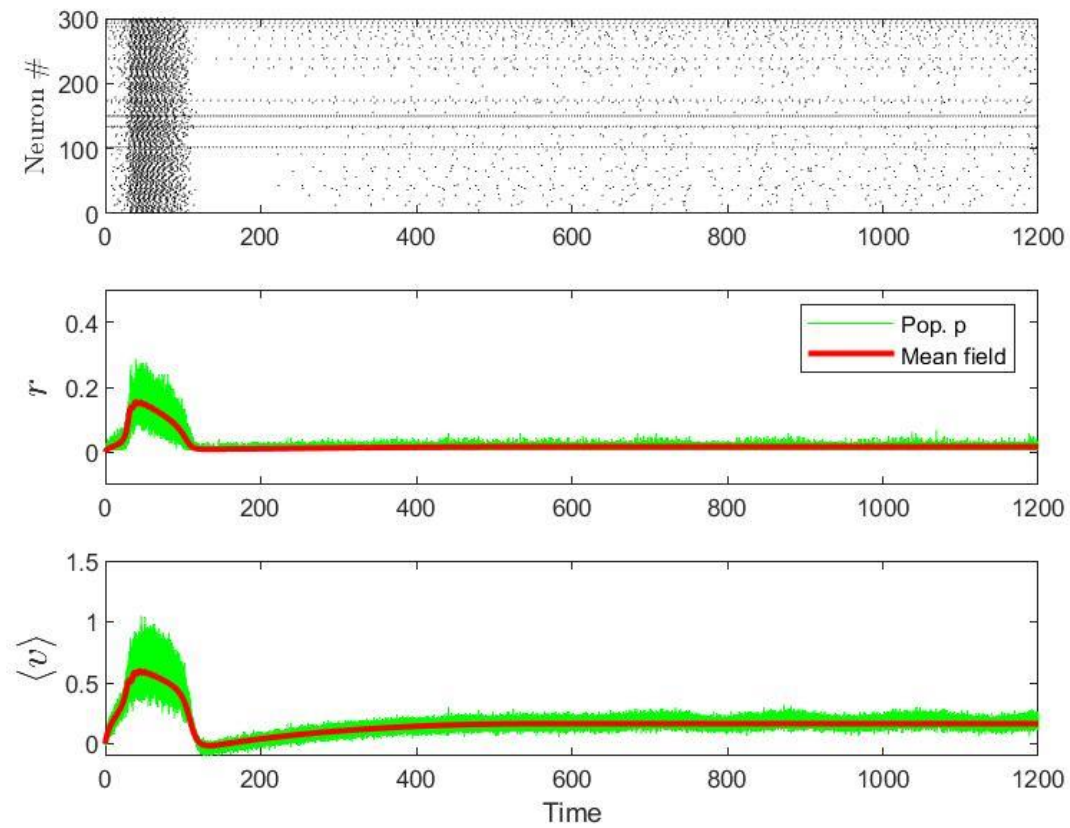
$$s'_p = -s_p / \tau_p^s + s_p^{jump} r_p \quad (35)$$

$$s'_q = -s_q / \tau_q^s + s_q^{jump} r_q \quad (36)$$

Most parameters used in two-coupled population is same as that of one population. The only parameter that differs are after-spike jump sizes  $\omega_m^{jump}$  and time constant  $a_m$

PARAMETER	VALUE	PARAMETER	VALUE
$\omega_p^{jump}$	0.0189	$\omega_q^{jump}$	0.0095
$a_p$	0.0077	$a_q$	0.077
$g_{p,p}^{syn}$	1.2308	$g_{q,q}^{syn}$	1.2308
$g_{p,q}^{syn}$	1.2308	$g_{q,p}^{syn}$	1.2308
$I_p^{ext}$	0	$I_q^{ext}$	0
$N_p$	8000	$N_q$	2000

Table 2: Dimensionless parameters for the two-couple population



*Fig.5 This network consists of 8000 strongly adapting neurons of population  $p$ . This is the behaviour of population when  $\bar{\eta} = 0.08$ . The first row of panels shows the raster plots of the spiking activity, the green line shows the population  $p$  and the red shows the corresponding variables in mean-field models. Parameters used are given in the table above and  $\Delta_p^\eta = 0.02$ .*

The figure shows that the dynamics of the two-population network are exactly described by the reduced mean-field description in the tonic firing (equilibrium points) and bursting (periodic orbits) regimes. The mean-field model for the network of two coupled Izhikevich populations involves more complicated bifurcations compared with that of the single-population network of strongly adapting Izhikevich neurons. Bifurcation analysis reveals that when the proportion of strong adapting neurons is  $\kappa = 0.8$ , the sequence of bifurcation is largely the same as when there is a single population of strongly adapting neurons

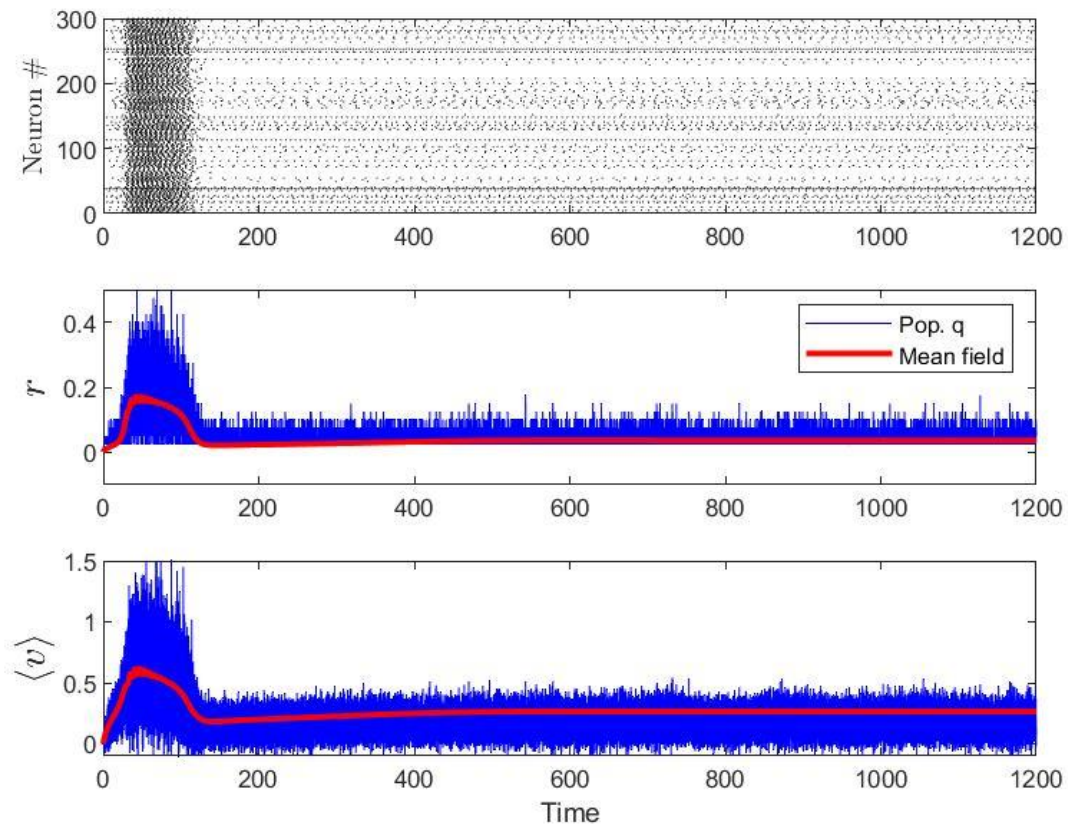


Fig.6 This network consists of 2000 strongly adapting neurons of population  $q$ . This is the behaviour of population when  $\bar{\eta} = 0.08$ . The first row of panels shows the raster plots of the spiking activity, the blue line shows the population  $p$  and the red shows the corresponding variables in mean-field models. Parameters used are given in the table above and  $\Delta_q^\eta = 0.02$ .

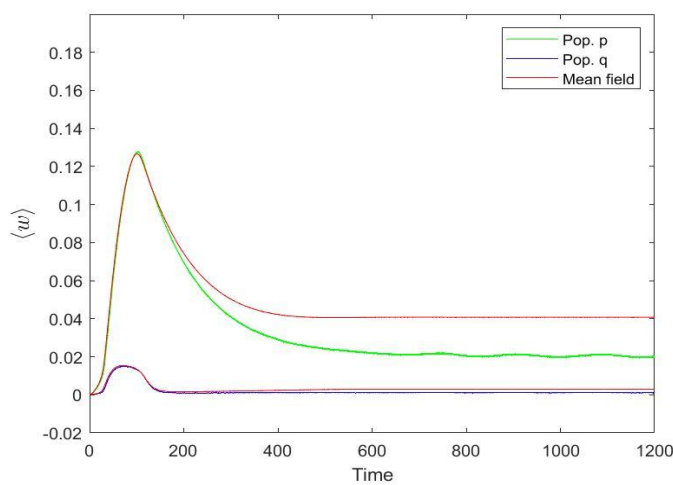


Fig.7 This shows the variation of mean-adaptation variable with time for 8000 neurons of population  $p$  and 2000 neurons of population  $q$ . The green and blue lines show the population  $p$  and  $q$  respectively. The red line shows the corresponding mean-field variable for each population

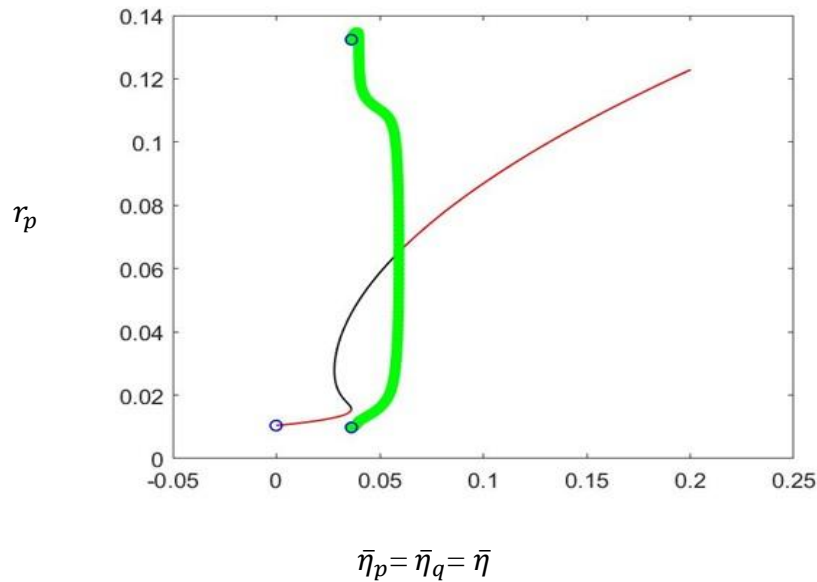


Fig 8. Figure shows the qualitative changes of the population firing rate  $r_p$  with respect to  $\bar{\eta}_p = \bar{\eta}_q = \bar{\eta}$  when  $\kappa$  is 0.5

This figure shows the one parameter bifurcation when the  $\kappa$  value is reduced to 0.5. The red and black lines correspond to the stable and unstable equilibrium points respectively. When  $\kappa = 0.5$  the system undergoes two saddle-node bifurcation and one Andronov-Hopf bifurcation. We can also see that the stable period behaviour is now initiated by what appears to be a saddle-node on an invariant circle bifurcation or homoclinic bifurcation and terminated by a supercritical Andronov-Hopf bifurcation at  $\bar{\eta}_p = \bar{\eta}_q \approx 0.05$ .

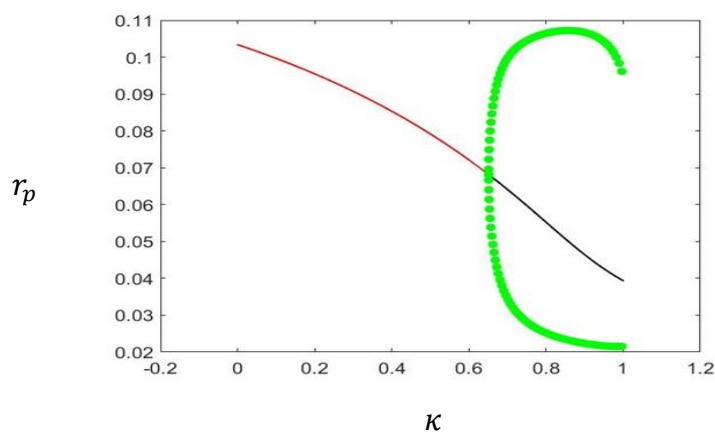


Fig 9. Indicates that stable bursting behaviour is more likely in a network with the higher proportion of strongly adapting neurons

#### **4.10 DISCUSSION**

We have derived a mean field model for a network of heterogeneous Izhikevich neurons which display spike frequency adaptation through a recovery variable. The mean-field models [2,10] have exhibited qualitative and quantitative agreement with the full network. The parameter values used in the numerical examples are a nondimensionalization of parameter values fit to actual neuronal data collected in the literature. Bifurcation analysis for the mean field system can be used to make predictions about the biological networks being studied. For example, how to understand the emergence of bursting in the CA3b region of the hippocampus based on experimental findings of neurons which display different amounts of spike frequency adaptation.

#### 4.11 REFERENCES

- [1] G. Deco, V. K. Jirsa, P. A. Robinson, M. Breakspear, K. Friston, The dynamic brain: from spiking neurons to neural masses and cortical fields, *PLOS Computational Biology* 4 (8) (2008) e1000092– e1000092. doi:<https://doi.org/10.1371/journal.pcbi.1000092>.
- [2] E. Montbrio, D. Pazó, A. Roxin, Macroscopic description for networks of spiking neurons, *Physical Review X* 5 (2015) 021028. doi:<https://doi.org/10.1103/PhysRevX.5.021028>.
- [3] E. M. Izhikevich, *Dynamical Systems in Neuroscience : the Geometry of Excitability and Bursting*, Computational neuroscience, MIT Press, Cambridge, Mass, 2007.
- [4] C. Ly, D. Tranchina, Critical analysis of dimension reduction by a moment closure method in a population density approach to neural network modeling, *Neural Computation* 19 (8) (2007) 2032–2092. doi:<https://doi.org/10.1162/neco.2007.19.8.2032>.
- [5] C. Bick, M. Goodfellow, C. R. Laing, E. A. Martens, Understanding the dynamics of biological and neural oscillator networks through exact mean-field reductions: a review, *Journal of Mathematical Neuroscience* 10 (2020) 9. doi:<https://doi.org/10.1186/s13408-020-00086-9>.
- [6] R. Gast, T. R. Knöösche, H. Schmidt, Mean-field approximations of networks of spiking neurons with short-term synaptic plasticity, *Physical Review. E* 104 (4-1) (2021) 044310–044310. doi:<https://doi.org/10.1103/PhysRevE.104.044310>.
- [7] W. Nicola, S. A. Campbell, Mean-field models for heterogeneous networks of two-dimensional integrate and fire neurons, *Frontiers in Computational Neuroscience* 7 (2013) 184. doi:<https://doi.org/10.3389/fncom.2013.00184>.
- [8] E. Ott, T. M. Antonsen, Low dimensional behavior of large systems of globally coupled oscillators, *Chaos* 18 (3) (2008) 037113. doi:<https://doi.org/10.1063/1.2930766>.
- [9] W. Nicola, S. A. Campbell, Bifurcations of large networks of two-dimensional integrate and fire neurons, *Journal of Computational Neuroscience* 35 (1) (2013) 87–108. doi:<https://doi.org/10.1007/s10827-013-0442-z>.
- [10] M. Dur-e-Ahmad, W. Nicola, S. A. Campbell, F. K. Skinner, Network bursting using experimentally constrained single compartment CA3 hippocampal neuron models with adaptation, *Journal of Computational Neuroscience* 33 (1) (2012) 21–40. doi:<https://doi.org/10.1007/s10827-011-0372>.

- [11] H. R. Wilson, J. D. Cowan, Excitatory and inhibitory interactions in localized populations of model neurons, *Biophysical Journal* 12 (1) (1972) 1–24. doi:[https://doi.org/10.1016/S0006-3495\(72\)86068-5](https://doi.org/10.1016/S0006-3495(72)86068-5).
- [12] G. Dumont, B. Gutkin, Macroscopic phase resetting-curves determine oscillatory coherence and signal transfer in inter-coupled neural circuits, *PLoS computational biology* 15 (5) (2019) e1007019–e1007019. doi:<https://doi.org/10.1371/journal.pcbi.1007019>.
- [13] P. Hemond, D. Epstein, A. Boley, M. Migliore, G. A. Ascoli, D. B. Jaffe, Distinct classes of pyramidal cells exhibit mutually exclusive firing patterns in hippocampal area ca3b, *Hippocampus* 18 (4) (2008) 411–424. doi:<https://doi.org/10.1002/hipo.20404>.

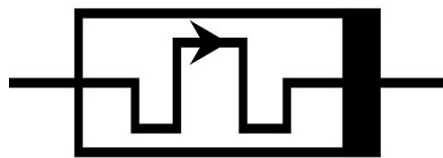
## CHAPTER - 5

### DYNAMICS OF MEMRISTIVE IZHIKEVICH NEURON

#### MODEL

##### 5.1 INTRODUCTION

A memristor is a two-terminal passive electrical component that functions as a basic non-linear circuit element that connects charge and magnetic flux. The memristor is a promising device in a wide range of analogue and digital applications, including memory chips, logic circuits, and neural networks [1].

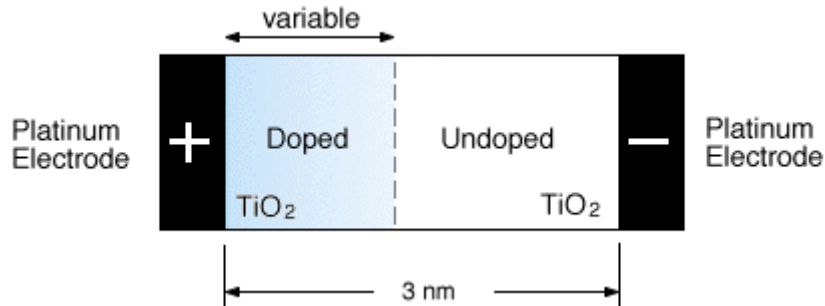


*Figure 5.1: Figure of memristor [12]*

Electronic circuits are currently constructed using three essential passive elements: resistor, capacitor, and inductor. The fourth essential constituent, known as the memristor, has just lately developed [2]. A memristor is a type of electronic component that can store and regulate electrical charge and current flow. It is short for "memory resistor" and was first theorized by Leon Chua in 1971 as the fourth fundamental passive circuit element, alongside resistors, capacitors, and inductors. Memristors have a unique property where their resistance changes in response to the amount of current that has already passed through it, meaning they can "remember" the amount of charge that has previously flowed through them. This makes them useful for building electronic circuits that can mimic the behavior of synapses in the human brain, leading to the development of new forms of artificial intelligence and advanced computing technologies. Members of an HP Lab submitted a report demonstrating the successful realisation of a nanoscale electronic component with observed physical properties that can be explained by the memristor theory. As shown in Figure 2, the HP memristor is a



solid state device composed of two Platinum electrodes and a nanometer-scale TiO<sub>2</sub> thin film having a doped and undoped area.



*Figure 5.2: Titanium dioxide memristor [13]*

The newly developed two-terminal passive element is known as a memristor because it exhibits characteristics of both a memory and a resistor. Memristors have demonstrated a number of exceptional qualities, including superior CMOS technology compatibility, a small device area for high-density on-chip integration, non-volatility, quick speed, low power dissipation, and great scalability [3]. A memristor's resistance, one of its fundamental characteristics, is influenced by the strength, direction, and duration of the voltage applied across its terminals. When the applied voltage is turned off, the memristor remembers its most recent resistance value and stores it till the next time the applied voltage is switched on. Moreover, it possesses additional qualities including pinched hysteresis and dynamical-negative resistance that might significantly affect nanoelectronics.

Since 2008, several applications for memristors have been suggested. Memristor-based Content Addressable Memories (MCAMs), which use a mix of memristor and MOS devices in memory chips, and Resistive Random Access Memory (RRAM) cell designs may both make use of memristors. To reduce the size and complexity of neuromorphic circuits, neural networks can effectively use the memristor's ability to "memorise" the current that passes through it and its direction. In the realm of logic circuits, a brand-new memristor-based IMPLY logic circuit was created. The ability to produce memristor-based logic on the same chip as memory cells makes it special.

Memristors (FPGAs) are also used in the design of crossbar-arrays, which are used in the switching blocks of Field Programmable Gate Arrays [4]. Thus, despite it taking a long time for memristors to go from a purely theoretical derivation to a practical implementation, these devices are now widely employed in non-volatile random-access memory and machine learning applications.

The non-volatile memristor's conductance is controlled by ion mobility, which is similar to how biological synapses and neurons function. These benefits have made the memristor an inevitable option as a component for both synthetic and natural neural networks.

## 5.2 MEMRISTOR PROPERTIES

### 5.2.3 Flux-Charge Relation

The flux and charge connection is stated as a function of charge in a charge-controlled memristor, whereas the relationship is expressed as a function of flux in a flux-controlled memristor [5]. Resistance is the equivalent of a linear (constant) memristor. The device behaviour is more complex if the relationship is nonlinear, thus the parameter in the memristor connecting  $q$  and  $\varphi$  is not constant [6].

The missing component between flux and charge is memristance  $M$ . The memristor is said to be charge-controlled with a memristance ' $M(q)$ ' given by:

$$M(q) = \frac{d\varphi}{dq} \quad (1)$$

The memristor is said to be flux-controlled with a memductance ' $W(\varphi)$ ' given by :

$$W(\varphi) = \frac{dq}{d\varphi} \quad (2)$$

Therefore, it can be derived as:

$$v = M(q) \cdot i \quad (3)$$

$$i = W(\varphi) \cdot v \quad (4)$$

$M(q)$  is theoretically a charge-controlled resistance since memristance has the same unit (Ohm) as resistance. The unit of conductance is likewise included in the memductance [7]. Memristance is the inverse of memductance,

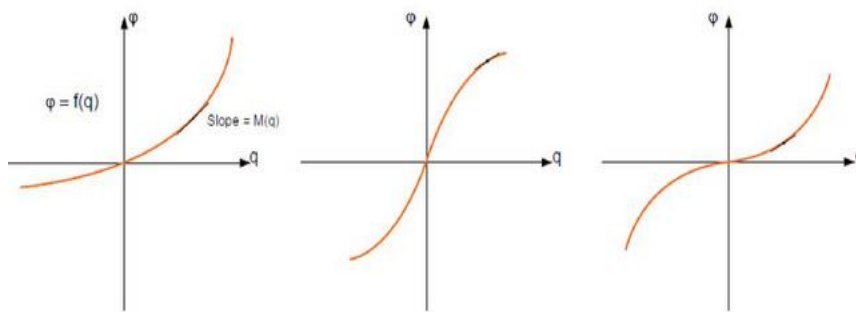
$$M = 1/W(\varphi) \quad (5)$$

A growing trend that is monotonic defines the  $q$ - $\varphi$  curve. The slope of this curve ( $q$ ) is defined as the memory  $M$ . As a result,  $M(q) \geq 0$  is always positive for the memory resistance. The passivity condition states that a memristor is a passive element if and only if the memristance has a non-negative value.

The memristor's immediate power output is provided by:

$$P(i) = M(q)i(t)^2 \quad (6)$$

passive device. This implies that it can only use power; it cannot create or store energy. Like a resistor, a memristor is completely dissipative [1].



*Figure 5.3 Three examples of charge-flux characteristics of the memristor, which all have monotonically increasing characteristics [13].*

### 5.2.2 Current-Voltage relation

The most crucial aspect of a memristor is its current-voltage characteristic, which displays a pinched hysteresis loop. The memristor is regarded as a basic component since the other three basic components cannot be combined to provide the I-V characteristic of the memristor [4]. Applying a periodic signal to a memristor will cause the current to be zero if the voltage is zero and vice versa. Therefore, the origin curve is always crossed by the voltage  $v(t)$  and current  $i(t)$  curves. The geometry of the pinched hysteresis loop will vary with frequency. As the frequency rises, the hysteresis loop contracts. If the frequency is raised to infinity, the memristor will act like a regular resistor [1].

Changes in the I-V characteristic's slope show a transition between different resistance states, with the resistance changing from positive to negative as the applied

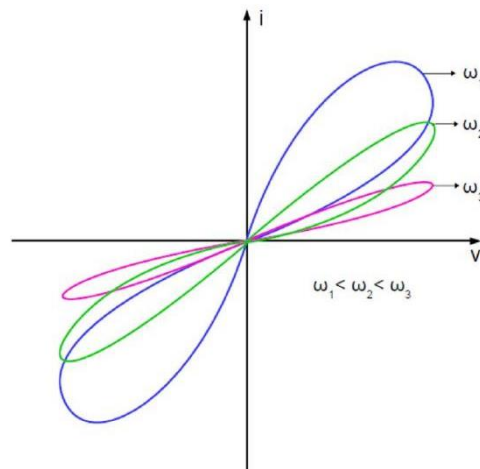


Figure 5.4 Current-voltage characteristics of the memristor [14].

voltage changes amplitude. The symmetrical voltage bias, which can compress to a straight line at high frequencies, causes double-loop I-V hysteresis [2].

### 5.2.3 Resistance – Time Relation

Fig. 4.5 shows the memristor's resistance versus time characteristic. The range of the instantaneous resistance is  $[R_{ON}, R_{OFF}]$ . The applied voltage affects the resistance values. The memory resistance reaches its extreme (highest or minimum) values for a sine-wave voltage with period  $T$  during the following time intervals:

$$t = (2n + 1)T/2 \quad [8].$$

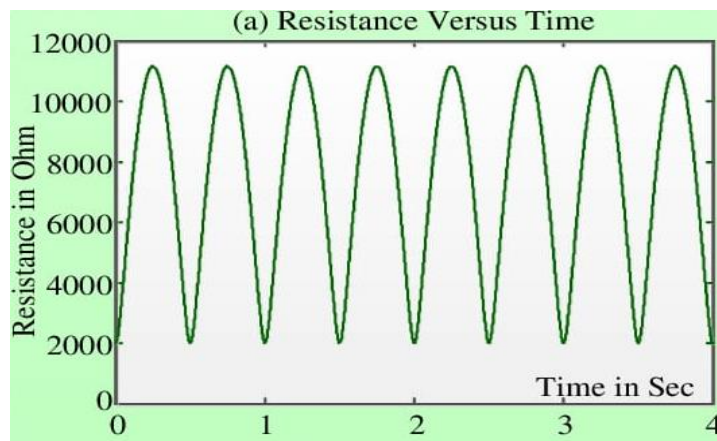


Figure 5.5 Resistance Versus Time plot of the memristor [15]

### 5.2.4 Resistance – Voltage Relation

Fig. 4.6 shows the resistance vs voltage characteristic. Initially, the memristor has a voltage of 0 volts, a current of 0 amps, and a resistance of  $R_i$  [8].

Additionally, the sign of  $v(t)$  determines the memristance value, or resistance  $[R_i, R_{OFF}]$  for  $v(t) < 0$  and  $[R_{ON}, R_i]$  for  $v(t) > 0$ . This is thus because resistance increases as voltage increases, but current follows voltage. The resistance reaches its peak,  $R_{OFF}$ , when the voltage falls to zero [2].

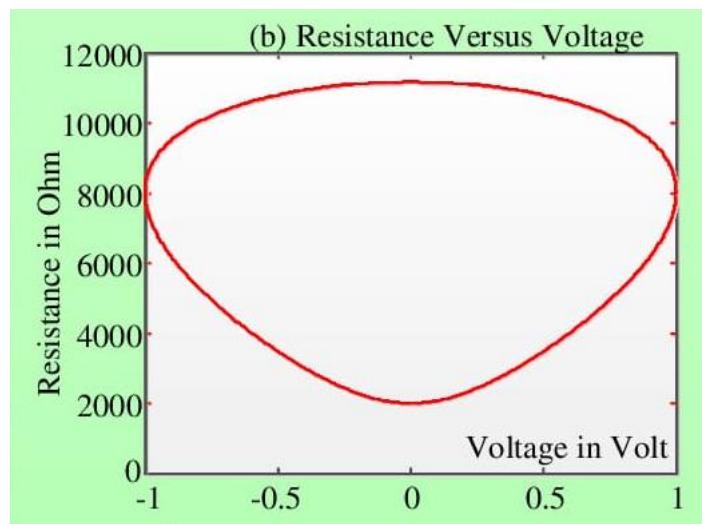


Figure 5.6 Resistance Versus Voltage plot of the memristor [15].

## 5.3 EXTENSION OF IZHKEVICH NEURON MODEL

### 5.3.1 The Network System

The Izhikevich neuron model which was formed by combining the best parts of Hodgkin-Huxley model and the integrate and fire neuron model was studied and the bifurcation was produced [9]. This model is now extended by adding an additional memristive variable  $\phi$  to the existing fundamental equations of the model. Now the following discontinuous ordinary differential equations (ODEs) define the network model for a population of Izhikevich neurons:

$$v'_k = v_k(v_k - \alpha) - \omega_k + \eta_k + I_{ext} + I_{syn,k} - k_1(\alpha_1\phi^2 + \beta\phi + \gamma)v_k \quad (7a)$$

$$\omega'_k = a(bv_k - \omega_k) \quad (7b)$$

$$\phi' = k_2v - k_3\phi \quad (7c)$$

$$\text{If } v_k \geq v_{peak}, \text{ then } v_k \leftarrow v_{reset} \text{ and } \omega_k \leftarrow \omega_k + \omega_{jump} \quad [10] \quad (7d)$$

The recovery variable  $\omega_k$  and the reset condition is same as that of the Izhikevich neuron model. But the membrane potential  $v_k$  is changed and an additional term called the magnetic flux term is added to the fundamental equation.

The neurons are connected together by synapses which is given by

$$I_{syn,k} = g_{syn} s(e_r - v_k) \quad [10] \quad (8)$$

Single exponential synapse is given by

$$s' = -\frac{s}{T_s} + \frac{s_{jump}}{N} \sum_{k=1}^N \sum_{t_k^j < t} \delta(t - t_k^j) \quad [10] \quad (9)$$

Both equation 3 and 4 is also mentioned in the previous chapter of Izhikevich neuron model. Eq (1-3) are the dimensionless parameters which helps in exploring the neurodynamics mathematically and numerically.

### 5.3.2 General Mean-field Description

In the limit  $N \rightarrow \infty$  the population density function becomes  $\rho(t, v, \omega, \eta, \phi)$  which gives the density of neurons. The continuity equation is given by

$$\frac{\partial}{\partial t} \rho(t, v, \omega, \eta, \phi) + \nabla \cdot J(t, v, \omega, \eta, \phi) = 0 \quad [10] \quad (10)$$

Where the probability flux will be defined as

$$\begin{aligned}
 \mathcal{J}(t, v, \omega, \eta, \phi) &= \begin{pmatrix} \mathcal{J}^v(t, v, \omega, \eta, s, \phi) \\ \mathcal{J}^\omega(t, v, \omega, \eta) \\ \mathcal{J}^\phi(t, v, \phi) \end{pmatrix} \\
 &= \begin{pmatrix} G^v(v, \omega, s, \eta, \phi) \\ G^\omega(v, \omega) \\ G^\phi(v, \phi) \end{pmatrix} \rho(t, v, \omega, \eta, \phi) \\
 &= \\
 &\left( \begin{array}{c} v(v - \alpha) - \omega + \eta + I_{ext} + g_{syn}s(e_r - v) - k_1(\alpha_1\phi^2 + \beta\phi + \gamma)v_k \\ a(bv - \omega) \\ k_2v - k_3\phi \end{array} \right) \rho(t, v, \omega, \eta, \phi)
 \end{aligned} \tag{11}$$

Boundary condition for the flux is given by

$$\mathcal{J}^v(t, v_{peak}, \omega, s, \eta, \phi) = \mathcal{J}^v(t, v_{reset}, \omega + \omega_{jump}, s, \eta, \phi)$$

The population firing rate will be thus rewritten as

$$\begin{aligned}
 r(t) &= \lim_{v \rightarrow v_{peak}} \iiint_{\eta\omega\phi} \mathcal{J}^v(t, v, \omega, s, \eta, \phi) d\phi d\omega d\eta \\
 &\equiv \iiint_{\eta\omega\phi} \mathcal{J}^v(t, v_{peak}, \omega, s, \eta, \phi) d\phi d\omega d\eta
 \end{aligned} \tag{12}$$

Mean membrane potential,

$$\langle v(t) \rangle = \iiint_{\eta\omega\phi} v \rho(t, v, \omega, \eta, \phi) d\phi dv d\omega d\eta \tag{13}$$

Mean adaptation variable,

$$\langle \omega(t) \rangle = \iiint_{\eta\omega\phi} \omega \rho(t, v, \omega, \eta, \phi) d\phi d\omega dv d\eta \tag{14}$$

Similarly, we can define the mean magnetic flux term as

$$\langle \phi(t) \rangle = \iiint_{v\omega\eta\phi} \phi \rho(t, v, \omega, \eta, \phi) d\phi d\eta d\omega dv$$

Derivating Eq(9) w.r.t t we get

$$\begin{aligned}\langle \omega \rangle' &= \iiint \int \omega \frac{\partial \rho(t, v, \omega, \eta, \phi)}{\partial t} d\phi d\omega dv d\eta \\ \langle \omega \rangle' &\approx \langle G^v(v, \omega) \rangle + \iiint \int \omega_{jump} J^v(v_{peak}, \omega, s, \eta, \phi) d\phi d\omega d\eta + \langle G^\phi(v, \phi) \rangle\end{aligned}\quad (15)$$

$G^\omega()$  function's linearity with respect to  $v$  and  $w$  is being taken into consideration and the population rate of firing defined in terms of flux and finally we get the ODE for the evolution of mean adaptation variable as

$$\begin{aligned}\langle \omega \rangle' &= G^\omega(\langle v \rangle, \langle \omega \rangle) + \omega_{jump} r(t) + G^\phi(\langle v \rangle, \langle \phi \rangle) \\ &= a(b\langle v \rangle - \langle \omega \rangle) + \omega_{jump} r(t) + k_2 \langle v \rangle - k_3 \langle \phi \rangle\end{aligned}\quad (16)$$

The Synaptic dynamics is rewritten in the terms of  $r(t)$  as

$$s' = -\frac{s}{\tau_s} + s_{jump} r(t) \quad [10] \quad (17)$$

The final mean-field model for the network of Izhikevich neurons includes the two equations (11) and (12) as a fundamental element. They are reliant on the mean membrane potential  $\langle v(t) \rangle$  and the population firing rate  $r(t)$  both are tow macro-variables.

### 5.3.3 Density in conditional form

Here we consider the population density function in the conditional form as same as that the Izhikevich neuron model. With the same assumption we describe the population firing rate in terms of conditional probability  $\rho(t, v|\eta)$  as

$$r(t) = \lim_{v \rightarrow v_{peak}} \int L(\eta) \rho^v(t, v|\eta) G^v(v, \langle \omega|v, \eta \rangle, \langle \phi|v, \eta \rangle, s, \eta) d\eta \quad (18)$$

Also, we assume

$$\begin{aligned}\langle \omega|v, \eta \rangle &= \langle \omega|\eta \rangle \quad [10] \\ \langle \phi|v, \eta \rangle &= \langle \phi|\eta \rangle\end{aligned}\quad (19)$$

which corresponds to a first order moment closure assumption [11]. We now have

$$r(t) = \lim_{v \rightarrow v_{peak}} \int L(\eta) \rho^v(t, v|\eta) G^v(v, \langle \omega|\eta \rangle, \langle \phi|\eta \rangle, s, \eta) d\eta \quad (20)$$



Similarly, we can rewrite the mean membrane potential using normalization condition on the marginal density of  $\omega$ ,

$$\int_{\eta} L(\eta) \int_{\nu} \nu \rho^{\nu}(t, \nu | \phi, \eta) \int_{\phi} \rho^{\phi}(t, \phi | \nu, \eta) d\phi d\nu d\eta \quad (21)$$

Integrate the continuity equation w.r.t.  $\omega$  and we using the population density in conditional form as in the Izhikevich neuron model we get

$$\frac{\partial \rho^{\nu}}{\partial t}(t, \nu | \phi, \eta) + \frac{\partial}{\partial \nu} [G^{\nu}(\nu, \langle \omega | \nu, \eta \rangle \langle \phi | \nu, \eta \rangle, s, \eta) \rho^{\nu}(t, \nu | \phi, \eta) \rho^{\phi}(t, \phi | \nu, \eta)] = 0 \quad (22)$$

Here the flux vanishes in the boundary  $\partial\omega$ .

The modified continuity equation using the moment closure assumption as in the equation (15)

$$\frac{\partial \rho^{\nu}}{\partial t}(t, \nu | \eta) + \frac{\partial}{\partial \nu} [G^{\nu}(\nu, \langle \omega | \eta \rangle \langle \phi | \eta \rangle, s, \eta) \rho^{\nu}(t, \nu | \phi, \eta) \rho^{\phi}(t, \phi | \nu, \eta)] = 0 \quad (23)$$

Steady state of solution of the solution of the system

$$\bar{\rho}^{\nu}(\nu | \eta) \bar{\rho}^{\phi}(\phi | \nu, \eta) \propto \frac{1}{\nu(\nu - \alpha) - \langle \omega | \eta \rangle + \langle \phi | \eta \rangle + \eta + I_{ext} + g_{syn} \bar{s}(e_r - \nu) - k_1(\alpha_1 \phi^2 + \beta \phi + \gamma) \nu k} \quad (24)$$

Where  $\langle \omega | \eta \rangle$ ,  $\langle \phi | \eta \rangle$ ,  $\bar{s}$  are the steady state values of  $\langle \omega | \eta \rangle$ ,  $\langle \phi | \eta \rangle$ ,  $s$

### 5.3.4 Lorentzian Ansatz

Expression for macroscopic variables  $\nu(t)$  and  $r(t)$  are simplified and Using the Lorentzian approach, we obtain the mean-field approximation for the Izhikevich network. Conditional probability  $\rho^{\nu}(t, \nu | \eta)$  in the form of Lorentzian distribution

$$\rho^{\nu}(t, \nu | \eta) = \frac{1}{\pi} \frac{x(t, \eta)}{[\nu - y(t, \eta)]^2 + x^2(t, \eta)} \quad [10] \quad (25)$$

Where  $x(t, \eta) =$  Half width half maximum

$y(t, \eta) =$  location of the center

$y(t, \eta)$  is the reason behind the only mean in principal value in the Lorentz distribution.

Therefore, mean membrane potential can be expressed in terms of  $y(t, \eta)$  as

$$\langle \nu(t) \rangle = \int_{\eta} y(t, \eta) L(\eta) d\eta \quad [10] \quad (26)$$

Under the given conditions as mentioned in the Izhikevich neuron model the population firing rate defined in the eq (15) can also be expressed in the form of Lorentzian coefficient as

$$r(\eta, t) = \lim_{v \rightarrow v_{peak}} \rho^v(t, v | \eta) G^v(v, \langle \omega | \eta \rangle \langle \phi | \eta \rangle, s, \eta) \quad (27)$$

$$= \lim_{v_{peak} \rightarrow \infty} \frac{1}{\pi} \frac{x(\eta, t)}{[v_{peak} - y(\eta, t)]^2 + x^2(\eta, t)} \left[ v_{peak} (v_{peak} - \alpha) - \langle \omega | \eta \rangle - \langle \phi | \eta \rangle + \eta + I_{ext} + g_{syn} s (e_r - v_{peak}) \right]$$

Then the total firing rate can also be expressed as

$$r(t) = \int_{\eta} r(\eta, t) L(\eta) d\eta = \frac{1}{\pi} \int_{\eta} x(\eta, t) L(\eta) d\eta \quad (28)$$

Now the Izhikevich neuron network explained in the equation (1-3) is reduced using mean-field reduction and thus we obtain the ordinary differential equations for the mean-field system as the follows

$$r' = \Delta\eta/\pi + 2r\langle v \rangle - (\alpha + g_{syn}s)r \quad (29a)$$

$$\langle v \rangle' = \langle v \rangle^2 - \alpha\langle v \rangle - \langle \omega \rangle + \bar{\eta} + I_{ext} + g_{syn}s(e_r - \langle v \rangle) - \pi^2 r^2 - k_1(\alpha_1\phi^2 + \beta\phi + \gamma) v_k \quad (29b)$$

$$\langle \omega \rangle' = a(b\langle v \rangle - \langle \omega \rangle) + \omega_{jump}\Gamma \quad (29c)$$

$$\langle \phi \rangle' = k_2\langle v \rangle - k_3\langle \phi \rangle \quad (29d)$$

$$s' = -s/\tau_s + s_{jump}\Gamma \quad (29e)$$

## 5.4 RESULT

In this chapter, an additional three variable memristive term has been added to already existing ordinary differential equations describing the Izhikevich model. Memristors can simulate the magnetic induction action brought on by neuron potential. Memristors are also advantageous in neural network applications because of their quick processing speed and energy-efficient computing. When the three variable memristive term was added, the three main governing equations explaining the Izhikevich neuron model has changed to four. We have successfully derived the equations explaining the dynamics of memristive neuron model similar to that of the original model.

## 5.5 REFERENCES:

- [1] V. Keshmiri, 'A Study of the Memristor Models and Applications', Linkoping University, 2014.
- [2] Mohanty, Saraju, Memristor: From Basics to Deployment. Potentials, IEEE, 32. 34-39, 2013 10.1109/MPOT.2012.2216298.
- [3] W. Xu, J. Wang, and X. Yan, 'Advances in Memristor-Based Neural Networks', Frontiers in Nanotechnology, vol. 3, 2021, doi: 10.3389/fnano.2021.645995.
- [4] Nafea, Sherief & Dessouki, Ahmed & El-Rabaie, El-Sayed. (2015). Memristor Overview up to 2015. 24. 79-106. 10.21608/mjeer.2015.64132.
- [5] B. Widrow, "An Adaptive ADALINE Neuron Using Chemical MEMISTORS," no. 1553–2, 1960.
- [6] Ho Yenpo & Huang, Garng & Li, Peng. (2011). Dynamical Properties and Design Analysis for Nonvolatile Memristor Memories. Circuits and Systems I: Regular Papers, IEEE Transactions on. 58. 724 - 736. 10.1109/TCSI.2010.2078710.
- [7] Frank Y. Wang, "Memristor for introductory physics", Preprint arXiv:0808.0286.2008.
- [8] A. G. Radwan, M. A. Zidan, and K. N. Salama, "On The Mathematical Modeling Of Memristors," in Proceedings of the International Conference on Microelectronics, 2010, pp. 284 –287.
- [9] E. M. Izhikevich, Dynamical Systems in Neuroscience : the Geometry of Excitability and Bursting, Computational neuroscience, MIT Press, Cambridge, Mass, 2007.
- [10] L. Chen, S. A. Campbell, Exact mean-field models for spiking neural networks with adaptation, <https://arxiv.org/pdf/2203.08341&hl=en&sa=X&ei=vRt3ZP2AEJCP6rQP6Iq6sAY&scisig=AGlGAw-9sFR1kvPc23cRQD4fB7BH&oi=scholar>
- [11] W. Nicola, S. A. Campbell, Mean-field models for heterogeneous networks of two-dimensional integrate and fire neurons, Frontiers in Computational Neuroscience 7 (2013) 184. doi:<https://doi.org/10.3389/fncom.2013.00184>.
- [12] Rambus Press. Perovskite memristor is just 5 nanometre thick , 2015 November 3 ; <https://www.rambus.com/blogs/perovskite-memristor-is-just-5-nanometers-thick-2/>

[13] Ahmed Ahmed Shaaban Dessouki, El-Sayed Mahmoud El-Rabaie, Sherief Nafea. Menoufia Journal of Electronic Engineering Research 24:79-106 ; July 2015  
DOI:10.21608/mjeer.2015.64132

[13] Silas Harvey , Memristor Training study-guidelines.

<https://images.app.goo.gl/ypL7kdrG1EP2n1Hp7>

[14] Silas Harvey , Memristor Training study-guidelines.

<https://slideplayer.com/slide/13852296/85/images/13/Memristor+properties-+Flux-charge+relation.jpg>

[15] S. P. Mohanty, "Memristor: From Basics to Deployment," in IEEE Potentials, vol. 32, no. 3, pp. 34-39, May-June 2013, doi: 10.1109/MPOT.2012.2216298.

<https://www.semanticscholar.org/paper/Memristor%3A-From-Basics-to-Deployment-Mohanty/148d4f260717d8002c0c973808dcb456c06e9bed>

## **CHAPTER -6**

### **DISCUSSION**

The relationship between brain characteristics and behaviour is frequently inferred from models in order to derive theories of cause and effect. They make predictions, which enables more focused experimentation. Models enable virtual experimenting, which facilitates the development of intuition. In order to regain lost control capacities including perception (for example, deafness or blindness), motor movement decision-making, and continuous limb control, biological neuron models seek to describe the mechanisms behind the nervous system's operation.

In this project we have chosen Izhikevich Neuron model for the study of neural dynamics of brain. The Izhikevich neuron model is used because of its simplicity and ability to accurately replicate the firing behavior of real neurons. It is a two-dimensional model that incorporates the essential dynamics of a neuron while minimizing complexity. The model can be easily tuned to represent a variety of neuron types, which makes it useful for simulating complex neural networks. Additionally, the Izhikevich model is computationally efficient, allowing for the simulation of large-scale networks and the exploration of complex neural dynamics. This has made it a popular choice for investigating the mechanisms of neural coding, plasticity and learning. Overall, the Izhikevich model provides a practical and biophysically realistic means of modeling the behavior of neural networks and understanding the underlying principles of neural computation.

For a network of heterogeneous Izhikevich neurons that exhibits spike frequency adaptation through a recovery variable, we have developed a mean-field model. We have learned the underlying properties of cortical neurons by assessing the population firing rate and bifurcation diagrams which was obtained through MATLAB software and XPPAUT.

Further we extended our project by adding a three variable memristive term to the already existing ordinary differential equation of the Izhikevich neuron model. The memristive Izhikevich neuron model is a type of artificial neuron model that incorporates a memristor into the traditional Izhikevich model. A memristor is a device that can remember the amount of current that has passed through it and modifies its resistance accordingly. The addition of a memristor to the Izhikevich model results in a more realistic neuron model that can exhibit the nonlinear dynamics and spiking behavior of real neurons. This can be particularly useful in

applications such as neuromorphic computing, where the goal is to mimic the behavior of biological neural networks using artificial systems.

One of the advantages of the memristive Izhikevich neuron model is its ability to exhibit a wide range of dynamical behaviors, including regular spiking, fast spiking, and bursting. This makes it a useful tool for studying the dynamics of neural networks and the effects of various neuromodulators and synaptic connections on network behavior. Another advantage of the memristive Izhikevich neuron model is its low power consumption, which makes it suitable for applications in energy-efficient neuromorphic computing systems. Additionally, the memristor-based implementation of the model can potentially be implemented using existing technologies, making it a practical option for real-world applications.

We brought a change from the Izhikevich neuron model to the memristive Izhikevich neuron model by incorporating a new variable which represents the electromagnetic induction or noise. We then described the behaviour of neuron by deriving the entire model and obtained the main five governing equations of the respective model which was found similar to that of the original model. We have generated the MATLAB code for the new memristive Izhikevich neuron model but we haven't obtained the stimulations for this model.

## MATLAB CODE FOR BASIC IZHIKEVICH MODEL

### ONE POPULATION

```
function x = cauchyrnd(mu,hw,varargin)
% x = cauchyrnd(mu,hw,M,N)
% to generate M*N random variables with Cauchy (/Lorentizan) distribution
% mu: the location parameter/ center
% hw: the scale parameter/ half width
%
% the cumulative distribution function (CDF):
%  $F = (x, \mu, hw) = 1/\pi * \arctan[(x-\mu)/hw] + 1/2$ 
% then,
%  $x = hw * \tan[\pi*(F - 1/2)] + \mu$ 
%
% F varies from 0 to 1. In code, we can replace F with values randomly
% sampled from the uniform distribution on (0,1)
%
% This technique is referred to as inverse transform sampling and is very
% useful for generating random variates from many distributions.
%
% ref.
% method 1: (seen in many places)
% https://math.stackexchange.com/questions/484395/how-to-generate-a-cauchy-random-variable
%
% method 2:
% without analytical expression of CDF (not verified yet)
% https://www.mathworks.com/matlabcentral/answers/80333-generate-number-from-a-probability-distribution
%
% method 3: generate fixed values (not proper)
% https://www.mathworks.com/help/stats/work-with-the-cauchy-distribution-using-the-t-location-scale-distribution.html
%
x = mu + hw.*tan(pi*(rand(varargin{:}) - 0.5));
end
t_net = avg_fired_time;

% figure(1)
% plot(t_net,R)
```

```

[up,lo]= envelope(R,300,'peak');
% hold on
% plot(t_net,up,'-r',t_net,lo,'-r')
% hold off

close all
figure(1)

%% size of the figure (width*height), position shown in the screen
fig=gcf;
fig.Position=[10,10,700,700];

%% raster plot,randomly selected neurons to plot
f1a=subplot(4,1,1);
N_rand = 300;
index_rand = ceil(N*rand(N_rand,1));
for j=1:N_rand
    select = find(firings(:,2)==index_rand(j));
    % the firing times of the jth neuron in the ylabel of raster plot,

    row_num = length(select);
    nn = j*ones(row_num,1);
    plot(firings(select,1),nn,'k','MarkerSize',0.5);

    hold on
end
ylabel('Neuron \#','Interpreter','LaTeX')
xlim([0,tend])

%% population firing rate
f1b=subplot(4,1,2);
tx=avg_fired_time;

plot(tx,R,'b')           % network
hold on
plot(t,rm,'r','LineWidth',2); % mean field

ylabel('$r(t)$','FontSize',14,'Interpreter','LaTeX')
xlim([0,tend])
hold off

```



```

%% mean membrane potential
f1c=subplot(4,1,3);
plot(time,v_mean,'b')      % network
hold on
plot(t,vm,'r','LineWidth',2) % mean field

ylabel('\langle v(t) \rangle','FontSize',14,'Interpreter','LaTeX')
xlabel('Time','Interpreter','LaTeX')
xlim([0,tend])
hold off

%% mean recovery variable

f1d=subplot(4,1,4);
plot(time,w_mean,'b')      % network
hold on
plot(t,wm,'r','LineWidth',2) % mean field model

% === Amplitudes of PO =====
long = length(time);
start = round(long*2/3);

up_w = max(w_mean(start:end))*ones(long,1);
lo_w = min(w_mean(start:end))*ones(long,1);

% plot(time,up_w,'m',time,lo_w,'m')
hold off
%=====

ylabel('\langle w(t) \rangle','FontSize',14,'Interpreter','LaTeX')
xlabel('Time','Interpreter','LaTeX')
xlim([0,tend])

% By Liang Chen
% May 4, 2022
%
% how to show the data distribution of the heterogeneous source
% histogram: the y-axis represents the number count or percentage of
% occurrences in the data for each column (bin).

clc
clear

```

```

close all

%% ===== Distribution 1 =====

N=10^4;
mu = 0.2;
hw = 0.02;

%% deterministic generation of a Lorentzian distribution
% A typo in [Montbrio2015], it should be "tan", not "atan"

eta_deter = zeros(N,1);
n = zeros(N,1);
for j=1:N
    n(j) = (2*j-N-1)/(N+1);
    eta_deter(j) = mu + hw*tan(pi*n(j)/2);
end

figure(1)
histogram(eta_deter,10000,'Normalization','probability');
% % 10000 bins, other options: 'Normalization','pdf'

% histogram(eta_deter,10000,'Normalization','probability',DisplayStyle='stairs');
% % default: DisplayStyle='bar'

xlim([-0.5,0.5])
% set(gca,'ytick',[]) % not show values on the y-axis
% yticklabels(yticks*100) % show value/%

ylabel('$p$', 'interpreter', 'latex', 'fontsize', 14)
xlabel('$\eta$', 'interpreter', 'latex', 'fontsize', 14)

%% random generation of a Lorentzian distribution

eta_rnd = cauchyrnd(mu,hw,N,1);

figure(2)
histogram(eta_rnd,20000,'Normalization','probability');
xlim([-0.5,0.5])
set(gca,'ytick',[])
ylabel('$p$', 'interpreter', 'latex', 'fontsize', 14)
xlabel('$\eta$', 'interpreter', 'latex', 'fontsize', 14)

```

```
%% ===== Distribution 2 =====
```

```
N=10^3;  
mu = 1.8;  
hw = 1.4;
```

```
%% deterministic generation of a bimodal Lorentzian distribution
```

```
n = zeros(N,1);  
eta_deter = zeros(N,1);  
for j=1:N/2  
    n(j) = (2*j-N/2-1)/(N/2+1);  
    eta_deter(j) = -mu + hw*tan(pi*n(j)/2);  
end
```

```
for j=N/2+1:N  
    n(j)=(2*j-3*N/2-1)/(N/2+1);  
    eta_deter(j) = mu + hw*tan(pi*n(j)/2);  
end
```

```
figure(3)  
h = histogram(eta_deter,800,'Normalization','probability');  
xlim([-20,20])  
set(gca,'ytick',[])  
ylabel('$p$', 'interpreter','latex','fontsize',14)  
xlabel('$\eta$', 'interpreter','latex','fontsize',14)
```

```
%% random generation of a bimodal Lorentzian distribution
```

```
eta_rnd1 = cauchyrnd(-mu,hw,N/2,1);  
eta_rnd2 = cauchyrnd(mu,hw,N/2,1);  
eta_rnd = [eta_rnd1; eta_rnd2];
```

```
figure(4)  
histogram(eta_rnd,2000,'Normalization','probability');  
xlim([-20,20])  
set(gca,'ytick',[])  
ylabel('$p$', 'interpreter','latex','fontsize',14)  
xlabel('$\eta$', 'interpreter','latex','fontsize',14)
```

```
%% ===== The End =====
```

```

% IC
rint = 0;
vint = 0;
wint = b*vint;
sint = 0;
sint = sint + (1-sint).*(sint>1); % to bound sm in [0,1];

[t,y] = ode45(@(t,y)
mf_sys(mu,hw,alpha,gsyn,er,a,b,wjump,tsyn,sjump,I,t,y)',[0,tend],[rint,vint,wint,sint]');
rm = y(:,1);
vm = y(:,2);
wm = y(:,3);
sm = y(:,4);
% By Liang Chen,
% Updated on Sep. 8, 2021
%
% introduce exact refractory time: from v(>= vpeak) to infinity, from -infinity to vreset
% dimensionless network model
%
%
%% all-to-all coupling with the same weights
SMAX = sjump*ones(N,N);
neff = N;
%
%% ICs of variables
v = zeros(N,1); % membrane potentials
%
v_mean = zeros(1,tend/dt+1);% evolution of the mean membrane potential
%
w = b*v; % recovery variable
w_mean = zeros(1,tend/dt+1);%
%
s = zeros(N,1); % synaptic gating variable (proportion)
s = s + (1-s).*(s>1); % to bound s in [0,1] because the synaptic
% current is g*s(t)*(er-v), not J*s(t) in [Montbrio2015]
sstore = zeros(1,tend/dt+1); % store the synaptic variable of the first neuron
% because of all-to-all connectivity, the
% synaptic variable of each neuron is the same
%
%% store the spike time and index of neuron to plot the rasterplot
firings = [];
%
%% Simulation, Euler integration
%

```

```

% fired_inf = find(v >= vinf);
for i = 1:tend/dt+1
    v_ = v;          % V_ at the time (i-1)*dt, V at the time i*dt
    w_ = w;
    s_ = s;
    sstore(i) = s(1);

    %%
    n_ref = find(v_ >= vreset & v_ <= vpeak); % neurons not in the refractory period
    v_mean(i) = mean(v_(n_ref)); % at the time (i-1)*dt
    w_mean(i) = mean(w_(n_ref));

    %%
    fired_inf = find(v_ >= vinf);
    firings = [firings; (i-1)*dt + 0*fired_inf, fired_inf];

    v(fired_inf) = -v_(fired_inf);
    w(fired_inf) = w_(fired_inf) + wjump;

    %%
    n_fired = find(v_ < vinf); % neurons not fire
    rhs = v_(n_fired).*(v_(n_fired) - alpha) - w_(n_fired) + eta(n_fired)...
          + I + gsyn*(er - v_(n_fired)).*s_(n_fired);
    v(n_fired) = v_(n_fired) + dt*rhs;
    w(n_fired) = w_(n_fired) + dt*(a*(b*v_(n_fired) - w_(n_fired)));

    s(n_fired) = s_(n_fired) + dt*(-s_(n_fired)/tsyn) + sum(SMAX(n_fired,fired_inf),2)/neff;
% row sum
    s = s + (1-s).*(s>1); % to bound s in [0,1];

end
%
%% calculation of the instantaneous firing rate
twin = 2*10^(-2);
fired_time = firings(:,1);
%
tmax = find(fired_time <= fired_time(end) - twin);
% tmax(1): the first time index when fired_time + twin > fired_time(end)
fired_num = zeros(tmax(end)+1,1);
fired_num(1) = length(find(fired_time < twin));
%
for i = 1:tmax(end)

```

```

    fired_num(i+1) = length(find(fired_time >= fired_time(i) & fired_time < twin +
fired_time(i)));
end
avg_fired_time = [0;fired_time(1:tmax(end))];
R = fired_num/twin/N; % the firing rate

```

```

%% ===== The End =====

```

```

function dy = mf_sys(mu,hw,alpha,gsyn,er,a,b,wjump,tsyn,sjump,I,t,y)

```

```

%

```

```

% The mean field model (no delay):

```

```

% r' = hw/pi + 2*r*v_mean - r*(g*s + alpha)

```

```

% v_mean' = -w_mean + mu + v_mean^2 + g*s*(er - v_mean)...

```

```

% - alpha*v_mean - pi^2*r^2

```

```

% w_mean' = a*(b*v_mean - w_mean) + wjump * r

```

```

% s' = -s/ts + sjump*r

```

```

%

```

```

% The heterogeneous source: the applied current

```

```

% the Lorentzian distribution: mu (center), hw (half width)

```

```

%

```

```

%=====

```

```

%

```

```

% if y(4) > 1

```

```

% y(4)=1;

```

```

% end

```

```

% it seems there is no effect on limit of s in [0,1]

```

```

% in our case, because mu, hw is small, s always stays in [0,1], but when

```

```

% mu=5, hw=1, s will >1.

```

```

dy(1) = hw/pi + 2*y(1)*y(2) - y(1)*(gsyn*y(4) + alpha);

```

```

dy(2) = y(2)^2 - alpha*y(2) + gsyn*y(4)*(er-y(2)) - pi^2*y(1)^2 -y(3) + mu + I;

```

```

dy(3) = a*(b*y(2) - y(3)) + wjump*y(1);

```

```

dy(4) = -y(4)/tsyn + sjump*y(1);

```

```

%

```

```

end

```

```

% Liang Chen, May 12, 2021

```

```

%

```

```

% Upper case for dimensional form, lower case for dimensionless form

```

```

% ref. [Nicola2013bif]

```

```

%

```

```

%% dimensional form of the Izhikevich neuron eq.(46)-(49)

```

```

% variables: V,W,S

```

```

%

```

```

% the recovery eq. of W
beta = -1;      % Resonator/Integrator Variable
TW = 200;      % time constant of the adaptation eq.
VR = -65;      % resting membrane potential
%
% the membrane potential eq. of V
C = 250;       % Capacitance
k1 = 2.5;      % spike width factor
VT = VR + 40 - beta/k1; % threshold
Er = 0;        % Reversal Potential
Gsyn = 200;    % maximal conductance
Iext = 0;      % extra current
%
% synaptic eq. of S
Sjump = 0.8;
Tsyn = 4;      % synaptic time constant
%
% the resetting rule
Vpeak = 30;
Vreset = -55;
Wjump = 200;
%
%
%% the dimensionless Izhikevich neuron eq.(50)-(53)
% variables
% v = 1 + V/abs(VR);
% w = W/(k*VR*VR);
% s = S;
%
% the recovery eq. of w
a = C/(TW*k1*abs(VR));
b = beta/(k1*abs(VR));
%
% the membrane potential eq. of v
alpha = 1 + VT/abs(VR);
gsyn = Gsyn/(k1*abs(VR));
er = 1+Er/abs(VR);
I = Iext/(k1*VR*VR);
%
% synaptic eq. of s
tsyn = Tsyn*k1*abs(VR)/C;
sjump = Sjump*C/(k1*abs(VR)); % different from [Nicola2013bif]
%
% the resetting rule

```

```

vpeak = 1 + Vpeak/abs(VR);
vreset = 1 + Vreset/abs(VR);
wjump = Wjump/(k1*VR*VR);
%

%% ===== The End =====

function varargout = plotshaded(x,y,fstr)
% copyright: http://jvoigts.scripts.mit.edu/blog/nice-shaded-plots/
%
% x: x coordinates
% y: either just one y vector, or 2xN or 3xN matrix of y-data
% fstr: format ('r' or 'b--' etc)
%
% example
% x=[-10:.1:10];plotshaded(x,[sin(x.*1.1)+1;sin(x*.9)-1],'r');

if size(y,1)>size(y,2)
    y=y';
end;

if size(y,1)==1 % just plot one line
    plot(x,y,fstr);
end;

if size(y,1)==2 %plot shaded area
    px=[x,fliplr(x)]; % make closed patch
    py=[y(1,:), fliplr(y(2,:))];
    patch(px,py,1,'FaceColor',fstr,'EdgeColor','none');
end;

if size(y,1)==3 % also draw mean
    px=[x,fliplr(x)];
    py=[y(1,:), fliplr(y(3,:))];
    patch(px,py,1,'FaceColor',fstr,'EdgeColor','none');
    plot(x,y(2,:),fstr);
end;

alpha(.2); % make patch transparent

end

% By Liang Chen, May 12, 2021
% Updated on June 1, Sep. 8

```



```

% Oct. 25: bursting
% May 20, 2022
%
% Simulation of the network of Izhikevich neurons
% dimensional form of eqs.
% heterogeneous parameters with the Cauchy/Lorentzian distribution
%
% ref: Liang Chen, Sue Ann Campbell, Exact mean-field models for spiking
%      neural networks with adaptation
% preprint: https://arxiv.org/abs/2203.08341
%
%=====
tic
clc
clear
%% values of the parameters
parameters

vpeak = 200; vreset = -vpeak;
vinf = 200; % represent the infinity, vpeak-vreset=vinf=200 in [DumontErmentrout2017]

N = 10^3; % number of cells

%% Euler integration parameters
dt = 10^(-3);
tend = 800;
time = 0:dt:tend;
%tend = Tend*k*abs(VR)/C; % Tend: dimensional; tend: dimensionless

%% heterogeneous parameter, Lorentzian distribution
mu = 0.12;          % centre
hw = 0.02;         % half width

% random generation
eta = cauchyrnd(mu,hw,N,1);

% or

% deterministic generation: typo in [Montbrio2015], "tan", not "atan"

% eta = zeros(N,1);
% for j=1:N
%   eta(j) = mu + hw*tan(pi/2*(2*j-N-1)/(N+1));

```

```

% end

%% mean-field model
% Izh_mf,          % Euler integration

Izh_mf_ode45      % ode45, efficient

%% network model

Izh_network3

%% save data:
save('Izh_mf_network.mat');

%% plot figures

fig_plot

toc
%% ===== The end =====

```

## **TWO POPULATION**

```

function x = cauchyrnd(mu,hw,varargin)
% x = cauchyrnd(mu,hw,M,N)
% to generate M*N random variables with Cauchy (/Lorentizan) distribution
% mu: the location parameter/ center
% hw: the scale parameter/ half width
%
% the cumulative distribution function (CDF):
%  $F = (x, \mu, hw) = 1/\pi * \arctan[(x-\mu)/hw] + 1/2$ 
% then,
%  $x = hw * \tan[\pi*(F - 1/2)] + \mu$ 
%
% F varies from 0 to 1. In code, we can replace F with values randomly
% sampled from the uniform distribution on (0,1)
%
% This technique is referred to as inverse transform sampling and is very
% useful for generating random variates from many distributions.
%
% ref.
% method 1: (seen in many places)
% https://math.stackexchange.com/questions/484395/how-to-generate-a-cauchy-random-variable

```

```

%
% method 2:
% without analytical expression of CDF (not verified yet)
% https://www.mathworks.com/matlabcentral/answers/80333-generate-number-from-a-probability-distribution
%
% method 3: generate fixed values (not proper)
% https://www.mathworks.com/help/stats/work-with-the-cauchy-distribution-using-the-t-location-scale-distribution.html
%
x = mu + hw.*tan(pi*(rand(varargin{:}) - 0.5));
end
%% population 1
figure(1)
% ===== raster plot =====
% randomly selected neurons to plot
subplot(3,1,1)
N1_rand = 300;
index_rand1 = ceil(N1*rand(N1_rand,1));
for j=1:N1_rand
    select1 = find(firings1(:,2)==index_rand1(j));
    % the firing times of the jth neuron in the ylabel of raster plot,

    row_num1 = length(select1);
    nn1 = j*ones(row_num1,1);
    plot(firings1(select1,1),nn1,'k','MarkerSize',0.1);

    hold on
end
ylabel('Neuron \#','Interpreter','LaTeX')
xlim([0,tend])
%
%===== the population firing rate =====
subplot(3,1,2)
fig(1)= plot(avg_fired_time1,R1,'g');    % network 1
hold on
fig(2)= plot(t,rm1,'r','LineWidth',2);    % mean field model 1
xlim([0,tend])
ylim([-0.1,0.5])
ylabel('$R$', 'FontSize',14,'Interpreter','LaTeX')
legend(fig(1:2),'Pop. p','Mean field')
%
%===== mean membrane potential =====
subplot(3,1,3)

```

```

plot(time,v1_mean,'g');      % network 1
hold on
plot(t,vm1,'r','LineWidth',2);  % mean field 1
xlim([0,tend])
ylim([-0.1,1.5])
ylabel('\langle v \rangle','FontSize',14,'Interpreter','LaTeX')
xlabel('Time')
%
%===== mean adaptation =====
% because wm evolves slow with time, but rm and vm fast, it is better to
% plot them respectively.
figure(2)
fig(1)=plot(time,w1_mean,'g');      % network 1
hold on
fig(2)=plot(time,w2_mean,'b');      % network 2
fig(3)=plot(t,wm1,'r');      % mean field 1
plot(t,wm2,'r')      % mean field 2
ylabel('\langle w \rangle','FontSize',14,'Interpreter','LaTeX')
xlabel('Time')
legend(fig(1:3),'Pop. p','Pop. q','Mean field')
xlim([0,tend])
ylim([-0.02,0.2])
%
%
%% population 2
figure(3)
% ===== raster plot =====
% randomly selected neurons to plot
subplot(3,1,1)
N2_rand = 300;
index_rand2 = ceil(N2*rand(N2_rand,1));
for j=1:N2_rand
    select2 = find(firings2(:,2)==index_rand2(j));
    % the firing times of the jth neuron in the ylabel of raster plot,

    row_num2 = length(select2);
    nn2 = j*ones(row_num2,1);
    plot(firings2(select2,1),nn2,'.k','MarkerSize',0.1);

    hold on
end
ylabel('Neuron \#','Interpreter','LaTeX')
xlim([0,tend])
%

```

```

%===== the population firing rate =====
subplot(3,1,2)
%tend_f = ceil(0.03*length(time)); % time span for fast variables
fig(1)= plot(avg_fired_time2,R2,'b');    % network 2
hold on
fig(2)= plot(t,rm2,'r','LineWidth',2);    % mean field model 2
xlim([0,tend])
ylim([-0.1,0.5])
ylabel('$r$', 'FontSize',14, 'Interpreter','LaTeX')
legend(fig(1:2), 'Pop. q', 'Mean field')
%
%===== mean membrane potential =====
subplot(3,1,3)
plot(time,v2_mean,'b');    % network 2
hold on
plot(t,vm2,'r','LineWidth',2)    % mean field 2
xlim([0,tend])
ylim([-0.1,1.5])
ylabel('$\langle v \rangle$', 'FontSize',14, 'Interpreter','LaTeX')
xlabel('Time')

%% === The End =====
% By Liang Chen
% Oct. 20, 2021
%
% Population 1: subscript 1; Population 2: subscript 2
%
%% Initial conditions
%
% IC % IC.m
%
rm1 = zeros(1,tend/dt+1);
vm1 = zeros(1,tend/dt+1);
wm1 = zeros(1,tend/dt+1);
%
rm2 = zeros(1,tend/dt+1);
vm2 = zeros(1,tend/dt+1);
wm2 = zeros(1,tend/dt+1);
%
sm1 = zeros(1,tend/dt+1);
sm2 = zeros(1,tend/dt+1);
% sm = sm + (1-sm).*(sm>1); % to bound sm in [0,1];
%

```

```

%%
p1 = N1/(N1+N2);
p2 = 1-p1;
%
%%
for i = 1:tend/dt
    gs1 = p1*gsyn11*sm1 + p2*gsyn12*sm2;
    gs2 = p1*gsyn21*sm1 + p2*gsyn22*sm2;
    %
    rm1(i+1) = rm1(i) + dt*(hw1/pi + 2*rm1(i)*vm1(i) - rm1(i)*(gs1(i) + alpha));
    vm1(i+1) = vm1(i) + dt*(vm1(i)^2 - alpha*vm1(i) + gs1(i)*(er - vm1(i)) - pi^2*rm1(i)^2-
wm1(i) + mu1 + Iext1);
    wm1(i+1) = wm1(i) + dt*(a1*(b*vm1(i) - wm1(i)) + wjump1*rm1(i));
    sm1(i+1) = sm1(i) + dt*(-sm1(i)/tsyn + sjump*rm1(i));
    sm1 = sm1 + (1-sm1).*(sm1>1); % to bound sm in [0,1];
    %
    rm2(i+1) = rm2(i) + dt*(hw2/pi + 2*rm2(i)*vm2(i) - rm2(i)*(gs2(i) + alpha));
    vm2(i+1) = vm2(i) + dt*(vm2(i)^2 - alpha*vm2(i) + gs2(i)*(er - vm2(i)) - pi^2*rm2(i)^2-
wm2(i) + mu2 + Iext2);
    wm2(i+1) = wm2(i) + dt*(a2*(b*vm2(i) - wm2(i)) + wjump2*rm2(i));
    sm2(i+1) = sm2(i) + dt*(-sm2(i)/tsyn + sjump*rm2(i));
    sm2 = sm2 + (1-sm2).*(sm2>1); % to bound sm in [0,1];
end

t = 0:dt:tend;

%% ===== The End =====
% Integration of the mean-field model of two populations
% By Liang Chen
% Nov. 5, 2021
%
%% initial conditions:
rint1 = 0;
vint1 = 0;
wint1 = 0;
sint1 = 0;
%
rint2 = 0;
vint2 = 0;
wint2 = 0;
sint2 = 0;
%sm = sm + (1-sm).*(sm>1); % to bound sm in [0,1];
%
%% integration

```

```

[t,y] = ode45(@(t,y)
mf_sys_2p(p1,p2,gsyn11,gsyn12,gsyn21,gsyn22,mu1,hw1,mu2,hw2,alpha,Iext1,Iext2,er,a1,a
2,b,wjump1,wjump2,tsyn,sjump,t,y)',[0,tend],[rint1,vint1,wint1,sint1,rint2,vint2,wint2,sint2]'
);
%
%% output of variables
rm1 = y(:,1);
vm1 = y(:,2);
wm1 = y(:,3);
sm1 = y(:,4);
%
rm2 = y(:,5);
vm2 = y(:,6);
wm2 = y(:,7);
sm2 = y(:,8);
% Liang Chen, May 12, 2021, Updated on Oct. 5, 2021
%
% the dimensional system eq.(59)-(62) in [Nicola2013Bif]
%
% dimensional system: Upper case
% dimensionless syst: lower case
% Population 1: subscript 1; Population 2: subscript 2
%
% I don't write the vector expression, e.g. v = [v1;v2], to avoid too large
% size of the matrix for Matlab.
%
%
%% Initial conditions
%
v1 = zeros(N1,1); v2 = zeros(N2,1);
w1 = zeros(N1,1); w2 = zeros(N2,1);
s11= zeros(N1,1); s12= zeros(N1,1);
s21= zeros(N2,1); s22= zeros(N2,1);
%
% evolution of the mean variables
v1_mean = zeros(1,tend/dt+1); v2_mean = zeros(1,tend/dt+1);
w1_mean = zeros(1,tend/dt+1); w2_mean = zeros(1,tend/dt+1);
%
% evolution of synaptic variables of the first neuron
% because of all-to-all connectivity, each neuron in the same group is the same
sstore11 = zeros(1,tend/dt+1); sstore12 = zeros(1,tend/dt+1);
sstore21 = zeros(1,tend/dt+1); sstore22 = zeros(1,tend/dt+1);

%% synaptic connection weight matrix (dimensionless)

```

```

smax11 = sjump*ones(N1,N1); % all-to-all coupling with the same weights
p11 = N1/N;
n11 = N1;

smax12 = sjump*ones(N1,N2);
p12 = N2/N;
n12 = N2;

smax21 = sjump*ones(N2,N1);
p21 = N1/N;
n21 = N1;

smax22 = sjump*ones(N2,N2);
p22 = N2/N;
n22 = N2;
%
%% store the spike time and index of neuron to plot the rasterplot
firings1 = [];
firings2 = [];
%
%% Simulation, Euler integration
%
for i = 1:tend/dt+1
    % restore the previous variable for the Euler integration
    v1_ = v1;          v2_ = v2;
    w1_ = w1;          w2_ = w2;
    s11_ = s11; s12_ = s12;    s21_ = s21; s22_ = s22;

    sstore11(i) = s11(1); % store the first neuron. because of all-to-all
        % connectivity, each neuron in the same group is the same
    sstore12(i) = s12(1);
    sstore21(i) = s21(1);
    sstore22(i) = s22(1);
    %
    %% neurons not in the refractory period vreset < v < vpeak
    n_ref1 = find(v1_ >= vreset & v1_ <= vpeak);
    v1_mean(i) = mean(v1_(n_ref1)); % at the time (i-1)*dt
    w1_mean(i) = mean(w1_(n_ref1));
    %
    n_ref2 = find(v2_ >= vreset & v2_ <= vpeak);
    v2_mean(i) = mean(v2_(n_ref2)); % at the time (i-1)*dt
    w2_mean(i) = mean(w2_(n_ref2));

    %% neurons fire at vinf

```



```

fired_inf1 = find(v1_ >= vinf);
firings1 = [firings1; (i-1)*dt + 0*fired_inf1, fired_inf1];
v1(fired_inf1) = -v1_(fired_inf1);
w1(fired_inf1) = w1_(fired_inf1) + wjump1;
%
fired_inf2 = find(v2_ >= vinf);
firings2 = [firings2; (i-1)*dt + 0*fired_inf2, fired_inf2];
v2(fired_inf2) = -v2_(fired_inf2);
w2(fired_inf2) = w2_(fired_inf2) + wjump2;

%% neurons not fire
n_fired1 = find(v1_ < vinf);
gs1 = p11*gsyn11*s11_(n_fired1) + p12*gsyn12*s12_(n_fired1);
rhs1 = v1_(n_fired1).*(v1_(n_fired1) - alpha) - w1_(n_fired1) ...
    + eta1(n_fired1) + Iext1 + gs1.*(er - v1_(n_fired1));
v1(n_fired1) = v1_(n_fired1) + dt*rhs1;
w1(n_fired1) = w1_(n_fired1) + dt*(a1*(b*v1_(n_fired1) - w1_(n_fired1)));
%
n_fired2 = find(v2_ < vinf);
gs2 = p21*gsyn21*s21_(n_fired2) + p22*gsyn22*s22_(n_fired2);
rhs2 = v2_(n_fired2).*(v2_(n_fired2) - alpha) - w2_(n_fired2)...
    + eta2(n_fired2) + Iext2 + gs2.*(er - v2_(n_fired2));
v2(n_fired2) = v2_(n_fired2) + dt*rhs2;
w2(n_fired2) = w2_(n_fired2) + dt*(a2*(b*v2_(n_fired2) - w2_(n_fired2)));
%
s11(n_fired1) = s11_(n_fired1) + dt*(-s11_(n_fired1)/tsyn) ...
    + sum(smax11(n_fired1,fired_inf1),2)/n11; % row sum
s12(n_fired1) = s12_(n_fired1) + dt*(-s12_(n_fired1)/tsyn) ...
    + sum(smax12(n_fired1,fired_inf2),2)/n12; % row sum
%
s21(n_fired2) = s21_(n_fired2) + dt*(-s21_(n_fired2)/tsyn) ...
    + sum(smax21(n_fired2,fired_inf1),2)/n21; % row sum
s22(n_fired2) = s22_(n_fired2) + dt*(-s22_(n_fired2)/tsyn) ...
    + sum(smax22(n_fired2,fired_inf2),2)/n22; % row sum
%
% s11 = s21 = s1; s12 = s22 = s2;
s = [s11 s12; s21 s22];
s = s + (1-s).*(s>1); % to bound s in [0,1]
% s11 = s11 + (1-s11).*(s11>1); % to bound s in [0,1];
% s21 = s21 + (1-s21).*(s21>1);
% s12 = s12 + (1-s12).*(s12>1);
% s22 = s22 + (1-s22).*(s22>1);

end

```

```

%% calculation of the instantaneous firing rate
twin = 2*10^(-2);
%% population 1
fired_time1 = firings1(:,1);
%
tmax1 = find(fired_time1 <= fired_time1(end) - twin);
% tmax(1): the first time index when fired_time + twin > fired_time(end)
fired_num1 = zeros(tmax1(end)+1,1);
fired_num1(1) = length(find(fired_time1 < twin));
%
for i = 1:tmax1(end)
    fired_num1(i+1) = length(find(fired_time1 >= fired_time1(i) ...
        & fired_time1 < twin + fired_time1(i)));
end
avg_fired_time1 = [0;fired_time1(1:tmax1(end))];
R1 = fired_num1/twin/N1; % the firing rate

%% population 2
fired_time2 = firings2(:,1);
%
tmax2 = find(fired_time2 <= fired_time2(end) - twin);
% tmax(1): the first time index when fired_time + twin > fired_time(end)
fired_num2 = zeros(tmax2(end)+1,1);
fired_num2(1) = length(find(fired_time2 < twin));
%
for i = 1:tmax2(end)
    fired_num2(i+1) = length(find(fired_time2 >= fired_time2(i) ...
        & fired_time2 < twin + fired_time2(i)));
end
avg_fired_time2 = [0;fired_time2(1:tmax2(end))];
R2 = fired_num2/twin/N2; % the firing rate

%% ==== end ====
% Mean-field model of two populations
% By Liang Chen
% Nov. 5, 2021
%
function dy =
mf_sys_2p(p1,p2,gsyn11,gsyn12,gsyn21,gsyn22,mu1,hw1,mu2,hw2,alpha,Iext1,Iext2,er,a1,a
2,b,wjump1,wjump2,tsyn,sjump,t,y)
%
% The mean field model (no delay):

```

```

% r'   = hw/pi + 2*r*v_mean - r*(g*s + alpha)
% v_mean' = -w_mean + mu + v_mean^2 + g*s*(er - v_mean)...
%           - alpha*v_mean - pi^2*r^2
% w_mean' = a*(b*v_mean - w_mean) + wjump * r
% s'     = -s/ts + sjump*r
%
% The heterogeneous source: the applied current
%   the Lorentzian distribution: mu (center), hw (half width)
%
%=====
%
gs1 = p1*gsyn11*y(4) + p2*gsyn12*y(8);
gs2 = p1*gsyn21*y(4) + p2*gsyn22*y(8);
%
% population 1
dy(1) = hw1/pi + 2*y(1)*y(2) - y(1)*(gs1 + alpha);
dy(2) = y(2)^2 - alpha*y(2) + gs1*(er-y(2)) - pi^2*y(1)^2 -y(3) + mu1 + Iext1;
dy(3) = a1*(b*y(2) - y(3)) + wjump1*y(1);
dy(4) = -y(4)/tsyn + sjump*y(1);
%
% population 2
dy(5) = hw2/pi + 2*y(5)*y(6) - y(5)*(gs2 + alpha);
dy(6) = y(6)^2 - alpha*y(6) + gs2*(er-y(6)) - pi^2*y(5)^2 -y(7) + mu2 + Iext2;
dy(7) = a2*(b*y(6) - y(7)) + wjump2*y(5);
dy(8) = -y(8)/tsyn + sjump*y(5);
%
end
% two populations of Izhikevich neurons
% strongly adapting, weakly adapting
%
% Liang Chen, May 15, 2021,
% updated on Oct. 5, 2021
%
% Upper case for dimensional form, variables: V, W, S
% lower case for dimensionless form, variables: v, w, s
% Iapp for dimensional syst. I for dimensionless syst.
%
%% Parameters for the dimensional form of the Izhikevich neuron eq.(59)-(62) in
[Nicola2013Bif]
%
% the recovery eq.
beta = -1;      % Resonator/Integrator Variable
VR = -65;      % resting membrane potential
%

```

```

% the membrane potential eq.
C = 250;          % Capacitance
k1 = 2.5;        % spike width factor
VT = VR + 40 - beta/k1; % threshold
Er = 0;         % Reversal Potential
%
% synaptic eq.
Sjump = 0.8;
Tsyn = 4;       % synaptic time constant
%
% the resetting rule
Vpeak = 30;
Vreset = -55;
%
N = 10^4; p1=0.8; p2=1-p1;
% Strongly adapting      Weakly adapting
N1 = N*p1;      N2 = N-N1; % number of neurons in each network
Gsyn11= 200; Gsyn12=200; Gsyn21= 200; Gsyn22=200; % maximal conductances between
populations: [SA-SA SA-WA; WA-SA WA-WA]
TW1 = 200;      TW2 = 20; % time constant of the adaptation eq.
Wjump1= 200;    Wjump2= 100;% adaptation jump
IEXT1 = 0;      IEXT2 = 0; % external current
%
%
%% Parameters for the dimensionless Izhikevich neuron eq.(50)-(53) in [Nicola2013Bif]
%
% the recovery eq.
a1 = C/(TW1*k1*abs(VR));    a2 = C/(TW2*k1*abs(VR));
b = beta/(k1*abs(VR));
%
% the membrane potential eq.
alpha = 1 + VT/abs(VR);
er = 1+Er/abs(VR);

gsyn11 = Gsyn11/(k1*abs(VR)); gsyn12 = Gsyn12/(k1*abs(VR));
gsyn21 = Gsyn21/(k1*abs(VR)); gsyn22 = Gsyn22/(k1*abs(VR));
%
% the external current
Iext1 = IEXT1/(k1*VR*VR);   Iext2 = IEXT2/(k1*VR*VR);
%
%
% synaptic eq.

```

```

tsyn = Tsyn*k1*abs(VR)/C;
sjump = Sjump*C/(k1*abs(VR)); % different from [Nicola2013bif]
%
% the resetting rule
vpeak = 1 + Vpeak/abs(VR);
vreset = 1 + Vreset/abs(VR);
wjump1 = Wjump1/(k1*VR*VR); wjump2 = Wjump2/(k1*VR*VR);
%
% variables
% v = 1 + V/abs(VR);
% w = W/(k1*VR*VR);
% s = S;
% T = C/(k1*abs(VR))*t % T for dimensional system, t for dimensionless
% system
%

%% === end ===
% By Liang Chen, May 12, 2021
% Updated on May 20, 2022
%
%
% Simulation of the two-population network of Izhikevich neurons
% dimensional form of eqs.
% heterogeneous parameters with the Cauchy/Lorentzian distribution
%
%
% ref: Liang Chen, Sue Ann Campbell, Exact mean-field models for spiking
% neural networks with adaptation
% preprint: https://arxiv.org/abs/2203.08341
%
%
%=====
tic
clc
clear
%% values of the parameters
parameters

a2 = 10*a1; % for dimensionless model
%but if from dimensionanl model, 10 times changes because of numerical error

vpeak = 200; vreset = -vpeak;
vinf = 200;

```

```

%% Euler integration parameters
dt = 10^(-3);
tend = 1200;
time = 0:dt:tend;
Tend = tend/(k1*abs(VR)/C);% Total simulation time, dimensional time (ms)

%% heterogeneous parameter, Lorentzian distribution

mu1 = 0.05;          mu2 = 0.05; % centre
hw1 = 0.02;          hw2 = 0.02; % half width

% random generation
eta1 = cauchyrnd(mu1,hw1,N1,1);
eta2 = cauchyrnd(mu2,hw2,N2,1);

% % deterministic generation, typo in [Montbrio2015], "tan", not "atan"
% eta1 = zeros(N1,1);
% for j = 1:N1
%   eta1(j) = mu1 + hw1*tan(pi/2*(2*j-N1-1)/(N1+1));
% end
%
% eta2 = zeros(N2,1);
% for j = 1:N2
%   eta2(j) = mu2 + hw2*tan(pi/2*(2*j-N2-1)/(N2+1));
% end

%% mean-field model

%Izh_SAWA_mf; % Euler method to integrate ODEs

Izh_SAWA_mf_ode % ODE45

%% network model (see details in run.m of the folder Simulation_QIF)

Izh_SAWA_network

%% save data

save('Izh_mf_network_SAWA.mat');

%% plot figures

fig_plot2

```

toc

%% ===== The end =====