



**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

MASTER THESIS

Bořek Šťastný

**Detection and analysis of polychronous
groups emerging in spiking neural
network models**

Department of Software and Computer Science Education

Supervisor of the master thesis: Mgr. Cyril Brom, Ph.D.

Study programme: Computer Science

Study branch: Theoretical Computer Science

Prague 2018

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In date

signature of the author

Title: Detection and analysis of polychronous groups emerging in spiking neural network models

Author: Bořek Šťastný

Department: Department of Software and Computer Science Education

Supervisor: Mgr. Cyril Brom, Ph.D., Department of Software and Computer Science Education

Abstract: How is information represented in real neural networks? Experimental results continue to provide evidence for presence of spiking patterns in network activity. The concept of polychronous groups attempts to explain these results by proposing that neurons group together to fire in non-synchronous but precise time-locked chains. Several methods for the detection of such groups have been proposed, however, they all employ extensive searching in network structure, which limits their usefulness. We present a new method by observing spiking dependencies in network activity to directly detect polychronous groups. Our method shows comparatively more efficient computation by trading off detection selectivity. The method allows for analysis of polychronous groups emerging in noisy networks. Our results support the existence of structure-forming properties of spontaneous activity in neural network.

Keywords: neural networks, polychronous groups, spiking neurons, auditory cortex

Contents

1	General Introduction	3
1.1	Aim of thesis	4
1.2	Structure of thesis	4
2	Introduction to neurobiology	5
2.1	Nervous system	5
2.2	Neurons	5
2.2.1	Cell structure	6
2.2.2	Resting membrane potential	6
2.2.3	Action potential	7
2.3	Synapses	8
2.3.1	Mechanism of chemical synapse	8
2.3.2	Synaptic plasticity	9
3	Introduction to neural modeling	11
3.1	Model detail	11
3.1.1	Sub-cellular modeling	11
3.1.2	Cellular modeling	12
3.1.3	Neural population modeling	12
3.1.4	Modeling requirements for developing spiking patterns	13
3.2	Spiking neural networks	13
3.2.1	Modeling of cortical areas	13
3.2.2	Neuronal dynamics	14
3.2.3	Synaptic dynamics	16
3.3	Network activity and polychronous groups	19
4	Existing methods for polychronous group detection	21
4.1	History and development of existing methods	21
4.2	Polychronous group scanning	21
4.3	Polychronous group detection	23
4.4	Discussion	25
5	Graph search based detection method	26
5.1	Requirements: polychronous group constraints	26
5.2	General concept	27
5.3	Method structure	30
5.4	Graph construction	31
5.5	Graph decomposition	32
5.6	Complexity of graph search based method	36
5.7	Noise in the network	37
5.8	Selectivity of group detection	38
5.9	Summary of presented methods	38

6 Spiking neural network simulation	40
6.1 Simulator requirements	40
6.2 SUSNOIMAC	40
6.3 Revising the simulator	41
6.3.1 Modeling of short term plasticity	41
6.3.2 Update of model connectivity	42
6.4 Results	42
6.5 Discussion	44
7 Analysis of polychronous groups in spiking neural networks	45
7.1 Related works	45
7.2 Polychronous groups emerging in spontaneous activity	48
7.2.1 Experiment method	49
7.2.2 Results	50
7.2.3 Discussion	52
8 Conclusion	54
Bibliography	55
List of Figures	60
List of Tables	61
Appendices	62
A User documentation	63
A.1 Environment version	63
A.2 Installing the implementation	63
A.3 Function call	63
A.4 Function parameters	63
A.4.1 Input parameters	63
A.4.2 Output parameters	64
A.5 Interpreting the output	64
A.6 Loading script	65

1. General Introduction

How neuronal activity represents information in a network has been widely debated topic and several conflicting theories have been proposed. We know that neurons develop action potentials when they receive enough stimulation and signal to other neurons via synapses but do individual spikes and their timing matter or is the frequency of neuronal activity the only significant form of coding? The beginning to this debate can be traced back to an experiment conducted by Edgar Adrian more than 90 years ago. Weights were hung from a frog muscle and the impulses produced by sensory nerve endings were recorded. Interestingly, as the weight (as so the stimulus) increased, the frequency of spikes recorded from sensory nerves increased as well (Adrian, 1926). This led to a conclusion that the frequency of spiking events has to play a role in information representation in neural networks. These ideas evolved into the classical firing rate theory (Shadlen and Newsome, 1998) of neuronal coding where neurons modulate their mean firing rates to transmit information. Due to the relative ease of experimental recording of firing rates, the firing rate theory was successfully used to describe the properties of many types of sensory and cortical neurons and their interactions (Deschenes, 1989; Fuglevand et al., 1993; Fuglevand and Segal, 1997; Tateno and Robinson, 2006; Enoka and Duchateau, 2017) In contrast, a number of scientists have argued that neural computation critically relies on the temporal coordination of spikes. Many observations of precisely repeating spiking patterns in the network activity have been reported (Abeles, 1991; Abeles and Gat, 2001; Lindsey et al., 1997; Prut et al., 1998; Villa et al., 1999; Mao et al., 2001; Ikegaya et al., 2004; Riehle et al., 1997; Beggs and Plenz, 2004). Recent *in vivo* research shows that neurons can generate these patterns with millisecond temporal precision (Chang et al., 2000; Tetko and Villa, 2001).

Perhaps the most researched concept involving precise spiking patterns is that of synfire chains (Abeles, 1991; Bienenstock, 1995; Diesmann et al., 1999; Ikegaya et al., 2004). First mentioned in (Abeles, 1982), synfire chains are layered feed forward structures naturally occurring in a network, where neurons from one layer fire synchronously to excite activity in the next layers. It has been shown that individual neurons can be part of many chains suggesting great capacity of coding in networks where synfire chains develop. (Bienenstock, 1995) A counter argument against the theory of synfire chains could be considered the findings by (Swadlow, 1974, 1985, 1988). There, conduction delays have been shown to be sometimes as small as 0.1 ms and other times as large as 44 ms, depending on the type and location of the neurons. This would contradict the premise of synfire chains where conduction delays between neurons from any two layers are expected to be equal or negligible to elicit synchronous activity.

More recently, Izhikevich proposed concept of polychronous groups (Izhikevich, 2006) where neurons activate not necessarily synchronously but rather in a precise, repeatable, time-locked patterns. He argued that polychronous firing is not an obstacle for representing information in a network but rather an advantage allowing for greater computational capacity. Early research in computer simulations (Izhikevich, 2006; Maier and Miller, 2008; Martinez and Paugam-Moisy, 2009) suggests that the number of polychronous groups representable in spiking

neural networks with conduction delays is extensive, far exceeding the number of neurons. Furthermore, the studies show (Izhikevich, 2006; Maier and Miller, 2008; Guise et al., 2015) that while the number of polychronous groups represented in a random network is small, polychronous groups readily emerge when the network is presented with various stimuli and synaptic weights are allowed to adapt using mechanisms of long term plasticity. This has been speculated as a possible mechanism how memories could be stored in the mammalian brain (Izhikevich, 2006).

While several methods for detection of polychronous groups appearing in simulated recorded activity have been proposed (Izhikevich, 2006; Martinez and Paugam-Moisy, 2009; Sun et al., 2015), they employ extensive search leading to computation complexity that prohibits their use in large neural network simulations (Izhikevich, 2006). Moreover none of these methods have been designed to operate on noisy networks, but noise in the form of spontaneous release of synaptic neurotransmitter has been shown to play an important role in development and formation of neural network structures (Andreae and Burrone, 2015; Hartmann et al., 2015).

1.1 Aim of thesis

The aim of this thesis is to adapt an existing spiking neural network model (Popelová, 2013) and to provide a software tool for polychronous groups detection. Algorithms for detection of stimuli-based polychronous groups as well as polychronous groups emerging during spontaneous activity will be implemented. The existing neural model will be revised and extended to provide more plausible computation and polychronous group detection. In addition to the software implementation, several experiments with varying model inputs will be performed to observe how polychronous groups develop in response to different stimuli.

1.2 Structure of thesis

In Chapters 2 and 3, the basic concepts of neurobiology and neural modeling are established. Chapter 4 analyses existing algorithms for polychronous group detection. A new method is proposed in Chapter 5 and advantages and disadvantages of presented methods are discussed. Chapter 6 describes the adaptation and revision of SUSNOIMAC simulator, which allowed for experimentation on emergence of polychronous groups in spontaneous activity. Methods and results of these experiments are discussed in Chapter 7. The thesis is summarized and concluded in Chapter 8.

2. Introduction to neurobiology

Before we delve into the field of computational neuroscience, we find it necessary to introduce some concepts of neurobiology, namely the basics of neural anatomy and physiology, which form the pillars of understanding of structure and function of nervous systems. After all, these concepts serve as the underlying model of all neural simulations and many scientists strive to replicate their mechanisms as closely as possible to further our understanding of human nature.

In this chapter we mention notable functions of the nervous system and describe its signature neuron cell. The basic mechanism of neuronal signaling, the development of action potential is presented. Finally, the synaptic connection between neurons is described.

2.1 Nervous system

Found in all vertebrates and most invertebrates, nervous system is a part of an organism that directly or indirectly controls most organs in its body. Typically composed of the central nervous system (the brain, brain stem and spinal chord) and the peripheral nervous system (the nerves), its functions are numerous and varied. It is responsible for transferring sensory information such as sight or hearing through sensory nerves to the specialized centers in the brain where these inputs get processed. It is also in the brain, where many of the high level functions of nervous system reside. Different areas allow for forming memories, learning, speech, instinctive behavior and self consciousness. The motor cortex, a part of the most notable outside area of the brain called cerebral cortex, is responsible for planning and execution of voluntary movements by sending signals through nerves to skeletal muscles. The nervous system also mediates many other systems in a body, for example it influences heart frequency, sleeps cycle or body temperature. Many centers of these autonomous functions reside in the brain stem, a part of brain adjoining and structurally continuous with the spinal chord.

2.2 Neurons

Neurons are the primary components of a neural system. They are cells able to transmit electrical and chemical signals to other cells in the system. The number of neurons in the nervous system is vast, 16 billions of neurons are reported in the cerebral cortex alone (Herculano-Houzel, 2009). The interconnected neuronal structures of nervous system are usually called neuronal networks. There are many types of neurons differing both in anatomy and their signaling dynamics. Here we focus on the basic properties of all neurons.

2.2.1 Cell structure

Notable parts of the neuron cell include the soma, dendrites and axon (see Figure 2.1). The soma is the compact body of the neuron, the nucleus and other organelles are situated here. The branch like structure of dendrites is the region where most of the signal inputs to the neuron occur. Finally, the axon is usually a long, cable like structure that can extend far from the soma and serves as a carrier of signals to other neurons. Neuron has usually only one axon, however, it typically branches out to allow for numerous connections to other cells. The nerves of the peripheral nervous systems contain bundles of axons wrapped in layers of connective tissue.

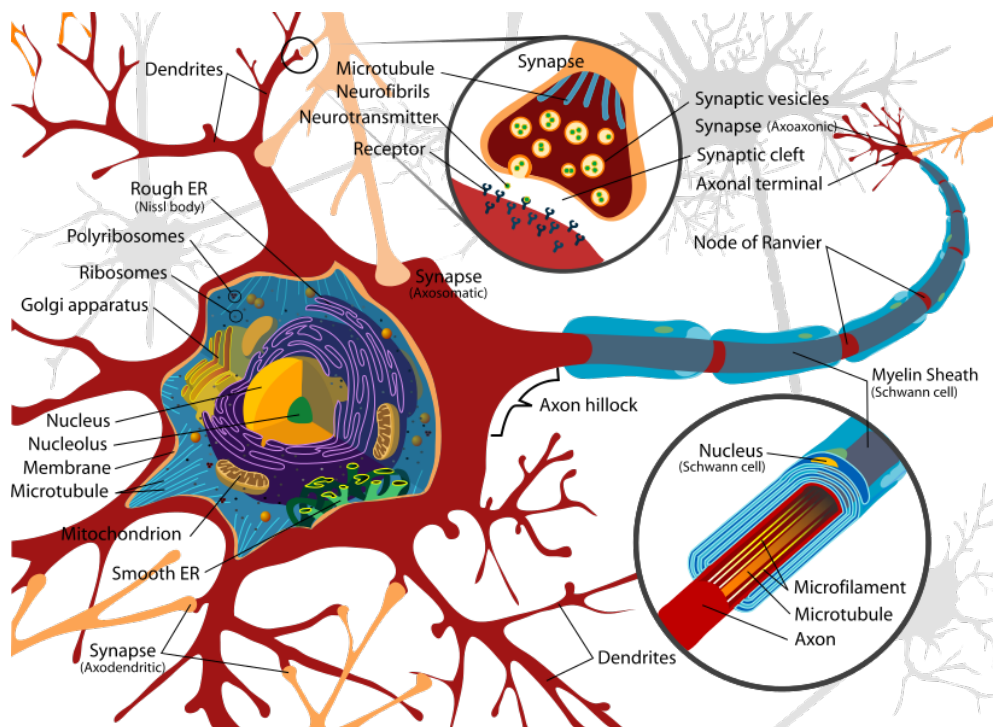


Figure 2.1: Diagram of a neuron, from (wikipedia.org, 2007).

2.2.2 Resting membrane potential

The most notable feature of the neuron cell is its excitable membrane. It is formed by lipid bilayer separating the inside of the cell from the extracellular space. Protein formations embedded in the membrane called ion pumps work to establish concentrations of potassium (K^+), sodium (Na^+) and calcium (Ca^{2+}) ions that differ inside and outside the cell. Selectively permeable ion channels (another example of membrane protein structure) allow specific flow of these ions according to the established concentration gradient. This movement of ions charges the membrane potential until an equilibrium is reached where the chemical gradient force equals the electrical force acting on the charged ions. The result of this interaction is a polarized membrane with resting membrane potential typically around -70 mV, greater concentration of K^+ ions inside the cell and greater concentration of Na^+ ion outside the cell.

2.2.3 Action potential

Signals from other neurons result in small changes of membrane potential and propagate to a place near the origin of the axon called the axon hillock. If these changes are too small, the membrane potential returns to its resting value. However, if this change in potential accumulates and crosses a certain threshold, a special type of ion channels, the so called voltage-gated Na^+ channels start opening (see Figure 2.2). These ion channels remain shut until the membrane potential reaches around -55 mV and then start letting positively charged Na^+ ions into the cell. This strong inward rush of ions temporarily positively charges the membrane to a peak value of about $+30$ mV. At this point the voltage gated Na^+ channels become inactive and at the same time, voltage gated K^+ channels open and let escape positively charged K^+ from the cell. These potassium ions now outside the membrane rapidly reverse the potential development back into negative values until their voltage gated channels too become inactive. This typically results in a small overshoot (hyperpolarization) before the membrane potential settles back on the resting value.

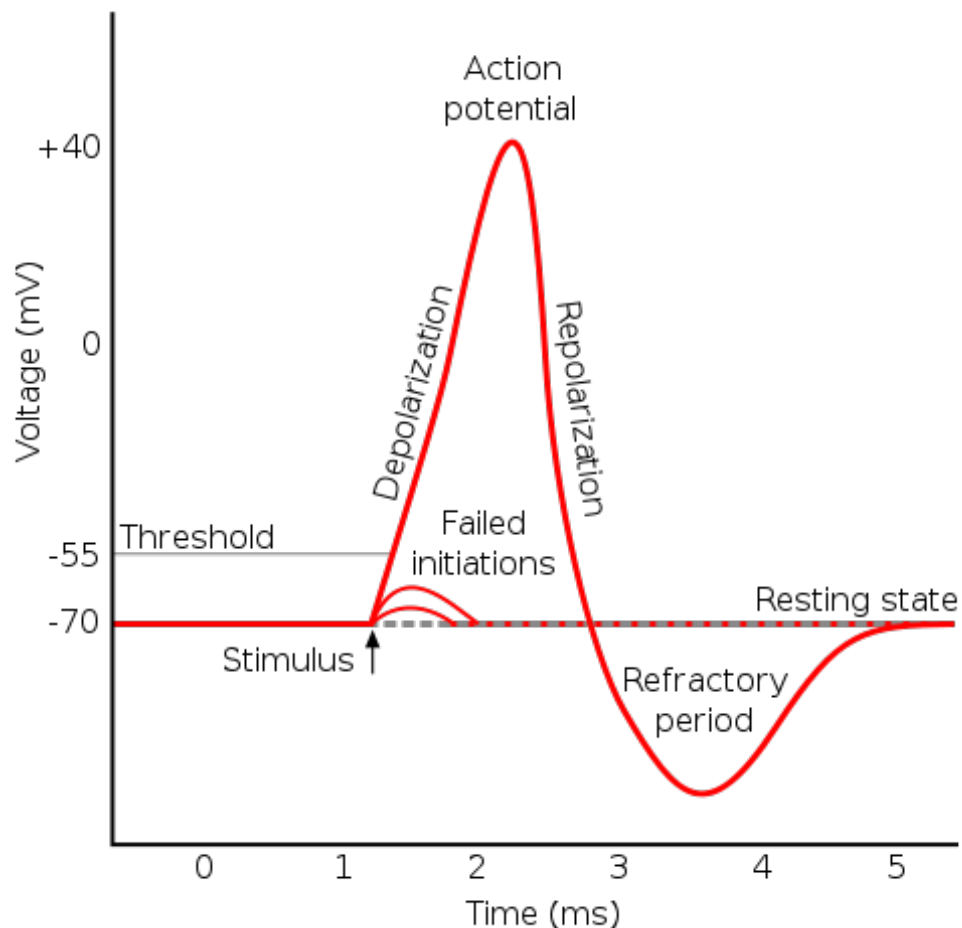


Figure 2.2: Diagram of action potential development, from (wikipedia.org, 2015).

This process, mediated by the voltage gated channels, is called action potential and is the mechanism of neuronal signaling. Once a part of membrane becomes positively charged during the development of an action potential, the inward rush of Na^+ ions pushes the voltage gated channels in neighboring regions of the membrane beyond threshold as well. This causes a chain reaction of action potential development along the axon until it reaches the axonal terminations near other cells. A neuron developing action potential along its axon is said to fire and the action potential is often called a spike. Once a neuron fires, the mechanisms for development of action potential get depleted and a neuron is not able to fire again for a short time, the so called refractory period.

One of the fundamental differences between neuron types is their firing characteristic. Regular spiking (RS) neurons can develop an action potential every 20-60 ms while fast spiking (FS) neurons require only 10ms (Subkhankulova et al., 2010). Yet another firing pattern exhibited by some neurons is the so called intrinsic bursting pattern (IB) where not a single spike but rather a tight group of spikes in a quick succession is produced followed by a period of inactivity. Further common firing patterns are the chattering (CH) and low threshold spiking (LTS), for more information regarding firing patterns see (Izhikevich, 2007). Finally we note that some neurons may develop action potentials without any stimulations or signaling from other neurons. As such, they are called spontaneously firing or spontaneously bursting neurons.

2.3 Synapses

The place where the axonal termination connects to another cell is called a synapse. Common are synapses to other neurons, muscle cells or gland cells (cells responsible for releasing hormones to the body). Two kinds of synapse connections exist. The first is the electrical synapse also called a gap junction. Electrical synapse allows direct connection between neurons and direct transmission of action potential. As this is quite rare form of connection, it is not discussed further in this text. By far more numerous is the second kind, the chemical synapse (see Figure 2.3).

2.3.1 Mechanism of chemical synapse

The chemical synapse consists of presynaptic area of axon termination, the gap between neurons called synaptic cleft and specialized, protein dense postsynaptic area called postsynaptic density. In this case, the cells are not directly connected, the width of the synaptic cleft is usually around 20-40 nm. Instead, the axon terminal contains vesicles with neurotransmitter, a chemical messenger that is released to the synaptic cleft upon arrival of action potential. The molecules of neurotransmitter briefly bind to specific receptors on the postsynaptic side of the connection causing changes to the postsynaptic membrane potential.

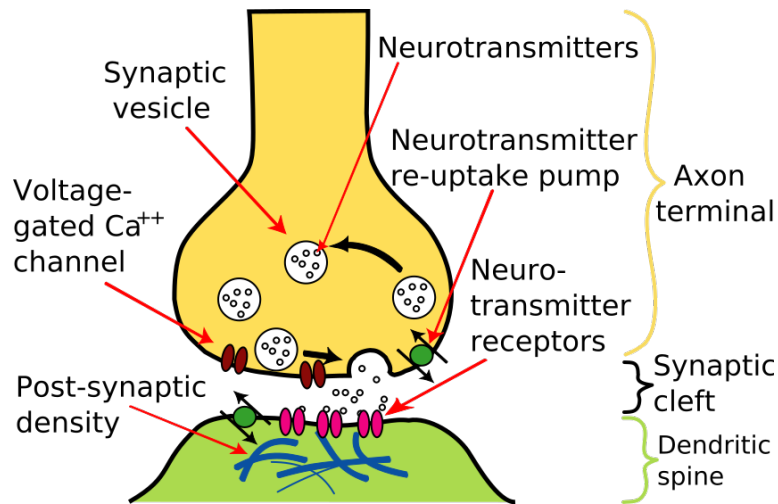


Figure 2.3: Diagram of a synapse, from (wikipedia.org, 2006).

If the release of neurotransmitter causes depolarization of the postsynaptic membrane, i.e. it charges the membrane positively towards the threshold of action potential, we speak of excitatory postsynaptic membrane potential (EPSP) and of excitatory synapse. By contrast, the release causing hyperpolarization of the membrane is said to produce inhibitory postsynaptic membrane potential (IPSP) and the synapse is called inhibitory. Neurons typically release either inhibitory or excitatory neurotransmitters in all of their synapses and are therefore also for simplicity called either inhibitory or excitatory neurons.

Some of the neurotransmitter release has been observed to happen spontaneously and not as response to any incoming signal. Only a small amount of neurotransmitter is released this way and resulting postsynaptic potentials are called miniature EPSPs/IPSPs or synaptic minis. It is generally considered a source of noise in the network activity, since the activity seemingly does not relate to signaling of input stimuli, however it has been shown to play an important role in information processing in the brain (Trapani and Nicolson, 2011) and network development (Andreae and Burrone, 2015).

2.3.2 Synaptic plasticity

The magnitude of change in the postsynaptic membrane potential due to the release of neurotransmitter is usually referred to as synaptic strength or synaptic weight. Synapses are believed to be plastic, meaning their strength can change over time. Mechanism of both short-term (lasting seconds to minutes) and long-term (hours to days) plasticity have been observed in chemical synapses. Short-term plasticity is thought to play important role in short-term adaptations to sensory inputs, transient changes in behavioral states, and short-lasting forms of memory. The short term increase of strength of synapse (facilitation) is attributed to the accumulations of the Ca^{+} ions in the axon terminal, which are responsible for the release of a neurotransmitter upon arrival of an action potential. Short term weakening of the synapse (depression) can be traced to insufficient re-uptake of the neurotransmitter to the presynaptic neuron and subsequent weaker release due to semi-depleted vesicles.

The long term plasticity is assumed to be the possible mechanism of memory retention and learning. Long term increase of synapse strength has been observed in hippocampus (a component of vertebrate brain) in a pair of neurons where the presynaptic neuron fires just before the postsynaptic neuron. Conversely, if the presynaptic neuron fired shortly after the postsynaptic spike, the synapse transmission would weaken. This timing of the firing of the two neurons (precisely in sequence or out of sequence) was found to be the reason behind these phenomena. The increase of synapse strength is usually referred to as long term potentiation (LTP) and the weakening is called long term depression (LTD). Long term plasticity in hippocampus relies on NMDA (N-methyl-d-aspartate) and AMPA (α -amino-3-hydroxyl-5-methyl-4-isoxazole-propionate) receptors and detailed mechanism can be found in (Citri and Malenka, 2008). Similar mechanisms of the spike timing dependent plasticity (STDP) have been also observed in cortical areas (D'amour and Froemke, 2015).

3. Introduction to neural modeling

The development of computational modeling greatly facilitated the research of neural structures. Assuming we have a model faithfully representing a biological network, an experiment that might require months of preparation and collecting of *in vivo* data can be completed in the matter of days. The more experiments are conducted, the better our understanding of the studied mechanisms becomes and the better simulation models can be constructed. This positive loop resulted in unprecedented growth of neuroscience in recent years.

An area where the models of neuronal networks are especially valuable is exploration of new theoretical concepts in order to explain various higher level functions of the brain. Each concept begins as a mere idea that needs to be validated and computational models provide means of rapid testing. This way, basic properties and dynamics can be observed much quicker compared to classical approach of observing *in vivo* recordings of neural activity. One such concept, the polychronous group (Izhikevich, 2006) was proposed as a possible mechanism of information representation in neural networks with encouraging initial results in computer models (Izhikevich, 2006; Maier and Miller, 2008; Chrol-Cannon et al., 2012).

In this chapter we attempt to provide an overview of different type of approaches to simulating real neural networks. As polychronous groups necessitate certain level of simulation detail, we remark on the choice of neural network simulations typically used when required to observe their formation. Such networks are called spiking neural networks and their structures and dynamics are described in greater detail. Finally we formalize the concept of polychronous group itself and define its basic properties.

3.1 Model detail

When designing neuronal network models, a compromise has to be usually made between the detail of network mechanics and computation efficiency. Existing models can be generally classified into three major categories based on simulation simulation detail.

3.1.1 Sub-cellular modeling

Sub-cellular modeling focuses on precise description of ion channels, whose actions make up the essential mechanism of cell to cell communication and signaling. Ever since the discovery of the excitable membrane, the knowledge of these mechanisms expanded and increasingly detailed and complex models were possible to construct. Nowadays with the fully atomistic description of structures of several ion channels (Doyle et al., 1998) and direct experimental observations of a single channel's action (Cha et al., 1999), the simulation detail has reached to modeling individual molecules behaving according to their chemical properties.

While these models provide unprecedented detail of the development transmission of action potential, their complexity is usually prohibitive for simulations where great number of neurons must be simulated (e.g. simulations of regions of mammalian cortex).

3.1.2 Cellular modeling

The next level of abstraction is modeling on the cellular level, where the elemental units of simulation are individual neurons and synapses. The actions of individual ion channels are aggregated and the main focus is on the resulting development of action potentials in neurons. This emphasis on the spiking nature of neurons gave name to the networks, where these neuron models are employed; such are called the spiking neural networks (SNN).

Various neuron models have been proposed with different levels of biological faithfulness and computational complexity. Perhaps the most influential were the findings of (Hodgkin and Huxley, 1952) which lead to creation of the famous Hodgkin-Huxley model. Based on their experiment on the squid giant axon, they were able to describe the development of membrane potential using differential equations for each type of ion channels. To this day, their model remains one of the most biologically faithful simulation techniques on the cellular level and is still used. However, its complexity still proved too high for large scale simulations. Thus, much research has been put into developing new models with the goal of making them simpler yet keeping the resulting neuronal behavior as faithful as possible. Perhaps the simplest model of a spiking neuron is the integrate and fire model. The development of membrane potential is described by a single equation which is essentially aggregating postsynaptic potentials coming from other neurons until a spike threshold is met. At that moment, a spike is generated, the membrane potential is reset and the neuron is prevented from firing during its refractory period. Such a simple approach is easily computable, but is not very precise in simulating actual neurons. One of the notable shortcomings is that its incapable of producing bursting or other more complex neuronal behavior and only simulations of regular spiking neurons are possible. The presented models and a model attempting to combine their strengths will be further discussed in section 3.2.

Similar to the neuron models, there are many approaches to simulating the synaptic connection between neurons ranging from a single weight parameter that is used as the final input to the postsynaptic neuron to full simulation of the post synaptic currents. Techniques for short-term and long-term plasticity have been developed, perhaps the most widely used is the simulation of spike timing dependency plasticity.

3.1.3 Neural population modeling

The category of neural population modeling encompasses a wide range of approaches that have been in literature called neural mass models, mean field models, neural field models or firing rate models. Generally, an attempt is made to describe the collective action of neural assemblies directly using some sort of population averaging. Collective state variables, typically defined as averages over

the group of cells, are used in order to describe the population activity directly in a single model. The reduction in model detail compared to spiking neural networks results in significant reduction in computational complexity which lead to widespread use in computer simulations. However such models cannot be used when precise spatial and temporal dependencies between neurons need to be observed (as is the case when observing synfire chains or polychronous activity) due to the averaging of neurons in a population.

3.1.4 Modeling requirements for developing spiking patterns

In order to observe the precise spiking patterns of polychronous groups, the simulation detail must allow for development of individual spikes in the network (Izhikevich, 2006). This naturally prohibits the use of population models. By contrast, sub-cellular modeling is usually too costly to simulate large networks and such level of simulation detail is generally not necessary. Thus, it is almost universal practice to remain within the realm of spiking neural networks which provide the best compromise between network size and faithfulness of neuron behavior. These can also model connection delays, which is essential of observing polychronous groups (Izhikevich, 2006).

3.2 Spiking neural networks

The observation of polychronous groups necessitates (in computer simulations) the use of spiking neural networks. Here we describe the typical techniques of modeling network structure and network dynamics in SNNs in greater detail. Later in the text when discussing computer simulations of behavior of polychronous groups, we assume the availability of simulation detail at least on the level displayed by SNNs.

3.2.1 Modeling of cortical areas

As already mentioned, the cerebral cortex contains distinct areas responsible for processing sensory information. This close connection to real world stimuli provides at least a basing understanding of what the function of these areas might be and what the inputs to the neural networks are. As a result, many of the models of real biological networks focus on modeling the cerebral cortex, especially the visual and auditory cortex.

Knowledge of neuronal structures of the network is essential when modeling real biological networks. Despite great efforts of researches, precise description of biological neural networks is impossible mainly due to sheer size of the networks (human cortex is reported to contain around 16 billion neurons (Herculano-Houzel, 2009)). Such complexity necessitates significant simplifications when designing the model structure. Fortunately many similarities exist between certain neurons. The behavior of the neuron usually depends on its experimentally observable properties and location in the network. This makes it possible to categorize neurons into so called neuronal types and assume that neurons of a single type exhibit similar behavior. Notable neuronal types found in cortical areas are:

- pyramidal neurons (p): they are the primary excitation units of the mammalian cortex. They show typically regular spiking, or sometimes chattering, or intrinsically bursting firing patterns (Contreras, 2004).
- spiny stellate neurons (ss): they are star-like shaped, excitatory neurons exhibiting regular spiking patterns(Contreras, 2004).
- basket interneurons (b): they exhibit fast spiking patterns and are one of the primary inhibitory neurons(Contreras, 2004).
- non-basket interneurons (nb): they are a group of morphologically different neurons including double-bouquet cells, neurogliaform cells and Martinotti cells and can show various firing patterns including low threshold spiking and latent spiking patterns.

Structure of the neural network of the cortical areas can be generally split into several layers. Typically, 6 layers are recognized (L1-L6) but layers 2 and 3 are sometimes considered as one layer because the distinction between these layers in real cortex is not clear. Types of neurons in different layers may differ in size and other parameters. For example, pyramidal neurons in L6 layer are usually different from pyramidal neurons in layer L4. This leads to definition of subtypes of neurons based on their location in network. Layer subtypes are denoted in this text by the type of the neuron and its location in cortical layer, for example, basket neuron from L5 is denoted b5.

A critically important description of the network structure is the connectivity between neurons (referred to as connectome). Unfortunately, this is possibly the least complete information about the network structure, because the total number of synapses greatly exceeds the number of neurons in a network. A following simplification can be made. Each neuron of a certain neuronal type has a typical number of synapses to other types. Neurons of the same type tend to project axons to specified layers where they form set number of synapses to other neurons in a determined radius. Which layers and how far the axons project can be estimated from available research data. Further information about this method of simulating the network connectome can be found in (Izhikevich and Edelman, 2008; Popelová, 2013)

3.2.2 Neuronal dynamics

When the structure of simulation model has been determined, the question of how to simulate the network dynamic arises. As mentioned, spiking neural networks employ neuron models to simulate development of action potential of neuron membrane. Models frequently used in computer simulations are presented below.

Hodgkin-Huxley model

The model of Alan Hodgkin and Andrew Huxley is mathematical model that describes how action potentials in neurons are developed and propagated. Often referred to as conductance-based model, it is based on results of voltage-clamp experiments on ion channels of the squid giant axon (Hodgkin and Huxley, 1952).

The general form of the models is described by four non-linear ordinary differential equations:

$$\begin{aligned}
C \frac{dV}{dt} &= -\bar{g}_K n^4 (V - E_K) - \bar{g}_{Na} m^3 h (V - E_{Na}) - \bar{g}_L (V - E_L) \\
\frac{dn}{dt} &= \alpha_n(V)(1 - n) - \beta_n(V)n \\
\frac{dm}{dt} &= \alpha_m(V)(1 - m) - \beta_m(V)m \\
\frac{dh}{dt} &= \alpha_h(V)(1 - h) - \beta_h(V)h
\end{aligned}$$

where V is the membrane potential and C is the membrane capacitance. Variables n , m , and h are dimensionless quantities between 0 and 1 representing potassium channel activation, sodium channel activation, and sodium channel inactivation, respectively. Values \bar{g}_L , \bar{g}_K , \bar{g}_{Na} are the maximal value of membrane conductance for the leak current, potassium current and sodium current, respectively. Parameters E_L , E_K , E_{Na} represent the respective reversal potentials. Functions $\alpha_n(V)$, $\alpha_m(V)$, $\alpha_h(V)$, $\beta_n(V)$, $\beta_m(V)$, $\beta_h(V)$ describe the transition between open and closed states for respective channels and were fitted using the data from the experiment. Precise description of these functions can be found in the founding work (Hodgkin and Huxley, 1952). The model shows highly detailed and realistic development of action potential. Further, due to the numerous parameters, many different neuronal types can be simulated displaying various firing patterns. However, due to its complexity, its use in large scale simulations is limited.

Leaky integrate and fire model

On the other side of the complexity spectrum of neuron models resides the Leaky integrate and fire model. It is given by a single differential equation with a spike generation criterion:

$$\begin{aligned}
C \frac{dV}{dt} &= -g_L (V - E_L) + I_{syn}(t) \\
\text{if } (V \geq V_t) &\text{ then } V \leftarrow V_r
\end{aligned}$$

where V represents the membrane potential, C is the membrane capacitance, g_L is the leak conductance, E_L is the leak potential. The neuron model integrates the synaptic inputs I_{syn} and when the membrane potential exceeds the spiking threshold V_t , the spike criterion is effectuated: a spike is registered and the membrane potential is reset to its resting value V_r . The simple nature of this model led to its widespread use in computer simulations even though its modeling capabilities are limited. The model can display only regular spiking pattern, the shapes of action potentials are not modeled and assumed to be identical every time a neuron fires and the spiking threshold and resting potential values are fixed. The result is limited biological plausibility of the leaky integrate and fire model.

Izhikevich model

Both previously mentioned models have their limitations, the Hodgkin-Huxley model is too costly to compute and the Leaky integrate and fire model is too simple to simulate diverse behavior of neuronal types. As a result, many scientists have tried to combine the strengths of both approaches to develop a model that inherits some of the advanced featured of Hodgkin-Huxley model while keeping the resulting model efficient. One of the successful models that emerged was the model by Izhikevich. It was published in multiple versions; we present here the generalized form found in (Izhikevich, 2007). The model is characterized by differential equations:

$$C \frac{dv}{dt} = k(v - v_r)(v - v_t) + u + I$$
$$\frac{du}{dt} = a(b(v - v_r) - u)$$

$$\text{if } (v \geq v_p) \text{ then } v \leftarrow c, \quad u \leftarrow u + d$$

where v is the membrane potential, u represents overall membrane recovery current, C is the membrane capacitance, v_r is the resting membrane potential, v_t is the instantaneous threshold potential, v_p is a spike cutoff value, I is the synaptic input current. Parameters a , b , c , d can be tuned to model different spiking patterns of neuronal types. The most frequent patterns of spiking activity (RS, FS, CH, IB, LTS, LS among others) can be modeled (Izhikevich, 2007). Additionally, it has been shown to require much less computation time than the Hodgkin-Huxley model (Izhikevich, 2007). This makes it one of the best models for simulating large scale realistic networks.

3.2.3 Synaptic dynamics

The most important attributes of synaptic transmission are the conduction delay and synaptic strength. The mechanism of chemical synapse is complicated and varies significantly between neurons in the nervous system. Moreover, the process is stochastic in nature and noise is introduced through the spontaneous release of neurotransmitter. Precise modeling of these processes is thus very complex and would be computationally costly considering the number of synapses in a typical neural network. This usually necessitates simplifications when modeling large networks such as visual or cortical areas of mammalian cortex.

Synaptic strength

One of the simplest synaptic models employs a single weight parameter w to represent the overall synaptic strength (the resulting EPSP/IPSP of synaptic transmission). The input $I_{syn}(t)$ of postsynaptic neuron n_p in time t is computed as:

$$I_{syn}(t) = \sum_{i=1}^k w_i$$

where w_i are the weights of synapses to neuron n_p transmitting signals in time t . Despite its simplicity, this synaptic model is used in many simulations of real neural networks (Izhikevich and Edelman, 2008; Phoka et al., 2012; Popelová, 2013). It is often used in conjunction with models of short-term and long-term plasticity which improve its limited plausibility (described in following sections). Also, mechanisms for spontaneous activity can be simulated by introducing random noise to neuronal inputs. More complex simulations of synaptic transmission exist, for example models simulating synaptic conductances. More information on synaptic modeling can be found in (Schutter, 2009).

Short term plasticity

The basic synapse model can be expanded by modeling short term synaptic plasticity. One of the successful approaches is the phenomenological model by (Markram et al., 1998) where n is the depression variable that models the availability of neurotransmitter and r the facilitation variable modeling the probability of release of single synaptic vesicle. The values $n(t)$ and $r(t)$ are changing in time and serve as scaling factors of the base synaptic weight. The development of the short term plasticity is given by the following differential equations.

$$\begin{aligned}\frac{dn}{dt} &= \frac{1-n}{D} - nr\delta(t-t_n) \\ \frac{dr}{dt} &= \frac{r_o-r}{F} + r_o(1-r)\delta(t-t_n)\end{aligned}$$

The parameter D is the time constant of neurotransmitter replenishment. Analogically, F is the time constant of the return of the release probability back to its baseline value r_o . The function δ is the Dirac delta function. Parameters D, F and r_o have been experimentally determined for number of cortical neurons (Markram et al., 1998). Each firing of a presynaptic neuron occurring at time t_n , decreases the depression variable n by nr , and increases the facilitation variable r by $r_o(1-r)$.

Most synapses exhibit prominently only one of the possible forms of short term plasticity (either facilitation or depression). This can be used to simplify the modeled dynamics and combine both phenomena into single variable x . Similarly to previous case, the synaptic weight gets scaled by the value of $x(t)$. The dynamic is given by the following differential equation.

$$\begin{aligned}\frac{dx}{dt} &= \frac{1-x}{\tau_x} \\ x &\leftarrow px \quad \text{when presynaptic neuron fires}\end{aligned}$$

The value of the scaling factor x tends to return to equilibrium $x = 1$ with the time constant τ_x and it is reset by each spike of the presynaptic cell to the new value px . Any value $p < 1$ decreases x and results in short-term synaptic depression, whereas $p > 1$ results in short-term synaptic facilitation (Izhikevich and Edelman, 2008). Simulation results suggest (Izhikevich and Edelman, 2008) that even the simplified model is capable of faithfully reproducing short term synaptic plasticity observed in recordings of real neurons.

Long term plasticity

The most popular form of long term synaptic plasticity is the spike timing dependent plasticity. Its general form can be found in (Song et al., 2000). As described in previous chapter, repeated arrivals of presynaptic spikes shortly before the postsynaptic spike lead to long-term potentiation (LTP) of the synapse, whereas a repeated spike arrival after the postsynaptic spike leads to long-term depression (LTD). This behavior can be implemented modeling value M , which is a modification value added to the synaptic weight. Let Δt be the time interval between presynaptic and postsynaptic spike, The value of M is computed as:

$$M(\Delta t) = \begin{cases} A_+ \cdot e^{\Delta t/\tau_+} & \text{if } \Delta t < 0 \\ -A_- \cdot e^{\Delta t/\tau_-} & \text{if } \Delta t > 0 \end{cases}$$

where A_+ , A_- are the maximum values of change, which occurs when neurons spike almost simultaneously, τ_+ and τ_- are parameters determined experimentally (typically around 20 ms).

Conduction delay

The transmission of signal from presynaptic to postsynaptic neuron is not instantaneous. Rather, the action potential developed at the axon hillock has to traverse the entire presynaptic axon, opening and closing voltage gated channels along its length. The speed of propagation ranges from 0.5 m/s for non myelinated axons of some cortical neurons to 120 m/s in sensory and motor pathways (Evarts, 1965; Swadlow, 1989). Further, typical chemical synapse introduces delay of 0.5-4 ms (Chun-Hua et al.). Finally, the action potential is propagated through the postsynaptic neuron's dendrite until it reaches the soma. The resulting conduction delays reach up to 10 ms in cortico-cortical connections and the cortico-spinal delays reach upwards of 70 ms (Swadlow, 1989).

The numerous properties of possible connections between neurons pose difficulty when deciding how to model conduction delays in the network simulation. Accepted practice is to note the structure of the nervous system the involved neurons belong to (e.g. cortex, spine) and then assume the most likely connection based on available research. A rough estimate of associated conduction velocity is made and the modeled delay is computed from the distance between neurons (Izhikevich, 2006; Popelová, 2013).

3.3 Network activity and polychronous groups

In spiking neural networks, individual neurons develop action potentials according to their neuron model and signal through synaptic connections. When observing network activity, the individual firing actions of neurons are collected in lists called spike trains. These records hold the most important attributes of all individual neuronal spikes: the involved neuron and the time of the firing. To avoid confusion, we provide a definition of neuronal spike as regarded in the context of neural computation and especially when discussing spiking patterns in a network.

Definition. Neuronal spike is an ordered pair (t, n) where n denotes the number of a neuron and t denotes the time of development of action potential at the axon hillock of the neuron n .

The observation that neural network activity exhibits patterns of spikes has been made decades ago (Abeles, 1982, 1991). Since then, it has been speculated that these patterns might explain the mechanisms behind information representation and processing in networks. (Abeles, 1991; Izhikevich, 2006; Guise et al., 2015). Researchers set to acquire evidence in the form of precise spike pattern repetition using both computer simulations and data from real networks with many positive results (Riehle et al., 1997; Villa et al., 1999; Chang et al., 2000; Tetko and Villa, 2001; Izhikevich, 2006). One of the newest concepts belonging to these spike patterns is the so called polychronous group (PG) introduced by Izhikevich in (Izhikevich, 2006). The novel idea behind the concept is that the temporal pattern of firing (when the neurons fire) is as important as the spatial component (which neurons fire) for information representation. Neurons typically require simultaneous arrival of multiple EPSPs to fire. If the EPSPs have to arrive at the same time, the presynaptic neurons must also fire according to the conduction delay. Conduction delays have been shown to be repeatable with sub-millisecond precision and varying widely between pairs of neurons (Swadlow, 1974, 1985; Shadlen and Newsome, 1998). Thus not synchronous, but rather precise, time locked activation of neurons would be the candidate of information representation in a network. This is significant departure from the more known theory of synfire chains (Abeles, 1991) where synchronous action is expected to transfer information through layers of neurons. Multiple experiments in computer simulations were conducted supporting the existence of polychronous groups (Izhikevich, 2006; Chrol-Cannon et al., 2012; Guise et al., 2015). We use the formal definition of polychronous group as found in (Martinez and Paugam-Moisy, 2009).

Definition. Let us have a set of trigger neurons $N = \{n_1, n_2 \dots n_k\}$. A k -triggered polychronous group is the set P of spikes in chain reaction following activation of trigger neurons at a time pattern $t_1, t_2 \dots t_k$. We call the initial spikes $TS = \{(t_1, n_1), (t_2, n_2) \dots (t_k, n_k)\}$ the spike trigger set. The size of a polychronous group P is the size of the set $|P|$.

It should be noted that polychronous groups as defined above exist even in absence of any network activity. One can imagine them as time-locked spike patterns supported by the structure of connection delays in the neural network that might emerge in neuronal activity if the trigger neurons are activated in a

precise sequence. Those groups are referred to as *supported polychronous groups* as per (Martinez and Paugam-Moisy, 2009).

However only a fraction of polychronous groups actually emerge during neural network activity. We call such groups *activated polychronous groups* (again in accordance with (Martinez and Paugam-Moisy, 2009)). We also follow their practice to consider a group activated if the recorded activity in a network contains the trigger spike set, because according to the definition once the trigger spikes occur, the whole polychronous group appears. However it may happen that the rest of the polychronous group fires slightly differently from the predicted structural pattern, especially in the presence of spontaneous activity and other forms of noise (Izhikevich, 2006).

4. Existing methods for polychronous group detection

To evaluate properties and emergence of polychronous groups in neural networks, a method capable of detecting such patterns needs to be employed. In this chapter, we present existing methods for polychronous method detection in spiking neural networks and make the distinction between scanning for all possible polychronous groups in network structure and detection of activated groups in recorded spike trains. Complexity analysis is carried out and existing algorithms are further discussed highlighting several problems in actual use.

4.1 History and development of existing methods

Since the conception of polychronous groups in the founding literature (Izhikevich, 2006; Martinez and Paugam-Moisy, 2009), several methods searching for polychronous groups were developed. Those methods reflect the authors' needs to establish polychronous group as a valid idea in the presence of similar concepts e.g. synfire chains while also trying to provide evidence of precise spatio-temporal patterns as an opposition to competing theories like the mentioned firing rate model. As a result, existing methods employ an extensive scanning of neural network structure for groups of neurons that might produce, if fired appropriately, a non-trivial chain reaction of spikes in a network. In this text we call such methods *polychronous group scanning*. It is important to make distinction between scanning algorithms and *polychronous group detection* algorithms that will be discussed later in the chapter. Scanning algorithms search for polychronous groups supported by network structure of synapse delays. In contrast detection algorithms search through recorded spike data for activated polychronous groups, meaning groups that actually appear during network activity.

Using scanning algorithms, researchers were able to show important facts about spiking network structure including that the number of polychronous group representable by a spiking neural network far exceeds the number of neurons, which might imply a vast capacity of network memory (Izhikevich, 2006), or that polychronous groups readily emerge in presence of spike timing dependent plasticity, further establishing the connection to representation of memories in the neural networks. (Izhikevich, 2006).

4.2 Polychronous group scanning

An example of an algorithm belonging to the group of polychronous group scanning is presented here. Since those algorithms show many similarities and general layout, for presentation the algorithm found in (Martinez and Paugam-Moisy, 2009) has been chosen, as it will later provide the most direct comparison to our proposed method. Further methods can be found in (Izhikevich, 2006) (Maier and

Miller, 2008). The variations between existing scanning algorithms are mentioned in the algorithm description.

Algorithm description

The scanning algorithm assumes the knowledge of neuronal model, network dynamics and synapse connectivity matrices:

$synapse_weight(N_i, N_j)$: weight of synapse from neuron N_i to neuron N_j

$synapse_delay(N_i, N_j)$: delay of synapse from neuron N_i to neuron N_j

The layout of a scanning algorithm is then as follows:

Algorithm 1 Scanning algorithm

```

1:  $PGs = \emptyset$ 
2: for all combination of  $s$  neurons out of  $n$  neurons of the network do
3:   for  $i := 1 \dots n$  do
4:      $NbTriggeringConnection_i := 0$ 
5:   end for
6:   for all trigger neurons  $TN_k, k := 1 \dots s$  do
7:     for all neurons  $N_i, i := 1 \dots n$  do
8:       if  $synapse\_weight(TN_k, N_i) \geq 0$  then
9:          $NbTriggeringConnection_i := NbTriggeringConnection_i + 1$ 
10:      end if
11:    end for
12:  end for
13:  for all neurons  $N_i, i := 1 \dots n$  do
14:    if  $NbTriggeringConnection_i \geq NbSpikesNeeded$  then
15:       $delay\_max := \max_s synapse\_delay(TN_k, N_i)$ 
16:      for all trigger neurons  $TN_k$  do
17:         $t_k := delay\_max - delay(TN_k, N_i)$ 
18:      end for
19:      Run simulation to calculate the  $PG$  with trigger neurons
20:       $\{NT_1, NT_2, \dots, NT_s\}$  firing with timing  $\{t_1, t_2 \dots t_s\}$ 
21:      if  $size(PG) \geq SizeNeeded$  then
22:        Save the spike trigger set of  $PG$  to  $PGs$ 
23:      end if
24:    end if
25:  end for
26: end for
27: return  $PGs$ 

```

It is assumed that neurons in the spiking neural network fire if they receive EPSPs simultaneously from at least $NbSpikesNeeded$ presynaptic neurons. General idea is to try every possible combination of s neurons to see if they could form a meaningful set of triggers and then verify if such triggers produce large enough polychronous group in the network. Algorithm proceeds by counting the number of postsynaptic connections on every neuron in the network coming

from the chosen neuron group. This number is stored for each neuron N_i into field $NbTriggeringConnection_i$. If there arises postsynaptic neuron with more than $NbSpikesNeeded$ connections, one can be sure that it fires if corresponding presynaptic neurons trigger in a sequence reflecting their delay so that the EPSPs arrive simultaneously. This prevents trying every possible timing pattern and saves computing time.

In the for-loop in step 13, every such postsynaptic neuron and resulting firing sequence or trigger neurons is considered. The network activity is simulated and the resulting polychronous group is observed. If it passes chosen size criterion, it is accepted into the solution set.

The network simulation of polychronous group chain reaction (step 19) can be carried out in many different approaches. It is mainly here where the existing algorithms differ the most. Izhikevich uses his model of differential equations (Izhikevich, 2007) in a discrete time simulation to simulate membrane potential and observe spiking activity (Izhikevich, 2006) while Martinez and Paugam-Moisy use event based simulation (Martinez and Paugam-Moisy, 2009). There, neurons are considered firing (and registering their EPSPs into calendar queue) if they either exceed given number of arriving spikes or if their membrane potential exceeds certain threshold. Important to note here is that the simulation is carried out without any noise in the network. This is because the simulation serves as criterion describing which spikes (and the involved neurons) belong to the resulting polychronous group and as such must be carried out deterministically.

Complexity

It is obvious that the complexity of the polychronous group scanning algorithm depends on the complexity of network simulation. However general statement about the layout can be made. Since $s \ll n$ and all n neurons must be checked for every combination of trigger neurons, the algorithm is at least $\Omega(n^{s+1})$. Further analysis of complexity can be found in (Martinez and Paugam-Moisy, 2009) where a proof can be found that using event based simulation and assuming realistic connectivity in real networks, the complexity of $O(n^{s+1})$ can be achieved.

Summary

To summarize thus far, regardless of which simulation method is used the basic premise of scanning algorithm remains the same: search through the synaptic delay structure of a network for activation pattern of neurons capable of producing non-trivial activity in the network. These patterns may or may not surface during network activity. Those that do, need to be detected by a separate class of algorithms called polychronous group detection.

4.3 Polychronous group detection

Despite the promising results of the scanning algorithms, there is not much research providing a complete solution of detection of polychronous groups actually occurring either during spiking network simulation or in vivo spike trains (activated polychronous groups). To our knowledge, the only algorithms are due to

(Martinez and Paugam-Moisy, 2009) and (Sun et al., 2015). They are a direct extension of scanning methods and require the use the scanning algorithm to create database of polychronous groups supported in the network prior to their execution. Again, both methods are quite similar in their basic idea and the method of (Martinez and Paugam-Moisy, 2009) was chosen as a direct continuation of presented scanning algorithm.

Algorithm description

Using the scanning method, a database of supported polychronous groups in the form of their corresponding spike trigger sets is created. Then the algorithm searches through the recorded activity comparing activated and database patterns.

Recorded spike trains are assumed in the form of matrix:

$Spike(m, i)$: time of m th spike of neuron N_i

Algorithm 2 Detection algorithm of Martinez and Paugam-Moisy

```

1: Run scanning algorithm to obtain database of supported PGs
2: for all PG in the database do
3:   retrieve trigger neurons  $\{NT_1, NT_2, \dots, NT_s\}$  from database
4:   retrieve trigger timings  $\{t_1, t_2, \dots, t_s\}$  from database
5:   for all  $m$  do
6:      $time := Spike(m, NT_1)$ 
7:     if  $\forall k \in [2; s], \exists l : Spike(l, k) = time + t_k - t_1$  then
8:       Save PG as activated
9:     end if
10:  end for
11: end for

```

For every known polychronous group in the database, the first neuron whose spiking activity belongs to the group is taken and its spiking record is ran through. Each spike in this record represents a potential placement of the polychronous group. To verify activation of the rest of the group, the spiking records of all other trigger neurons are checked if they contain spikes according to the precise firing pattern associated with the spike trigger set of the group. If so, the activation of the polychronous group is detected and is reported.

Complexity

For every polychronous group in the database, the firing of the first neuron of the trigger set is fixated. On average, the spiking record of every neuron contains $O(S/n)$ spikes where S is the number of all recorded spikes. Thus $O(S/n)$ fixations are possible on average. The rest of the spike trigger set can be verified in constant time Assuming that P denotes the number of polychronous group in the database, resulting computational complexity of the detection algorithm is $O(P * S/n)$. Further details can be found in (Martinez and Paugam-Moisy, 2009).

4.4 Discussion

Martinez and Paugam-Moisy conclude that their proposed method is applicable both to spiking neural network simulation and also to analysis of real data (Martinez and Paugam-Moisy, 2009). This claim certainly holds true for the detection part of the algorithm if we already know possible polychronous groups supported in the network structure and we are able to simultaneously record large number of neurons. While difficult to provide, such recordings are becoming realistically obtainable through advancing methods of data acquisition. However, we remind that the detection algorithm can be used only in tandem with the scanning algorithm and thus inherits any problems and restrictions of that algorithm.

And it is precisely the part of scanning for polychronous groups that we believe to be problematic. In the case of computer simulations, the complexity of scanning algorithm prohibits analysis of larger networks. Because of that, scanning methods have been restricted to use with network consisting of at most several hundreds of neurons. However, much larger simulations already exist, not to mention the size of real mammalian cortex.

The proposed application for real networks presents even more obstacles. Certainly infeasible is exciting real neurons and observing resulting responses especially considering the great number of combinations that must be explored. More realistic could be simulating real neurons in a computer simulation. However the current knowledge of neuronal and synaptic dynamics still doesn't allow for precise mapping between real and simulated neurons in non-trivial networks. For example, we still have only a very incomplete description of cortex connectome. More importantly, the simulation detail would have to be very high for faithful mapping between neurons increasing computational complexity.

Finally, none of the existing methods account for noise in the network in the form of spontaneous release of neurotransmitter despite being reported as important part of neural network dynamics. (Andreae and Burrone, 2015; Hartmann et al., 2015).

5. Graph search based detection method

We aim to address the issues of existing detection methods by proposing a new method that does not need to employ a scanning phase to detect activated polychronous groups. Rather, we attempt to directly interpret recorded data and detect activated groups based of the spiking structure as this approach avoids the restrictions of the scanning algorithm. We begin this chapter by discussing the definition of polychronous groups. We establish the concept of a spike dependency graph and general ideas behind the method. Two distinct algorithm phases are described in greater detail. Discussed are method applications in noisy networks and selectivity of the algorithm. Finally, our method is compared with existing algorithms.

5.1 Requirements: polychronous group constraints

Following the definition of polychronous group strictly, we notice that the definition allows for structures that do not actually correspond to the general idea of neural activity that begins with a group of triggering spikes which are then followed by a fixed chain reaction propagating in the network. For example, no restriction is posed on the delay between triggering spikes. Presumably, trigger sets spanning absurdly large periods of time are of little use for analysis of spiking activity being extremely unlikely to ever occur in the spiking data. This constraint is posed implicitly on the polychronous groups detected by the algorithm of Martinez since the only combinations of neurons that are considered have all trigger neurons connected to the same postsynaptic neuron. In order to activate the postsynaptic neuron, they must spike within a time window no larger than maximum synaptic delay in the network. The period between the first and last spike of a polychronous group is a good candidate for a user selectable parameter depending on their intended use of detected polychronous groups. We denote this parameter *dmax*.

To ensure non-triviality of polychronous groups, methods for polychronous group detection usually allow either groups larger than specified minimum size as in (Martinez and Paugam-Moisy, 2009) or require the longest path along activation chain to be at least a set number (Izhikevich, 2006). Both constrains are valid criteria and either can be chosen based on the computation complexity and the intended use for the detected polychronous groups. For our method, we have chosen to detect groups where a path can be found between two spikes longer than a number *pathLength* specified by the user.

As for the size of spike trigger set, the scanning algorithm fixes its size in the beginning and returns only trigger sets of that size. If different sizes are required, the user must run the scanning algorithm for every size specified. With increasing size, the complexity of algorithm increases and so the size is usually kept low. For example (Martinez and Paugam-Moisy, 2009) searches for trigger sets of size 3, (Maier and Miller, 2008) uses sets of size 2 to explore properties of polychronous groups. Reasonable might be to require trigger sets of size in a certain range [*minSize*, *maxSize*] selectable by the user.

To recapitulate, it is reasonable to require a detection method to report polychronous group such that:

- the delay between the first and the last spike of a trigger group is at most $dmax$
- the size of trigger set is in the interval $[minSize, maxSize]$
- the longest path along the activation chain is longer than $pathLength$ parameter

where algorithm parameters $path$, $dmax$, $minSize$, $maxSize$ are user selectable.

5.2 General concept

Let us first consider a spike from the recorded activity in a network without noise, we deal with possible sources of noise later in the chapter. Such spike is typically result of previous network activity in the form of spikes arriving from presynaptic neurons. Different types of neurons behave differently and have different spiking characteristics, however, general assumption is that in order to provoke spike or bursting activity on postsynaptic neuron, several EPSPs must arrive together within a short period of time called *jitter*. (Izhikevich, 2006; Martinez and Paugam-Moisy, 2009). To achieve that, the spiking sequence of presynaptic neurons must happen in precise time sequence matching the synapse delays.

Every pair of recorded spikes of connected neurons that corresponds to the synapse delay between them thus represents causality in the spiking structure since when a presynaptic neuron fires, it propagates change in membrane potential along its synapses to every postsynaptic neuron. The presynaptic spike contributes to the postsynaptic spike and the postsynaptic spike depends on it. The level of contribution and dependence may vary between presynaptic neurons and alone they may have not been enough. However, one can be sure that together they were able to produce the postsynaptic spike (because it was indeed recorded). In another words, this precisely timed spike set is enough to propagate activity further and thus directly corresponds to a part of polychronous group. This leads us to a following definition of spike dependency graph.

Definition (Spike dependency graph). Let $S = \{(n, t); \text{neuron } n \text{ fired at time } t\}$ be a set of neuronal spikes representing recorded activity in spiking neural network. Spike dependency graph is a directed graph $G = (V, E)$ where $V = S$ i.e. graph vertices are the recorded spikes. Let $s_1 = (n_1, t_1) \in V$, $s_2 = (n_2, t_2) \in V$. Set E consists of directed edges $e = (s_1, s_2)$ such that there exists synapse between n_1 and n_2 and presynaptic spike of neuron n_1 in time t_1 elicits an EPSP on neuron n_2 contributing to generation of postsynaptic spike in time t_2 .

Further in the text in the context of spike dependency graph, we use terms spike and graph vertex interchangeably since they describe the same concept.

Comparing definitions of polychronous group and spike dependency graph we obtain weak condition for detecting polychronous groups:

Lemma 5.1. Let $G = (V, E)$ be a spike dependency graph corresponding to recorded activity in a network. Let P be the set of vertices of connected subgraph of G , $TS \subseteq P$. Let P satisfy condition:

$$\forall s, s' [s \in P \setminus TS \wedge (s', s) \in E \implies s' \in P]$$

Then P is subset of polychronous group with spike trigger set TS .

Polychronous groups that do not follow such condition may be present in the spiking recording. Some of the predecessors of a spike may not have played any role in the development in cascading activity at all and merely got connected because the neurons happened to trigger in convenient time. Consider the spike dependency graph from figure 5.1. At first glance, we might recognize that it contains polychronous group $P = \{1, 2 \dots 17\}$ with spike trigger set $TS = \{1, 2, 3, 4, 9\}$.

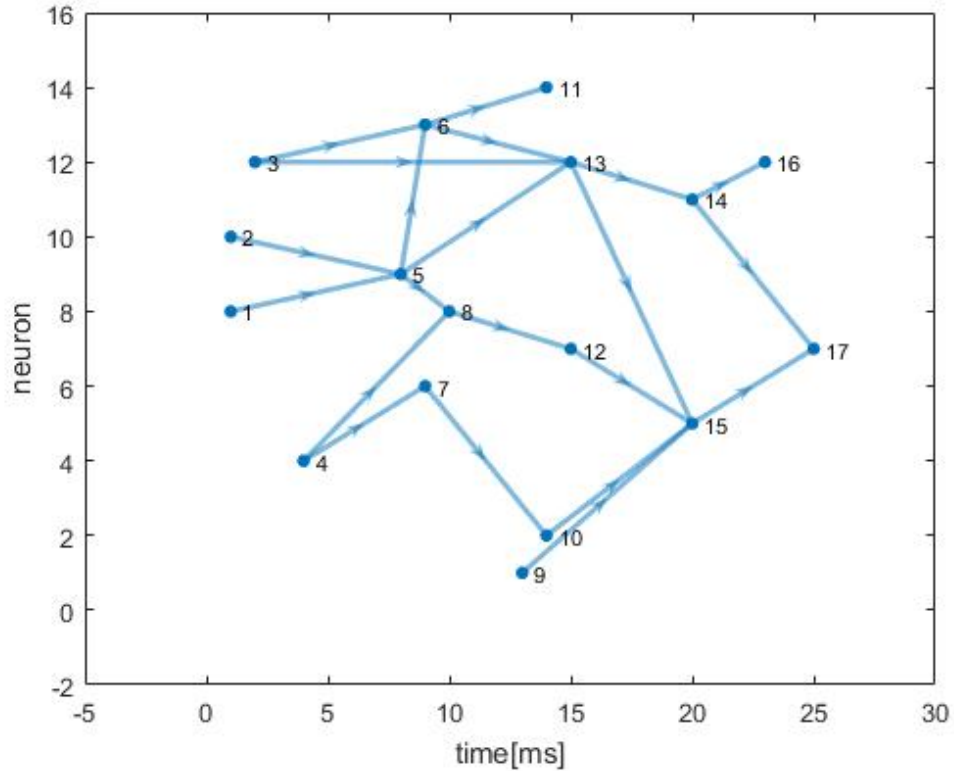


Figure 5.1: Example of a spike dependency graph.

However, this is not the only polychronous group present in the graph. Figure 3.2 shows a subgraph highlighted green. Let us imagine that only the spike set $TS' = \{1, 2, 3\}$ has fired in isolation. Surely the activity must cascade through spikes $C = \{5, 6, 11, 3, 14, 16\}$. This is because the set respects every dependency in the firing cascade and no other influence is needed to activate respective neurons. We can successfully prove the existence of polychronous group $P' = TS' \cup C$ with trigger set TS' and report it.

In contrast, were the set of spikes $TS'' = \{1, 2, 4, 9\}$ to trigger in isolation, one might be able to observe response as in figure 5.3 highlighted red. This

might be the case if spike 15 wouldn't depend on spike 13 meaning neuron 5 would not require neuron 12 to spike precisely 5 ms earlier in addition to other activity coming from neurons 1, 2 and 7 (producing spikes 9, 10 and 12). However as this information is not present in the spiking data, we have to expect that the dependency indeed exists. If the user requires polychronous groups with maximum path length at least 4, this polychronous group cannot be reported based on the structure present in the recorded data even though in reality the maximum path is 6 (from spike 1 to spike 17). However we are still able to at least report an approximation of such group in the form of the previously mentioned group P and its trigger set $TS'' \subseteq TS$. We acknowledge this limitation and focus on detecting as many (approximate) groups as possible.

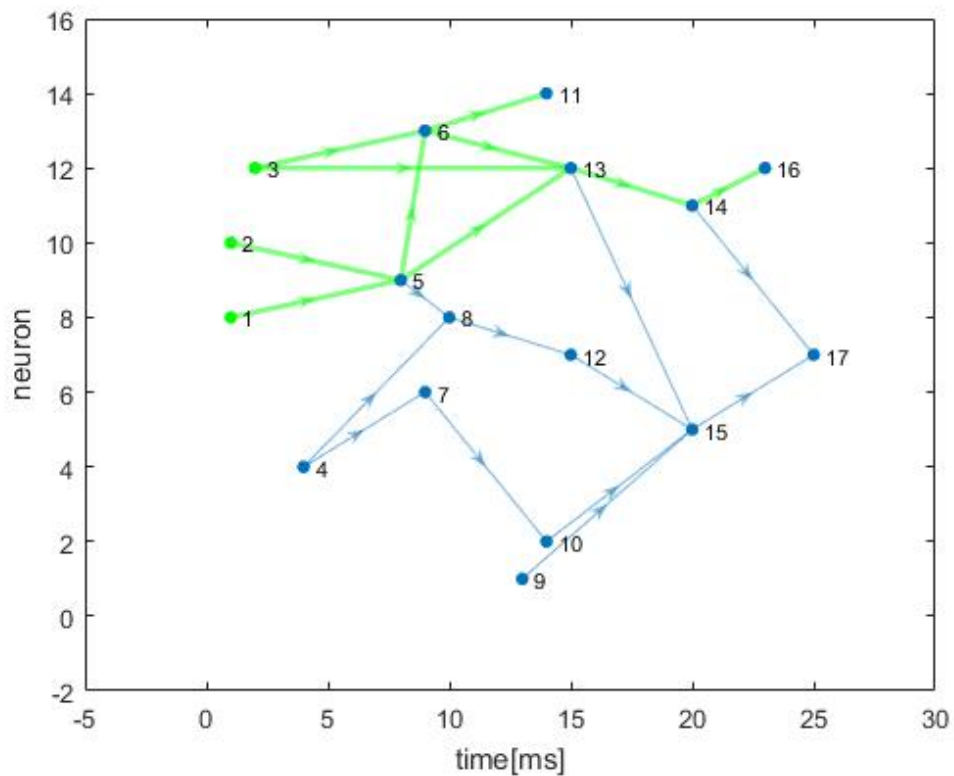


Figure 5.2: Example of a detected activated polychronous group

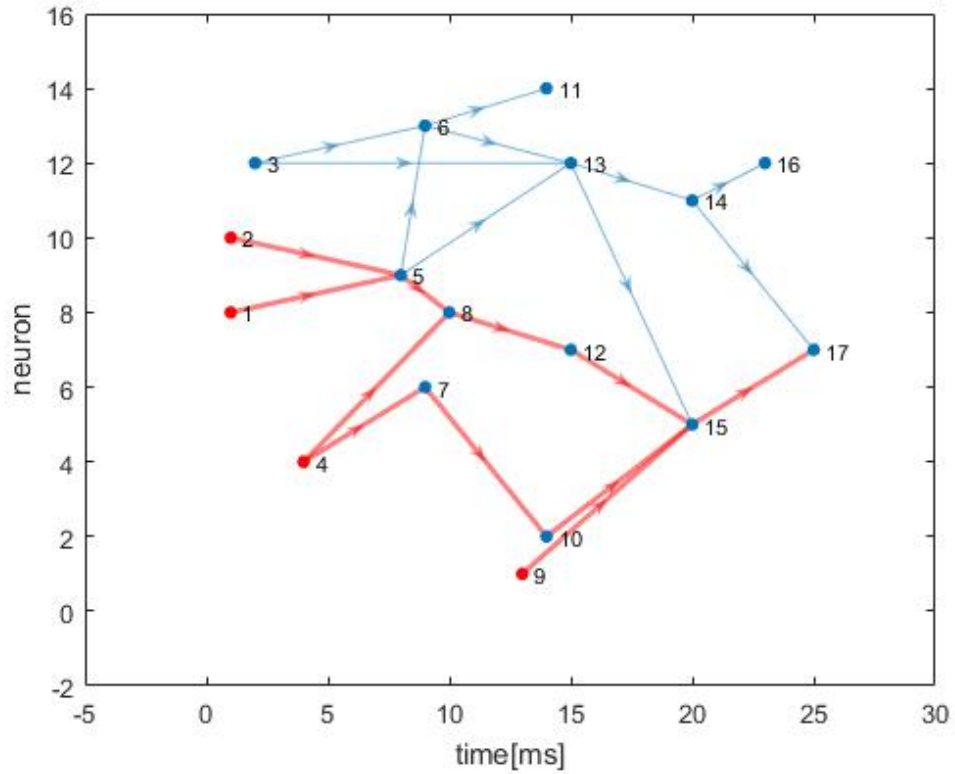


Figure 5.3: Example of a possible activated group that cannot be reported

5.3 Method structure

Instead of scanning for all the possible polychronous groups in a network like in the method of Martinez, we try to directly interpret recorded data. This skips the scanning phase of algorithm or rather performs scanning implicitly using the structure present in the spike recording. The method can be split into two distinct phases of graph construction and graph decomposition. During the first phase, the spike dependency graph is constructed from the recorded spike data. The algorithm then proceeds with the graph decomposition phase where it applies lemma 5.1 to detect as many polychronous groups as possible and report their trigger sets. Any additional constraints that may be posed on the polychronous groups and their trigger sets are applied during this phase. For better clarity, both phases are presented separately, each with its own set of associated variables and procedures.

5.4 Graph construction

Let us begin with the description of the first phase, the construction of a spike dependency graph.

Description of variables

We assume spike list and synapse connectivity matrices:

spikes : list with fields *neuron* and *time* describing individual recorded spikes, sorted by *time* field

synapse_weight(N_i, N_j) : weight of synapse from neuron N_i to neuron N_j , value is zero if no synaptic connection exist between neurons

synapse_delay(N_i, N_j) : delay of synapse from neuron N_i to neuron N_j , value is zero if no synaptic connection exist between neurons

Algorithm phase results in a spike dependency graph in the form of neighbors lists:

successors(s) : list of spikes directly dependent on spike s

predecessors(s) : list of spikes on which spike s directly depends

Algorithm description

Algorithm 3 Construction of spike dependency graph

```
1: for all spikes  $s$  do
2:    $successors(s) := \emptyset$ 
3:    $predecessors(s) := \emptyset$ 
4: end for
5: for all spikes  $s$  do
6:    $preTime = s.time$ 
7:    $preNeuron = s.neuron$ 
8:   if  $preNeuron$  is excitatory then
9:     for all  $postNeuron$ :  $synapse\_delay(preNeuron, postNeuron) > 0$  do
10:      for  $jitter := 0 \dots \maxJitter$  do
11:         $postTime := preTime + synapse\_delay + jitter$ 
12:        if  $\exists s2$ :  $s2.neuron = postNeuron$  and  $s2.time = postTime$  then
13:           $successors(s).add(s2)$ 
14:           $predecessors(s2).add(s)$ 
15:          break
16:        end if
17:      end for
18:    end for
19:  end if
20: end for
21: return  $successors, predecessors$ 
```

For every spike of presynaptic neuron, we check if its postsynaptic connections fired as well. If they happen to fire just after arrival of spike from presynaptic neuron than there is spiking dependency between them and edge is added to the graph. We allow for a short gap between supposed and real postsynaptic spiking in the parameter *maxJitter* which is equivalent to *jitter* in the polychronous group scanning algorithms.

Precise value of *maxJitter* should be tuned by the user and set with the predicted neuronal model in mind. Larger values lead to more interconnected and perhaps more faithful polychronous groups however care must be taken not to create too many false edges. False edges lead to connecting otherwise separate groups together and prevent the algorithm from individually reporting these groups.

The use of *maxJitter* parameter is based on the convention mentioned in (Martinez and Paugam-Moisy, 2009; Maier and Miller, 2008). However we note that stronger EPSP probably influences the postsynaptic neuron longer than a weaker one and thus an arbitrary, fixed value might not be ideal to determine the spike dependency. Moreover, the relation is likely not simple. For simplicity, we have decided not to model such dynamics, however, this allows for possible improvement of the method in the future.

5.5 Graph decomposition

Once the dependency graph is created, the algorithm proceeds with the graph decomposition phase. We present our algorithm description in two steps. The general idea of the algorithm are briefly laid out followed by a detailed description of the algorithm phase.

We are tasked with detecting groups passing the path criterion meaning we need to find a path between two vectors in the group of at least a given size. An existence of a single pair of spikes would suffice to accept the polychronous group. An important observation is made in (Izhikevich, 2006) and (Martinez and Paugam-Moisy, 2009) stating that if the trigger set of polychronous group is activated, the rest of the group will trigger as well. This reduces the problem of detecting the whole polychronous group to detecting the trigger set and proving that there is large enough subset of the chain reaction present in the recorded data to pass path criterion. For that, the condition from lemma 5.1 is used.

An effective way to do this is to take any spike in the graph, let us call this spike a root, and look at its predecessors. Lemma 5.1 gives that if we take all the predecessors as a spike trigger set, than together with the root spike we have found a subset of polychronous group. The maximal path length is 1. To obtain larger paths, new valid trigger set can be generated by replacing any of the trigger spikes with all of its predecessors. At one point, the trigger sets start being sufficiently far away from the root spike to be reportable while still satisfying condition from lemma 5.1 along the way, thus forming a polychronous group.

General idea is then to consider every spike in the graph as root spike. For every root spike, the recursive exploration of possible trigger sets is done as described above. Care must be taken not to repeatedly visit already explored trigger sets and to prevent the exploration from being exponential. A hash map containing a list of already explored trigger sets can be effectively used to cut recursion

and achieve polynomial algorithm with respect to the number of polychronous groups found in the recorded data.

Chronological order is used to process the root spikes in a graph. That is because single trigger set may be obtainable from multiple root neurons. Let us assume this trigger set has been already accepted during exploration from previous root spike. Having already considered the trigger set before, it would be pointless to re-explore from this point, the only novel information being larger maximum path than before due to the new root spike being successor of the former one. This can lead to large saving in computation time.

Description of variables

Let us now describe the phase of graph decomposition in full detail. The algorithm phase requires spike dependency graph connection data in variable:

predecessors : adjacency lists of incoming edges obtained from graph construction phase

successors : adjacency lists of outgoing edges from graph construction phase

We output spike trigger sets of polychronous groups into variable:

accepted : hash set data structure containing accepted spike trigger sets

We output only trigger spike sets that fit the parameters:

minSize : minimal size of trigger set

maxSize : maximal size of trigger set

pathLength : path length required

dmax : maximal delay between first and last spike in the trigger set, it is required that *dmax* is larger than maximum synapse delay in a network

In the algorithm we further make use of the variables:

max_path(v) : shortest path from root spike to spike vertex v

visited(v) a flag indicating if the vector was visited during search

ingraph : list of all vertices visited during search

vertex : current spike vertex during search

timeLimit : time limit on exploration

newTriggerSet : new candidate trigger spike set

rejected : hash set data structure containing rejected spike trigger sets

Finally, to implement breadth first search, we use the data structures:

queue : queue for searching through subgraph

exploreList : queue for trigger spike set exploration

Algorithm description

During the algorithm we consider every spike vertex in the graph as a root vertex. The first part of algorithm computes the longest path from any spike in the predecessor subgraph to the root spike as these would be the candidates for longest paths in the detected polychronous groups. If the algorithm proceeds from the latest to the earliest spike in the subgraph than once the maximum path is computed for any vector, all of its successors have already had the maximum their own maximum path computed. This is because the spikes are ordered by the time of their occurrence and synapse delays are positive only.

Next comes the exploration of possible trigger sets itself. It is started by queuing the root vector as a trivial spike trigger set. Trigger set dequeued from *exploreList* is modified to obtain new candidate spike trigger sets. In the for cycle on line 36, every member is one by one replaced by all its predecessors in the spike dependency graph. The resulting new trigger sets are tested if they still fall into the time limit of search. If not, the search is stopped there and the set is not placed into the queue for further exploration. The exploration is also stopped if the new trigger spike was already created by a differ parent set and thus is already recorded either in accepted or rejected spike trigger set hash maps. Finally, if the newly created trigger set represents novel polychronous group, it is tested for size and path length, either accepted or rejected and placed into an appropriate list.

The exploration may possibly take too long to compute, since polychronous groups may span long periods of time in the recorded activity. This can be reduced by limiting the simulation time depth of search by the user. This significantly speeds up the computation but potentially causes the algorithm to miss valid polychronous groups. However, the reliability of the activation of the polychronous group was observed very high only for the initial part of activation chain and then deteriorating with time as polychronous groups convolve with others (Izhikevich, 2006). Thus it may be prudent to set an upper limit on a polychronous group time span anyway. This is also the practice in (Martinez and Paugam-Moisy, 2009).

Finally, the reason why the list of rejected spike trigger groups is reinitialized for every new root vertex, is that the rejection might have happened due to the path criterion. With a later root spike, the trigger set may increase its longest path from the new root vertex compared to the previous one, thus having a new chance of being accepted.

Algorithm 4 Decomposition of dependency graph

```
1: accepted =  $\emptyset$ 
2: for all spikes s do
3:   max_paths(s) := 0
4:   visited(s) := false
5: end for
6: queue =  $\emptyset$ 
7: for all spikes root do
8:   queue.enqueue(s)
9:   shortest_paths(s) := 0
10:  ingraph := root
11:  visited(s) := true
12:  while queue is not empty do
13:    vertex = queue.dequeue()
14:    for all predecessors p of vertex do
15:      if visited(p) == false then
16:        visited(p) := true
17:        ingraph.add(p);
18:        if vertex.time - p.time  $\leq$  timeLimit then
19:          queue.enqueue(p)
20:        end if
21:      end if
22:    end for
23:  end while
24:  sort ingraph in descending order by spike time
25:  for all spikes v in ingraph do
26:    if v == root then
27:      max_paths(v) := 0
28:    else
29:      max_paths(v) :=  $\max_{s \in \text{successors}(v)} (\text{max\_paths}(s)) + 1$ 
30:    end if
31:  end for
```

Algorithm 4 Decomposition of dependency graph (continued)

```
32:   rejected =  $\emptyset$ 
33:   exploreList.enqueue(root)
34:   while exploreList is not empty do
35:     triggerSet := exploreList.dequeue()
36:     for all spikes s of triggerSet do
37:       newTriggerSet := triggerSet \ {s}  $\cup$  predecessors(s)
38:       if  $\min_{v \in \text{newTriggerSet}}(v.time) \geq \text{root.time} - \text{timeLimit}$  then
39:         if newTriggerSet  $\notin$  accepted  $\cup$  rejected then
40:           exploreList.enqueue(newTriggerSet)
41:           if  $\text{minSize} \leq |\text{newTriggerSet}| \leq \text{maxSize}$ 
42:             and  $\text{max}_{v \in \text{newTriggerSet}(v)}(\text{max\_paths}(s))$  then
43:               accepted.add(newTriggerSet)
44:             else
45:               rejected.add(newTriggerSet)
46:             end if
47:           end if
48:         end for
49:       end while
50:     for all spikes v in ingraph do
51:       max_paths(v) := 0
52:       visited(v) := false
53:     end for
54: end for
55: return accepted
```

5.6 Complexity of graph search based method

During the first phase of algorithm i.e. the graph construction, we run through every spike and look for all possible synaptic connections for spike response in order to detect spiking dependency. Since *maxJitter* is a constant, we get complexity of $O(cS)$ where *c* is the maximum number of connections of a neuron in neural network and *S* is the number of recorded spikes.

For the part of graph decomposition, we consider every spike as a root spike and run breadth first search into predecessor subgraph to compute maximal paths of predecessor spikes. Since the maximum depth is limited, at most c^l edges and vertices are explored, *l* being the time limit of search depth. The complexity of the computation of longest paths is then $O(c^l S)$. Then, the exploration of spike trigger sets is carried out. Every spike trigger set is explored at most once for each root. If we assume that the size of any trigger set is much lower than number of spikes recorded spikes and that we use appropriate hashing function to avoid frequent collisions, the complexity of exploration of algorithm is on average $O(PG * S)$, *PG* being the number of polychronous groups present in the recorded data.

Since probability of synaptic connection between cortical neurons diminishes with increased distance and there is only finite number of neurons in an area of network, many network models consider maximum number of connection per neuron to be constant. Moreover, only a fraction of possible synaptic connection results in dependency in the graph, due to the precise temporal requirement thus limiting vector degree. The complexity of the whole algorithm can be estimated as $O(cS + c^lS + PG * S) = O(PG * S)$ for realistic networks.

5.7 Noise in the network

So far, we have been dealing with noiseless networks. As noise is considered any spiking activity that is not result of incoming spikes from recorded neurons. The problem with noise in the network is that lemma 5.1 relies on causality of spikes. If there are edges created in the graph leading to spikes that occur without involvement of other recorded neurons, the lemma no longer holds true and this might lead to reporting wrong or altered polychronous groups. Most common sources of noise are:

- spontaneously firing or bursting neurons
- artificial inputs (e.g. electrode stimulation)
- activity coming from non recorded neurons (e.g. different parts of cortex)
- synaptic minis

The common denominator of all these phenomena is that they are capable of producing spiking activity in the network on their own without involvement of recorded neurons. However their different origins may prove useful in reducing their influence on detection algorithm.

A list of potentially spontaneously firing neurons can be created and used to filter recorded spiking activity; no edges will be added leading to these spikes during graph construction phase. Especially in the case of computer simulation, complete information about neurons in the network is available and the behavior may be detected even in the case of complicated neuron models. This could be done as a part of preprocessing before actual simulation.

In the case of artificial inputs to the network (stimulating with electrodes), one can simply filter the initial spikes that are obviously result of stimulation. Again, no edges are added leading to this spike vertices.

Activity coming from non recorded neurons generally falls into two categories. Activity coming from other parts of cortex is best treated as inputs to the network since it is hopefully not numerous, this of course requires precise tracking of such inputs. Activity from the region of interest is typically simulated in full and every neuron is recorded. This is probably not the case in real network recordings, however, the more neurons are recorded, the better the performance of the algorithm will be.

Perhaps the most interesting source of noise in the network are the miniature postsynaptic potentials caused by spontaneous release of neurotransmitter. Many functions have been attributed to spontaneous activity, among them keeping a

sort of base line activity in a network. It has been repeatedly reported that synaptic minis play a crucial role in forming of shaping neural connections (Andreae and Burrone, 2015). To distinguish between miniature and normal releases of neurotransmitter, it is generally sufficient to consider the size of generated post-synaptic potential, being much smaller in the case of minis compared to regular EPSP/IPSP.

5.8 Selectivity of group detection

It must be re-emphasized that using the graph search method, it is impossible to report every polychronous group present in the spiking activity due to the weak nature of lemma 5.1 This may potentially cause difficulties with application of the method to analysis of spiking activity, for example the method is obviously not fit to report absolute number of activated polychronous groups present in the data, since some of the groups may be unintentionally connected and reported only as a union of separate groups. Still, many valid groups can be successfully detected by the method which might prove useful in different scenarios especially involving large networks, where no other methods are computable in reasonable time.

5.9 Summary of presented methods

Finally, methods for polychronous group detection are compared (see Table 5.1 for comparison of method properties) and we summarize the analysis of presented methods.

Table 5.1: Comparison of presented methods for detection of polychronous groups.

Properties	Martinez detection	Graph detection
network dynamics	required	not required
synapse connections	required	required
PG scanning	required	not required
trigger set size	increases complexity	implicitly
scanning complexity	$O(n^{s+1})$	$O(PG * S)$
detection complexity	$O(PG * S/n)$	$O(PG * S)$
detection	full	partial

The main improvement introduced by graph search method is the lifting of several prohibitive requirements placed upon the knowledge of neural network in order to detect activated polychronous groups. This leads to much easier application to different simulation and potentially application to analysis of read data. Only a rough idea of neuronal and synapse dynamics is required to tune jitter parameter. Full knowledge of synapse connections in networks is required by both methods though.

Another improvement is the omission of scanning phase of the existing detection methods. The complexity of existing scanning algorithm required before

actual detection increases sharply with the network size prohibiting application on larger networks (as discussed in previous chapter).

It must be noted, that the result of graph search is an approximation of the full detection of activated groups in the network. Certain groups cannot be reported individually based solely on the information contained in the spiking structure. However every reported group satisfies the definition of polychronous group due to the application of lemma 5.1. By contrast, the detection using scanning method uses pre-computed information in the form of database of supported polychronous groups to exactly detect all activated groups.

Finally, existing methods of detection of activated polychronous groups can not be used while working with noisy networks, e.g., networks exhibiting spontaneous activity of synapses. While this poses some complication for the graph search method, it is capable of detecting activated polychronous groups even in the presence of noise.

6. Spiking neural network simulation

In order to conduct experiments analyzing behavior of polychronous groups emerging in neural networks, we needed to be able to simulate simulate such networks with sufficient level of detail. As discussed earlier, polychronous groups necessitate the use of spiking network simulations. In this chapter, we describe our choice of simulator and describe our involvement in revising its parameters and network dynamics to improve the realism and plausibility of simulated activity. Results of the applied modifications are discussed. Simulations of neural networks are used to provide data for experiments presented in the next chapter.

6.1 Simulator requirements

When introducing the concept of polychronous groups, Izhikevich formulated the minimal required properties of networks that are able to exhibit formation of such groups. As these were pivotal for the choice of our network simulator, we provide a recapitulation here. Namely the are:

- The networks simulation needs to allow for development of spikes of individual neurons. Appropriate neuron model should be used for plausible results.
- Conduction delay must be simulated and has to be non-trivial (i.e. not unitary). Emergence of polychronous groups relies on the delay structure of the network, its concept being both spatial and temporal pattern of spiking activity.
- Simulation of synaptic dynamics must include simulation of some for of long term plasticity to allow for formation and development of polychronous groups.

6.2 SUSNOIMAC

To satisfy our the requirements on the simulator, we have decided to adapt the SUSNOIMAC simulator presented in (Popelová, 2013). The name stands for Simulator Using Spiking Neurons Originally Intended for Modeling Auditory Cortex and the notable features of the simulator are:

- The simulator belongs to the category of spiking neural networks and uses Izhikevich neuron model for realistic development of action potential
- Simulations of large scale networks are possible thanks to its high computational efficiency resulting from parallel implementation. Models with 100 thousand neurons and 21 millions of synapses have been successfully simulated.

- It allows for high flexibility in network structure definition (neurons, their types and density and synaptic connection with given weights and delays) and high flexibility in definition of inputs.
- High amount of detail of synaptic dynamics is achieved by implementing models of spike-timing dependent plasticity and spontaneous activity in a form of spontaneous synaptic release
- The simulator is able to batch process multiple experiments aiding in the experiment design.

As such, the simulator satisfies all requirements posed above. Moreover, its capabilities allow for large scale, physiologically faithful simulations of mammalian cortex as demonstrated in (Popelová, 2013). Last but not least, we were able to closely cooperate with its creators; this greatly facilitated its adaptation for our intended use.

6.3 Revising the simulator

Upon completion in 2013, the simulator was suitable for faithful simulations developing advanced network properties such as tonotopy of the auditory cortex (Popelová, 2013). However, due to time constraints, certain features were missing in the simulator, notably a mechanism for simulating short term plasticity. In addition, certain neuronal types showed non physiological firing frequency. Following are the applied methods addressing presented issues.

6.3.1 Modeling of short term plasticity

To improve the realism of simulated network activity, we have decided to implement a mechanism for short-term plasticity, which was missing in the original version of simulator. When deciding on the possible simulation method, the overall focus of the simulator had to be considered. The simulator was designed to allow for efficient stimulations of large networks. Thus the implemented mechanism needed to be simple enough not to worsen its computation performance.

We have decided to implement the simplified model of short term plasticity as presented in section 3.2.3. We have used following parameters published in (Izhikevich and Edelman, 2008) to differentiate the effect of short-term plasticity in different types of neurons. Synapses of neuronal types not in the table are assumed not to exhibit short term plasticity. The implementation parameters are in Table 6.1.

Table 6.1: Parameters used in the implementation of short-time plasticity based on (Izhikevich and Edelman, 2008).

neuronal types (to) (from)	p,ss (RS)	b (FS)	nb (LTS)
p,ss (RS)	$\tau_x=150, p=0.6$	$\tau_x=150, p=0.6$	$\tau_x=100, p=1.5$
b (FS)	$\tau_x=150, p=0.6$	$\tau_x=150, p=0.6$	no plasticity

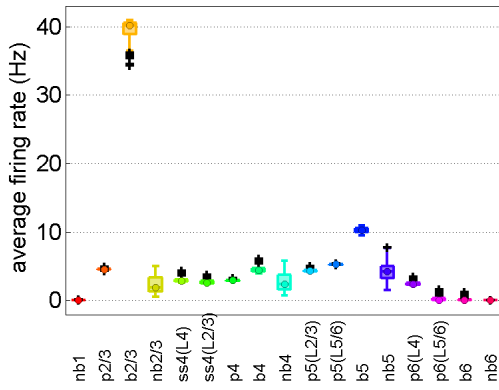
6.3.2 Update of model connectivity

The SUSNOIMAC simulator is suitable for modeling of the primary auditory cortex of mammalian brain. One of the greatest challenges when creating realistic models of real networks is the acquisition of structure data. The types of neurons in the network, their location in the network structure and synaptic connections must be known. The research of these network parameters is far from complete, sometimes with conflicting results.

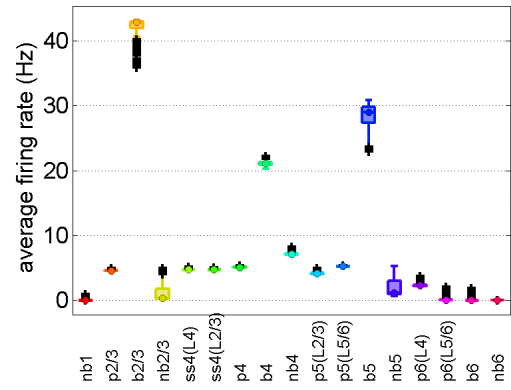
The precise parameters used in the SUSNOIMAC simulation of the primary auditory cortex are specified in (Popelová, 2013, pp. 103-107). Resulting network activity was observed and compared to data from real networks (Popelová, 2013, Chapter 8). Model showed generally very plausible behavior developing many properties observed in real networks. Notable exceptions were the firing rates of neuronal types b2/3 and b5 (banded neurons in layers 2/3 and 5). Typical firing frequencies in the simulated model were 40-45 Hz for b2/3 neurons and 15-30 Hz for b5 neurons. By contrast, real network observations report values of 3.5-5 Hz for b2/3 neurons and 6-8 Hz for b5 neurons (Li et al., 2014; Sakata and Harris, 2012). Although comparing absolute values of firing rates can be deceiving, reasons to suspect the simulation values are twofold. Firstly, the b2/3 neurons fire more frequently than b5 neurons in the model, while it is contrariwise in both studies. Secondly, all other neuronal types in the simulated model show plausible firing rates in the range of 0.5-20 Hz agreeing with observations in (Sakata and Harris, 2012). To resolve the non-physiological firing rates of certain types of neurons (most noticeably b2/3), the connectome was revised in cooperation with Institute of Experimental Medicine, Academy of Science of, Czech republic. The original and updated connectivity data used in the revised model can be found in the electronic attachment. The change in the firing rate of different neurons is documented in the next section.

6.4 Results

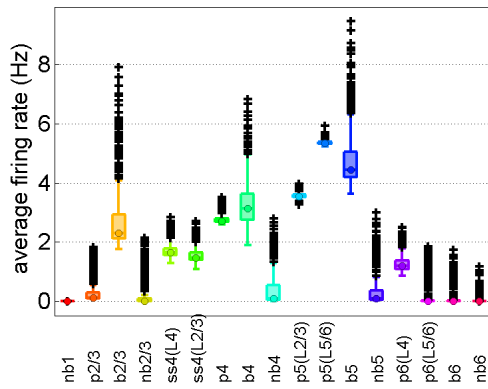
The network activity was observed both without and with input noise. 50000 neurons were simulated. Only spontaneous activity was present in the network without input noise. Simulated model time was 10 minutes in each experiment. Further details and simulation properties setup was as described in (Popelová, 2013, chapter 7). Activity in model without revisions (Figures 6.1a,b), with revised connectome (Figures 6.1c,d) and with both revised connectome and implemented short term plasticity was recorded (Figures 6.1e,f). Significant decrease in firing rate of b2/3 and b5 neurons was observed in the simulations using revised connectome (Figures 6.1c,d,e,f) compared to simulations using original connectome (Figures 6.1a,b). The mean firing rate of b2/3 decreased from 40 Hz to around 3 Hz in the simulation without noise input and from 45 Hz to 5 Hz in the simulation with noise. Similarly, the mean firing rate of b5 decreased from 10 Hz to around 5.5 Hz in the simulation without noise input and from 30 Hz to 8 Hz in the simulation with noise. Addition of the short-term plasticity had further, but smaller effect on firing rate in the network. Notably, it led to further decrease in firing rates of b2/3 and b/5 neurons in the simulation with input noise, from 5 to 3 Hz and from 8 Hz to 4 Hz respectively (See figures 6.1d,f).



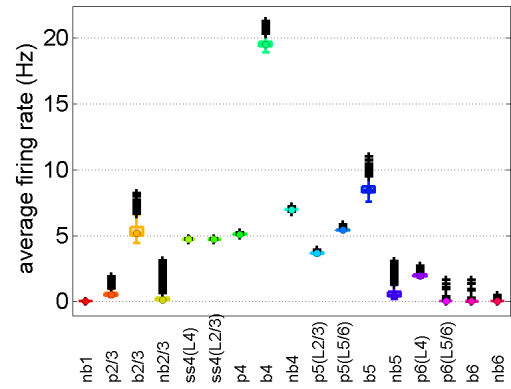
(a) No revision, no noise



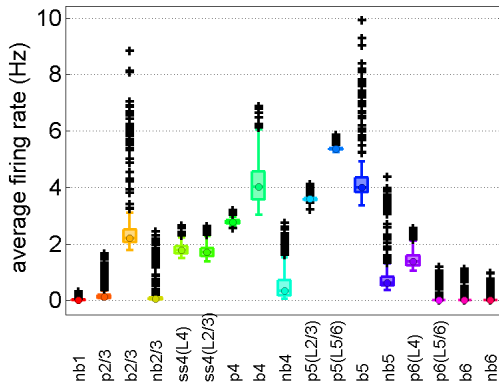
(b) No revision, input noise



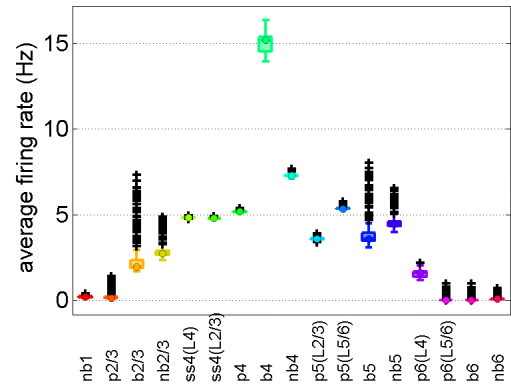
(c) Revised connectome only, no noise



(d) Revised connectome only, input noise



(e) Revised connectome + plasticity, no noise



(f) Revised connectome + plasticity, input noise

Figure 6.1: A comparison of the mean network activity between networks without revision (a, b), with revised connectome (c, d) and networks with model implementing short-term plasticity and revised connectome (e, f). First column (a, c, e) displays simulations without input noise, second column (b, d, f) with input noise.

6.5 Discussion

The simulation results suggest that firing rates observed in the revised model improved both absolutely and relatively when using the revised connectome, i.e. firing rates of both concerned neuronal types are now close to values observed in real networks (Lindsey et al., 1997) and type b5 neurons now appear more active than type b2/3 neurons (Sakata and Harris, 2012). Other neuronal types showed comparatively minor changes in their firing rates as stayed within their expected ranges from (Sakata and Harris, 2012).

The addition of mechanism for short term plasticity had further minor but positive effect on firing rates in the network. Its effect was noticeable in the simulations with input noise where it further reduced firing rates of concerned neuronal types closer to expected values. More importantly, the implemented mechanism increases the realism of simulated model.

Both revisions improved behavior of the network activity in the simulated model. This was important for conducting plausible experiments on behavior in polychronous groups, since the emerging polychronous groups depend closely on the network structure and dynamics.

7. Analysis of polychronous groups in spiking neural networks

Before any analysis of emergence and behavior of polychronous groups in neural networks was possible, a compilation of published research was needed to establish basic knowledge of the subject. Being a relatively new concept, not many results have been published so far, however this situation changes by the year. In this chapter, we present in detail the most influential results published so far. We remark on the the lack of research of behavior of polychronous groups in noisy networks, i.e networks where spontaneous synaptic activity occurs, despite it being important mechanism of formation real neural networks. Finally we conduct an experiment exploring emergence of polychronous groups in spontaneous activity in network subjected to noisy inputs. The results are presented and discussed.

7.1 Related works

The article (Maier and Miller, 2008) provides an overview of various characteristics of polychronous groups emerging in a minimal model of spiking neural network able to exhibit such groups. The network model obviously plays a major role in the results of the experiments and in this case it had following properties:

- Integrate and fire neuronal model
- Varied size, up to 1500 neurons
- Variable connectome with up to 30000 synapses
- Synapse delays based on distance between neurons
- Long term plasticity using STDP

Following figures are due to the authors of the experiments. The structure of synaptic delays in a spiking network was examined for supported polychronous groups using scanning method similar to the one presented in chapter 4. Figures 7.1 and 7.2 summarize the idea that the number of supported groups in a network grows as the size of the network increases. The data suggest that the relation seems to be linear in case of increase of the number of neurons. As for increasing the number of synapses, the rate of increase is even faster suggesting possible exponential growth. These findings are consistent with those of (Izhikevich, 2006).

The final figure 7.3 compares the number of groups present in network where synapse delays are chosen randomly and those where the neurons are connected deterministically meaning the delays reflect the distances between neurons. The data suggest that that the number of polychronous groups was much higher for the network with the proportionally chosen delays.

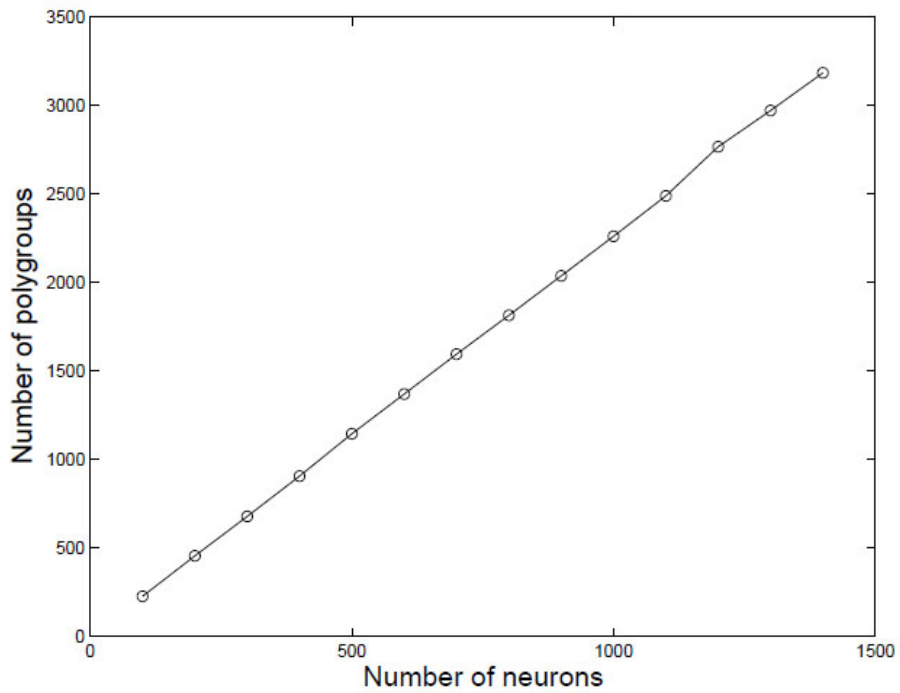


Figure 7.1: A plot showing the number of polychronous groups as a function of N , the number of neurons in the system (Maier and Miller, 2008).

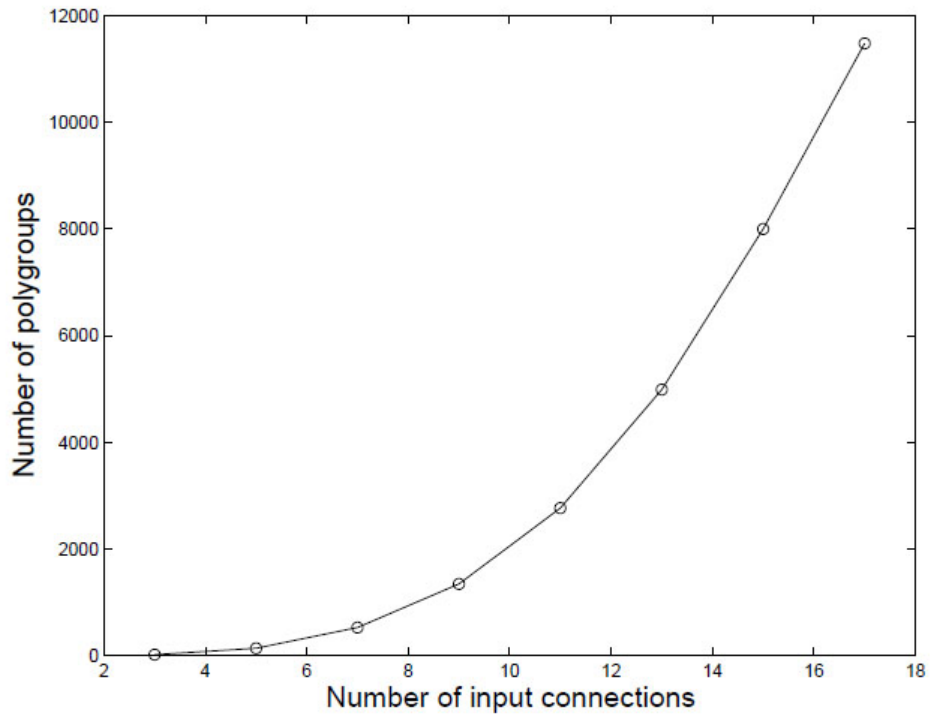


Figure 7.2: The number of polychronous groups as a function of m , the number of input connections to each neuron in the network (Maier and Miller, 2008).

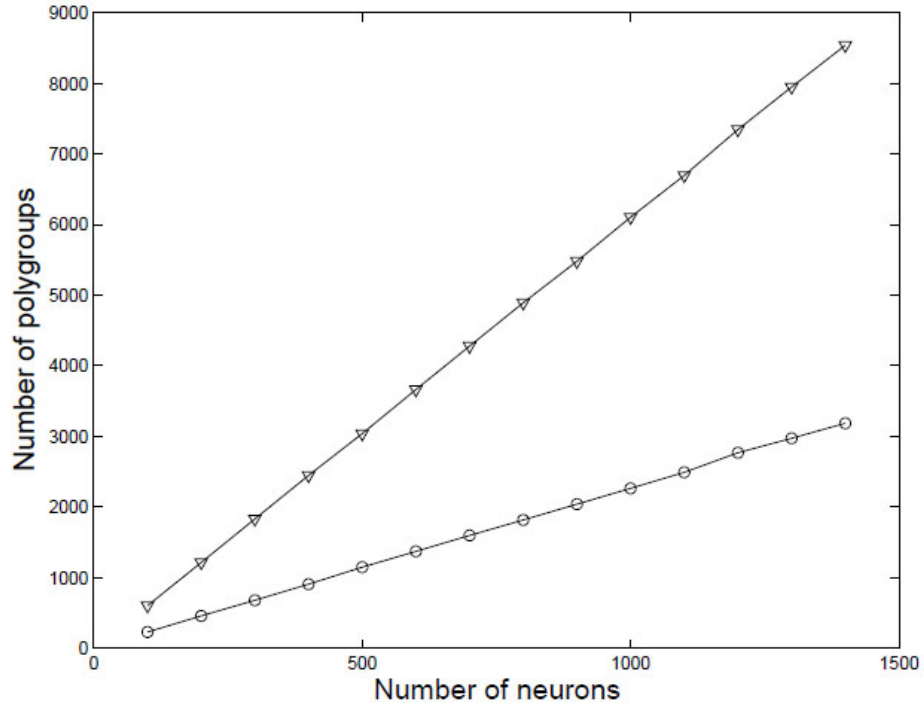


Figure 7.3: The number of polychronous groups as a function of N , the number of neurons in the system, for networks with random vs. deterministic delays. The random network graph is marked by small circles, the deterministic graph by inverted triangles (Maier and Miller, 2008)

Another research article (Chrol-Cannon et al., 2012) set to explore the emergence of polychronous groups under varying input patterns. The simulation setup was following:

- Izhikevich neuron model
- 1000 neurons
- 0.1 connection probability (100000 synapses)
- Synapse delay in $\{0, 20\}$ ms for excitatory, in $\{0, 1\}$ ms for inhibitory neurons
- Long term plasticity using STDP

The experiments were carried out with varying number of input patterns presented to the network (1, 2, 4, 8 inputs). Each input pattern consisted of 100 spike trains fed to 100 pre-selected input neurons chosen at the beginning of the simulation. Each input pattern lasted a whole second before switching to the next one. After such stimulation, the network structure was examined for supported polychronous groups using scanning algorithm. Authors compared experiments where 1, 2, 4 or 8 patterns were repeatedly alternated throughout the duration of the simulation. The results are summarized in Figure 7.4.

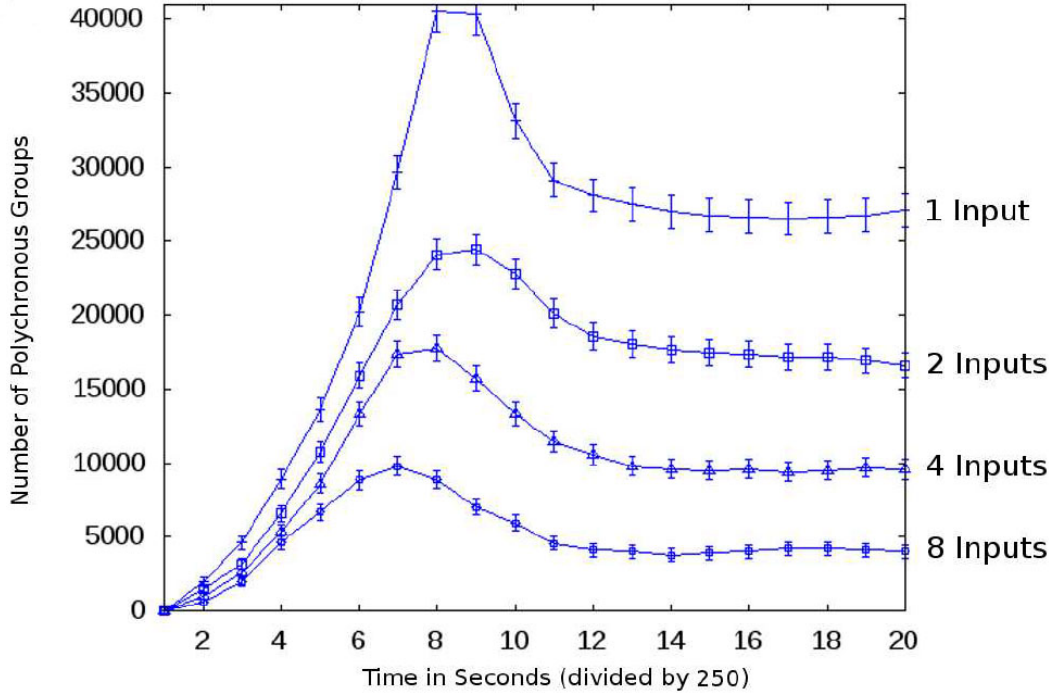


Figure 7.4: Emergence of groups over a 5,000 second time period while being stimulated by; 1, 2, 4 or 8 alternating patterns. Snapshots of numbers of groups within the network are taken at 250 second intervals (Chrol-Cannon et al., 2012).

The main finding was that the number of supported polychronous groups seems to decrease with growing number on presented input patterns. It was hypothesized that this development might be due to the competition of input patterns in the network structure.

7.2 Polychronous groups emerging in spontaneous activity

So far, the behavior of polychronous groups in noisy networks developing spontaneous activity is not documented. We believe that this is due to the lack of methods are capable of detecting polychronous group even in the presence of noise. As mentioned earlier, spontaneous activity is an important element in development and function of the network and models missing mechanisms of its development lack the biological plausibility of more realistic models like the SUS-NOIMAC simulator described in previous chapter. Our graph search method detection is, to the best of our knowledge, the only method capable of analyzing data from such advanced simulators.

We decided to evaluate the polychronous groups present in spontaneous activity in a network. So far, we are not aware of any research analyzing polychronous groups emerging in spontaneous activity. The published research so far, presented in the previous section, helped us form the basic assumption about the behavior of polychronous group in such conditions. We hypothesized that spontaneous activity can be viewed of as a self imposed input to the network structure repeatedly occurring with intensity and frequency oscillating around a baseline given by

parameters of minis. We assumed that the greater the number of supported PGs in a network, the more polychronous groups would get activated by random input such as spontaneous activity and thus present in the recorded data. We aimed to explore the development of polychronous groups present in network subjected to noise input of varying intensities to observe potential differences in network structure.

In detail, we hypothesized that:

- Noise would behave as large number of input stimuli presented to network inputs
- The number of supported PGs would drop gradually during simulation (as per (Chrol-Cannon et al., 2012))
- As a result, the number of activated PGs during spontaneous activity would drop as well
- Input noise intensity could play a role in the rate at which PG count diminishes

7.2.1 Experiment method

We used the revised SUSNOIMAC simulator to simulate a network consisting of 5000 neurons with topology found in primary auditory cortex. Detailed description of network model can be found in (Popelová, 2013, Chapter 7). Experiment was conducted in 2 phases. First phase was a general adaptation of network to the presented noisy input that was carried out in a simulation of 10 minutes of model time. All throughout this period, the synaptic weights were allowed to adapt using spike-timing-dependent plasticity. Changing values of synaptic weights were saved every minute to allow for examination of network state.

Second phase consisted of simulating 10 seconds of emerging spontaneous activity using saved network states from previous phase. We call these alternate reality experiments since they allow for a different network development while beginning at exactly the same state as the saved network snapshot. This time the STDP was disabled so that the activity better represents the saved network and not a new development. The overall experiment structure is summarized in figure 7.5.

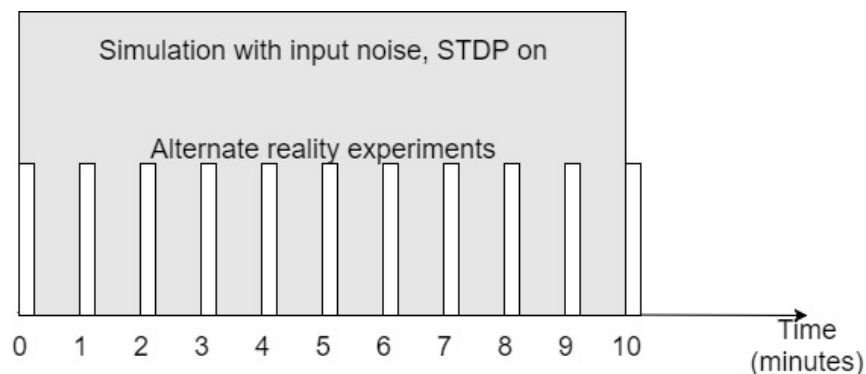


Figure 7.5: The structure of conducted experiment. Snapshots of the adapting network were taken every minute for later simulation of spontaneous activity.

When a SUSNOIMAC simulation starts, there is an erroneous behavior of neurons caused by the initial development of membrane potential of neurons. The simulation settles quickly and normal network characteristics are established after 2 seconds of simulation. To eliminate this boundary effect of simulation start, the first 5 second of recorded spontaneous activity is discarded and only the remaining 5 seconds are used for the analysis to avoid possible misleading data, see figure 7.6.

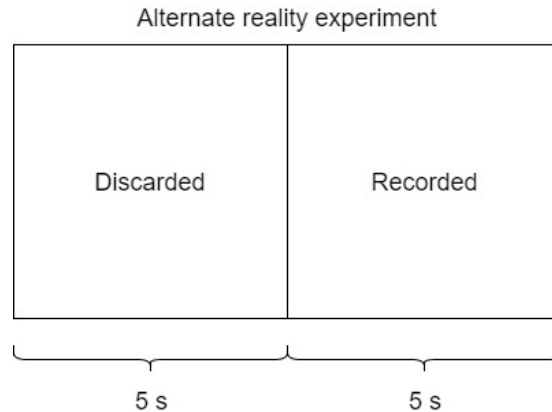


Figure 7.6: The first 5 second of recorded spontaneous activity from the alternate reality experiment is discarded to reduce boundary effect of simulation start.

Noise levels of 0.0025, 0.01 and 0.5 were tried in separate simulations (representing inputs in the form of minis, weak noise and strong noise respectively). Each simulation was repeated 10 times for each level of noise and observed values were averaged respectively.

The numbers of activated polychronous groups in the simulations were detected using the graph search method of detection. For further analysis, we have decided to observe the number of groups that occur multiple times during the simulation suggesting their strong support in the network structure. As the graph search method may miss activation of some groups, we needed to combine our method with the algorithm of Martinez. Our graph search method was used to provide approximation of the database of polychronous groups present in the network. Such database would be unobtainable using existing scanning methods. The method of Martinez (without the initial scanning phase) was then used to precisely count the number of occurrences of activated polychronous groups.

7.2.2 Results

A significant drop in the amount of PGs present in spontaneous activity was detected by the algorithm (see figure 7.7). More gradual descent was expected, however recorded data are feasible and seem to agree with existing experiments (Chrol-Cannon et al., 2012). Faster descent rate was expected for stronger noise levels, but no difference was detected; all experiments reach the reduced values of activated polychronous groups by the end of the first minute (figure 7.7).

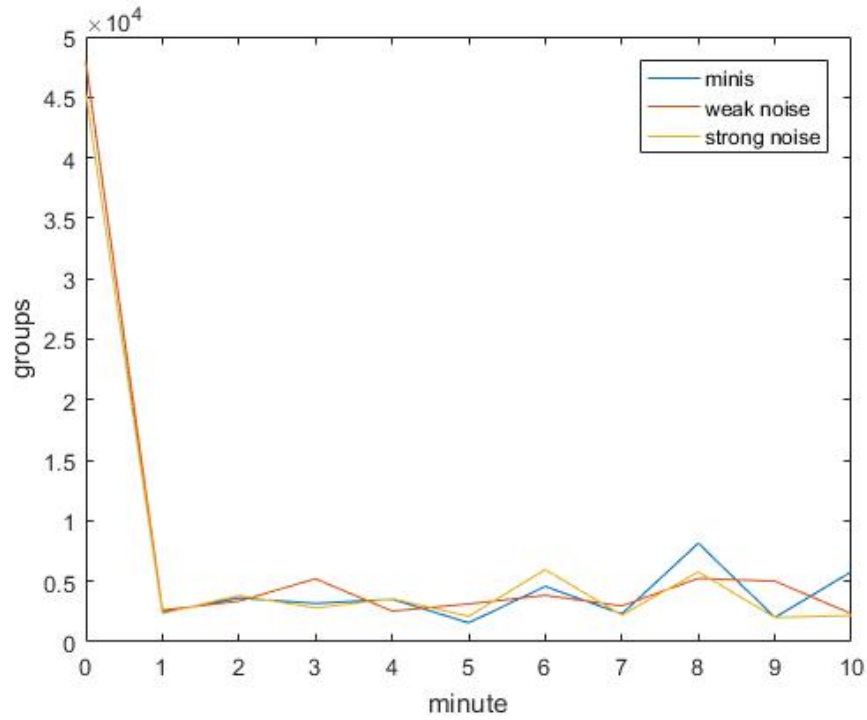


Figure 7.7: The numbers of detected activated polychronous groups during 5s of spontaneous activity during the alternate reality experiments. The x-axis denotes the time of the snapshot of adapting network defining the synaptic weights used in alternate reality experiment.

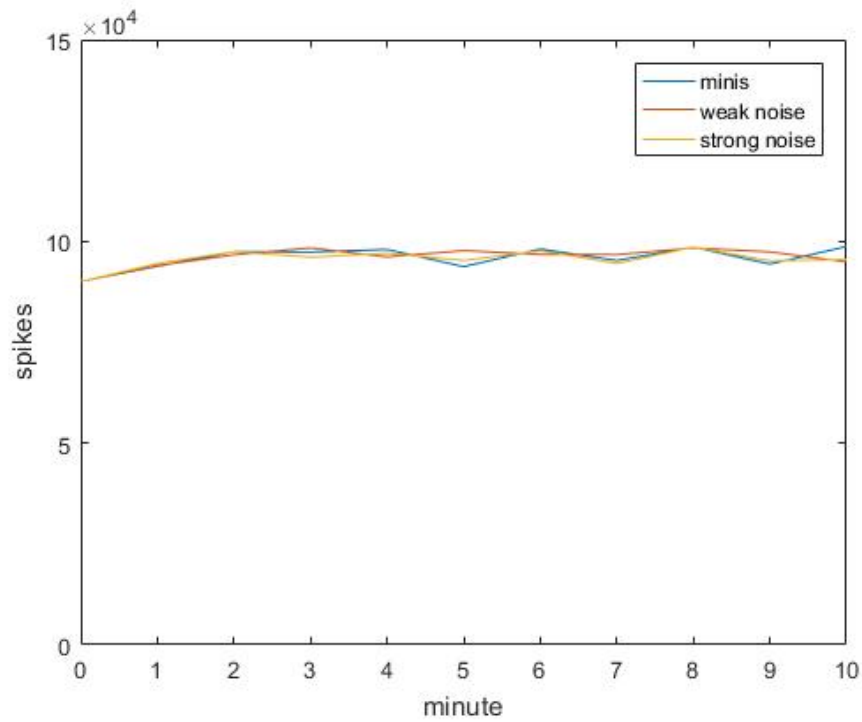


Figure 7.8: The numbers of spikes recorded during 5s of spontaneous activity. The x-axis denotes the time of the snapshot of adapting network defining the synaptic weights used in alternate reality experiment.

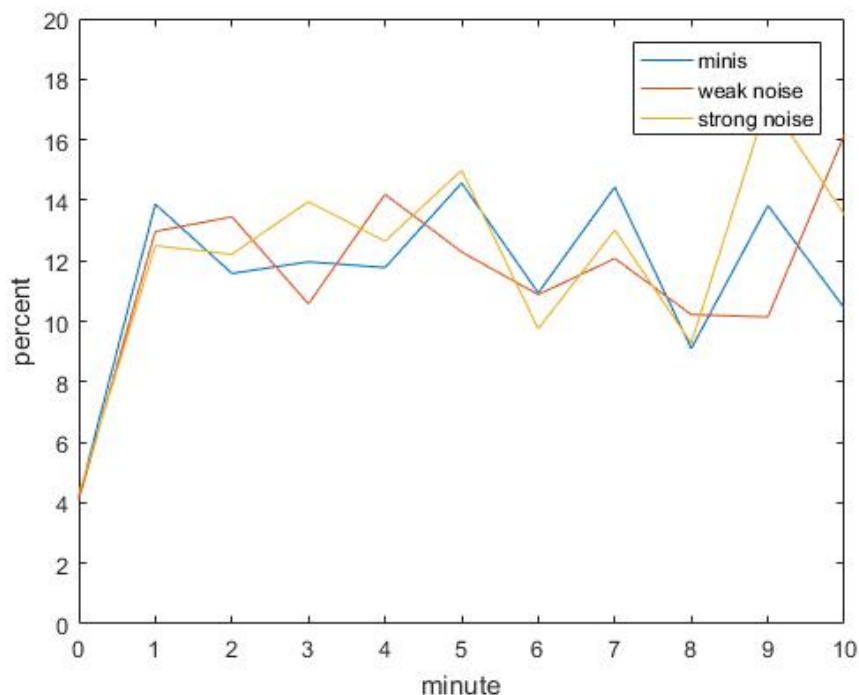


Figure 7.9: The percentages of recurring polychronous groups during 5s of spontaneous activity. The x-axis denotes the time of the snapshot of adapting network defining the synaptic weights used in alternate reality experiment.

Similar number of spikes were recorded in the 10s alternate reality experiments for the initial weights as for the subsequent minutes (roughly 95000 spikes, see figure 7.8) and similar number of spikes were discarded from the first five seconds of every simulation. The difference in the number of activated PGs thus indeed seems to be the result of STDP adaptation to the noisy input and not a side effect of change in overall network activity.

The data suggest, that while the number of activated polychronous groups in the adapted network sharply decreases, those that are present actually reappear with higher probability during a period of time in the adapted network. The percentage of polychronous groups that occurred more than once during analyzed period went from 5% to around 14% (see figure 7.8).

7.2.3 Discussion

Despite the exposition to a noisy input, the network structure seems to be developing strong support for certain polychronous groups. This is supported by the increased reappearance of polychronous in spontaneous activity in adapted network compared to the initial state with randomized weights. This occurred, even when adapting without additional input (inputs were minis) supporting existence of the self-organizing mechanisms of spontaneous activity in network.

Our data seem to support the hypothesis that input noise behaves similarly to a large number of input stimuli regarding adaptation of polychronous groups. The study (Chrol-Cannon et al., 2012) observed decrease of supported polychronous groups when increasing the number of inputs presented to the network. Similarly in our experiment, a sharp decrease in activated polychronous groups was

observed between the adapted (to a large number of stimuli) and randomized networks (no stimuli yet presented for adaptation). Interestingly, no significant difference was found between different intensities of input noise. This suggests that it is the number of input stimuli, rather than their strength, that has greater effect on adaptation of polychronous groups.

Another possible explanation of the similarities between varying levels of input noise could be that even self-organization resulting from spontaneous activity is enough to saturate the supporting capacity of the network. Polychronous groups have to compete for network support and if there exists a maximum support capacity of a network, stronger input might just replace the existing groups faster while keeping the overall number the similar.

We speculate that besides the self-organization mechanism of spontaneous release of neurotransmitter, its activity can actually have positive effect on capacity of support in the neural network. By introducing weak inputs to the network, the supported polychronous groups get weaker thus potentially allowing for additional supported groups in the network. However, our experiment provides only indirect support and further examination would be necessary.

8. Conclusion

In this thesis, we have presented a new method for polychronous group detection based on the concept of spike dependency graph in the recorded activity of a neural network. Our approach avoided the universally employed extensive scanning for polychronous groups resulting in several improvements over existing methods. Most notably, it reduced the computation complexity allowing the use with large neural network simulations. It also allows for detection of activated polychronous groups in noisy networks, which is a feature missing in the existing methods. Finally, comparatively less detailed description of the neural network dynamics is required further facilitating its use with various neural networks. The trade-off for these improvements is the reduced sensitivity of detection.

Using the presented detection method, we were able to conduct experiments exploring emergence of polychronous groups in spontaneous activity. We have shown that input noise leads to reduced overall activation of polychronous groups suggesting saturation of the supporting capacity of a network. While the overall number of activated groups decreased, groups that remained in the recorded activity appeared with higher probability hinting a development of network support for certain groups. Our findings seem to support the claim that noise and spontaneous activity contribute to self-organizing behavior of neural networks.

To conduct the experiments, an advanced simulation model of auditory cortex (SUSNOIMAC) was adapted. We were able to revise the simulated network structure and enrich the simulated synaptic dynamics by implementing short-term plasticity. The revised simulation model exhibits more realistic firing rates of basked interneurons.

As discussed earlier, the single jitter parameter is not optimal when modeling spike dependency. More complex criteria of spike dependency could be developed in the future to improve the detection performance of proposed method. More experiments examining emergence of polychronous groups can be conducted, especially with polychronous groups representing various sensory stimuli.

Bibliography

- ABELES, Moshe. *Local cortical circuits*. Springer-Verlag Berlin Heidelberg, 1982. ISBN 978-3-642-81708-3.
- ABELES, Moshe. *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press, 1991. ISBN 0521376173.
- ABELES, Moshe, GAT, I. Detecting precise firing sequences in experimental data. *Journal of neuroscience methods*, **107**(1):141–154, 2001. ISSN 0165-0270.
- ADRIAN, Edgar D. The impulses produced by sensory nerve endings. *The Journal of physiology*, **61**(1):49–72, 1926. ISSN 1469-7793.
- ANDREAE, Laura C, BURRONE, Juan. Spontaneous neurotransmitter release shapes dendritic arbors via long-range activation of nmda receptors. *Cell reports*, **10**(6):873–882, 2015. ISSN 2211-1247.
- BEGGS, John M, PLENZ, Dietmar. Neuronal avalanches are diverse and precise activity patterns that are stable for many hours in cortical slice cultures. *Journal of neuroscience*, **24**(22):5216–5229, 2004. ISSN 0270-6474.
- BIENENSTOCK, Elie. A model of neocortex. *Network: Computation in neural systems*, **6**(2):179–224, 1995. ISSN 0954-898X.
- CHA, Albert, SNYDER, Gregory E, SELVIN, Paul R, BEZANILLA, Francisco. Atomic scale movement of the voltage-sensing region in a potassium channel measured via spectroscopy. *Nature*, **402**(6763):809–813, 1999. ISSN 0028-0836.
- CHANG, EY, MORRIS, KF, SHANNON, R, LINDSEY, BG. Repeated sequences of interspike intervals in baroresponsive respiratory related neuronal assemblies of the cat brain stem. *Journal of neurophysiology*, **84**(3):1136–1148, 2000. ISSN 0022-3077.
- CHROL-CANNON, Joseph, GRÜNING, André, JIN, Yaochu. The emergence of polychronous groups under varying input patterns, plasticity rules and network connectivities. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2012. ISBN 1467314900.
- CHUN-HUA, Yuan, JIANG, Wang, BIN, Deng. Effects of chemical synapse type and time delay on vibrational resonance in coupled neurons. In *Control Conference (CCC), 2012 31st Chinese*, pages 7453–7456. IEEE. ISBN 988156381X.
- CITRI, Ami, MALENKA, Robert C. Synaptic plasticity: multiple forms, functions, and mechanisms. *Neuropsychopharmacology*, **33**(1):18–41, 2008. ISSN 0893-133X.
- CONTRERAS, Diego. Electrophysiological classes of neocortical neurons. *Neural Networks*, **17**(5):633–646, 2004. ISSN 0893-6080.
- D’AMOUR, James A, FROEMKE, Robert C. Inhibitory and excitatory spike-timing-dependent plasticity in the auditory cortex. *Neuron*, **86**(2):514–528, 2015. ISSN 0896-6273.

- DESCHENES, Michael. Short review: Rate coding motor unit recruitment patterns. *The Journal of Strength and Conditioning Research*, **3**(2):34–39, 1989. ISSN 1064-8011.
- DIESMANN, Markus, GEWALTIG, Marc-Oliver, AERTSEN, Ad. Stable propagation of synchronous spiking in cortical neural networks. *Nature*, **402**(6761):529, 1999. ISSN 0028-0836.
- DOYLE, Declan A, CABRAL, Joao Morais, PFUETZNER, Richard A, KUO, Anling, GULBIS, Jacqueline M, COHEN, Steven L, CHAIT, Brian T, MACK-INNON, Roderick. The structure of the potassium channel: molecular basis of k⁺ conduction and selectivity. *science*, **280**(5360):69–77, 1998. ISSN 0036-8075.
- ENOKA, Roger M, DUCHATEAU, Jacques. Rate coding and the control of muscle force. *Cold Spring Harbor Perspectives in Medicine*, page a029702, 2017. ISSN 2157-1422.
- EVARTS, Edward V. Relation of discharge frequency to conduction velocity in pyramidal tract neurons. *Journal of neurophysiology*, **28**(2):216–228, 1965. ISSN 0022-3077.
- FUGLEVAND, Andrew J, SEGAL, Steven S. Simulation of motor unit recruitment and microvascular unit perfusion: spatial considerations. *Journal of Applied Physiology*, **83**(4):1223–1234, 1997. ISSN 8750-7587.
- FUGLEVAND, ANDREW J, WINTER, DAVID A, PATLA, AFTAB E. Models of recruitment and rate coding organization in motor-unit pools. *Journal of neurophysiology*, **70**(6):2470–2488, 1993. ISSN 0022-3077.
- GUISE, Mira, KNOTT, Alistair, BENUSKOVA, Lubica. Enhanced polychronization in a spiking network with metaplasticity. *Frontiers in computational neuroscience*, **9**, 2015.
- HARTMANN, Christoph, LAZAR, Andreea, NESSLER, Bernhard, TRIESCH, Jochen. Where’s the noise? key features of spontaneous activity and neural variability arise through learning in a deterministic network. *PLoS computational biology*, **11**(12):e1004640, 2015. ISSN 1553-7358.
- HERCULANO-HOUZEL, Suzana. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in human neuroscience*, **3**, 2009.
- HODGKIN, Alan L, HUXLEY, Andrew F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, **117**(4):500–544, 1952. ISSN 1469-7793.
- IKEGAYA, Yuji, AARON, Gloster, COSSART, Rosa, ARONOV, Dmitriy, LAMPL, Ilan, FERSTER, David, YUSTE, Rafael. Synfire chains and cortical songs: temporal modules of cortical activity. *Science*, **304**(5670):559–564, 2004. ISSN 0036-8075.
- IZHIKEVICH, Eugene M. Polychronization: computation with spikes. *Neural computation*, **18**(2):245–282, 2006.

- IZHIKEVICH, Eugene M. Solving the distal reward problem through linkage of stdp and dopamine signaling. *Cerebral cortex*, **17**(10):2443–2452, 2007. ISSN 1460-2199.
- IZHIKEVICH, Eugene M, EDELMAN, Gerald M. Large-scale model of mammalian thalamocortical systems. *Proceedings of the national academy of sciences*, **105**(9):3593–3598, 2008. ISSN 0027-8424.
- LI, Ling-yun, XIONG, Xiaorui R, IBRAHIM, Leena A, YUAN, Wei, TAO, Huizhong W, ZHANG, Li I. Differential receptive field properties of parvalbumin and somatostatin inhibitory neurons in mouse auditory cortex. *Cerebral cortex*, **25**(7):1782–1791, 2014. ISSN 1460-2199.
- LINDSEY, Bruce G, MORRIS, Kendall F, SHANNON, Roger, GERSTEIN, GL. Repeated patterns of distributed synchrony in neuronal assemblies. *Journal of Neurophysiology*, **78**(3):1714–1719, 1997. ISSN 0022-3077.
- MAIER, Willard L, MILLER, Bruce N. A minimal model for the study of polychronous groups. *arXiv preprint arXiv:0806.1070*, 2008.
- MAO, Bu-Qing, HAMZEI-SICHANI, Farid, ARONOV, Dmitriy, FROEMKE, Robert C, YUSTE, Rafael. Dynamics of spontaneous activity in neocortical slices. *Neuron*, **32**(5):883–898, 2001. ISSN 0896-6273.
- MARKRAM, Henry, WANG, Yun, TSODYKS, Misha. Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences*, **95**(9):5323–5328, 1998. ISSN 0027-8424.
- MARTINEZ, Regis, PAUGAM-MOISY, Helene. Algorithms for structural and dynamical polychronous groups detection. *Artificial Neural Networks ICANN 2009*, pages 75–84, 2009.
- PHOKA, Elena, WILDIE, Mark, SCHULTZ, Simon R, BARAHONA, Mauricio. Sensory experience modifies spontaneous state dynamics in a large-scale barrel cortical model. *Journal of computational neuroscience*, **33**(2):323–339, 2012. ISSN 0929-5313.
- POPELOVÁ, Markéta. Software tool for modelling coding and processing of information in auditory cortex of mice. Master’s thesis, 2013.
- PRUT, Yifat, VAADIA, Eilon, BERGMAN, Hagai, HAALMAN, Iris, SLOVIN, Hamutal, ABELES, Moshe. Spatiotemporal structure of cortical activity: properties and behavioral relevance. *Journal of neurophysiology*, **79**(6):2857–2874, 1998. ISSN 0022-3077.
- RIEHLE, Alexa, GR N, Sonja, DIESMANN, Markus, AERTSEN, Ad. Spike synchronization and rate modulation differentially involved in motor cortical function. *Science*, **278**(5345):1950–1953, 1997. ISSN 0036-8075.
- SAKATA, Shuzo, HARRIS, Kenneth D. Laminar-dependent effects of cortical state on auditory cortical spontaneous activity. *Frontiers in neural circuits*, **6**, 2012.

- SCHUTTER, Erik De. *Computational modeling methods for neuroscientists*. The MIT Press, 2009. ISBN 0262013274.
- SHADLEN, Michael N, NEWSOME, William T. The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *Journal of neuroscience*, **18**(10):3870–3896, 1998. ISSN 0270-6474.
- SONG, Sen, MILLER, Kenneth D, ABBOTT, Larry F. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, **3**(9):919–926, 2000.
- SUBKHANKULOVA, Tatiana, YANO, Kojiro, ROBINSON, Hugh PC, LIVESEY, Frederick J. Grouping and classifying electrophysiologically-defined classes of neocortical neurons by single cell, whole-genome expression profiling. *Frontiers in molecular neuroscience*, **3**, 2010.
- SUN, Haoqi, YANG, Yan, SOURINA, Olga, HUANG, Guang-Bin. Runtime detection of activated polychronous neuronal group towards its spatiotemporal analysis. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2015. ISBN 1479919608.
- SWADLOW, Harvey A. Systematic variations in the conduction velocity of slowly conducting axons in the rabbit corpus callosum. *Experimental neurology*, **43**(2):445–451, 1974. ISSN 0014-4886.
- SWADLOW, HARVEY A. Physiological properties of individual cerebral axons studied in vivo for as long as one year. *Journal of Neurophysiology*, **54**(5):1346–1362, 1985. ISSN 0022-3077.
- SWADLOW, HARVEY A. Efferent neurons and suspected interneurons in binocular visual cortex of the awake rabbit: receptive fields and binocular properties. *Journal of neurophysiology*, **59**(4):1162–1187, 1988. ISSN 0022-3077.
- SWADLOW, Harvey A. Efferent neurons and suspected interneurons in s-1 vibrissa cortex of the awake rabbit: receptive fields and axonal properties. *Journal of Neurophysiology*, **62**(1):288–308, 1989. ISSN 0022-3077.
- TATENO, T, ROBINSON, HPC. Rate coding and spike-time variability in cortical neurons with two types of threshold dynamics. *Journal of neurophysiology*, **95**(4):2650–2663, 2006. ISSN 0022-3077.
- TETKO, Igor V, VILLA, Alessandro EP. A pattern grouping algorithm for analysis of spatiotemporal patterns in neuronal spike trains. 1. detection of repeated patterns. *Journal of Neuroscience Methods*, **105**(1):1–14, 2001. ISSN 0165-0270.
- TRAPANI, Josef G, NICOLSON, Teresa. Mechanism of spontaneous activity in afferent neurons of the zebrafish lateral-line organ. *Journal of Neuroscience*, **31**(5):1614–1623, 2011. ISSN 0270-6474.
- VILLA, Alessandro EP, TETKO, Igor V, HYLAND, Brian, NAJEM, Abdellatif. Spatiotemporal activity patterns of rat cortical neurons predict responses in

a conditioned task. *Proceedings of the National Academy of Sciences*, **96**(3): 1106–1111, 1999. ISSN 0027-8424.

List of Figures

2.1	Diagram of a neuron, from (wikipedia.org, 2007).	6
2.2	Diagram of action potential development, from (wikipedia.org, 2015).	7
2.3	Diagram of a synapse, from (wikipedia.org, 2006).	9
5.1	Example of a spike dependency graph.	28
5.2	Example of a detected activated polychronous group	29
5.3	Example of a possible activated group that cannot be reported	30
6.1	A comparison of the mean network activity between networks without revision (a, b), with revised connectome (c, d) and networks with model implementing short-term plasticity and revised connectome (e, f). First column (a, c, e) displays simulations without input noise, second column (b, d, f) with input noise.	43
7.1	A plot showing the number of polychronous groups as a function of N , the number of neurons in the system (Maier and Miller, 2008).	46
7.2	The number of polychronous groups as a function of m , the number of input connections to each neuron in the network (Maier and Miller, 2008).	46
7.3	The number of polychronous groups as a function of N , the number of neurons in the system, for networks with random vs. deterministic delays. The random network graph is marked by small circles, the deterministic graph by inverted triangles (Maier and Miller, 2008)	47
7.4	Emergence of groups over a 5,000 second time period while being stimulated by; 1, 2, 4 or 8 alternating patterns. Snapshots of numbers of groups within the network are taken at 250 second intervals (Chrol-Cannon et al., 2012).	48
7.5	The structure of conducted experiment. Snapshots of the adapting network were taken every minute for later simulation of spontaneous activity.	49
7.6	The first 5 second of recorded spontaneous activity from the alternate reality experiment is discarded to reduce boundary effect of simulation start.	50
7.7	The numbers of detected activated polychronous groups during 5s of spontaneous activity during the alternate reality experiments. The x-axis denotes the time of the snapshot of adapting network defining the synaptic weights used in alternate reality experiment.	51
7.8	The numbers of spikes recorded during 5s of spontaneous activity. The x-axis denotes the time of the snapshot of adapting network defining the synaptic weights used in alternate reality experiment.	51
7.9	The percentages of recurring polychronous groups during 5s of spontaneous activity. The x-axis denotes the time of the snapshot of adapting network defining the synaptic weights used in alternate reality experiment.	52

List of Tables

5.1	Comparison of presented methods for detection of polychronous groups.	38
6.1	Parameters used in the implementation of short-time plasticity based on (Izhikevich and Edelman, 2008).	41

Appendices

A. User documentation

Here we provide the user documentation for the implementation of our graph search method. Input and output parameters are defined. Interpretation of output is provided. The application of our method on data from SUSNOIMAC simulation is facilitated by provided loading scrip which is described at the end of the guide.

A.1 Environment version

The implementation was developed and tested in Matlab version 2016b.

A.2 Installing the implementation

The implementation of the graph search method can be found in the source folder of electronic attachment. This folder needs to be added to the Matlab path by following command:

```
>> addpath('path_to_implementation_folder');
```

A.3 Function call

The main function of the graph search detection method is contained in the file graphDetection.m and is called from the Matlab environment as:

```
>> PGs = graphDetection(spikes, excitatory, delayLists, ...  
                        weights, weightLimit, jitter, ...  
                        timeLimit, maxDelay, maxPath, ...  
                        minSize, maxSize);
```

A.4 Function parameters

A.4.1 Input parameters

The function requires 11 input parameters. Their order is fixed and they can be split into 3 categories:

Network data

Let us assume that a network has n neurons and s spikes have been recorded.

- spikes - list of spikes in the network (recorded activity) in the form of $s \times 2$ matrix, each row contains one spike, first column represents time of spike, second column denotes neuron number, list of spikes must be in ascending order by the time of spike

- excitatory - mask denoting excitatory status of neuron in the form of $1 \times n$ vector, field containing 1 represents inhibitory neuron, field containing 0 represents excitatory neuron
- delayLists - connectivity data in the form of $n \times 1$ cell array D, cell $D\{i\}$ contains a $m \times 2$ matrix D_i containing synaptic delays of neuron i to its postsynaptic connections, each row of D_i represents one synaptic connection, first column of D_i is the number of postsynaptic neuron, second column of D_i is the conduction delay
- weights - synaptic weights of the network simulation in the form of $n \times n$ matrix W , $W(i, j)$ denotes the weight from presynaptic neuron i to postsynaptic neuron j , $W(i, j) = 0$ if there is no connection between neurons

Graph construction parameters

- weightLimit - minimum weight of synapses able to form a connection in the spike dependency graph, serves to filter weak synapses, must be in range $[0, wMax]$, where $wMax$ is maximum weight allowed in the network
- jitter - jitter parameter of allowed concurrency of spikes

Graph decomposition parameters

- maxPath - length of path required to be present in the group graph for it to be reported
- minSize - minimum allowed size of spike trigger set
- maxSize - maximum allowed size of spike trigger set
- timeLimit - maximum allowed time difference between first and last spike belonging to the detected group, serves to cut long polychronous groups
- maxDelay - maximum allowed time difference between first and last spike of the trigger set of detected group, must be greater than the maximal conduction delay in the network

A.4.2 Output parameters

- PGs - detected activated polychronous groups

A.5 Interpreting the output

When the main function finishes its computation, it stores the detected in the output parameter **PGs**. Its content can be observed by typing the parameter name to the console. An example output might look like :

```
>> PGs

PGs =

    1x3 cell array

    Columns 1 through 3

    [3x2 double]    [4x2 double]    [4x2 double]
```

We can interpret the output as follows. The method detected 3 groups and reported them in cell array of size 1×3 . Each cell contains corresponding spike trigger sets. To obtain the spike trigger set of first detected group we query the console with command:

```
>> PGs{1}

ans =

         0         586
         4        3032
        58         412
```

The first column of the output matrix represents the firing pattern of the spike trigger set, the second column the trigger neurons. Each row represents one trigger spike. For example, the first row represents the spike of neuron 586 at a time 0.

A.6 Loading script

To facilitate loading data from the SUSNOIMAC simulator, a loading script was created. For full guide to the SUSNOIMAC simulator, see (Popelová, 2013, Chapter 3). For our purposes we need the following simulation files:

- *myname_firings.csv*
- *myname_neuronsStatistics.csv*
- *myname_synapsesStatistics.csv*
- *table1.csv*

The loading script is called `loadAll.m` and has to be placed in the same folder as the simulation files. Before running the loading script, the filename variable in the Matlab workspace needs to be defined:

```
>> filename = 'myname';
```

Finally, the loading script is called as:

```
>> loadAll;
```

The script loads the network parameters into Matlab workspace satisfying the input parameter definitions of the detection method.