UNIVERSITY OF PAVIA – IUSS SCHOOL FOR ADVANCED STUDIES PAVIA

Department of Brain and Behavioral Sciences (DBBS)
MSc in Psychology, Neuroscience and Human Sciences

Simple models to generate integer ratios between temporal intervals

Supervisors:

Prof. Laura Ferreri

Prof. Andrea Ravignani

PhD student Yannick Jadoul

Thesis written by

Chloé Marie Amélie Camille Coissac

512839

Academic year 2023-2024

Abstract

Rhythm links the structure of music with sounds in nature, reflecting common patterns across species. An intriguing commonality found across diverse musical traditions and species is the categorical distribution of rhythms. Such categories often conform to small integer ratios: intervals whose relative durations can be expressed as a fraction of two integers. This thesis explores the generation of rhythmic patterns which display integer ratios, using three models: a Markov-chain, a spiking neural network, and a model inspired by crickets' rhythmic stridulation. Each model uncovers key mechanisms behind the emergence of integer ratios in rhythmic behavior. The Markov chain model demonstrates how simple stochastic processes can generate these ratios and how the models' parameters control their presence and variation. The spiking neural network model shows that a great variety of rhythmic patterns related to integer ratios can emerge from intrinsic neural properties. Finally, the cricket model reveals how basic biological feedback loops can produce rhythmic categories and integer ratios. Our findings suggest that rhythmic patterns in music and animal vocalization may stem from shared, fundamental processes, providing benchmarks in the study of the evolution of rhythm and musicality.

Aknowledgment

First, I would like to express my sincere gratitude to my supervisor Andrea Ravignani, and co-supervisor Yannick Jadoul, for their guidance and benevolence throughout this journey. Their knowledge and experience have contributed both to the achievement of this master thesis and to my growth as an aspiring scientist. I am also extremely thankful to the other lab members who shared my time in Rome: Jelle Van Der Werff, Olga Metelko, and Teresa Raimondi. Their support, warmth, and collaboration have greatly enriched my experience.

My appreciation also goes to my family and friends spread out around this world. Each of you has been a source of support and inspiration paving the way to this achievement. I want you to know that I will always wake up with a whiff of nostalgia on a Saturday morning.

Index

Introduction

Music has a profound impact on human experience (Levitin, 2008). From the rhythmic beat that gets our feet tapping to the melodies that bring tears to our eyes, music holds a special place in our mind. We are left wondering: where did it all begin?

This thesis seeks to contribute to this ongoing exploration of the origins of music. Using models based on Markovian principles, the chirping behavior of an Amazonian cricket and the firing pattern of artificial neurons, we aim to get some insights into the underlying mechanisms of this behavior.

Uncovering the enigma of music is a subject that stirred scholars from diverse disciplines. In recent decades, there has been a growing interest in approaching this question from an evolutionary point of view. Psychologists have delved into human development. They adopted an ontogenetic lens to explore how newborns respond to music. This investigation has shed light on the genesis of musical cognition within humans (Háden et al., 2024; Trehub, 2003). Anthropologists, on the other hand, have dug into the history of *Homo sapiens*, seeking archaeological remnants of early musical expressions in our ancestral past (Conard et al., 2009; Killin, 2021). These two complementary approaches have offered insights into the origins of music. However, to fully understand this phenomenon, we should also consider looking beyond human-specific studies. Research on animal vocalizations and rhythmic behaviors can provide valuable clues about the evolution of music.

The study of rhythm and sound in animals is indeed relevant to the study of music due to the distinction between two notions: *Music* and *Musicality* (Honing et al., 2015). *Music* is a human production and presents important variations across cultures. Due to these important variations regarding its forms and meanings across societies, it doesn't reach consensus on a definition (Trehub et al., 2015). Nonetheless, in a broad perspective, it can be described as a social and cultural construct developed upon our faculties of musicality (Honing et al., 2015). *Musicality*, in contrast, can be defined as a set of traits constrained by our cognitive systems. This set of traits allow humans to produce and perceive music (Honing et al., 2015). These traits are not exclusively present in humans, but spread in various other species (Gregorio et al., 2021; Huang & Yin, 2023; Roeske et al., 2020). They have combine in a unique way in humans to develop our musicality (Fitch, 2006).

Progress in understanding the origins of music cognition could therefore benefit from identifying similar traits in other species (Bouwer et al., 2021; Kotz et al., 2018). It could provide insights into the evolutionary processes that led to human cognition. Species sharing these traits help us understand the circumstances of their development (Ravignani et al., 2019). When similar traits are found in related species, it often indicates analogous evolution. In contrast, traits shared with distant species suggest homologous evolution due to environmental constraints (Fuhrmann et al., 2014). This comparative approach can help us piece together the building blocks of human musicality (Honing et al., 2015).

When a trait has been identified as universal, through behavioral studies across several cultures and species, the following step is the construction of models reproducing this trait. Modelisation can provide informations on the mechanisms that constitute the trait and thus, inform us on the cognitive processes involved. In addition, the comparison between several models explaining the same trait, can provide a more detailed explanation of the mechanism. Finally, models enable the exploration of the behavior beyond real world constraints. This extended investigation allows for a deeper understanding of the factors contributing to this trait (Zandberg et al., 2021). Thus, modeling universal traits is an important step to uncover the origins of musical cognition.

This project aims to compare simple computational models for one musical features found across-culture and beyond humankind: temporal sequences with inter onset intervals (IOIs) in relation to integer ratios (Gregorio et al., 2021; Raimondi et al., 2023; Roeske et al., 2020). The literature proposes several models for generating musical rhythm with IOIs in relation to integer ratios. These models encompass approaches based on neural resonance theory (E. Large & Grondin, 2008) such as The Gradient Frequency Neural Networks (GrFNNs) (E. W. Large et al., 2023) and others based on musical knowledge, such as rhythmic syntactic trees (Rohrmeier, 2020). Most of the models are based on the assumption of the perception of a rhythmic hierarchy (e.g. beat and meter). However, it remains uncertain whether rhythmic hierarchies are perceived in non-human species. Consequently, it is unclear if these models can account for the creation of rhythmic patterns with intervals related to integer ratios beyond humans.
Moreover, despite many rhythm models in the literature, few studies compare their unique contributions to the comprehension of rhythmic behaviors.

We will, in consequence, investigate the following questions: How do simple models produce integer ratios? What do their variations reveal about universal biological mechanisms underlying rhythmic behaviors? Our hypothesis is that simple models can generate temporal sequences with inter-onset intervals related to integer ratios due to emergent properties of the models, without the need for perfect integer ratio values as inputs.

Chapter 1 starts by an overview of the important concepts to understand what are inter onset intervals related to integer ratios in rhythmic contexts. Then, it presents the main methodology used to analyze the presence of integer ratios in temporal sequences. Following this, the chapter reviews evidences of the presence of these ratios across musical cultures and diverse species. Finally it explores different hypotheses regarding the presence of these ratios from a cognitive and neural point of view.

Chapter 2 delves into the role of computational models in rhythm research, introducing existing models for generating and analyzing rhythms and identifies gaps in the current approaches. It also emphasizes the need for models that can account for universal rhythmic patterns beyond humankind, and the importance of comparing different models to gain deeper insights into rhythmic mechanisms.

Chapter 3 presents three rhythmic models that address gaps identified in chapter 2. These models are expected to provide simple and biologically grounded explanations of how IOIs related to integer ratios are produced across cultures and species.

Chapter 4 and 5 detail the methodology used to analyze the behavior of our three models of interest and report the results of these analyses.

Chapter 6 gives a summary of the results obtained through our different simulations and compares the behavior of the different models. Each result is linked with the hypothesis regarding the presence of integer ratios presented in chapter 1. Finally, the strengths and limits of this study are presented along with proposals for future research.

Chapter 1: Integer ratios background

1.1 Inter-onset interval and integer ratios in rhythmic contexts.

Temporal sequences capture the chronology of events. In other words, it represents the order in which events happen over time. Rhythmic sequences represent a specific type of temporal sequence. They refer to sequences of events that follow a structure pattern (London, 2004). For example, the ticking of a clock creates an isochronous rhythmic sequence, with each tick happening at regular intervals.

In temporal sequences, the interval between the onset of an event and the onset of the next event is called Inter Onset Interval (IOIs). Researchers analyze IOI patterns to characterize the structure of these sequences. IOIs can be analyzed in terms of their absolute duration (Horr & Di Luca, 2015). For instance, analyzing how IOIs between beats (i.e. the regular rhythmic pulses in music) lengthen in a musical piece that slows down can reveal the phrase-final lengthening principle (Todd, 1985). This principle describes how the final beat or note is extended to signal the end of a phrase.

In addition to the absolute duration of the IOIs, researchers often calculate the ratio between IOIs (Roeske et al., 2020; Rohrmeier, 2020). This analysis provides a way to investigate if similar underlying patterns might be present in different temporal sequences. For example, if we compare the temporal sequence of two metronomes with different paces, we will find that the two temporal sequences have different absolute durations of IOIs. However, if we examine the ratios of IOIs of the two temporal sequences, they will both exhibit the same isochronous underlying pattern (i.e., events that are equally distributed in time).

Rhythmic categories classify patterns based on the proportional relationships between IOIs. This helps researchers understand similarities and differences across sequences. In humans, these categories are often tied to beat and meter (Rohrmeier, 2020), which reflect top-down cognitive processes where individuals infer temporal regularities from acoustic signals. However, if we construct our classification only on the ratios produced between two subsequent IOIs (Roeske et al., 2020) we can avoid references to beat and meter and use this categorisation to identify the rhythmic patterns even in species that do not perceived beat and meter as humans do (Gregorio et al., 2021; Raimondi et al., 2023; Roeske et al., 2020).

Among the possible ratios, small integer ratios (e.g. 1:1, 1:2, 2:1, 1:3…) are frequently suggested to be perceptually preferred because of their inherent mathematical simplicity (E. W. Large & Snyder, 2009; Sethares, 2014). For example, the ratio 1:1 represents two intervals of equal length, while the ratio 1:2 represents an interval that lasts twice as long as the previous one. These simple integer ratios are the building blocks of more complicated, intricate temporal patterns. Thus, to understand complex structure, a first step is to identify basic ratios within temporal sequences. This observation highlights the need for systematic methods to analyze these ratios. Methodologies have been developed to assess the presence of these ratios in rhythmic patterns.

1.2 Methodologies to identify and analyze integer ratios

To assess the presence of integer ratios in a temporal sequence, a methodology has been fine-tuned in Roeske et al., (2020). They had obtained from their sequence analysis, pairs of adjacent IOIs. In each pair they calculated the ratio between the two IOIs following the formula :

$$r_i = \frac{IOI_i}{IOI_i + IOI_{i+1}}$$

Where r_i is the *i*th ratio and IOI_i is the *i*th interval.

It is a straightforward and mostly used one, because it normalizes the ratio on a continuous scale between 0 and 1. The middle of this scale (0.5) represents isochrony, indicating equal interval spacing (see figure 1).

To define if a ratio is an integer (on-integer), they divided the ratio space into bins of on- and off-integer ratio (see vertical lines in figure 1). For example, integer ratios from isochrony and under are 1:1, 1:2, 1:3 in terms of length of interval. In fraction, this corresponds to $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$. Starting from there, they defined non-integer ratios as the middle point between two integer ratios. Doing so, they defined non-integer ratios as 1:1.5, 1:2.5, 1:3.5 in terms of length of interval. This corresponds to $\frac{1}{2.5}$, $\frac{1}{3.5}$, $\frac{1}{4.5}$ in terms of fraction. They then set boundaries to distinguish between integer ratios and non-integer ratios, defining what is on-integer ratio versus off-integer ratio. These delimitation boundaries were set as the middle point between an integer ratio and the next non-integer ratio, so they correspond to 1:1.25,

1:1.75, 1:2.25, 1:2.75 in terms of length of interval. This corresponds to $\frac{1}{2.25}$, $\frac{1}{2.75}$, $\frac{1}{3.25}$, $\frac{1}{3.75}$, $\frac{1}{4.25}$ (Figure 1).

According to the grid, if a ratio falls between the two boundaries separating an integer ratio from the two adjacent non-integer ratios (see dark gray area on figure 1), it's an on-integer-ratio. If a ratio falls beyond the boundaries separating an integer ratio from its two adjacent non-integer ratios (see light gray area on figure 1) , it's an off-integer-ratio. Each ratio is assigned a weight based on the size of the bin it falls into. Roeske then performs a bootstrap on the weighted lists of on-integer and off-integer ratios. If the resulting bootstrap distributions of the two lists do not overlap, the ratios under investigation are classified as significantly present.

This methodology also uses probability density function (PDFs) to visually represent the distribution of the ratios (Figure 1). This representation helps understanding the distribution of the ratios. We can visually observe how ratios cluster around integer values, creating visually identifiable peaks in the distribution. This visual representation also helps the identification of similarities and differences in the ratio production between different sequences.

One question that could arise regarding this methodology is its sensitivity to variability within IOI sequences. As the standard deviation of the IOI distribution increases, the spread of the resulting ratio distribution will increase. How much can the spread of the IOI distribution increase while still producing clusters of integer ratios that are detected as significant by this methodology? This will be one of the secondary questions explored in this thesis.

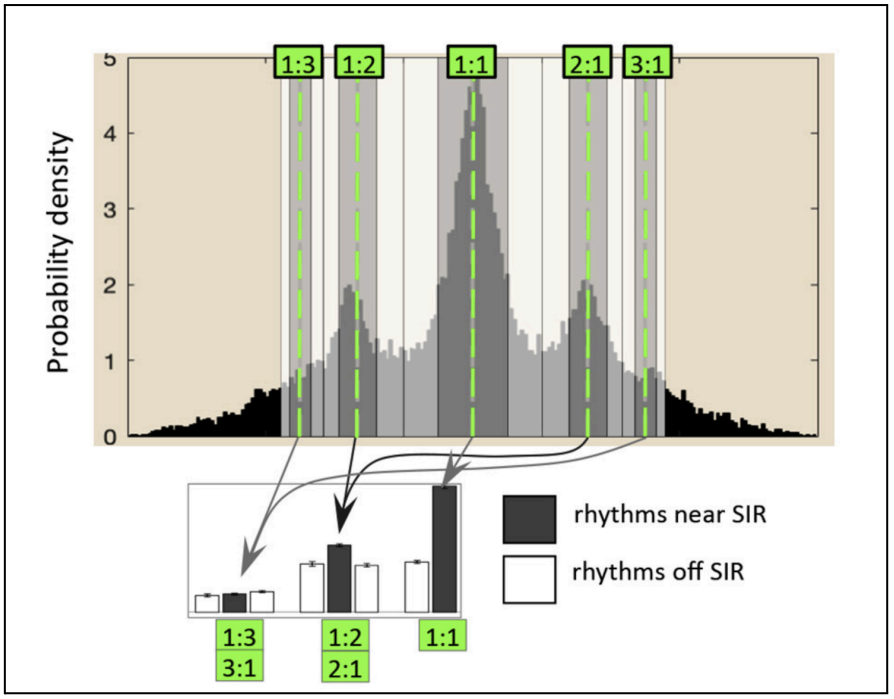


Figure 1: Graphical representation of the Inter-Onset Interval (IOI) ratios from an Indian raga dataset (Roeske et al., 2020). The black histogram displays the probability density of IOI ratios within the analyzed sequences. Dashed green lines indicate the positions of integer ratios. Dark gray areas highlight bins corresponding to on-integer ratios, while light gray areas represent off-integer ratios bins. The accompanying bar plot shows the counts of on-integer and off-integer ratios for each considered integer ratio. White bars depict the number of ratios near a simple integer ratio (within on-integer bins), while dark bars represent the number of off-integer ratios.

1.3 Presence of rhythmic categories on integer ratios

The study of IOIs ratios across musical traditions and animal calls has revealed intriguing patterns. The ratios of two adjacent intervals in these temporal sequences are not continuously distributed. They correspond to discrete categories. Moreover, these discrete categories predominantly align with simple integer ratios. The presence of these categories in acoustic signals in humans and other species suggests that it might be a crucial characteristic of sound structure. Thus, the mechanism involved in their creation could be rooted in fundamental aspects of cognition and shared across species.

Cross-cultural studies on the mental representation of rhythms (Jacoby et al., 2024) revealed that humans have systematic bias (i.e., priors) to produce rhythm that aligns with small

integer ratios (i.e., 1:1, 2:1, 3:1). In the same way, studies focusing on rhythmic production in children showed that this bias toward the production of basic small integer ratios appeared at an early stage of life (Nave et al., 2024). These findings suggest that despite the diversity of musical styles, there is a common cognitive principle governing rhythmic structures across cultures: The perception and production of temporal sequences with inter onset intervals in relation to integer ratios. This bias is likely to be rooted in basic human cognition as it appears at an early stage of life.

Similarly, researcher investigated the presence of rhythmic categories in primates (Gregorio et al., 2021; Raimondi et al., 2023). These researches revealed that lemurs *Indri Indri* and gibbons exhibit rhythmic categories in their calls. More specifically, both species presented the isochronous categories (i.e., 1:1). However, only the Indri Indri exhibited the 1:2 ratios. These discoveries suggest that use of rhythmic categories and its underlying mechanisms might be an ancient trait already present in a common ancestor of these three primates.

Moving away from the primate families, researchers still identified rhythmic categories that correspond to simple integer ratios, as exemplified by bird songs (Roeske et al., 2020). Thrush nightingales also possess rhythmic categories. They present a significant cluster on isochrony (i.e., 1:1) and a bias toward the 1:2 and 1:3 ratios. This finding indicates that the use of rhythmic categories extends beyond primates. It is therefore possible that rhythmic categories are widespread across the animal kingdom and have developed independently in different species. This would indicate again that they may play a central role in the construction of vocalizations.

The discovery of rhythmic categories in relation to simple integer ratios in both humans and non-humans species, such as songbirds and singing primates reveals a fascinating dimension of vocalization. Categorical rhythm could be a crucial component of vocal production and could reflect a shared evolutionary mechanism for rhythm. These findings underscore the need for further explorations into rhythmic patterns across a broader range of animal calls, which may offer deeper insights into the evolution and universality of rhythmic processing across species (Honing et al., 2015).

We will now review some hypotheses on the presence of rhythmic categories across human cultures and other species.

1.4 Theoretical perspectives on their presence

Rhythmic categories

Rhythmic categories play a crucial role in facilitating synchronization. Synchronization can be defined as the process of the alignment of one's action with others (e.g., in chorusing or conversational contexts). In humans, individuals rely on the structure of rhythmic patterns to elaborate predictions about upcoming events(Large & Jones, 1999). These predictions allow them to synchronize their actions in groups. The conjunction of these two processes is referred to as entertainment. Entrainment in turn, through maintain synchronization, improves group coordination. Coordination can be defined as the organization and harmonious interaction of group members in order to achieve a shared goal. For example in an orchestra, each musician entrains to the beat to synchronize their actions. This entertainment ensures the coordination between the different parts of the symphony, which enables an harmonious final outcome.

It seems that some similar principles are present in the animal kingdom. Some of the species in which rhythmic categories have been identified also exhibit song coordination. For example, the lemur *Indri Indri,* presents in his vocalization the 1:1 and 1:2 ratios. This lemur is also a singing primate and displays coordinated duet and chorusing. Therefore, rhythmic categories may also help coordination in such species. However, the way these lemurs synchronize to achieve coordination do not involve entrainment to the beat as humans do. Humans are the only ones among primates to be able to synchronize their action by entraining themselves to external stimuli (Merker et al., 2009).

Regular rhythmic patterns also simplify cognitive processing. They provide a predictable structure that the brain can easily interpret and anticipate. When rhythms are categorized into familiar patterns, the cognitive load required to process and respond to these patterns is reduced (E. W. Large & Jones, 1999). For example, Cason and Schön, (2012) found that rhythmic priming improved phonological processing. Participants detected a specific phoneme more quickly when it was presented on-beat with a rhythmic prime, compared to when it was off-beat. This demonstrates how rhythmic alignment can enhance cognitive efficiency (Cason & Schön, 2012).

Rhythmic categories also significantly aid in learning and memory. The repetitive patterns serve as mnemonic clues, helping individuals to remember information more effectively

(Essens & Povel, 1985; E. W. Large & Jones, 1999, p. 19; Purnell-Webb & Speelman, 2008). The structure of rhythmic sequences also enhances neural processing by providing a regular temporal structure during encoding (E. W. Large & Jones, 1999). This structure aids in memory reinforcement, improving long-term retention and recognition performance (Essens & Povel, 1985; Purnell-Webb & Speelman, 2008).

It is also plausible to hypothesize that in animals, rhythmic patterns play a crucial role in learning and memory. The songs of many birds are culturally transmitted by learning (Roeske et al., 2020). They learn their songs through the imitation of rhythmic patterns heard from adult conspecifics. Indeed, discrete entities are often simpler to communicate and learn compared to those from a continuous range (Rosch et al., 1976).

Taken together, these findings suggest that categorical rhythm may share cognitive purposes across humans and other species that exhibit rhythmic patterns.

Convergence toward integer ratios

One question in the study of rhythm is why these categories correspond to integer ratios. The Neural Resonance Theory (NRT) (E. Large & Grondin, 2008) try to link this bias with the brain's oscillatory activity. Biological neurons in the brain possess a rhythmic activity at various frequencies and therefore can be compared to an oscillator network. The neural oscillators of this network tend to synchronize to external stimuli. This synchronization operates through several mechanisms.

One of these mechanisms is phase locking. It represents a process where the timing of one signal oscillations synchronized with another signal to match his phase (i.e., the position of a point in time in a waveform cycle) and frequency (Pikovsky et al., 2001). In a phase-locked loop, the system continually adjusts its output to match the timing of an input signal. This process keeps both signals in synchrony and allows the system to maintain a stable rhythmic pattern. Phase locking between systems is often observed at integer ratios, because they represent straightforward, predictable relationships between periodic events, making them easier to maintain in both biological and mechanical systems (Patel et al., 2005; Pikovsky et al., 2001). However, phase locking can also occur at non-integer ratios. This is especially true in complex oscillatory systems or those with multiple interacting components (Pikovsky et al., 2001). Phase locking is not exclusive to neural systems. It can also occur in other biological systems through mechanisms such as basic error correction feedback (Sismondo,

1990). These kinds of mechanisms are memoryless (i.e., not able to predict) and constantly adjust their discrepancy with the desired outcome based on real-time perceptive feedback from the stimuli. Sismondo's research on cricket chirping behavior (Sismondo, 1990) offers an example of it. It shows how a cricket can achieve stable phase locking with an external stimulus through a simple feedback process. When the cricket hears a stimulus, it adjusts its chirp period for the next chirp based on whether it was ahead or behind.

However, the NRT theory also involves more complex principles of entertainment (E. Large & Grondin, 2008). The neural oscillators respond more strongly (i.e., resonate) to stimuli that match their frequency but also its harmonics. Harmonics are related to each other by integer ratios (Pikovsky et al., 2001). The ability of these oscillators to entrain at multiple time scales of an external stimuli through non-linear oscillation is supposed to be linked with the perception of hierarchical rhythm structure in humans. Consequently, models based on this theory imply the perception of beat and meter (E. Large & Grondin, 2008; E. W. Large et al., 2023) and might not stand to explain the presence of integer ratios in other species. However, the study of the capacity of a single neuron to produce rhythmic patterns corresponding to integer ratios could inform us on the role of neural oscillators beyond humans.

We have now assessed the presence of integer ratios across musical culture and diverse species and expose different hypotheses regarding their presence from a cognitive and neural point of view. The next chapter will introduce the importance of modelisation to uncover biological mechanisms and review some models of rhythmic production.

Chapter 2: Rhythmic modeling

2.1 Role of computational models in uncovering biological mechanisms

Computational models play a crucial role in uncovering the mechanisms behind biological processes. By simulating complex biological systems, these models help researchers explore hypotheses, predict outcomes, and understand the underlying principles of various phenomena. To understand how an observed system operates internally, one must construct a mechanism that exhibits the same behaviors. In the context of rhythm perception and production, computational models can provide significant insights into the neural and cognitive mechanisms involved, testing the different hypothesis about the emergence of rhythm categories on integer ratios (Vernes et al., 2021).

For example, behavioral studies of western and central South Pacific whales pointed out distinct patterns in song evolution between these two populations (Payne & McVay, 1971). In the northern populations, songs evolved gradually. Conversely, in the southern populations, periodic revolutions occurred as songs are rapidly adopt from neighboring populations. Captive studies on smaller cetaceans have already demonstrated their capacity for vocal learning (i.e., individuals modifying their signals based on experience with others) (Janik & Slater, 2000). However, Humpback whales, with their complex and evolving songs, were an extraordinary case study. The study of these animals through controlled field experiments was challenging. To overcome this issue, researchers modeled the mechanism of song transmission present in humpback whales (Zandberg et al., 2021). Thanks to this modelisation, they discovered that the size of the population and the way individuals interacted can explain the variation observed in the speed at which whale songs evolved between the two hemispheres. Their findings show the power of computational models to uncover underlying mechanisms (i.e., vocal learning and cultural transmission in animal populations). Their study offers insights that extend beyond what behavioral data alone have provided. Therefore, this approach can also be applied to test alternative hypotheses regarding the production of rhythmic categories corresponding to integer ratios.

2.2 State of the art of the actual models for rhythmic production and perception

In the current literature on rhythmic modeling, one category of models is based on musicological principles. These kinds of models are based on musical knowledge and assume perception of beat and meter. Syntactic rhythmic trees are an example of them. One example of it is the syntactic tree model developed by Martin A. Rohrmeier (Rohrmeier, 2020). This model is a formal approach to understand rhythm, and it is based on principles taken from linguistics and music theories. It put the accent on the hierarchical and syntactic pattern that underlie rhythmic patterns in music. It represents rhythms reusing syntactic tree diagrams (see figure 2). Each node in the tree corresponds to a rhythmic event or a group of events. Higher-level nodes represent larger rhythmic groups like meters, and lower-level nodes represent individual beats or subdivisions. To determine how to group rhythmic events, the syntactic rhythmic tree employs rules to analyze rhythmic patterns. Following the same idea, by applying these syntactic rules, the model could generate new rhythmic patterns. Starting from a root node representing an entire measure, and recursively applies grouping and subdivision rules to create detailed rhythmic sequences. This generative approach ensures that the resulting rhythms adhere to the hierarchical and syntactic principles embedded in the model. For example, consider a generative application for a simple 4/4 measure. The syntactic rhythmic tree for this rhythm would start with a root node representing the entire measure. This root node would branch into four sub nodes, each representing a quarter note. Further subdivisions could represent eighth notes or sixteenth notes within each quarter note. By varying the subdivision rules and grouping structures, the model can generate a wide range of rhythmic variations while maintaining a coherent overall structure.

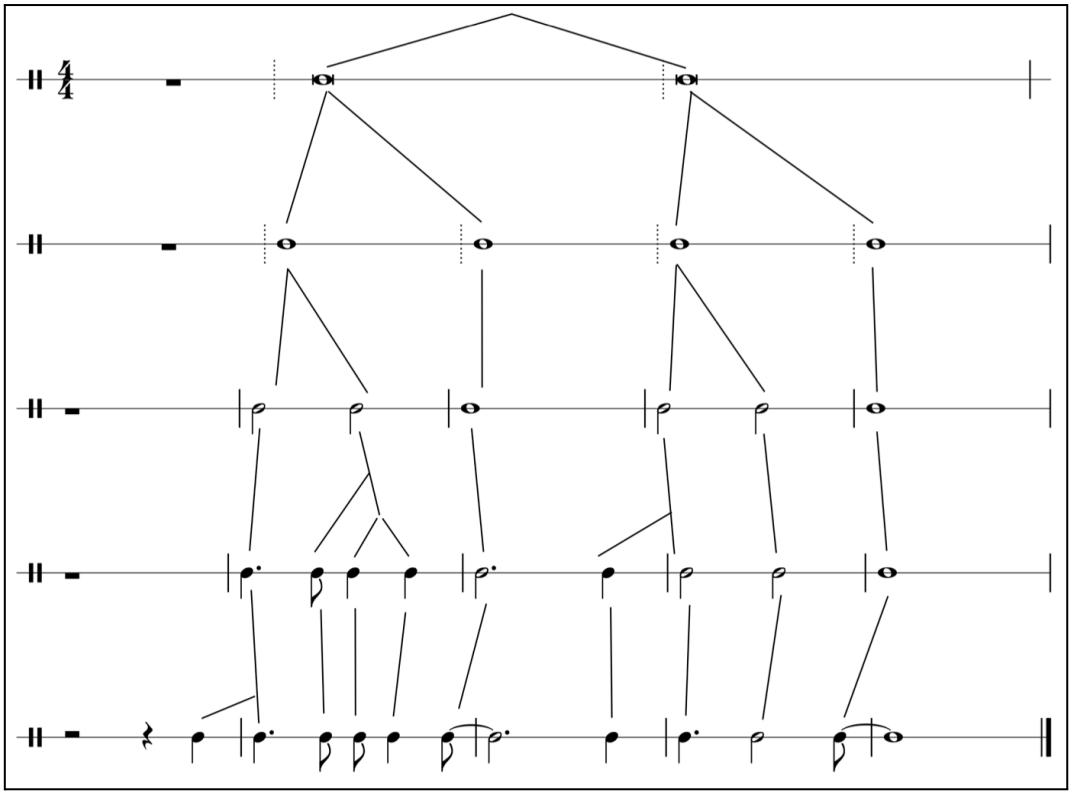


Figure 2: A rhythmic analysis of the first four bars of the melody of the Jazz standard “Blue Bossa”. (Rohrmeier, 2020) © M. Rohrmeier. Licensed under CC BY 4.0.

To enhance the diversity of rhythm generation, algorithms based on musical knowledge are often used in conjunction with Markov chains (Pachet, 2010; Temperley, 2010). Musical knowledge-based algorithms encompass the rules that handle the overall musical structure. Meanwhile, Markov chains manage probabilistic transitions between rhythmic states, focusing on the microstructure of rhythmic patterns providing some variability (see chapter three for a detailed explanation). This dual approach ensures that the rhythm sequence exhibits both coherence and variability.

The formalism and generative rules employed in musicology modeling bias the production of inter-onset intervals toward integer ratios. Indeed, the concept of meter in western musical tradition divides the temporal space into equal parts corresponding to integer units of time. Then, the rules to generate rhythm corresponding to this meter are based on the division or multiplication by an integer or integer ratios of this metrical grid. For example, a metrical grid compose of four units of time, can be only compose of IOIs express as simple fraction of the grid (e.g., 1/4, 1/8, 1/16, etc) which correspond to integer ratios (1:1, 2:1, 4:1, etc.).

Therefore, rohrmeier's syntactic rhythmic tree model provides a powerful tool to analyze and generate rhythmic patterns in music. The model aligns with some cognitive theories of rhythm perception, suggesting that listeners perceive rhythms in a structured and organized manner. In addition, it provides an explanation for the emergence of rhythmic categories related to integer ratios. However, there are some limits to the use of a musicology based model. The theories that constitute the model are linked with specific cultures. Doing so, they are mostly accurate only within the specific musical contexts they have been developed upon. In addition, this kind of model also offers limited insight in the biological bases of the production rhythmic categories.

Another category of rhythmic generative models are coming from the neural resonance theory. As explained in the previous chapter, these kinds of models are based on coupled oscillators that aim to imitate the brain's neural rhythms. The Gradient Frequency Neural Networks (GrFNNs) is one of the models based on this theory (E. W. Large et al., 2023). The model is composed of a group of oscillators with each a specific fundamental frequency. Together, they form a gradient of frequencies and allow the model to respond to a great range of frequency input. The oscillators that have a fundamental frequency close to the stimulus frequency will enter in resonance and respond more strongly to the stimulus and synchronize their phase with it. Each of the oscillators that have adapted their phase to match the stimulus will in turn influence neighboring oscillators. This process will continue until the entire network attains a stable rhythmic pattern. To achieve and maintain this stable rhythm, the model also relies on non linear oscillation. With some oscillators locking their frequencies on harmonic frequencies of the stimulus.

As explained in the previous chapter, The GrFNNs model and other models based on the NRT theory explain the formation of rhythmic categories based on integer ratios due to inherent properties of the oscillators (E. Large & Grondin, 2008; Pikovsky et al., 2001). The GrFNNs model demonstrates how neural oscillation can entrain to external rhythms, and produce rhythmic categories corresponding to integer ratio. However, the model is highly tied to human cognition and assumes the perception of a hierarchical rhythmic structure. In doing so, it might not be applicable to non-human species.

To overcome these issues and still have simple biological mechanisms explaining the emergence of integer ratio it's possible to focus on simpler mechanisms such as error correction models. This mechanism might explain rhythm synchronization through phase

locking without relying on complex cognitive processes (Sismondo, 1990). Another solution could be to focus on the firing property of a single neuron.

From this review, it came out that many models rely on cultural or perceptual knowledge specific to human rhythms. In contrast, our goal is to explore simple models that can generate integer ratios independently of particular musical traditions or high order cognitive processes (i.e., the perception of hierarchical structure of rhythm). This approach aims to provide a more universal framework to understand rhythm production and the creation of rhythmic categories related to integer ratios.

2.3 Need for comparative analysis and model evaluation.

Numerous rhythmic models are available, but few studies have systematically compared them to assess the different insights they provide. Comparing models can reveal how various approaches explain the same phenomena through different underlying processes. For instance, neural oscillation models might provide biological explanations to the creation of hierarchical rhythmic structure in humans (E. Large & Grondin, 2008), while rule-based models explain this hierarchy in terms of syntactic rules present in a given musical tradition (Rohrmeier, 2020).

In addition, comparing simple models against more complex ones help assess the relevance of the added complexity. Following the principle of Occam's razor (Duignan, 2024), simple models should be favored unless the added complexity yields a significant improvement. For example, comparing a basic rhythmic model that predicts rhythmic patterns using a simple statistical (Pachet, 2010) approach against a more complex model incorporating advanced machine learning techniques such as neural networks (Karbasi et al., 2021). In this case, the simpler model acts as the null hypothesis, and the complex model is evaluated to determine whether its additional complexity significantly enhances the accuracy of rhythmic predictions.

Each model may have its strengths and limitations depending on the specific aspect of the rhythmic behavior we are interested in (e.g., biological, cognitive or cultural). For example, the strength of neural oscillation models might stand in their biological plausibility (E. Large & Grondin, 2008). Conversely, they might have difficulties in accurately producing the rhythmic structure of a specific musical tradition. Syntactic rhythmic trees, for their part,

might excel in generating rhythms for a specific musical tradition (Rohrmeier, 2020), but lack biological basis.

Each model might apply to different scales of the rhythmic behavior. Models that rely on beat and meter perception might explain the complex hierarchical structure of rhythm present in humans (E. W. Large et al., 2023; E. W. Large & Snyder, 2009). Simpler models might be able to account for humans and non-human species, providing insights into the evolutionary aspects of rhythm production.

Overall, comparative analysis of available models is essential to advance our understanding of rhythmic behavior.

This chapter has pointed out some gaps in the modeling literature of rhythm. There is a need for simple models that can account for the creation of rhythmic sequences with intervals related to integer ratios that can be applied across human musical traditions and species. These simple models would ideally have biological plausibility. They should also not refer to any ability to perceive beat or meter. In the next chapter, we will delve into the presentation of three different models that satisfied these characteristics.

Chapter 3: Selection and description of the models

3.1 Introduction

This chapter will present our three computational models of interest: Markov chains, the cricket model and spiking neural networks (SNNs). Markov chains, though their memoryless properties, provide a structured approach to modeling rhythmic transitions and are effective in capturing the emergence of specific integer ratios within musical contexts. The cricket model, based on insect synchronization behavior offers ecological validity and insights into cross-species rhythmic behaviors. SNNs simulate neuronal dynamics to understand how intrinsic connectivity and firing properties lead to rhythmic patterns. This third model is aligned with neural entrainment theory. Collectively the three models contribute to unraveling the mechanisms underpinning rhythmic complexity across various biological and computational frameworks.

3.2 Markov chain model

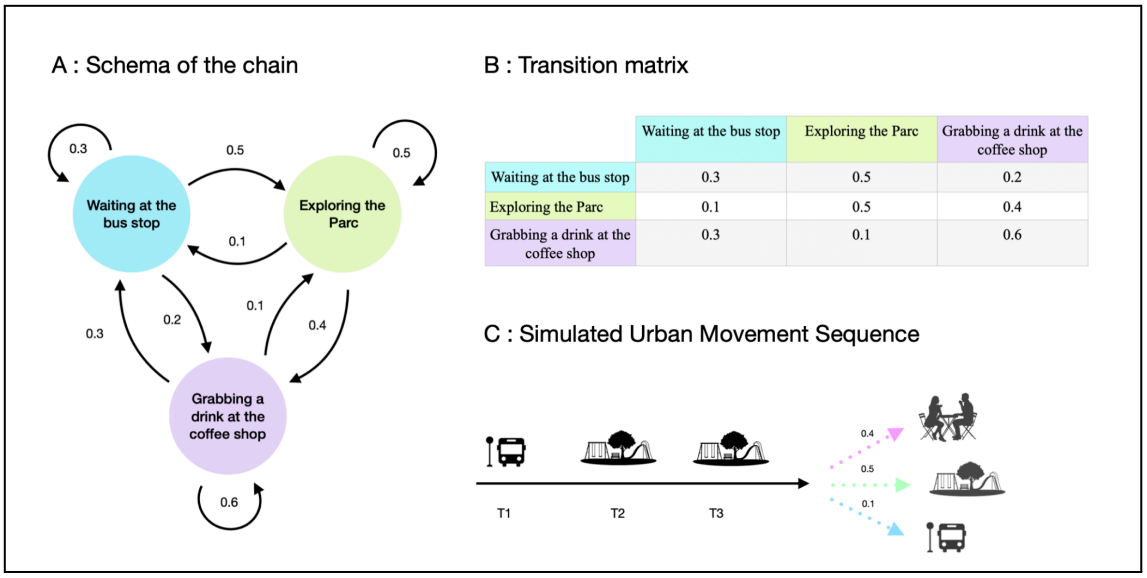
- **General explanation**

Our initial focus is on a first-order discrete-time Markov chain, a fundamental model in the realm of stochastic processes. Markov chains describe sequences of potential events, highlighting transitions between different states. These transitions obey specific probability rules, where the likelihood of moving to a certain state depends solely on the preceding state (Norris, 1997; Sneyd et al., 2022):

$$Pr = (X_{n+1} = x \mid X_n = x_n)$$

To make it more concrete, imagine yourself in a crowded city center with people randomly walking around. We can capture and predict the random paths of the people, by looking at it through the lens of a Markov chain. Within this framework, we can delineate a series of "states" representing different activities or locations within the city center, such as waiting at a bus stop, exploring a park, or grabbing coffee at a café. The transitions between these states occur probabilistically, mirroring the diverse factors influencing people's movements (e.g. the time of the day or the weather). This probability of transiting from one place to another is

captured within a transition matrix. This matrix details the likelihood of moving from one state to another, based on the current state. For example, if someone is at a bus stop, the transition matrix might indicate that it has 20% of chance of heading to the café, but 50% chance of exploring the park, and only 30% of chance to stay at the bus stop. Importantly, the assumption made here is that the probability to transition to a new state is only dependent on the current state. In other words, in a Markov chain, the prediction of the next state should be as good as the one that could be made knowing the process's full history. In this example, it would mean that each person in our city only cares about where they are now and not where they come from when deciding where to go next. From this set of states and probabilities we can extract a Markov chain model as presented in figure 1. This representation allows us to understand the complex movements of individuals in the city center, and can capture someone's journey in a sequence of probable locations.



	Waiting at the bus stop	Exploring the Parc	Grabbing a drink at the coffee shop
Waiting at the bus stop	0.3	0.5	0.2
Exploring the Parc	0.1	0.5	0.4
Grabbing a drink at the coffee shop	0.3	0.1	0.6

Figure 1: An example Markov chain model of urban dynamics, illustrating the transition probabilities and a random sequence gets generated by the Markov chain. (A) Diagram of the chain illustrating the transition between the different states. (B) Table of the different transition probabilities. (C) Illustration of a temporal sequence showing a possible the transitions over time between the different locations.

- **Rationale for exploring this model**

Using stochastic models to generate rhythmic behavior, and even more so, musical rhythmic behavior, may initially seem counterintuitive given the perception of music as a structured rather than a random art form. However, several types of time sequences have been successfully modeled using random processes (e.g., stock prices). Music and musical rhythm

are no exception, as they also deal, to some extent, with randomness. Transitions between microscopic musical states (e.g., one note or rhythm to another) can be described by probabilistic laws. There is not just one way to transition from one note to another within a specific musical style, but rather a set of possibilities with different probabilities of occurrence, thus allowing for a balance between structure and randomness in the compositional process. Examples from composers like Xenakis demonstrate how stochastic processes can imbue compositions with emotion, challenging initial expectations of mechanistic music (Harley, 2010). The global process is predictable even if the composed elements are random.

There may be concerns about the applicability of Markov chains at a macroscopic scale of music composition (e.g. musical phrases), due to their memoryless nature. However, they are effective to represent musical mechanisms and in fact, have been widely used to identify musical composers or even generate improvisation (Temperley, 2010).

All of these are important points for us here. First, Markov chains have already successfully been used to model some features of musical processes. They are memoryless; therefore, they do not refer to any beat or metric perception. This allows us to have a model of a mechanism that could fit as well for humankind rhythmic production as for non-human species. The fact that the model is only valid on a smaller scale and could lack coherence on a larger scale doesn't concern us for this specific search. We are interested in which integer ratios the mechanism can produce, and while integer ratios can occur at larger scales, we are specifically interested in integer ratios between adjacent intervals. Finally, a Markov chain is a simple model with an easily interpretable and explainable mechanism, and we can easily control the different parameters which explicitly map onto properties of the resulting sequences. Indeed, concerning the production of integer ratios, if we define our states already in relation to integer ratios, it will obviously produce what we are looking for. However, we will be able to finely control some other parameters. Specifically, we will be interested in this model in the spread of the distribution that still allows for a significant presence of integer ratios (see Chapters 4 and 5).

- **Detailed model setup and dynamics**

We programmed and simulated our Markov chain model using Python. The aim of our program is to obtain sequences of IOIs, using a Markov chain, with duration that canform integer ratios. The different states of the chain represent the different duration of IOIs that our

model can produce. Each state corresponds to a normal distribution with a specific mean (μ) and standard deviation (σ). At each step, a random value will be selected in the normal distribution corresponding to the current state of the chain. The value selected at each step will be saved in a list and thus create our sequence of IOIs. From this sequence of IOIs, we can calculate the different rhythm ratios between adjacent IOIs, and extract a density function and its graphical representation to quantify the presence of integer ratios, as explained in Chapter 2.

For each simulation of our model, creating a sequence of IOIs, it is possible to set its parameters: the number of states, the parameters of the normal distribution corresponding to each state (μ,σ), the starting probabilities of the chain and the transition matrix between the different states. In addition, the script allows us to obtain its stable distribution using an empirical or analytical method. A detailed description of the model computation is present in Appendix A. Based on this, we can have a look at what our model can produce. Figure 2 illustrates the process of our Markov chain models with two states to obtain a sequence of IOIs.

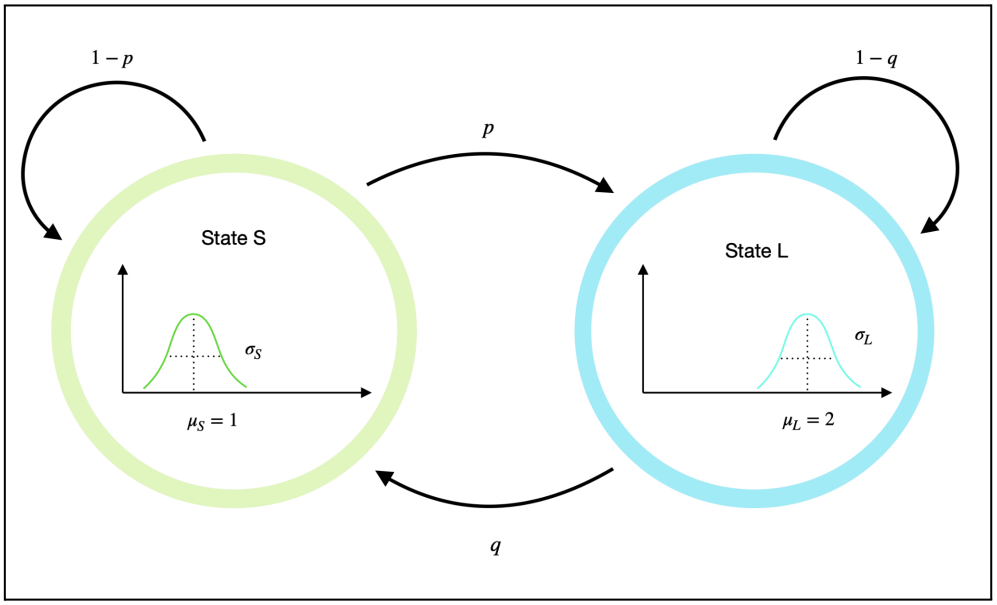


Figure 2: Diagram of a 2-state Markov chain, generating a sequence of IOIs. Several parameters of the Markov process are shown, including the transition probabilities p and q, and the mean and standard deviation of each state's normal distribution (μ_S, σ_S) and (μ_L, σ_L). This model can easily be extended to more states, by providing additional transition probabilities and parameters for a new state's distribution of generated IOIs.

In Figure 3 below, we can see the resulting density plot for a sequence of 10000 steps created by this two state Markov chain model with these specific values for the transition matrix and the normal distributions.

A: Transition matrix

	S	L
S	0.5	0.5
L	0.5	0.5

B: Normal distributions

(μ_S, σ_S)	(1 , 0.1)
(μ_L, σ_L)	(2 , 0.2)

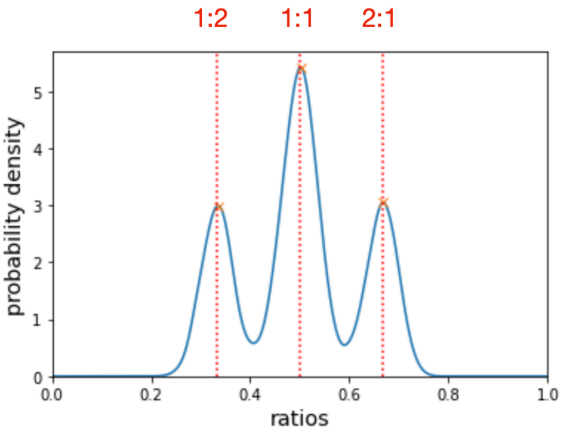


Figure 3: The estimated probability density function resulting from the simulation of the Markov chain shows three distinct peaks, corresponding to three distinct integer ratios (1:1, in the middle; 1:2 and 2:1, respectively left and right).

We can anticipate the presence of the three peaks in the density function in Figure 3 mathematically. As we have set our mean of normal distribution respectively on 1 and 2, we are inducing the production of specific integer ratios. One possibility is that the Markov chain is in state S and stays in S, or is in state L and stays there, producing two similar IOIs and leading to a ratio close to 1:1. Similarly, if the Markov chain switches from S to L, it will produce two IOIs close to a 1:2 ratio, and a transition from L to S will lead to a 2:1 ratio. As all transitions have equal probability, with twice the possibilities of producing a 1:1 ratio compared to the two other ones, the peak for isochrony (1:1) is more prominent than the two smaller peaks for the other integer ratios.

Observing this, we can play with the transition probabilities to disadvantage isochrony production in our model. For example, by drastically reducing the probability of staying in one state as shown in Figure 4.

A: Transition matrix

	S	L
S	0.1	0.9
L	0.9	0.1

B: Normal distributions

(μ_S, σ_S)	(1 , 0.1)
(μ_L, σ_L)	(2 , 0.2)

C: Resulting ratio density plot

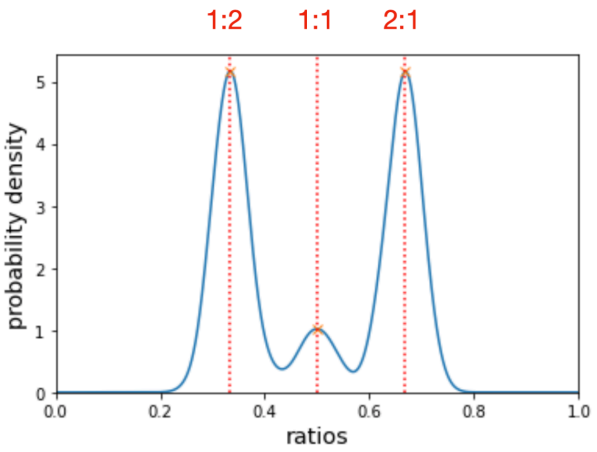


Figure 4: When the transition probabilities *p* and *q* are increased, the Markov chain remains less often in the same state S or L. The resulting probability density now shows much larger peaks around the 1:2 and 2:1 ratios.

Doing some calculation, we could predict the outcome of our model mathematically based on our states and the probabilities transitions. But it becomes more interesting to look at the spread of the states' distributions, and how this will impact the significance of our results. For example, how broadly can a state's generated IOIs be spread can to still be close enough to detect the underlying integer ratios and be considered as statistically significant? And what happens if the spread of the two IOIs distributions start to overlap? Figure 5 shows an example of this exploration that we will be digging into in the next chapters.

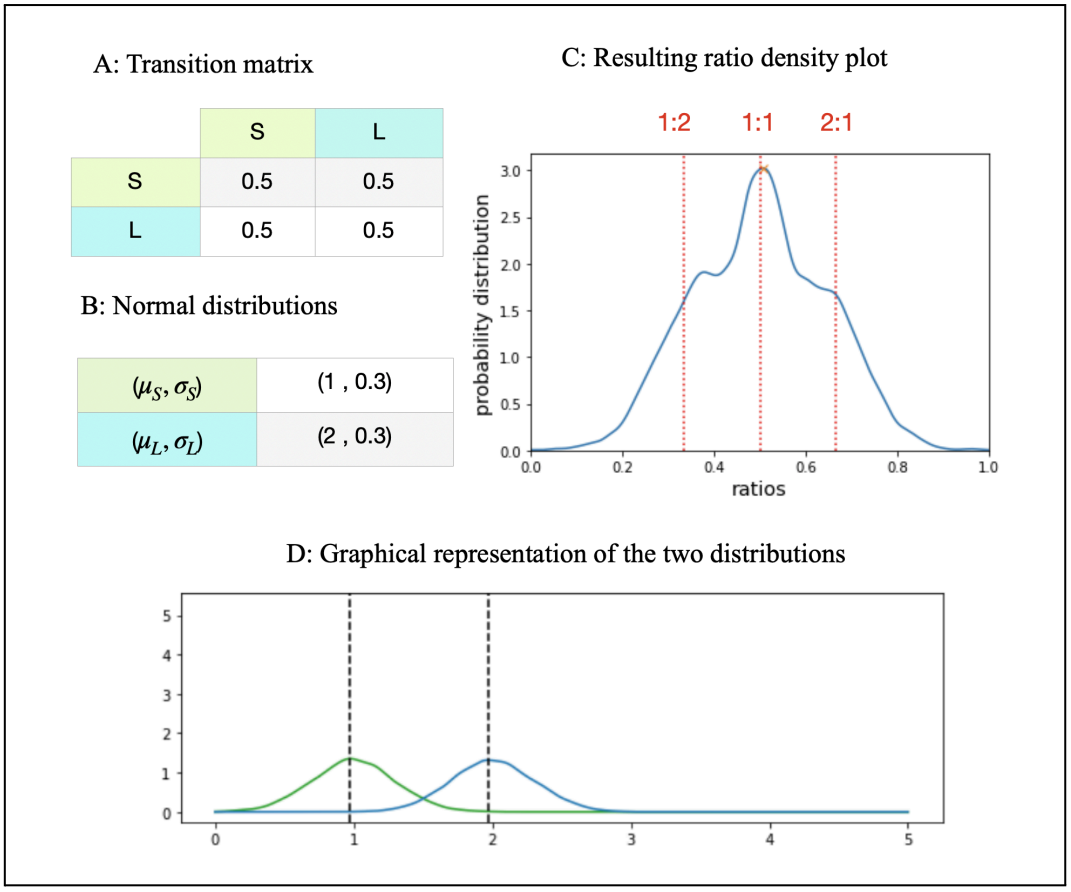


Figure 5: When the spread of the states' two IOI distributions (i.e., σ_S and σ_L) increases, the resulting density of rhythm ratios does not show three separate peaks anymore. Rather, because of the increased noise when sampling IOI, the resulting ratio peaks also become wider and overlap.

- **Parameters of interest and their relevance**

Several parameters can be varied in our simple Markov chain model. However, as we laid out above, two parameters of interest will be of particular interest: the standard deviation (σ) of the normal distribution of the states, and will be explored in the experiments in this thesis. Additionally, we will also explore how changes in the transition probabilities between the states affect the behavior of the model and the produced integer ratios.

We will be limiting our investigation in this search to a two state Markov chain, but given the flexibility of the model, additional states could easily be integrated in future experiments. Finally, a Markov chain model also has probabilities for the state in which the simulation will first be. Even though our model implementation allows for the selection of starting probabilities, the number of steps we are simulating in each sequence is much higher than the number necessary to reach the stationary distribution; i.e., simulations of this simple 2-state

Markov chain get quickly to a point where the initial state ceases to matter. Thus, this parameter should not impact the results and is left aside in our experiments.

3.3 Spiking neural network model

- **General explanation**

Our second model of interest will be a spiking neural network. Artificial neural networks (ANNs) are models inspired by the organizational principles of biological neural networks. They consist of units known as artificial neurons interconnected by edges of different weights, analogous to the neurons in the brain connected to each other by synapses of different strength (see Figure 6).

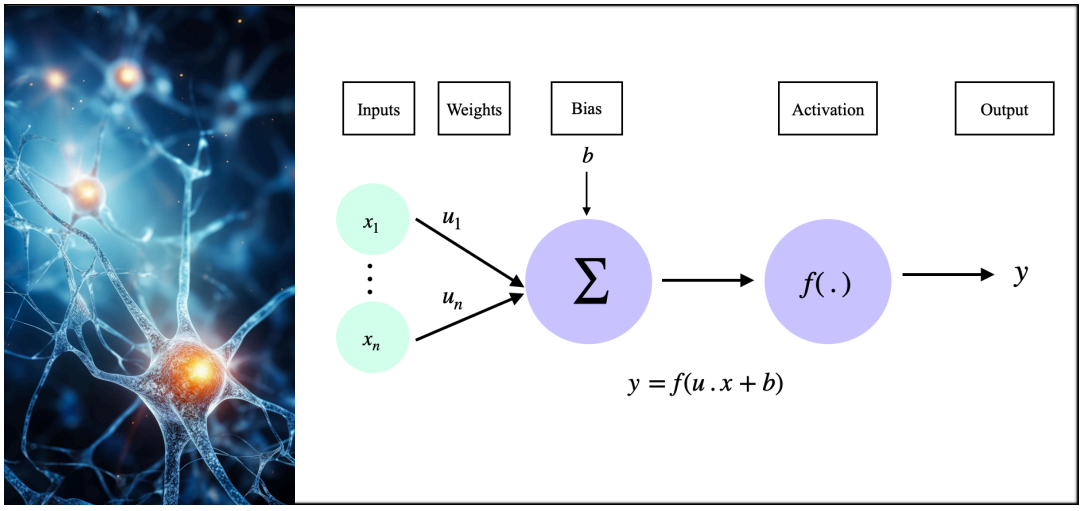


Figure 6: Illustration of a biological neural network[1] (left) form the inspiration for the modeled structure and behavior of artificial neural networks (right).

The initial input of ANNs are usually features of external data encoded as real numbers. To determine the output of a neuron, the input numbers are sum according to their weight with the addition of a bias. The final sum passes through an activation function to generate the output. This final output represents the accomplishment of the task (e.g. recognition).

ANNs form the basis of the recent surge in deep learning models which have reached state-of-the-art results in several artificial intelligence tasks. ANNs are widely used for tasks such as classification, predictive modeling, and adaptive control, aiming to estimate outcome probabilities based on input datasets (Bishop, 2016; Narendra & Parthasarathy, 1990). They undergo training to optimize connection weights primarily through backpropagation methods

[1]https://www.vecteezy.com/free-photos/neuron Neuron Stock photos by Vecteezy

and have evolved significantly over the past decades, transitioning from networks with a single layer between input and output, to deep neural networks with many more layers and non-linear transformations. This increasing complexity in combination with larger amounts of training data has strongly improved their performance over the years.

However, despite the original inspiration to imitate brain architecture, ANNs still lack biological plausibility in terms of structure, computation, and learning rules. This limitation has driven the development of spiking neural networks (SNN), which seek to partially address these challenges (Yamazaki et al., 2022).

SNNs aim to bridge the gap between neuroscience and computation, using more realistic neuron models to carry out the computation. In traditional ANNs, nodes are connected to each other by each of different strength passing real numbers, mirroring the connections between neurons through multiplication. But actually, in the brain, neurons communicate with each other with spikes trains. A spike is emitted when the voltage of a neuron membrane depolarizes after having reached a threshold value. After that, the membrane goes to a refractory period where it cannot receive input anymore and returns to its rest state ready to react to incoming stimulus again.

SSNs aim to model this biological reality much closer. The model considers the properties of the neuron, and assigns a value representing its electrical potential (that changes according to its mathematical model), the kind of synaptic connection, and simulates the changing neuron in continuous time rather than discrete time. The main idea is that an artificial neuron of SNNs fires when its membrane reaches a threshold value, sending a signal to neighboring neurons which then adjust their potential in response to this signal. Figure 7 depicts a schematic of such a spiking neural network (Dutta et al., 2020). While the structure of an SNN resembles the architecture of a traditional ANN, the information gets propagated in a fundamentally different manner.

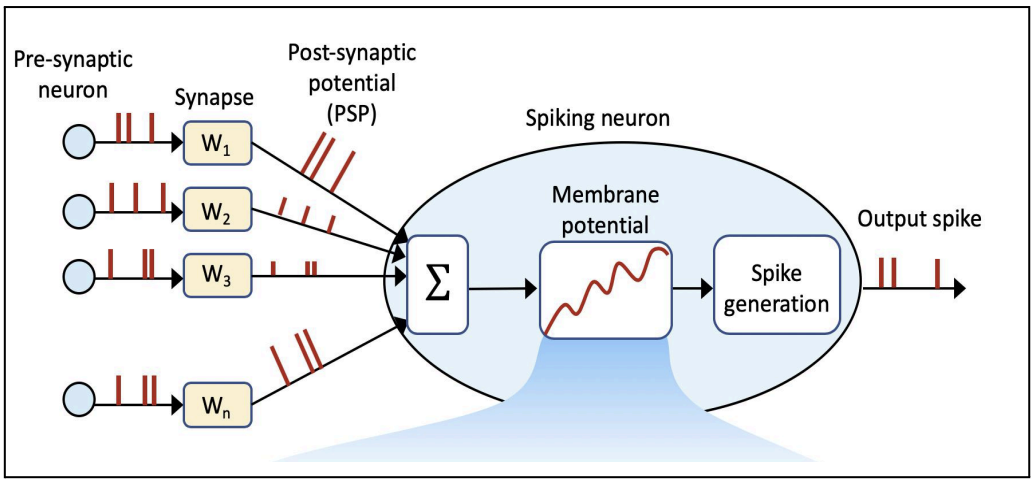


Figure 7: Diagram of a spiking neuron (Dutta et al., 2020). While the overall structure resembles the one in Figure 2, information gets propagated in spike trains rather than real number values.

Due to their similarities with biological neural networks, SNNs are also used to test biological hypotheses and model brain networks (Susi et al., 2021). This kind of use is of great interest for us, as it means that this model can be a plausible representation of a biological one and potentially provide insights into biological processes producing integer ratios.

- **Rationale for exploring this model**

As already touched upon above, a model based on a spiking neural network would be really relevant for investigating integer ratios in temporal sequences generated by biological processes. SNNs are models biologically plausible and specialize in the treatment of temporal information. We can explore simple neuronal organization, through their inherent connectivity and firing properties, can autonomously generate temporal patterns that adhere to integer ratios. While our experiments do not directly involve external rhythmic adjustment, they align with the spirit of neural entrainment theory by exploring how neurons synchronize and organize their activity to encode temporal information. Moreover, SNNs allow for the construction of simple organization of neurons, which can comprise just one or two units and effectively produce different kinds of temporal sequences.

- **Detailed model setup and dynamics**

A widely used tool for spiking neural networks today is the NEST library (Espinoza Valverde et al., 2024), which can be accessed via a standalone application or a Python package. This Python library is ideal for us to use it in our simulations. It provides a wide range of models: 50 models of neurons and 10 different synaptic models and even allows different neurons and

synaptic models to coexist, like in a biological neural network.

For our simulation, we aim to have a structure relatively simple. To achieve this, we choose the *leaky integrate-and-fire alpha-shaped input currents* neuron[2] (Espinoza Valverde et al., 2024). The leaky integrate-and-fire family is a widely used because it combines computational simplicity and biological plausibility. The model assumes a fixed threshold value, a fixed refractory period and don't present any adaptation mechanisms. This model is one of the simpler ones available, but it already has a great range of parameters that we can explore in our experiments and simulations.

Now, we will delve into the equations that define the model. The membrane potential evolves according to :

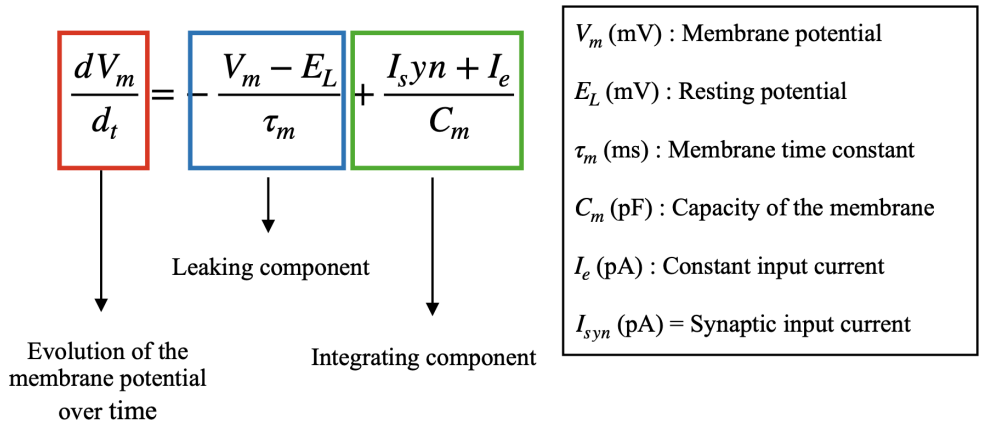


Figure 8: Membrane potential evolution over time in the *leaky integrate-and-fire alpha-shaped input currents*.

Based on the equation in Figure 8, the neuron emits a spike at time step $t^* = t_{tk+1}$ if $V_m(tk) < V_{th}$ and $V_m(tk+1) \geq V_{th}$.Subsequently, the membrane potential gets reset to $V_m(t) = V_{rest}$ for $t^* \leq t \leq t^* + t_{ref}$ and is clamped to V_{rest} during the refractory period.

The leaking component indicates that the higher the membrane potential (V_m), the greater the leak of the neuron. We can control the degree of this leakage by varying the value of the

[2] (*iaf_psc_alpha – Leaky integrate-and-fire model with alpha-shaped input currents — NEST Simulator Documentation*, s. d.):

membrane time constant (τ_m). As shown in the equation, higher values of τ_m result in a less powerful leakage component because it appears in the denominator. This leads to a more efficient charging, leading to a higher firing rate. Conversely, lower values of τ_m increases the leakage during the neuron's charging phase, leading to an elongation of the charging time and the reduction of the firing rate. If τ_m is too low, the neuron may not be able to fire at all, as the membrane starts leaking more than it gets chaged: In this case, when the membrane potential reaches a certain point, the leakage becomes equal to the charging rate, preventing further increase in potential (see Figure 9 A, B, C). This limit can be demonstrated by finding the point where the derivative of the membrane potential becomes zero, given a specific set of parameters in the equation.

The integrating component indicates that the membrane potential (Vm) is influenced by the external input current (I_e). We can see this by varying the value of I_e: As we can observe inFigure 4, higher values of I_e result in a stronger integration of the external current, leading to a more rapid increase in V_m and therefore a higher speed of firing. Conversely, lower values of I_e result in a weaker integration, slowing down the increase in V_m and consequently the firing rate (see Figure 9 D, E, F). If I_e is too low – less than 375.1pA –, the neuron is not able to reach the threshold to fire (see Figure 4 F) . This limit can be demonstrated again by finding the point where the derivative of the membrane potential becomes zero, given the specific parameters of the equation.

The upper limit for the firing rate is defined by the absolute refractory period of our neuron (i.e. 2ms). This period means that after firing, our neuron is not able to fire again for 2ms. Thus, our neuron cannot fire at a rate higher than 500Hz. We recorded the number of spikes from a neuron under various input currents and determined that the actual limit firing rate of this model, given the default parameters, is 475Hz for an input current of 37706 pA.

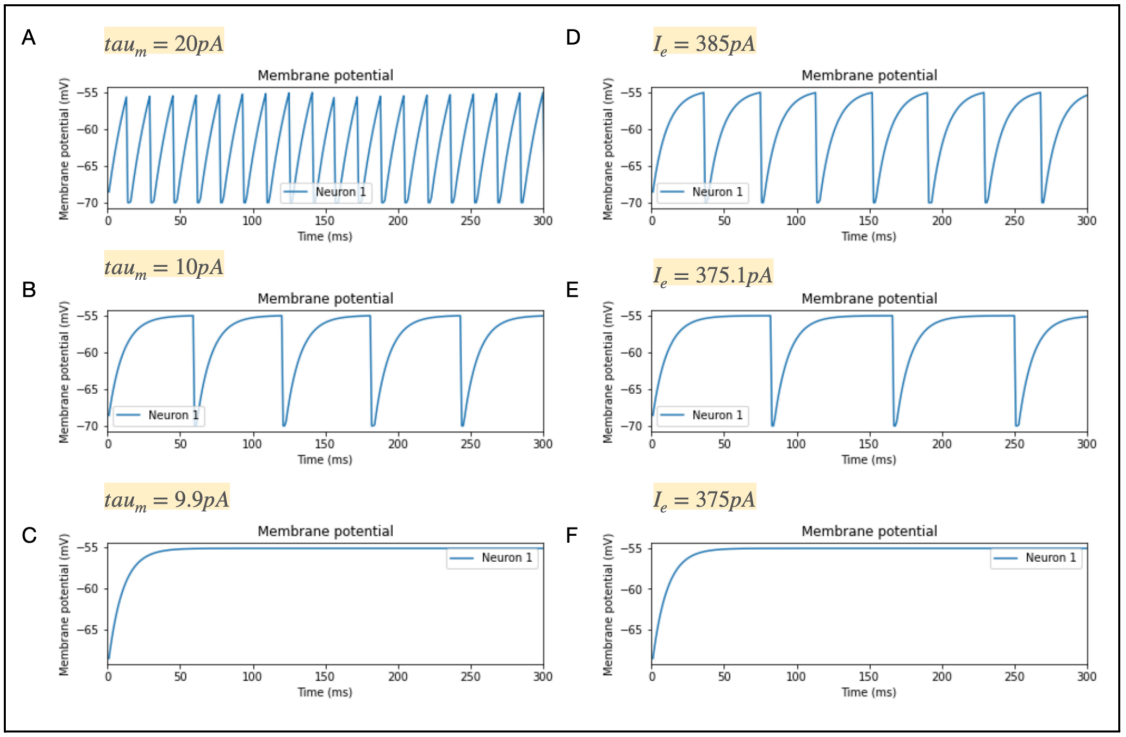


Figure 9: Voltmeter recording of neuron membrane potential at different membrane time constant (τ_m) and input current (I_e). The smaller τ_m, the less often a neuron will fire. The higher I_e, the more often a neuron will fire.

Now, let focus on the synaptic input current component of our model, I_{syn}. The individual postsynaptic currents (PSCs) are given by:

$$i_{syn,X}(t) = \frac{e}{\tau_{syn,X}} t e^{-\frac{t}{\tau_{syn,X}}} \Theta(t)$$

Where X represents either excitatory ($X = ex$) or inhibitory ($X = in$) presynaptic neurons and $\Theta(t)$ is the Heaviside step function (i.e., $\Theta(t)$ is 0 if $t < 0$ and 1 if $t \geq 0$).

As a result, the total charge transferred by a single postsynaptic current (PSC) is determined by the synaptic time constant according to:

$$q = \int_0^\infty i_{syn,X}(t)dt = e\tau_{syn,X}$$

It can be visualized in a more concrete way, by looking at the area under the curve represented in Figure 5. This curve shape can be modulated through the variation of the rise time of the excitatory synaptic alpha function (τ_{syn}). The higher τ_{syn}, the slower the curve of the input current to reach its maximum. So, for a higher τ_{syn} the transmission of the input current will be longer in time and the amount of current received by the neuron will be greater (see Figure 10). These factors will significantly influence the neuron's behavior in two ways. Firstly, a higher synaptic efficacy will result in more input charge accumulating on its membrane. Secondly, if the efficacy is sufficient to sustain the current transmission for longer than the neuron's refractory period, the neuron will have a chance to partially benefit from the incoming spikes after its recovery period. This has a profound impact on the firing behavior of our neuron, especially in the context of a simple two-neuron model, as we will explore later.

When multiple spikes arrive over time, the model initially scales each input current by its respective weight and then accumulates them over time following :

$$I_{syn} = \sum_{j} w_j \sum_{k} i_{syn,} X(t - t_j^k - d_j)$$

Where j indexes either excitatory ($X = ex$) or inhibitory ($X = in$) presynaptic neurons, k indexes the spike times of neuron j, and is the delay from neuron.

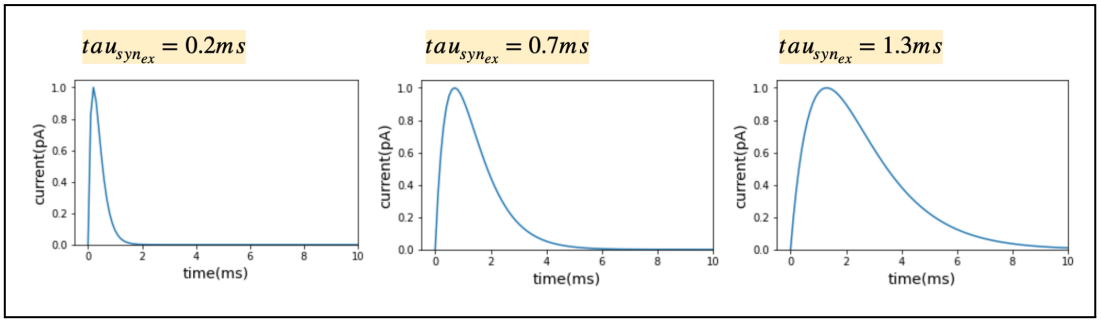


Figure 10: Graphical representation of the transfer of charge Induced by postsynaptic current for different values of $tau_{syn_{ex}}$.

As an example, we can examine the recorded neuron membrane potential over time in two simple examples of a spiking neural network.

First, we have a single neuron which receives an artificial spike train at several different rates and weights as presented in figure 11, A. We can notice that the higher the rate of the input spike train is, the higher will be the firing rate of the neuron: Compare the phase the beginning, where the neuron is freely firing at its base rate, with the two phases where the neuron is exposed to incoming spike train (red and green spikes and highlighted regions in the plot in Figure 11). We can see that in the last phase (blue spikes and region in Figure 11), as the neuron is subject to inhibitory incoming spikes, this discharges the neuron over time, causing a decrease in his firing rate.

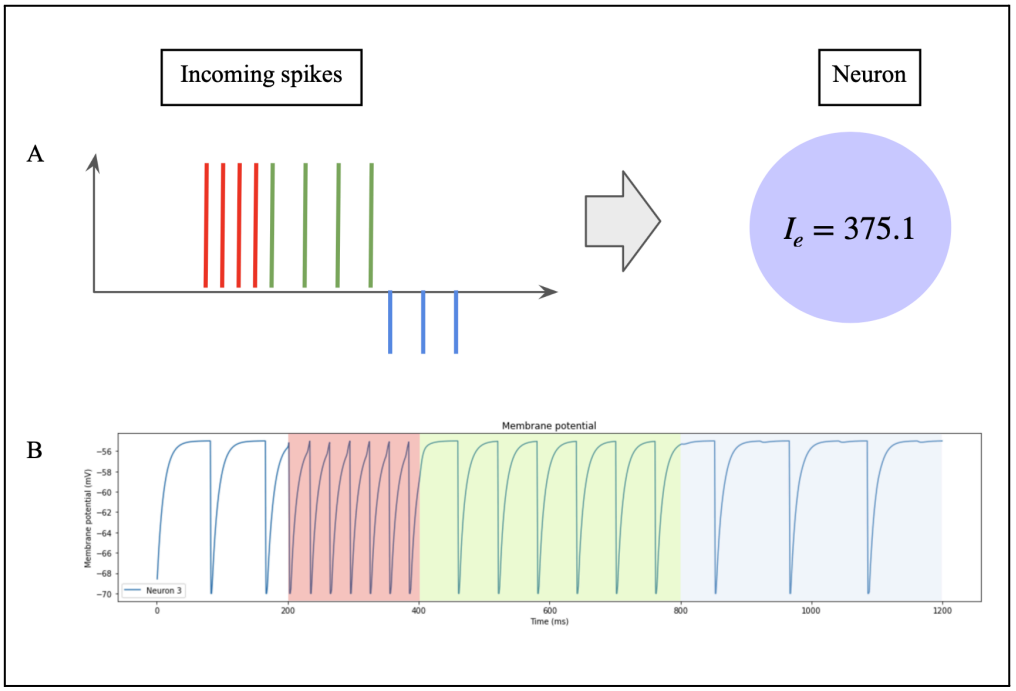


Figure 11: Response of a single neuron to artificial spike train with varying rate and weight. With (A) diagram of the input spike train and the spiking neuron; and (B) a voltmeter recording of the membrane potential of the neuron.

In the second example, we have two neurons with different firing rates (see Figure 12 A). The first neuron is connected as input to the second one, influencing its behavior. We can see that varying the weight of the connection between the neurons influences the behavior of the second neuron. In the case of a connection with a weight of 200, the second neuron synchronizes to the first neuron's firing pattern (see Figure 12 B). In the case of a connection weight of 800 however, the second neuron starts to produce an independent and irregular pattern of firing (see Figure 12 C). This is an interesting finding to explore further in the later experiments with SNNs. We can also observe that the rising time of the excitatory synaptic

alpha function plays a crucial role in the resulting behavior of the second neuron, creating an irregular firing behavior even for a connection weight of 200 (see Figure 12 D). This can be explained by the phenomenon described in the earlier paragraph. As such, this rising time parameter is also of great interest for later exploration.

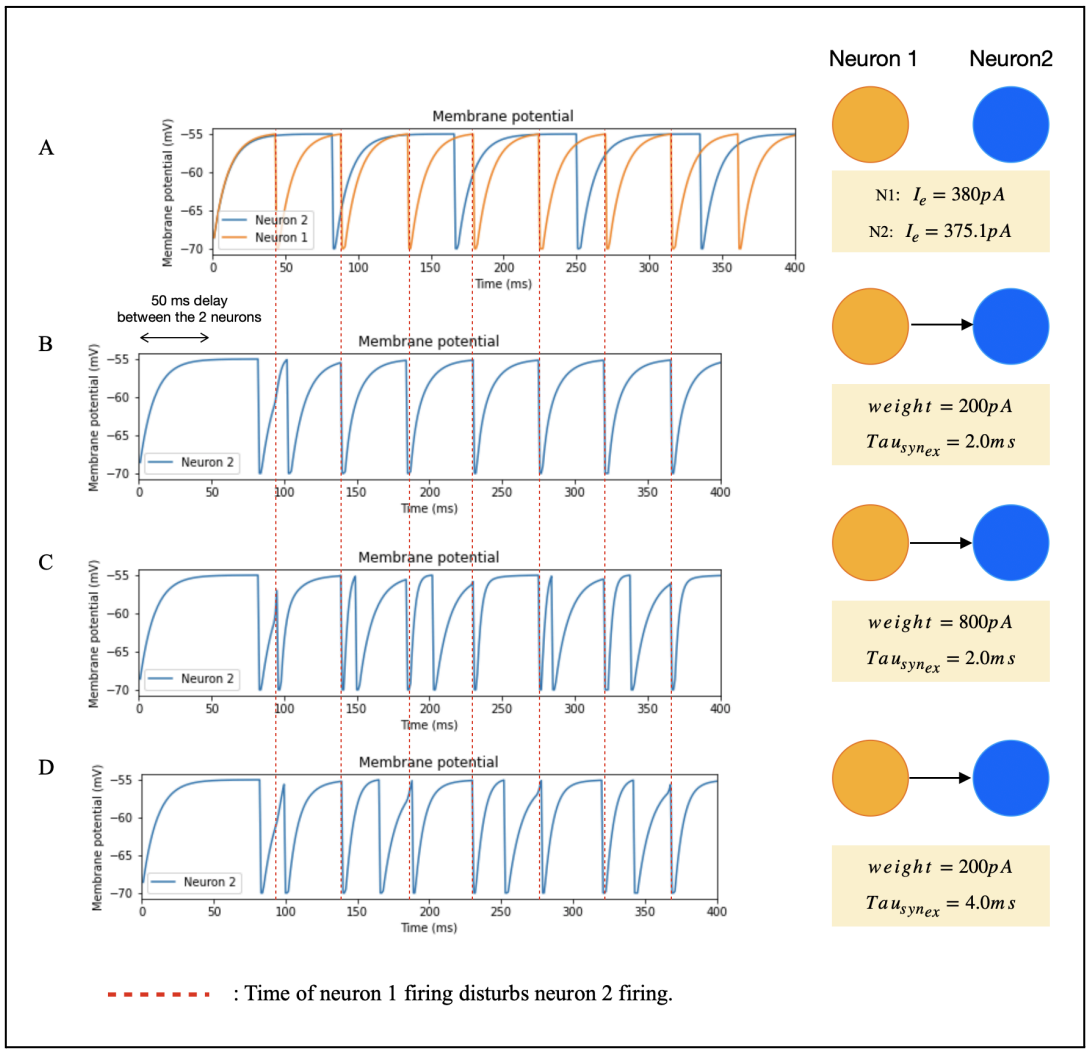


Figure 12: Simulations investigating the influence of synaptic weight and rising time on neuronal firing behavior in a two-neuron configuration show that both parameters strongly influence the system's behavior. With respectively: (A) Voltmeter recording of the membrane potential of the neuron 1 and 2. The two neurons are independent. Red lines represent the firing time of the neuron 1. (B) Voltmeter recording of the membrane potential of the neuron 2. Neuron 2 aligns with Neuron 1 firing pattern at a connection weight of 200 and tau_syn of 2.0. (C) Voltmeter recording of the membrane potential of the neuron 2. Neuron 2 exhibits independent and irregular firing at a connection weight of 800 and tau_syn of 2.0. (D) Voltmeter recording of the membrane potential of the neuron 2. Neuron 2 exhibits independent and irregular firing at a connection weight of 200 and tau_syn of 4.0

In the spiking neural networks implemented in NEST, it is possible, using NEST's *spike_recorder* functionality, to record the spiking time of the different components over time. These spiking times form a temporal sequence of events, from which we will extract the IOIs sequence. Based on this IOI sequence we then explore the presence of integer ratios. The model also allows us to provide specific spike trains to our neurons with defined time-domain characteristics (see Chapter 4).

- **Parameters of interest and their relevance**

During our exploration and implementation of the spiking neural network models, we identified four parameters worth investigating in this experiment. As mentioned earlier, the connection weight and the rise time of the excitatory synaptic alpha function appear to be promising candidates for modifying the intrinsic isochronous behavior of a neuron. Therefore, we will investigate these two parameters. Additionally, we are interested in studying the fundamental frequency of the neuron to determine whether slight variations in this parameter between otherwise identical neurons are sufficient to account for the generation of diverse integer ratios.

3.4 Cricket model

Figure 13: Photograph of a cricket of the species *Mecopoda elongata* (species *L*)[3], a species from the same genus as and morphologically similar to the *Mecopoda*, species *S* studied by Sismondo (1990) and modeled in the current section.

[3] Image source: Len Worthington, https://commons.wikimedia.org/wiki/File:Mecopoda_elongata_(Walker,_1871)_Large_Dry_Leaf_Katydid_80_mm_(18727981610).jpg

- **General explanation**

Our third and final model of interest will be based on the acoustic behavior of a *Mecodopa* species *S*, a Tropical Katydid. The species's rhythmic behavior has been studied and modeled in a study by Sismondo, (1990)which will serve as the basis for our further analysis of integer ratios in this model of cricket chirping.

Several kinds of insect tend to produce rhythmic signals that are synchronous or regularly alternating with those of conspecific individuals. The genus Mecopoda is one of them. It is a good candidate for a general model of insect synchronization behavior. Even more interesting for us, in the experimental and modeling results reported by Sismondo, (1990) the crickets seems to demonstrate integer ratios between certain intervals.

To obtain a model of this behavior, artificially generated chirps were played with different phases and intensities to disturb the empirical chirping behavior of one Mecodopa cricket. Every 8 to 12 natural chirps of the cricket, an artificial chirp was emitted, randomly shifted with the one of the cricket. Then, the period of time from the last unperturbed chirp until the next one was measured and compared to the averaged time of its three last unperturbed chirps. From this the study was able to extract the phase-response curve of the cricket hearing an external stimulus with variable period (see Figure 14).

The study defines the phase of the stimuli ϕ wrt. the last unperturbed chirp as

$$\phi = [t(s) - t(0)] / T_0$$

and the normalized perturbed period of the cricket chirp T, as a function of the phase between the chirp and the stimulus, is given by

$$T = [t(p) - t(0)] / T_0$$

In these two formulas, $t(s)$ is the time of the stimulus, $t(0)$ the time of the last chirp, and $t(p)$ the time of the next chirp. T_0 represent the intrinsic period of the cricket taken in the experiment as the average last three unperturbed periods.

The horizontal central line on the graphical representation of the PRC curve represents the intrinsic period of the cricket. We can see that two branches emerge symmetrically from this central line, with a discontinuity on the horizontal axis. We could describe this representation as if when the stimuli arrive close by after his chirp, the cricket thinks it was in advance and will extend its next chirp period (left branch). Conversely, if the stimuli arrive close to its next chirp, the cricket thinks it is late, and he will shorten his next period (right branch). Finally,

the discontinuity represents the limit point between his two adaptation strategies: reducing or prolonging his next period to be on time with the next external stimuli.

In a second part of the experiment (Sismondo, 1990), the cricket was not exposed to only one shifted stimuli, but to a continuous isochronous stimulus from which he gradually varied the period from 500 ms to 6s. The ratio between the normalized period of the stimulus over the period of the cricket chirps was calculated and results showed that the cricket is able to show a stable phase-locked response to this stimulation. Going from one chirp every 5 stimuli (5:1 ratio) to at least 2 chirps for one stimulus (1:2 ratio), passing among others the ratios 3:1, 2:1, 3:2, 1:1, and 2:3. All of these are integer ratios, and are thus relevant to our research question. The graphical representation of this phase-locked behavior can be observed in Figure 14 B.

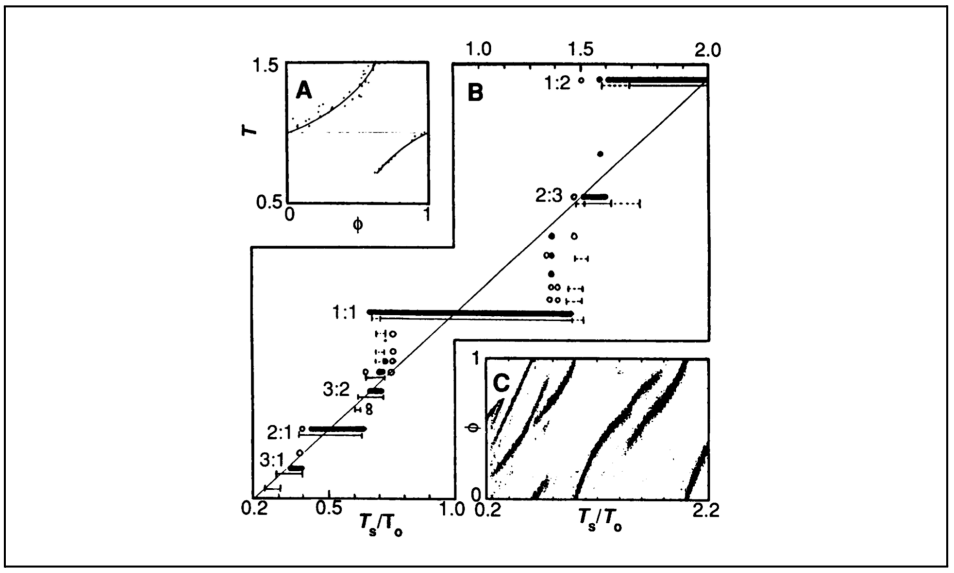


Figure 14: Graphical representation from Sismondo, (1990). (A) Phase response curves (PRC) representing the adaptation length of the cricket chirping period depends on the phase between its last chirp and the next stimuli. The dotes represent the experimental data and the dark line the PRC modeling from Sismondo, (1990). (B) Plot of Cantor function showing phase-locked modes of the cricket chirps in entrainment to artificial stimuli from Sismondo, (1990). (C) Rate-phase plot showing the phase-locked modes of the cricket chirps in entrainment to artificial stimuli from Sismondo, (1990).

In summary, these experiments on the Mecopoda crickets revealed intricate synchronization behaviors with artificial stimuli at integer ratios. It also provides valuable insights into the mechanism of this behavior that enable us to develop a computational model of this behavior for our own experiments. In addition, the author also introduced a mathematical analysis of a

modeled cricket, in order to further investigate the crickets' behavior. We interpret the succinct description of the model and implement the mechanisms in our own, new computer simulation.

- **Rationale for exploring this model**

The Mecopoda crickets are a biological organisms that produces phase-locked chirps which display integer ratios. The behavior is representative of general insect synchronization patterns, providing a robust foundation for modeling. Using a biological model ensures that our models maintain a degree of ecological validity. It can potentially offer insights applicable to other species or biological contexts. In addition, its complex behavior can be reduced into a simple computational model.

- **Detailed model setup and dynamics**

We first programmed a function in Python that fitted the graphical representation of the PRC curve provided by Sismondo, (1990). We added a parameter to the function that enables us to control for the noise present in the PRC curve (see Appendix A).

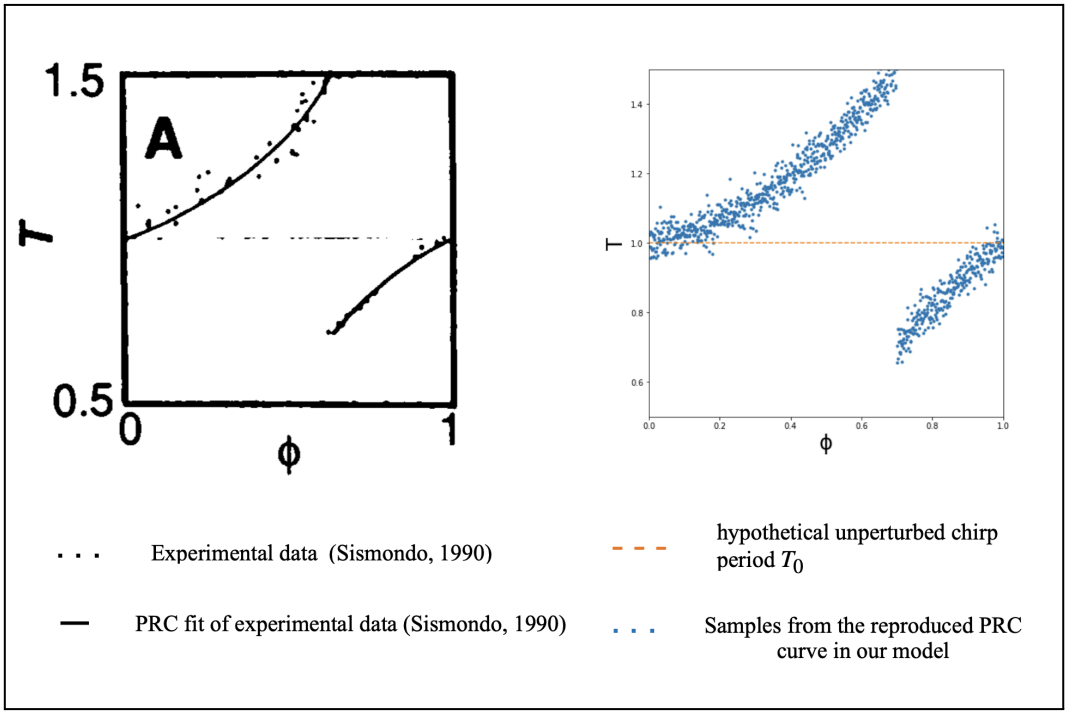


Figure 15: Phase response curves (PRC) representing the adaptation length of the cricket chirping period depends on the phase between its last chirp and the next stimuli. The left plot presents the PRC

curve fitted by Sismondo (1990). The right plot shows our reproduction of this PRC curve, visually matching the left plot.

From this PRC curve reproduction, we are able to obtain the length perturbed chirp period of the cricket for any phase value between the last undisturbed chirp of the cricket and the external stimulus.

We implemented a second function allowing us to compute, over time, the variation of the adapted cricket chirping period exposed to a sequence of external stimuli. In Sismondo, (1990), detailed modeling and calculations regarding the adaptation of a cricket's chirping period in response to stimuli are provided only for situations where the cricket hears a single stimulus with a certain phase shift relative to its last chirp. However, there is no detailed data on how the chirping period adapts when the cricket encounters continuous stimuli presented at an isochronous rate, such as when it hears two stimuli before its next chirp. In our model, if the cricket hears a stimulus, it adjusts its chirping period accordingly. If a second stimulus occurs before it can chirp again, it "forgets" the previous adjustment and re-adapts based on the new stimulus. Thus, the cricket responds only to the most recent stimulus.

Running a computer simulation of this implementation gives us a temporal series of the chirping events of the cricket when it is exposed to specific stimuli (see Figure 16). This function also computes a time series corresponding to the external stimuli events (see Figure 16). From this time series, we can extract IOIs sequences, and then calculate the ratio by dividing the period of the isochronous stimulus by the cricket's IOIs (to compare our results to the previous experiment; (Sismondo, 1990)), or using the formula from (Roeske et al., 2020) presented in Chapter 1 (and explained in more detail in Chapter 4).

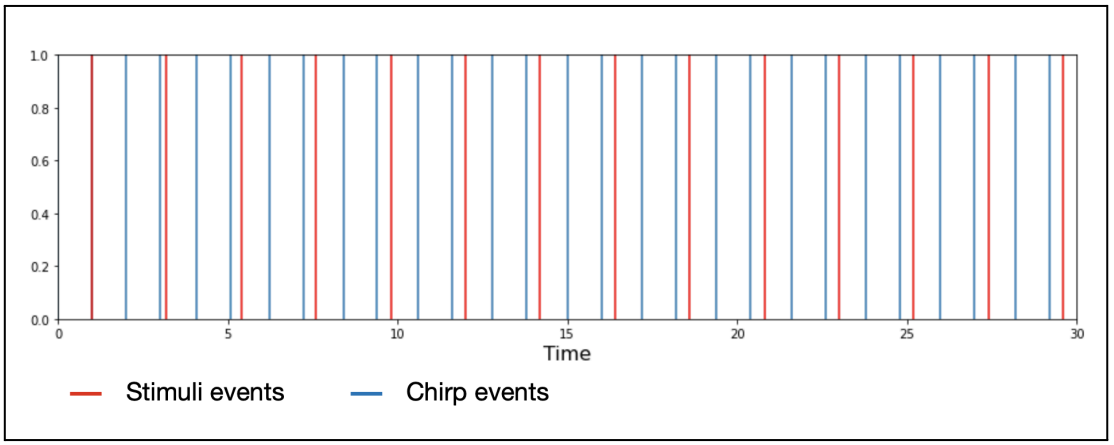


Figure 16: Example of resulting temporal sequence stimuli and cricket chirps obtained by a simulation of our model. Red lines represent the times at which a stimuli event occurs. Blu lines represent the time at which a chirp event occurs.

To validate our implementation of the cricket against previous results (Sismondo, 1990), we simulated the behavior of the cricket using our two functions for the same range of stimuli period (from 0.2 to 2.2 times the cricket's intrinsic period, as presented in Figure 17.). We visualized our results in the same way for comparison (see Figure 17 B), and observed some differences with the results presented in the original research (see Figure 17 A). For a high stimuli period, our simulation showed an oscillation of the cricket between 2 interval periods. Further investigation of the model showed that this difference stems from the fact that our artificial cricket instantaneously returns to its fundamental frequency when not being perturbed again. Likely, it takes the actual crickets longer before returning to their intrinsic period (Sismondo, 1990). To overcome this issue, we add a "relaxation" parameter to our model, which we can adjust to make our simulated cricket return faster or slower to its intrinsic period when a chirp interval is not perturbed by a stimulus. A relaxation rate of 1, corresponds to the cricket returning instantaneously to its natural chirping period after a perturbed period. A relaxation rate of 0.2 corresponds to the cricket, after being perturbed by an external stimulus, adapting its period at a rate of 20% to get closer to its natural period at each of its chirps. We simulated the experiment again with different relaxation rate and found a good fit with the original results for a relaxation rate that diminishes by 20% the perturb period of the cricket returning to his fundamental period when not being perturbed (see Figure 17 C).

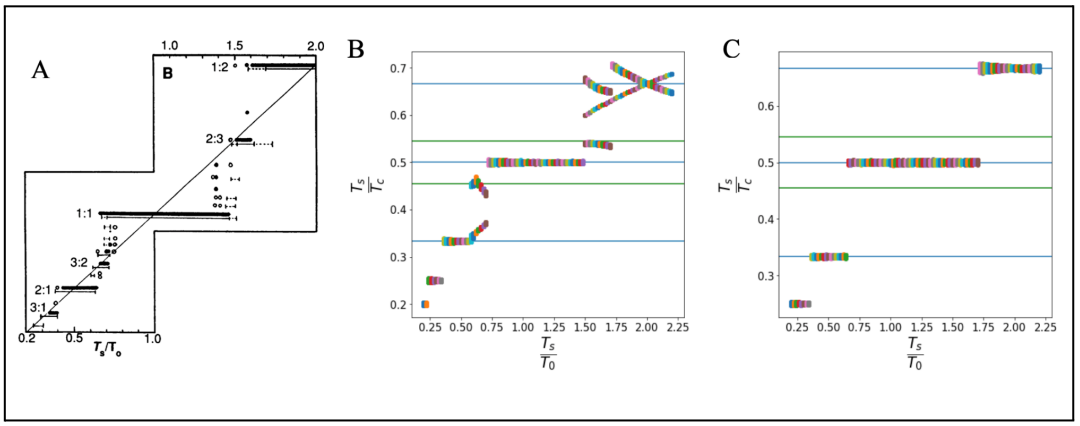


Figure 17: Reproduction of the plot of cantor function from Sismondo, (1990), showing phase-locked modes in entrainment to artificial stimuli. (A) Plot of Cantor function showing phase-locked modes in

entrainment to artificial stimuli from Sismondo, (1990). (B) Results obtain with our simulated cricket without relaxation rate. (C) Results obtain with our simulated cricket presenting a relaxation rate of 20%

Finally, in the experiment by (Sismondo, 1990), the behavior of the cricket was only observed when subject to an isochronous stimulus. However, the way our model is implemented will allow us to give to the cricket any kind of time series as stimulus and simulate the resulting behavior.

- **Parameters of interest and their relevance**

All the parameters of the PRC curve could be varied, but we will keep them fixed in our experimental simulations. Our first parameter of interest will be the period of the external stimuli. As we saw with the experiment conducted by Sismondo, the kind of integer ratio produced emerges from the relation between the stimuli period and the cricket chirp. Our second parameter of interest will be the relaxation rate of the cricket. As we saw during our own attempt to reproduce the previous results with our computational model, the production of integer ratio also depends on the relaxation rate of the cricket.

Chapter 4: Methods

To investigate the behavior of the various models we described in the previous section, we analyze the temporal sequences they receive as input and the ones they produce as output. There are two special types of time event sequences used in the simulations: isochronous sequences and sequences generated by a Poisson process.

In the subsequent sections, we outline the calculation of inter-onset intervals (IOIs) from the time event sequences and the methodologies for calculating and analyzing ratio sequences. We then describe how we compare the production of ratios and integer ratios between the models. Finally, we list all the simulations that we perform with the models, including detailed configurations and key parameters of interest, including the specific assumptions relevant to each simulation.

4.1 Sequences of events

Two of our models of interest function based on the input of a time event sequence. In this thesis, we use two types of time event sequences as inputs: isochronous and exponentially distributed.

4.1.1 Isochronous sequences

An isochronous sequence corresponds to events that are equally distributed in time. These types of sequences represent a highly regular rhythm and are common in music and other rhythmic phenomena. For example, a metronome produce an isochronous sequence, or the underlying beat of music is typically isochronous. All rhythm ratios calculated from the IOIs of a perfectly isochronous sequence are exactly 0.5. In other words, for an isochronous sequence the ratio distribution has a single, very sharp peak right corresponding to a 1:1 integer ratio.

Isochronous sequences are use to stimulate crickets (Sismondo, 1990), with periods ranging from 0.2 to 2.2 seconds. Our simulations of the cricket model use isochronous stimuli within the same range.

In the spiking neural model, a neuron with a continuous internal current input, not influenced by any external factors, exhibits an isochronous firing pattern. Therefore, simulations where one neuron serves as the input to another correspond to an isochronous stimulation. The lower limit of input current that allows a neuron to fire is 376pA (see Chapter 3, Section 3.3),

corresponding to an isochronous firing frequency of 117 Hz. The maximum average firing rate of a neuron, constrained by the absolute refractory period of this model of neuron (2ms), is 475 Hz (i.e., 475 spikes per second; see Chapter 3, Section 3.3).

4.1.2 Poisson sequences

A Poisson sequence is a random sequence generated by a Poisson process, which represents random events in time whose IOIs are exponentially distributed. In a Poisson process, events occur independently at a constant average rate. The probability of an event occurring in a given time interval only depends on this rate, not on previous events. In contrast to the highly structured isochronous sequence, a Poisson sequence's events are randomly spaced in time, so a Poisson sequence is a highly random temporal sequence. Through computational simulations of this kind of time sequence, we found that the resulting ratio density distribution was equal to a uniform distribution (Jadoul et al., in prep.). In other words, for a large enough number of sampled intervals, a Poisson sequence have a completely flat distribution without any integer ratio peaks.

Poisson processes also describe highly random temporal processes in nature, such as the exponential radioactive decay.

We use this type of input time series in our cricket and spiking neuron models as a baseline sequence that represents random input. Since the ratio distribution of a Poisson process is uniform, this approach allows us to observe and measure how our models introduce regularity and transform a random stimulus input into a ratio density that clusters around integer ratios.

4.2 Sequences of IOIs

4.2.1 Time event reference for the extraction of IOIs

Except for the Markov chains, our models produce sequences of events in time. As we are interested in integer ratios between intervals, these event times need to be converted into IOI sequences for our further analyses. By doing so, we obtain a measure of the intervals between consecutive events, which allows us to calculate the proportional relationships between event durations.

The first step involves selecting which events to considere for thee construction of the IOI sequences. For both models, we can either consider only the events of the output unit of the model (the spiking times of neuron 2 or the cricket's chirps) or the combination of events produced by each component of the model (the spiking times of both neuron 1 and 2, of the Poisson generator and neuron, or the times of the external stimuli and cricket chirps). The former can be referred to as an *output event sequence*, and the latter as a *merged event sequence*. In this work, we focus on the output event sequence, but in the discussion, we explore the valuable insights that further research incorporating merged event sequences could provide.

4.2.2 Calculation of IOIs

In the case of the Markov chain, the IOIs are directly given by the interval selected in the normal distribution of the state.
In the case of the two other models, the IOI sequence is calculated as followed:

$$IOI_i = t_{i+1} - t_i$$

In this formula, t_i is the time at which event i occurs in the temporal sequence.

4.2.3 Control of the length of the sequence

To ensure enough data points to analyze a simulation, we need to make sure that our sequences of IOIs reach a minimum length. We set a minimum length of 100 IOIs for our sequences. This threshold was chosen to provide a sufficiently large sample size to capture a representative density function of the ratio distribution, allowing for accurate analysis of the ratios produced by the network.
For the Markov chain model, we can control the length of the IOI sequences by setting the desired number of steps in our chain. In a similar way, we can set the desired number of chirps of the cricket model.
In the spiking neural network, we cannot directly control the length of the IOI sequences by specifying the number of spikes a neuron should emit before stopping the simulation. Instead, we have control over the duration of the simulation. A simulation of 10s ensures that even at his lowest firing rate, a neuron fire more than 100 times in a simulation.

4.3 Sequences of ratios

4.3.1 Single ratios sequence

a. Ratio calculation

The ratio sequence is calculated using the formula by (Roeske et al., 2020) presented in chapter 2:

$$r_i = \frac{IOI_i}{IOI_i + IOI_{i+1}}$$

Here, IOI_i is the *i*th interval as described previously.

b. Calculation of the ratio density functions

Once the ratio sequence is calculated, we approximate and visualize a continuous probability distribution based on these ratios. To do so, we use a kernel density estimation (KDE) to estimate the probability density function (PDF) of the observed ratios in a non-parametric way with an automatic bandwidth determination. We calculate it using *gaussian_kde* from the *scipy.stats* Python package (with the default bandwidth selection, following Scott's rule). In addition, we use the *find_peaks* function from *scipy.signa*l Python package to know the number of peaks present in the KDE and the value of the ratio corresponding to the peaks. Following Roeske et al. (2020), we use a minimum prominence threshold of 0.07 to minimize spurious detection of peaks. This informs us on the number of cluster ratios present in the distribution and their localization.

Based on this KDE, we can also calculate the cumulative density function (CDF), for some precise parameters that require a complementary look. It gets calculated by numerically integrating the KDE over the desired range of values. By observing the shape of the CDF, we can appreciate the concentration of ratios around integer ratios: For example, a sharp rise indicates a higher concentration of ratios around a specific value, whereas a flat part of the CDF indicates a low density of values in that region.

c. Ratios sequence weighted count on and off-integer ratios

Using the method presented in Chapter 1 (Roeske et al., 2020), we calculate the weighted count of on-integer and off-integer ratios for all sequences.

This weighted count is use to estimate the amount of ratio around the perfect value of the integer ratio by dividing the weighted on-integer count by the sum of the weighted on- and

off-integer counts. The closer to 1 the result of the division is, the more concentrated will be the probability density function on the integer ratio of interest.
In the case of the Markov chain model, we apply the complete statistical analysis, as used in the literature (Roeske et al., 2020) and as presented in chapter 1, to evaluate the significance of a ratio cluster around integer ratios. This allows us to see what is the largest spread that the normal distributions can have while still producing a significant proportion of integer ratio according to this method.

d. Differential entropy

Differential entropy is the generalization of (Shannon) entropy for a continuous probability distribution. Just like entropy in the case of a discrete probability distribution, continuous entropy quantifies the level of uncertainty or disorganization in the data. A high value of entropy, corresponds to a less organized distribution, i.e. with high uncertainty. Conversely, a low entropy value indicates a more organized distribution, with a higher predictability. In our case, since the rhythm ratio is always between 0 and 1, the maximum entropy corresponds to a uniform distribution and is equal to 0, and the more a distribution is organized, the more negative its differential entropy is. For example, the distribution resulting from a Poisson process have a high entropy, close to 0, whereas a distribution with one or a few very sharp peaks have a very negative differential entropy value.
In our analyses, we calculate the differential entropy of our ratio sequence using the *differential_entropy* function from the *scipy.stats* python package.

4.3.2 Exploration of ratio sequences production per a couple of parameters

In order to explore the behavior of the different models and what effect their parameter values have on the presence of integer ratios in the produced temporal sequences, we compute the measures described above and display them in heatmaps. In each such heatmap, we can explore all combinations of a range of values for two of a model's parameters. The resulting visualization presents us with an overview of the models' global behavior and whether and how their output is affected by the different parameters.

We compute a heatmap of the entropy produced by each couple of parameters. This allows us to quantitatively assess the level of organization or uncertainty in the ratio sequences produced by the full range of combinations of two different parameters. If a certain combination of

parameters produce a very low entropy score, the square of the heatmap corresponding to the tested parameter pair is white.

We compute a heatmap of the number of peaks produced by each couple of parameters. This allows us to perform a visual analysis of the number of clusters in the ratio distributions produced by the models for the interaction of two parameters. This heatmap is complementary to the one showing entropy, as a low entropy can be the result of several sharp peaks or of a single slightly broader one. Additionally it can also provide a general overview of the rhythm ratio clusters, even if they are not detected around the selected ratios of interest (i.e., 1:1, 1:2, etc.; see below).

We compute a heatmap of the fraction of on-integer ratios for each integer ratio of interest (1:1, 1:2, 1:3, 2:1, 2:3, 3:1, 3:2). This allows us to perform a visual analysis of the areas under which the pairs of parameters produce integer ratios. If a certain combination of parameters does not produce any rhythm ratios around our ratio of interest, the square of the heatmap corresponding to the tested parameter pair is left blank. If an integer ratio is present, the corresponding area of the heat map will be colored according to how many of the ratios are close to the ratio of interest. The more concentrated the distribution is around the integer ratio, the closer to 1 it is the fraction of on-integer ratio and the more intense the color. This heatmap enables us to evaluate which parameter setups allow the production of integer ratios by the model, and determine how narrow the range of parameters is for producing the different kinds of integers.

We compute a heatmap of the statistical significance of on-integer ratios for each integer ratio of interest (1:1, 1:2, 1:3, 2:1, 2:3, 3:1, 3:2). This allows us to perform a visual analysis of the areas under which the pairs of parameters produce integer ratio clusters which can be identified as statistically significant. If a certain combination of parameters does not produce a statistically significant ratio cluster around our ratio of interest, the square of the heatmap corresponding to the tested parameter pair is left blank. If it is significant, the corresponding area of the heat map it is colored in green with an intensity matching the magnitude of the p_value (i.e. dark green: $p < 0.001$, medium green: $p < 0.01$, light green: $p < 0.05$). This heatmap enables us to evaluate which parameter setups allow the production of a statistically significant cluster of integer ratios by the model, and determine how narrow the range of parameters is for producing the different kinds of integers.

Finally, we computed a heatmap of the differential entropy produced by each parameter combination. This heatmap allows for a visual analysis of the level of organization or uncertainty in the ratio distributions produced by the model. The entropy heatmap high entropy indicates a less organized distribution with greater uncertainty, while low entropy signifies a more structured distribution. The entropy heatmap provides an overall measure of how tightly the data are organized. This offers a broader perspective on the distribution of rhythm ratios, even when the clusters do not perfectly align with the ratios of interest.

4.3.3 Comparison of input-output convergence towards integer ratios

As the Poisson sequences result in a uniform ratio distribution, the simulations including the use of this kind of sequence allow us to examine the intrinsic tendency of our model to produce integer ratio. Put differently, since a Poisson sequence as input to these models does not have any integer ratios, this analysis provides insight into how much each model generates structured output sequences.

4.3.4 Comparison of the production of integer ratios between models

Apart from the Markov chain model, which has its own specific assumptions and testing methods, we aim to perform simulations for the cricket and spiking neural network models that are as similar as possible. This approach aims to facilitate a direct comparison between the two models. Both the cricket and neuron models consist of two units: an input unit and an output unit, where the input unit influences the behavior of the output unit. We calculate the ratio sequences generated by the output units in each case. This conceptual similarity between the two types of models, allow us to set up the models and analyze their resulting sequences in a consistent manner. This way, we can easily compare the results and graphs obtained for both models, as well as their convergence towards integer ratios when using Poisson sequences as input.

4.4 Simulations

For each model, we conduct several explorations of the parameter space, following the principle outlined above. Here, we list the details of the simulations conducted for each model, including the configurations used, the parameters varied and in which range, and the specific assumptions made.

4.4.1 Markov chain with 2 states: Transition probabilities

In a first experiment, we run our Markov chain model with two states, Long and Short (S and L). Both states output intervals sampled from a normal distribution, respectively with means 1 and 2 and standard deviation 0.1 and 0.2. In our exploration of the parameter range, we vary the probabilities p and q, which jointly make up the transition matrix, from 0 to 1. Here, p is the transition probability from states S to L, and q is the transition probability from L to S. As such, the probability in state S to stay in S equals 1 - p, and the probability in L stay in state L is 1 - q. We pick 101 equally spaced values from 0 to 1, for both p and q, resulting in more than 10000 simulations with unique parameter combinations, which each generated 1000 intervals.
We expect to observe differences in the cluster of integer ratios produced by the model depending on the value of the transition probabilities. When p and q are close to 1, the Markov chain constantly jumps from state S to state L and back, leading to the production of mostly 1:2 and 2:1 ratios. Conversely, with p and q close to 0, the chain mainly stays in the same state, which should result in mainly 1:1 ratios being produced. If p is close to 0 and q close to 1, we mainly expect samples from state S and some transitions to state L leading to a high production of 1:1 ratio and a little production of 1:2 and 2:1. As such, the Markov chain is the most interpretable of the three models and therefore this experiment can help us identify the minimum and maximum values of transition probabilities needed for resulting integer ratio clusters which can be identified as statistically significant. In general, the simplicity of the model helps us to better understand the extracted measures, and prepares for later interpretation of the cricket and spiking neural network model results.

4.4.2 Markov chain with 2 states: Distributions' standard deviation

In the next experiment, we again use a Markov chain simulation with two states, S and L, generating normally distributed intervals with means respectively 1 and 2. However, we now fix both the probabilities p and q at 0.5. Instead, we vary the standard deviation of the Markov chain's states' distributions: The standard deviation of S, σ_S, vary logarithmically between 0.04 and 0.2, and that of L, σ_L, from 0.08 to 0.4. Both parameters take 100 unique values, again resulting in 10000 Markov chain simulations. This experiment aims to help us understand what is the maximum spread allowed in the normal distributions of each state to still obtain statistically significant integer ratio clusters on 1:1, 1:2, and 2:1. We expect that as the distribution becomes more dispersed, the proportion of on-integer ratios will decrease until the cluster is no longer identified as significant.

4.4.3 Spiking neural network with 2 neurons: Input currents

Next, we investigate a spiking neural network composed of two neurons. An input neuron firing at a regular rate is connected to a second neuron, which also has its own intrinsic regular firing rate. In this configuration, the spiking behavior of the input neuron affects the spiking behavior of the output neuron (see Chapter 3, Section 2). The connection weight between the two neurons will be fixed at +150 pA, a typical default value of the NEST simulator. The maximum firing rate of our neuron is 475 Hz with an input current of 37706 pA. However, we choose to lower this upper range to 550 pA, as it allows us to more clearly observe the interesting area of integer ratios produced by our model. Beyond 550 pA, up to 37706 pA, the model only generates isochronous firing patterns, without producing any other ratios. Therefore, we vary the input current of the two neurons from 376 pA to 550 pA, in 101 linear steps.

We expect that the higher the input neuron's firing rate compared to the one of the output neuron, the more it will perturb the output neuron's isochronous fundamental frequency. Consequently, we expect that a higher input rate will result in a higher chance of observing the presence of other ratios than isochrony.

4.4.4 Spiking neural network with 2 neurons: Connection weight and synaptic rise time

We then conduct a variant of the previous experiment, but the input current of the two neurons is now fixed at 376pA. Instead, we vary the connection weight between the 2 neurons conversely from 0pA to 10000 pA, in 101 linear steps. 10000pA may seem high, compared to the typical default weight of connections between two neurons in the NEST simulator, but it helps capture the full range of behavior of this model. We also vary the rising time of the input synaptic current of the output neuron from 1ms to 8ms, in 101 linear steps; this is an average range for a rising time of the input synaptic current of neurons.

We expect that as the connection weight between the two neurons increases, the output neuron's firing behavior will deviate from isochrony, potentially reducing the prevalence of isochronous patterns. This change may increase the possibility of observing other kinds of integer ratios. A similar effect is expected with changes to the rising time of the input synaptic current.

4.4.5 Spiking neural network with Poisson generator: Input current and Poisson rate

The final spiking neural network experiment uses a network composed of one Poisson generator and one neuron. The Poisson generator, which produces a Poisson sequence of excitatory input spikes, will be connected as input to the neuron. In our experiment, this neuron has a fixed intrinsic regular firing rate of 376pA. We vary the rate of the Poisson generator from 16 to 475 Hz, in 101 linear steps. The second varying parameter is the input current of the output neuron, again from 376 pA to 550 pA, in 101 linear steps.

We know that, given it is run for long enough, a Poisson generator produces sequences with a uniform distribution of rhythm ratios between adjacent intervals. As such, this experiment will demonstrate how the intrinsic properties of a spiking neuron, a model of a biologically basic mechanism, can create integer ratios in a signal that initially did not exhibit them.

4.4.6 Cricket model with isochronous stimuli: Relaxation rate and stimulus period

Finally, we also explore two versions of the cricket model. The first experiment with the cricket model consists of an isochronous stimulus and the modeled chirping behavior of a cricket when exposed to external stimuli, as explained in Chapter 3. The isochronous stimuli vary from 0.2 to 2.2 times the natural chirping period of the cricket (which is, for simplicity, set to be 1 unit long), as in the experiment by Sismondo (1990). We vary the rate of the stimulus in 101, linear steps. Additionally, we vary the relaxation rate of the cricket from 0.2 to 1, in 101, linear steps.

We expect that varying the period of the external isochronous stimuli will result in the production of isochronous sequences of chirp, as the cricket will lock his period based on the stimuli period. However, for some specific periods of stimuli, we may observe the emergence of two peaks in the rhythm ratio distribution if the cricket oscillates between two different periods of chirp to match with the stimuli. Additionally, we expect that decreasing the relaxation rate of the cricket will remove the potential oscillation between two periods and thus in those cases lead to sharper and fewer peaks in the rhythm ratio distributions.

4.4.7 Cricket model with Poisson stimuli: Relaxation rate and stimulus rate

Finally, we explore a variant of the cricket model where a Poisson sequence is used as stimulus for the cricket. The average period of the exponentially distributed stimuli again vary from 0.2

to 2.2 in 101, linear steps. The natural chirping period of the cricket is again taken as 1 unit, as before. As in the previous experiment, we vary the relaxation rate of the cricket from 0.2 to 1 in 101, linear steps.

We expect to find similar results to those of the previous experiment, but with less well-defined clusters on integer ratios due to the random nature of the external stimuli. Analogously to our rationale for introducing a Poisson generator in the spiking neural networks, this experiment will allow us to more closely assess how much regularity the cricket model introduces to a fully random Poisson input sequence.

Chapter 5: Results

We now present the results obtained from each simulation outlined in the previous chapter. The seven experiments explore the different models and their parameter ranges, and aim to provide insight into the models' overall behavior and the effect of the parameters.

5.1 Markov chain with 2 states: Transition probabilities

Simulations with varying transition probabilities between the two states revealed that different probabilities resulted in distinct peaks in the integer ratio distributions. Markov chains are reasonably simple models which can often be analyzed analytically. However, with this computational model our main interests are both to validate our analytic reasoning, and better understand the methods employed to analyze the ratio sequences which we later use for less simple models.

Differential entropy of rhythm ratio distribution

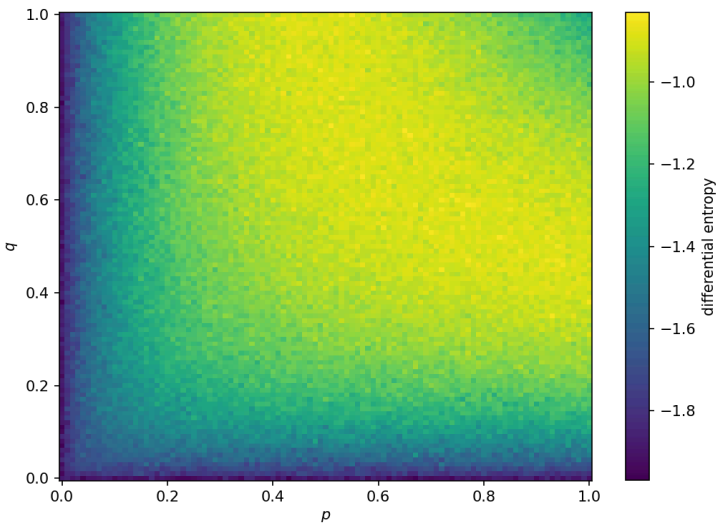


Figure 1: Heatmap of the estimated differential entropy of the distribution of rhythm ratios for varying transition probabilities of a 2-state Markov chain model. The heatmap shows the empirically estimated entropy for all combinations of 101 values of the p and q parameters of the Markov chain.

The entropy heatmap reveals that a reduced transition probability from one or both states leads to highly negative entropy values (ranging from -1.6 to -1.8 or lower). This decrease in entropy indicates a more structured outcome. This is expected because reducing the transition probability for one or both states causes the system to remain in the same state more often, making successive events more likely to be the same. This leads to a high likelihood of

isochronous rhythms and reduces the possibility of producing other ratios, such as 1:2 or 2:1. Conversely, increasing the transition probabilities results in a higher entropy, indicating a more uniform outcome. The model is more likely to switch between states, increasing the chances of achieving ratios 1:2 and 2:1 in addition to 1:1. Additionally, the upper right corner of the heatmap shows a slight reduction in entropy, indicating a return to a slightly more organized outcome. When the transition probabilities are very high, the likelihood of consecutive identical states becomes very low, eliminating the 1:1 ratio, increasing the concentration of ratios around 1:2 and 2:1 ratios, and reducing entropy. Generally speaking, higher transition probabilities increase the diversity of integer ratios produced by the model, with an optimal range before the disappearance of the 1:1 ratio due to excessively high transition rates.

Number of peaks in rhythm ratio distribution

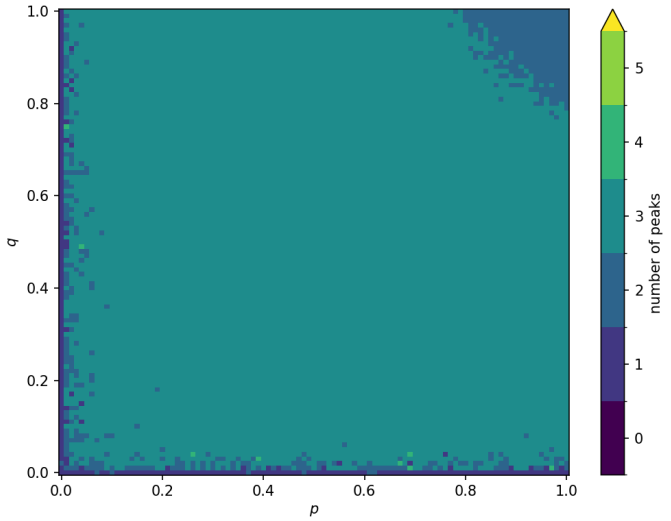


Figure 2: Heatmap of the number of peaks of the KDE distribution of the rhythm ratios for a 2-state Markov chain model. The heatmap shows the number of peaks for transition probabilities p and q both varying between 0 and 1 (101 steps).

The entropy results are mirrored in the heatmap of the number of peaks. At very low probabilities for state transitions, the number of peaks is one, indicating a single cluster (i.e., isochrony) which corresponds with low entropy. As the transition probabilities increase, the number of peaks rises, quickly reaching a large stable region of parameter combinations with three peaks. Indeed, slightly higher transition probabilities than 0 allows for different integer ratios than just 1:1 to appear (i.e., 1:1, 1:2, and 2:1 in our Markov chain); this corresponds to the increased entropy observed earlier. As transition probabilities become very high, the

number of peaks decreases, happening concurrently with the slight reduction in entropy observed previously. Indeed, in this case, high transition probabilities lead to the disappearance of the isochronous cluster: when p and q are close to 1, the Markov chain constantly jumps from state S to state L and back, leading to the production of mostly 1:2 and 2:1 ratios. So, at some point, the model is passing from a distribution of 3 ratios (1:1, 1:2, 2:1) to one with only 2 peaks (i.e., 1:2, 2:1), which also slightly reduces the total entropy of the model.

Fraction of on-integer ratios

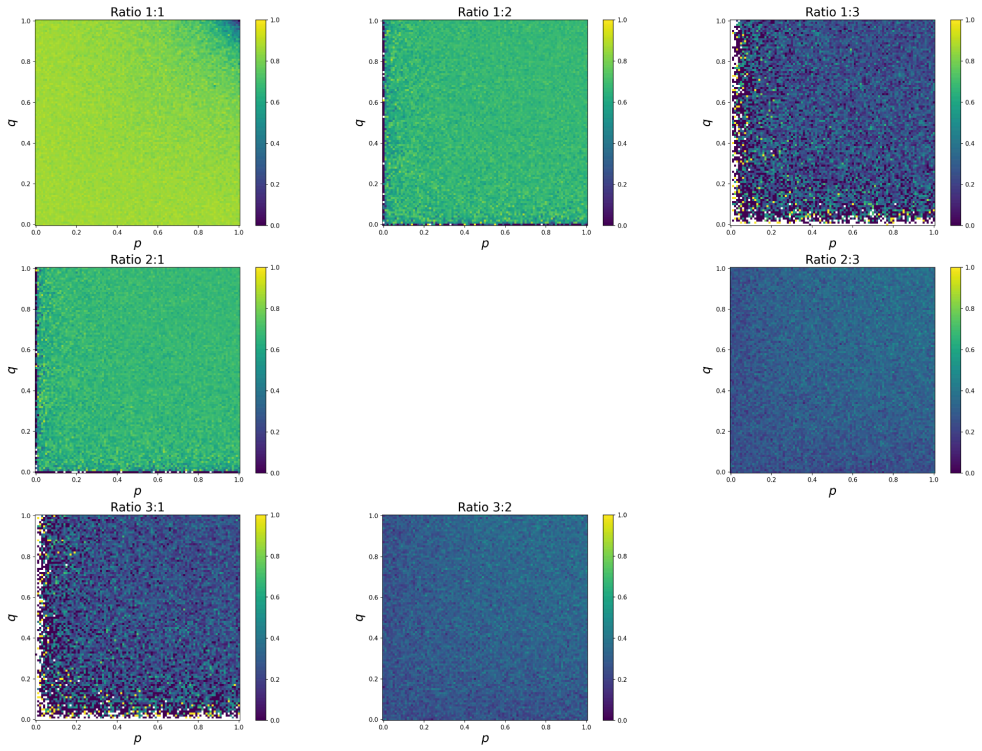


Figure 3: Heatmaps of the fraction of on-integer ratios in the sequences resulting from 2-state Markov chain model simulations. As before, transition probabilities vary between 0 and 1 (in 101 steps).

The detailed results for the fraction of on-integer ratios zoom in on 7 integer ratios of interest: 1:1, 1:2, 1:3, 2:3, 2:1, 3:1, and 3:2. The results show a high and sharp cluster of isochrony (1:1) across all tested parameters but a slight decrease for high transition probabilities (starting around p and $q > 0.8$). This aligns with previous reasoning and observations that very high transition probabilities make consecutive identical states very unlikely, drastically reducing the 1:1 cluster. Conversely, at very low transition probabilities, there is a very little

formation of 1:2 or 2:1 ratio clusters, but these clusters appear for the rest of the parameter pairs tested. Additionally, there is an absence of any ratios close to 3:1 and 1:3 at low transition probabilities, but a low presence of these ratios for the other probabilities tested. Finally, the same goes for ratios 3:2 and 2:3, but in a more consistent manner across probabilities. This can be explained by the way the Markov chains are set up, with a 1:2 ratio between the mean intervals of the S and L states: Any rhythm ratios close to 1:3 and 3:1, 2:3 and 3:2 are due to random fluctuations of the underlying normal distribution of the S and L states and extreme values getting sampled from these distributions. The 2:3 and 3:2 integer ratios are more consistent, because their rhythm ratios fall respectively between 1:2 and 1:1, and between 2:1 and 1:1. For this reason, intermediate rhythm ratios are more readily generated by the Markov chains.

Statistical significance of integer ratios

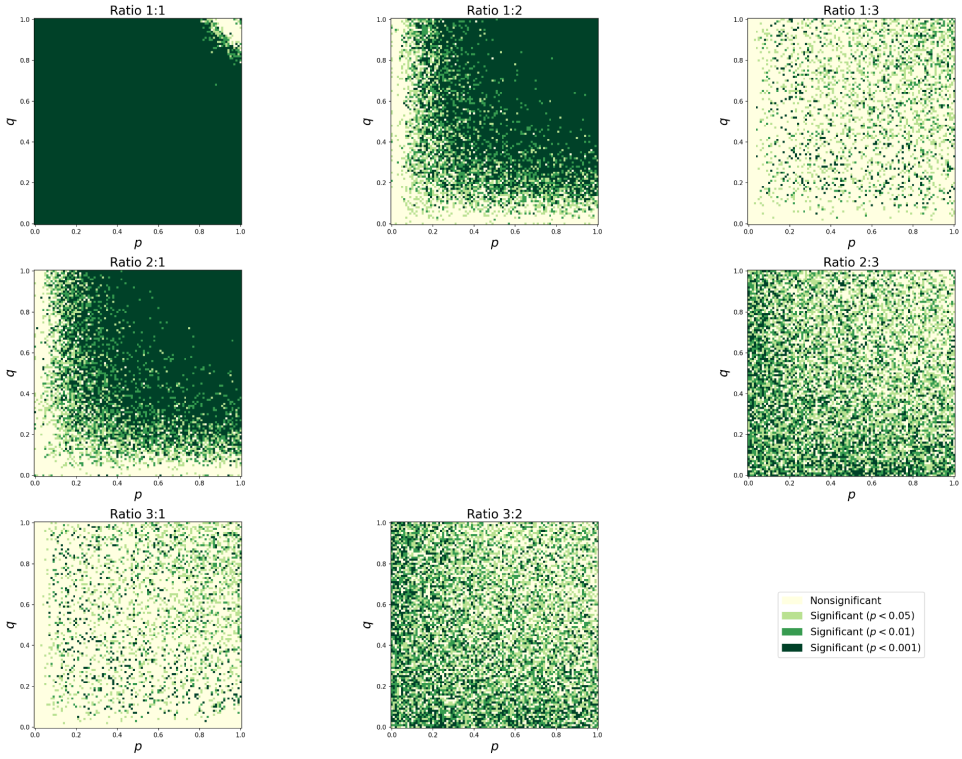


Figure 4: Heatmaps of the statistical significance of the presence of different integer ratios in simulations with varying transition probabilities for a 2-state Markov chain model. As before, transition probabilities vary between 0 and 1 (in 101 steps). Three different p values are considered $p < 0.05, p < 0.01, p < 0.001$.

The significance results across almost all tested ratios mirrored the fraction of on-integer results: The presence of 1:1 ratios is highly significant except for high p and q transition probabilities, and the presence of 1:2 and 2:1 becomes significant once p and q are both above approximately 0.1 and the Markov chain starts to transition often enough between the two states. Ratios 1:3 and 3:1 follow the results from the previous subsection, with sporadic significant values, following the random fluctuations of the data. However, for the 2:3 and 3:2 ratios, the number of significant simulations is much higher than overall expected. This can partially be explained by the variation in samples drawn from the states' normal distributions, but might also indicate that the test cannot completely capture the presence of more complex integer ratios than 1:n or n:1 ratios. We further investigate the influence of the distributions' standard deviations in the next section's experiments. However, our results indicate the applied statistical tests (Roeske et al., 2020) accurately capture the main features of the model's behavior.

- **Conclusion**

Overall, this experiment's findings show that changing transition probabilities in a two-state Markov chain greatly affects the production of integer ratios. The highest diversity occurs at intermediate transition probabilities. Low transition rates result in only isochronous clusters, while high rates cause the isochronous cluster to disappear.

5.2 Markov chain with 2 states: Distributions' standard deviation

In this simulation, we vary the standard deviations of the normal distributions for both states of the two-state Markov chain model. The goal is to explore how increasing the spread affects the clusters of integer ratios, and to determine the maximum allowable spread for these clusters to still be identified as statistically significant. The transition probabilities are p and q are both kept fixed to 0.5, right in the middle of the range explored in the previous section.

Differential entropy of the rhythm ratio distribution

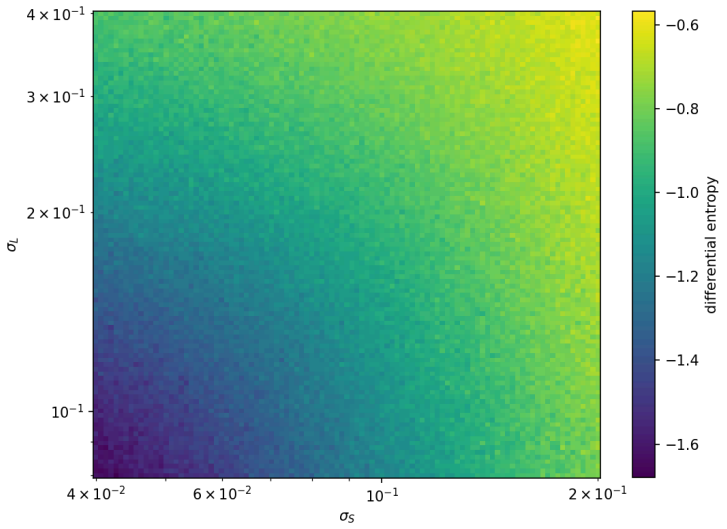


Figure 5: Heatmap of the estimated differential entropy of the distribution of rhythm ratios generated by a 2-state Markov chain model for a broad range of standard deviations of the states' generated normal intervals. The normal distributions' standard deviation ranges from 0.04 to 0.2 and 0.08 to 0.4, for states S and L respectively, with each taking 101 values evenly spaced on a logarithmic scale.

The entropy heatmap shows that as the standard deviation of both states increases, the outcome ratio distribution becomes less organized, corresponding to higher (i.e., less negative) entropy values: The differential entropy is lower than -1 around $\sigma_S = 0.05$ and σL = 0.10 and becomes as high as -0.6 for $\sigma_S = 0.2$ and $\sigma_L = 0.1$. Indeed, as the spread of the distribution increases, the sample range in which the IOIs are randomly chosen becomes larger, leading to an increased probability range for the ratio distribution outcome and leading to a flatter, less organized probability distribution.

Number of peaks in rhythm ratio distribution

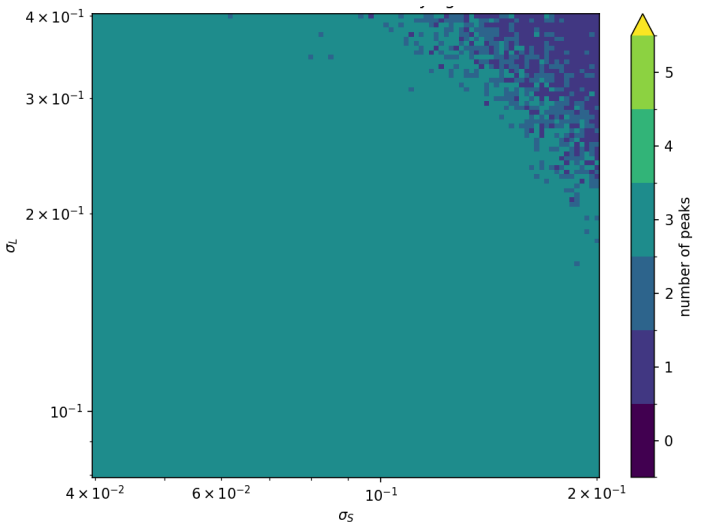


Figure 6: Heatmap of the number of peaks in the KDE rhythm ratio distributions, when varying the states' standard deviation of a 2-state Markov chain model. Again, the 101 values for σ_S and σ_L are equally spaced on a logarithmic scale.

We observe that the number of peaks transitions from three to one as the σ values for both states increase. This widening of the distributions causes the three clusters to become less distinct and merge into two or even one large cluster. As the separate peaks merge into a single very broad cluster, the model's entropy continues to grow as the spread increases and the distribution gets closer and closer to the uniform distribution (see previous graph).

Fraction of on-integer ratios

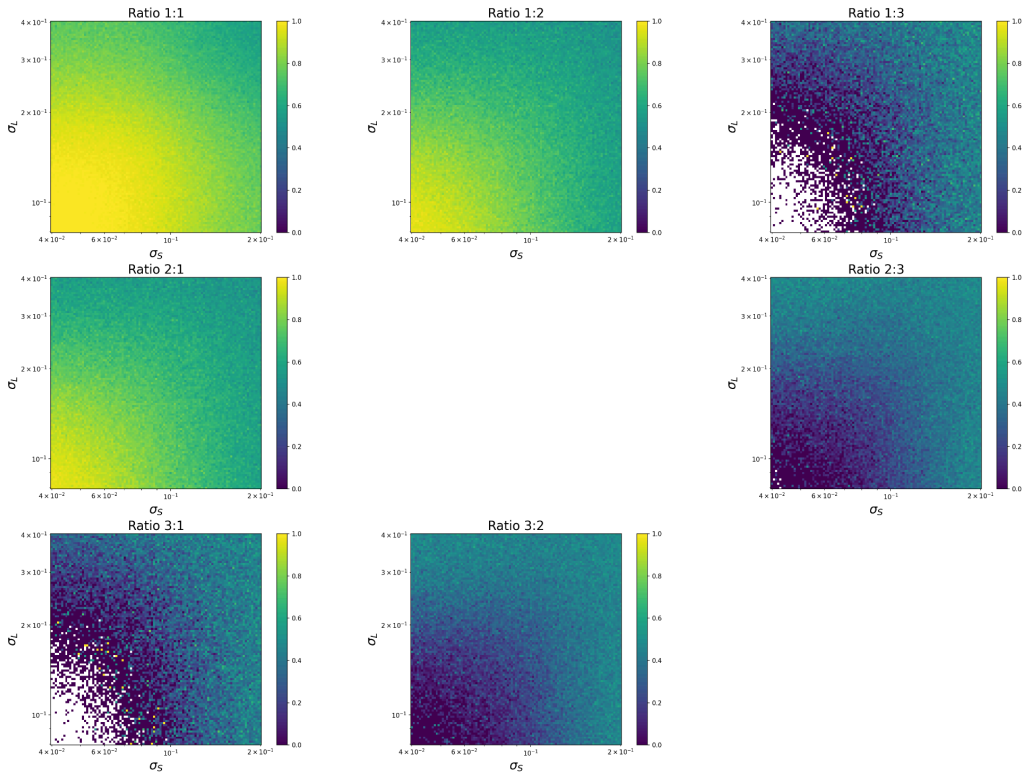


Figure 7: Heatmaps of the fraction of on-integer ratios, when varying the states' standard deviation of a 2-state Markov chain model. Again, the 101 values for σ_S and σ_L are equally spaced on a logarithmic scale.

The fraction of non-integer ratios aligns with previous observations. As the spread of the distributions increases, the fraction of non-integer ratios diminishes for the 1:1, 1:2, and 2:1 ratios. This indicates that the clusters for these ratios become less sharp as the spread increases, and relatively more ratios fall into the off-integer bins surrounding the on-integer bins. However, there are some interesting additional observations. First, the white area in the 1:3 and 3:1 distributions suggests that with very little spread, these ratios are very rare to be formed at all, as for these simulations no ratios fall into the on- or off-integer bins. Secondly, the 1:3, 3:1, 3:2, and 2:3 ratios show an increase in non-integer ratios as the spread grows. This may suggest that increasing the spread allows for a broader range of possible ratios by offering a wider range of IOI values within each state. However, all these ratios converge towards a ratio of 0.5 for high standard deviations, corresponding to approximately equal bin counts for on- and off-integer bins. This again is a consequence of the fact that the distribution becomes higher in entropy and closer to uniform.

Results statistical significance of on-integer ratios

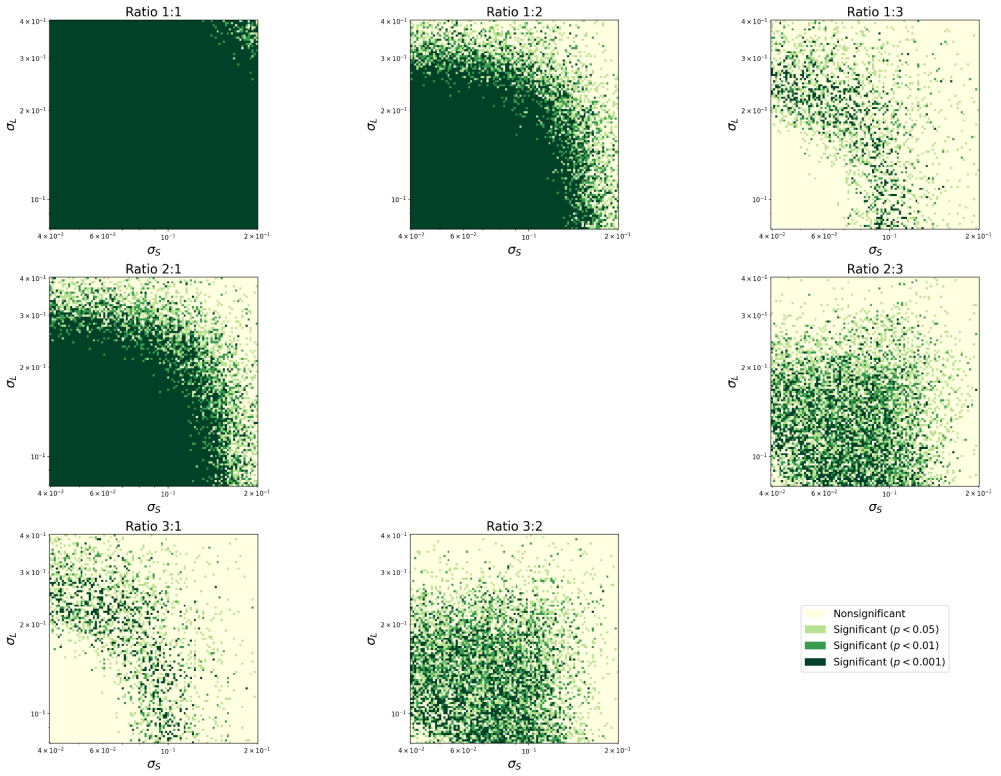


Figure 8: Heatmaps of the statistical significance of on-integer ratios, when varying the states' standard deviation of a 2-state Markov chain model. Again, the 101 values for σ_S and σ_L are equally spaced on a logarithmic scale. Three different p values are considered $p < 0.05$, $p < 0.01$, $p < 0.001$.

The statistically significant heatmaps (see Figure 8) closely mirror the results observed in the fraction of on-integer heatmaps (see Figure 7). The 1:1 ratio is most robust against IOIs being drawn from very broad normal distributions: Only for the highest standard deviations values that were tested, the random noise starts to overpower the underlying 1:1 signal. The same goes for the 2:1 and 1:2 ratios, yet an increased spread of the sampled IOIs seems to have a stronger influence on the statistical tests, as there is a larger region of the parameter space where these ratios are nonsignificant. These significant simulations also exhibit decreasing p-values as the spread of the distribution increases.

For the other ratios, 3:2 and 2:3, 3:1 and 1:3, the situation is different, as these are other ratios than the ratios between mean S and L intervals generated. These different ratios are only found to be significant for specific parameter combinations forming an arc across the heatmap. This occurs because, at low distribution spreads, only very sporadically integer ratios are produced in this region, when a strong outlier gets sampled in one or both of the S and L states. As the spread increases, it allows for the creation of a significant amount of

ratios in this area. However, when the spread becomes too large, the number of simulations producing significant clusters decreases, and p-values continue to drop as the spread widens.

- **Conclusion**

This simulation shows that increasing the standard deviations in the two-state Markov chain model leads to a loss of clarity in the integer ratio clusters. As the spread grows, distinct clusters begin to merge, resulting in fewer, less defined groups. This is reflected in higher entropy values and a reduction in the fraction of on-integer ratios, particularly for ratios like 1:1, 2:1, and 1:2. While some other integer ratios, such as 3:2 and 2:3, or 3:1 and 1:3, remain significant for specific parameters, they also become less distinct as noise increases. Ultimately, excessive spread results in the merging of distinct patterns into a poorly defined isochronous cluster.

5.3 Spiking neural network with 2 neurons: Input currents

In these simulations, we vary the intrinsic input current of two neurons in a two-neuron model.

The heatmap with the fractions of on-integer ratios (see figure 9 below), shows a fraction equal to 1 for isochrony for almost all the tested couples of parameters. It indicates that this configuration primarily produces isochrony across the tested range of currents.

Nevertheless, some regions, specifically between 380pA - 420pA for neuron 1 and 380pA - 400pA for neuron 2 (e.g. see red box in figure 9), present some variations. In these regions, the fraction of on-integer ratios diminish for isochrony and become present for the ratios 1:2, 2:1, 2:3 and 3:1. This indicates production of ratios less centered around isochrony, and the presence of diverse kinds of simple integer ratios. A grid of KDE density plots corresponding to the area within the red box provides a clearer view of the emerging patterns in these areas (see Figure 10 below). We can observe through these KDE density graphs that, around these values of input currents, the isochrony clusters become less well-defined and other peaks in the ratio's distribution are emerging.

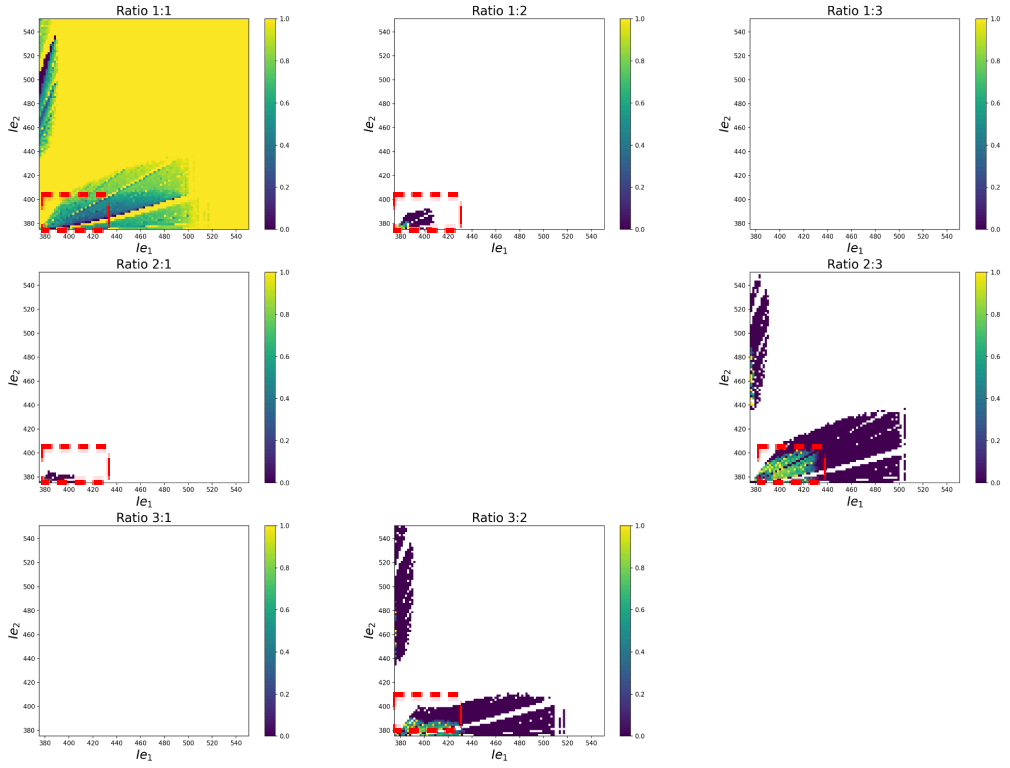


Figure 9: Heatmaps of the fraction of on-integer ratios for a 2-neurons spiking neural network model for a broad range of simulations with varying constant input current. The uncolored regions correspond to simulations where no ratios are present in the on- or -off integer bin. The red boxes correspond to a region where a great variety of integer ratios are produced; a probability density grid of this box is shown in figure 10.

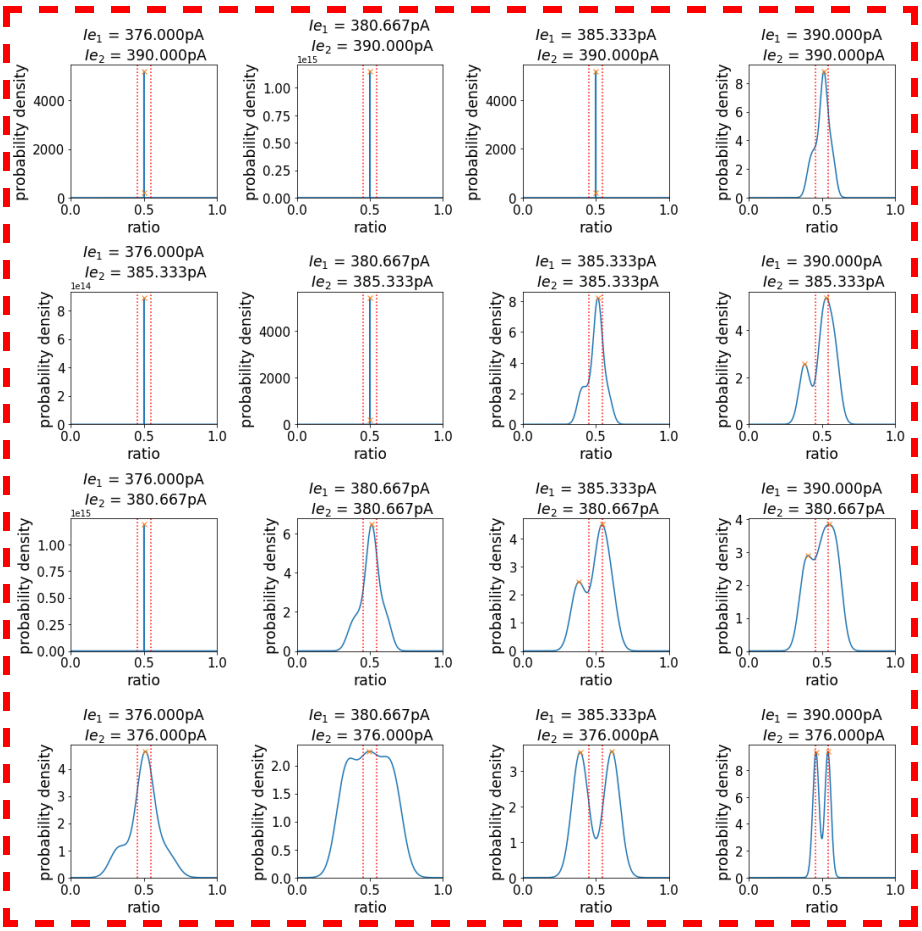


Figure 10: Grid of KDE density plots showing the distribution of ratios for neuron 1 and neuron 2 input currents varying from 376 pA to 390 pA. The red dotted lines represent the non-integer bin for isochrony (1:1). The orange crosses in the distribution correspond to the identified peaks.

Examining in more detail the region with isochrony (i.e., where the on-integer fraction of the 1:1 ratio is equal to 1), we can notice variations in the other results. For example, in the heatmaps with differential entropy (see Figure 11) and the number of peaks (see Figure 12), we can observe differences in the shape of the distribution within this seemingly uniform area.

The entropy heatmap shows a low entropy score for the area from $I_{e1} = I_{e2}$ to $I_{e1} = I_{e2} + 60pA$ (see, for example, the orange box in Figure 11). In the same area, the number of peaks graph indicates the presence of several peaks. A closer look at this in the orange box with the help of a KDE density grid, we can see that the distribution is close to perfect isochrony, with small peaks detected at the base of the 1:1 distribution.

Conversely, the entropy heatmap shows a higher entropy score for the rest of the tested parameters of this uniform isochronous area (see, for example, the purple box in Figure 11),

even though several peaks are still present. The KDE density plot within this purple box shows an area where all probability mass is situated inside the on-isochrony bin, but is more spread than the one examined in the orange box. This distribution, centered on the isochronous region, is also still split into several peaks but the peaks are more equal in height.

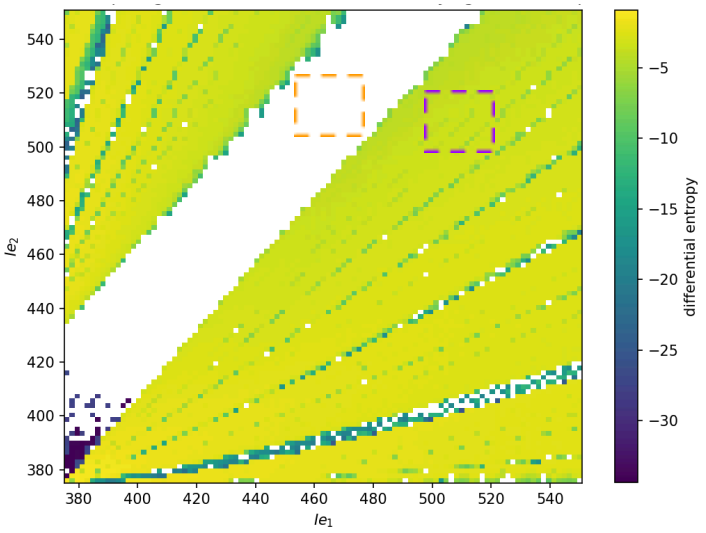


Figure 11: Heatmap of the differential entropy of KDE distribution for a 2-neurons spiking neural network model for a broad range of simulations with varying constant input current. The uncolored regions correspond to simulations with an extremely low differential entropy estimate. The orange box corresponds to an isochronous region with a very low entropy score; a probability density grid of this box is shown in Figure 13. The purple box corresponds to an isochronous region with a high entropy score (>-5); a probability density grid of this box is shown in Figure 14.

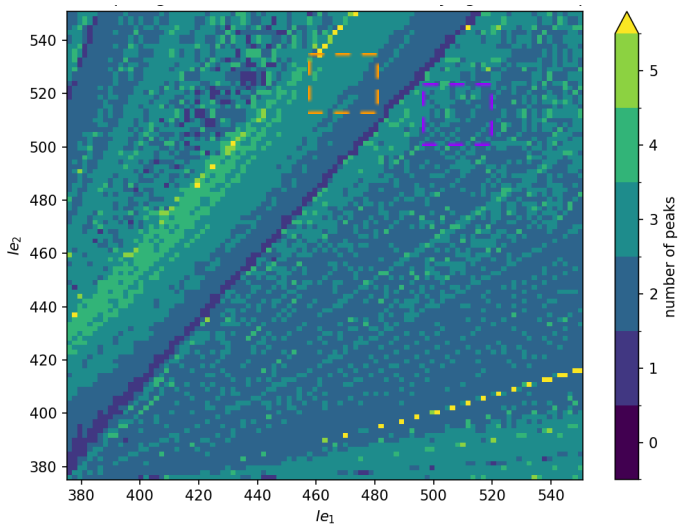


Figure 12: Heatmap of the number of peaks in KDE distribution for a 2-neurons spiking neural network model for a broad range of simulations with varying constant input current. A probability density grid of the orange box is shown in Figure 13. A probability density grid of the purple box is shown in Figure 14.

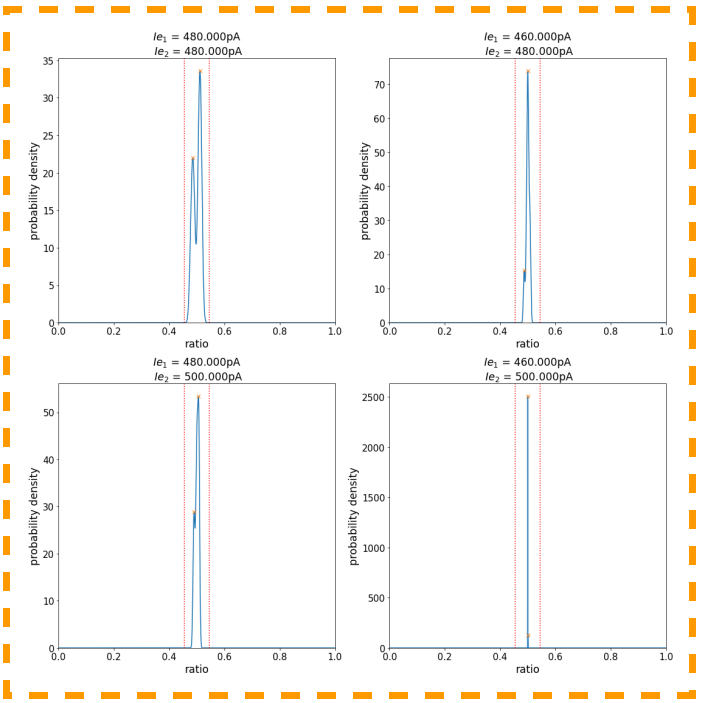


Figure 13: Grid of probability density plots showing the distribution of ratios for neuron 1 input current varying in 450 pA to 470 pA and neuron 2 in 500 to 520 pA. The red dotted lines represent the range of the on-integer bin for isochrony (1:1). The orange crosses in the distribution correspond to the identified peaks.

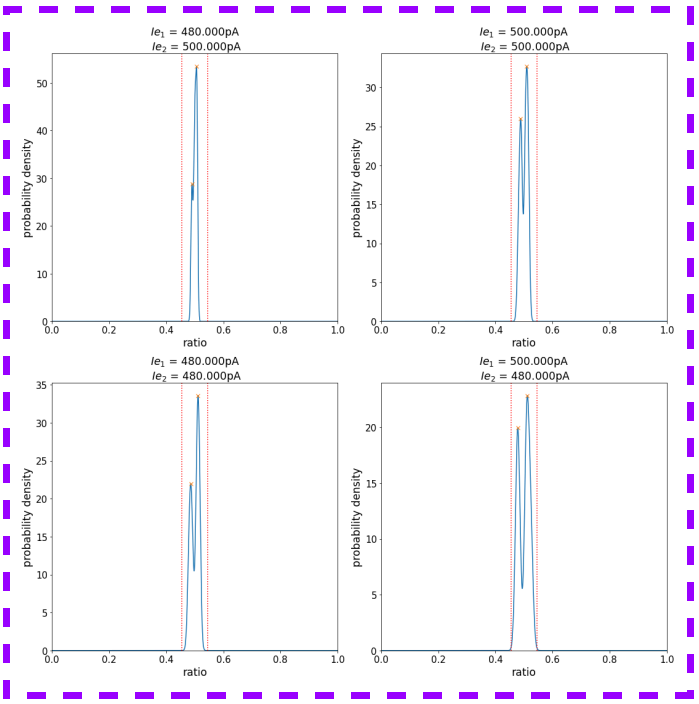


Figure 14: Grid of probability density plots showing the distribution of ratios for neuron 1 and neuron 2 input currents varying from 480 pA to 500 pA. The red dotted lines represent the on-integer bin for isochrony (1:1). The orange crosses in the distribution correspond to the identified peaks.

- **Conclusion**

In summary, in this simulation, varying the intrinsic input current of the neurons in a two-neuron model primarily produces isochrony across the tested range of currents. However, specific regions (especially 380pA - 420pA for neuron 1 and 380pA - 400pA for neuron 2) exhibit less isochronic and more diverse ratios. The KDE probability density plots in these regions of the parameters space confirm the presence of different ratio clusters when isochrony is less dominant.

5.4 Spiking neural network with 2 neurons: Connection weight and synaptic rise time

In the next simulation, we keep the model with two neurons, but instead vary the weight of connection between the two neurons, as well as the synaptic rise time of the output neuron. Different values for these parameters have a strong impact on the firing patterns of the output neuron.

At a first glance, we can see that as connection weight increases, the overall probability density of the time sequence ratios produced by the output neuron converges towards a

distribution with a clear peak around isochrony. This convergence is accompanied by a reduction in the prominence of other clusters within the distribution. This pattern is clearly visible in the following density graphs that track KDE evolution across different rising times for three different increasing connection weights (see Figure 15, below).

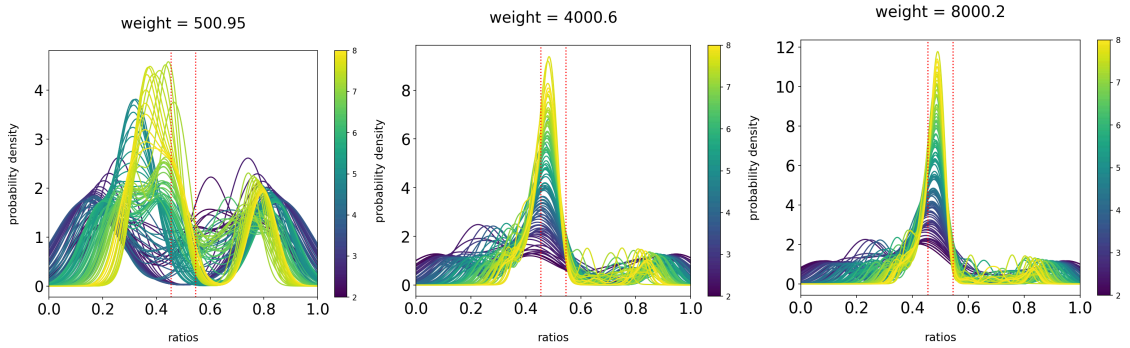


Figure 15: Probability density function of the ratio distribution for varying rising times [2–8 ms] at three connection weights [500, 4000, 8000]. The red dotted lines represent the non-integer bin for isochrony (1:1).

Similarly, as the rising time of the synaptic function increases, the isochrony clusters within the KDE distribution become more pronounced, with a corresponding decrease in the presence of non-isochronic clusters. This effect is particularly evident in the KDE density plots across different rising times for all tested connection weights (see Figure 16, below).

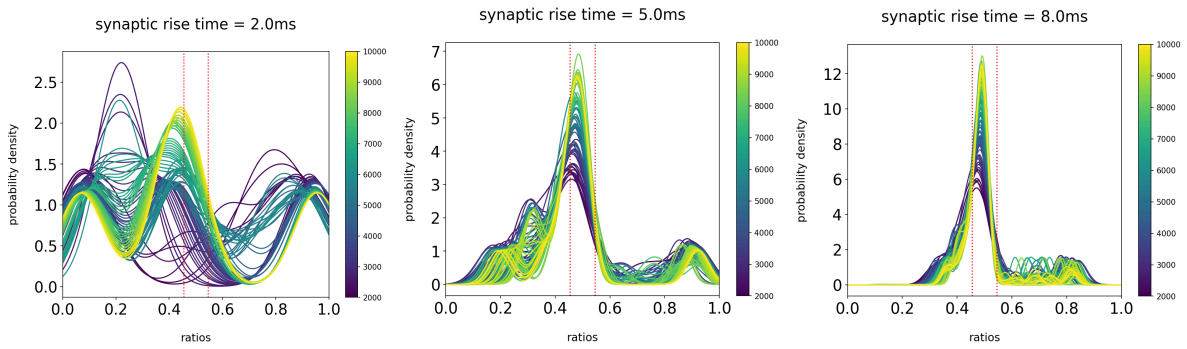


Figure 16: Cumulative distribution functions of the KDE rhythm ratio density function for varying connection weights [1–10000] at three chosen values for the synaptic rise time [2, 5, 8 ms]. The red dotted lines represent the non-integer bin for isochrony (1:1).

When examining the interaction between connection weight and rising time, through a heatmap of differential entropy, a more nuanced understanding emerges. The interplay between these two parameters creates a complex, striped pattern within the parameter space,

where discrete bands or ridges appear, as reflected by a notable drop in entropy within these regions (see the uncolored region and yellow stripes in Figure 17). These organized areas also correspond to higher fraction of on-integer ratios in the ratios of interest (see yellow stripes, Figure 18), indicating that the model exhibits self-organizing behavior on integer ratios in response to specific combinations of connection weight and rising time.

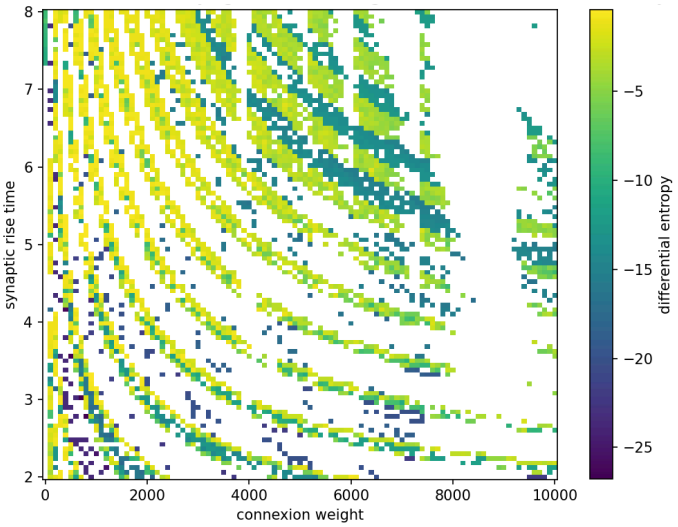


Figure 17: Heatmap of the differential entropy of rhythm ratio distribution for a 2-neurons spiking neural network model for a range of simulations with varying connection weight and synaptic rise time. The uncolored regions correspond to simulations with a very low entropy value.

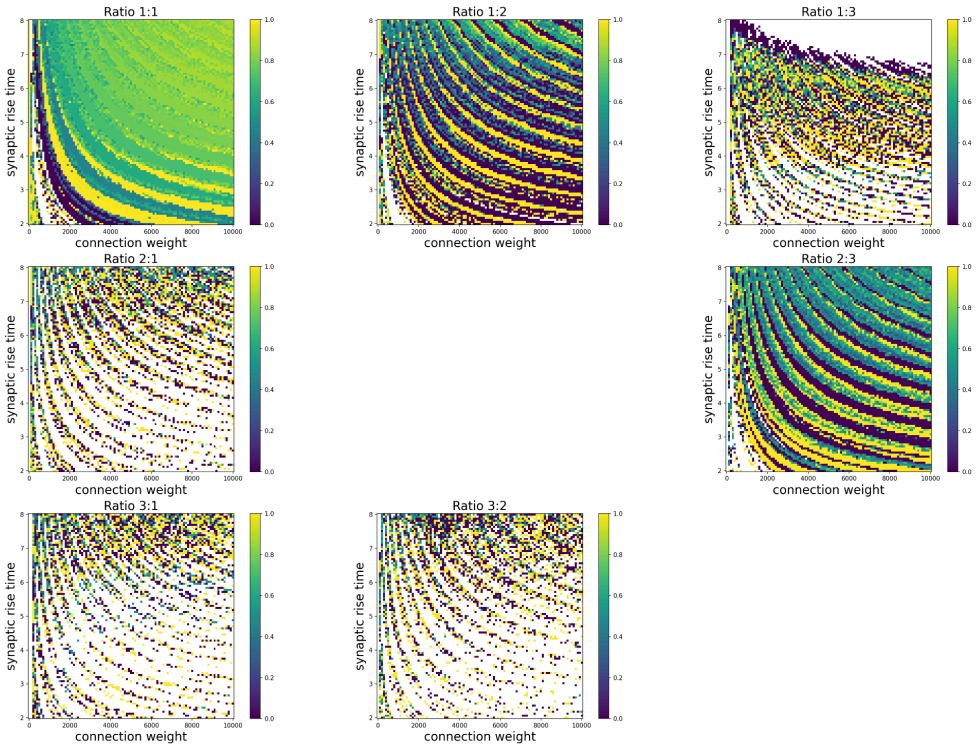


Figure 18: Heatmaps of the fraction of on-integer ratios for a 2-neurons spiking neural network model for a broad range of simulations with connection weight and synaptic rise time. The uncolored regions correspond to simulations where no ratios are present in the on- or -off integer bin.

The heatmap with the number of peaks adds another layer to our analysis (see Figure 19). It reflects the same organized pattern observed in the other graphs, highlighting a structured output within the parameter space. As we enter one of these organized discrete bands, the number of peaks generally stabilizes at around three, dropping to one between the bands.

A closer examination of a grid of KDE density plots of these bands (see Figure 20, corresponding to the purple box in Figure 19) reveals that while the number of peaks and the entropy level remains constants, the peaks of the KDE density shift away from our ratios of interest (e.g. 1:2, 2:1) to other value of ratios, sometimes beyond even beyond the 1:3 and 3:1 ratios we are analyzing. This can be seen by the presence, in some of the density plots, of peaks outside the two red lines that delimit the lower and upper bounds of the 1:3 and 3:1 ratios. This indicates that the system remains organized within these bands, but generates clusters outside our primary area of interest. These observations underscore the complexity of the model, revealing its potential to generate a broader range of temporal sequences containing rhythm ratios beyond the simple ones we focussed on here.

The regions between the organized bands correspond to higher levels of entropy. They are also associated with fewer peaks. It suggests that in these areas, the model generates more uniform distributions, which may prevent the emergence of distinct clusters.

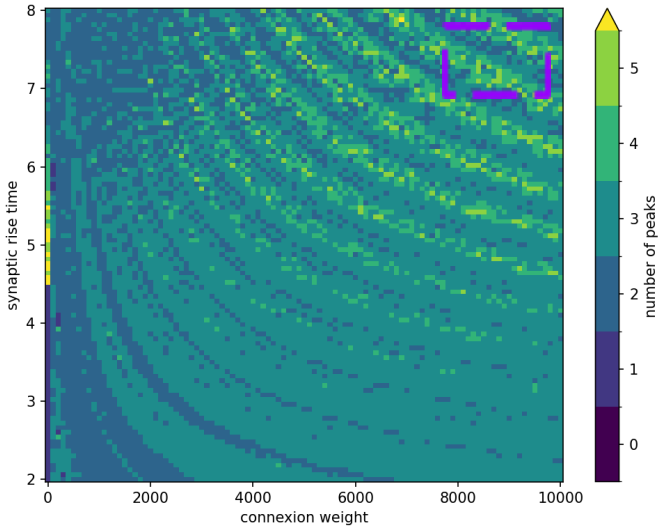


Figure 19: Heatpmap of the number of peaks in the KDE rhythm ratio distribution for a 2-neurons spiking neural network model for a range of varying connection weight and synaptic rising time values. A probability density grid of the purple box is shown in Figure 20.

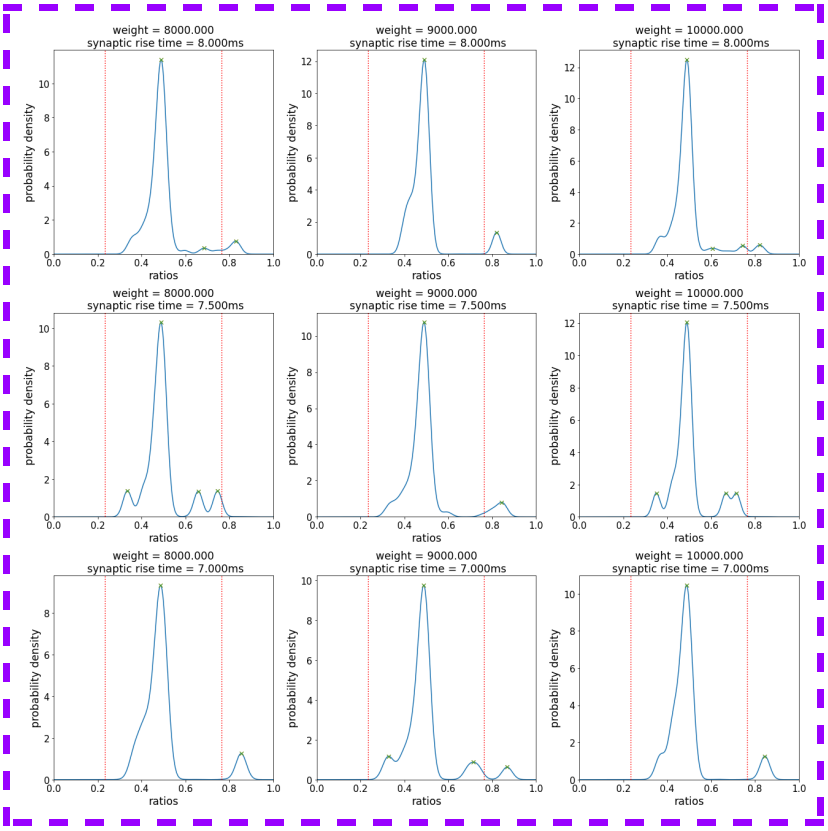


Figure 20: Grid of KDE density plots showing the distribution of ratios for connection weight varying in 8000 to 10000 and the synaptic rise time from 7ms to 8ms.The red dotted lines represent the lowest on-integer limit (i.e., low bin limit of 1:3) and highest on-integer limit (i.e., high bin limit of 3:1) for our ratios of interest. The orange crosses in the distribution correspond to the identified peaks.

- **Conclusion**

This model demonstrates that, by manipulating the connection weight and the rise time of the excitatory synaptic function, it is possible to achieve temporal sequences characterized by distinct integer ratios. The results support the hypothesis that simple neural models can generate these ratios through emergent properties, even when they only receive isochronous sequences as inputs. The interaction between connection weight and rising time produces a characteristic banded pattern in the output, alternating regions where specific integer ratios dominate and regions where the model produces a less organized outcome with clusters merging around isochrony. However, the detection of peaks outside the range of interest

suggests that the model also has the capacity to produce more complex or maybe less regular rhythmic patterns, providing a deeper insight into the range of behaviors that simple and minimalistic models like this one can produce.

5.5 Spiking neural network with Poisson generator: Input current and Poisson rate

In the third and last exploration of spiking neural networks, we connect a single neuron to the output of a Poisson generator. This setup allows us to examine how our model intrinsically transforms an input with a uniform rhythm ratio distribution, and which small integer ratios get created out of such uniform input. We vary both the rate of the Poisson generator and the intrinsic input current of the neuron.

When looking at the fraction of on-integer heatmaps, it seems that this kind of architecture makes the model produce gradually an increasing sharp cluster around isochrony as the intrinsic input current of the neuron increases, while gradually decreasing the presence of the 2:1 and 1:2 ratios and maintaining the presence of the 3:2 and 2:3 ratios.

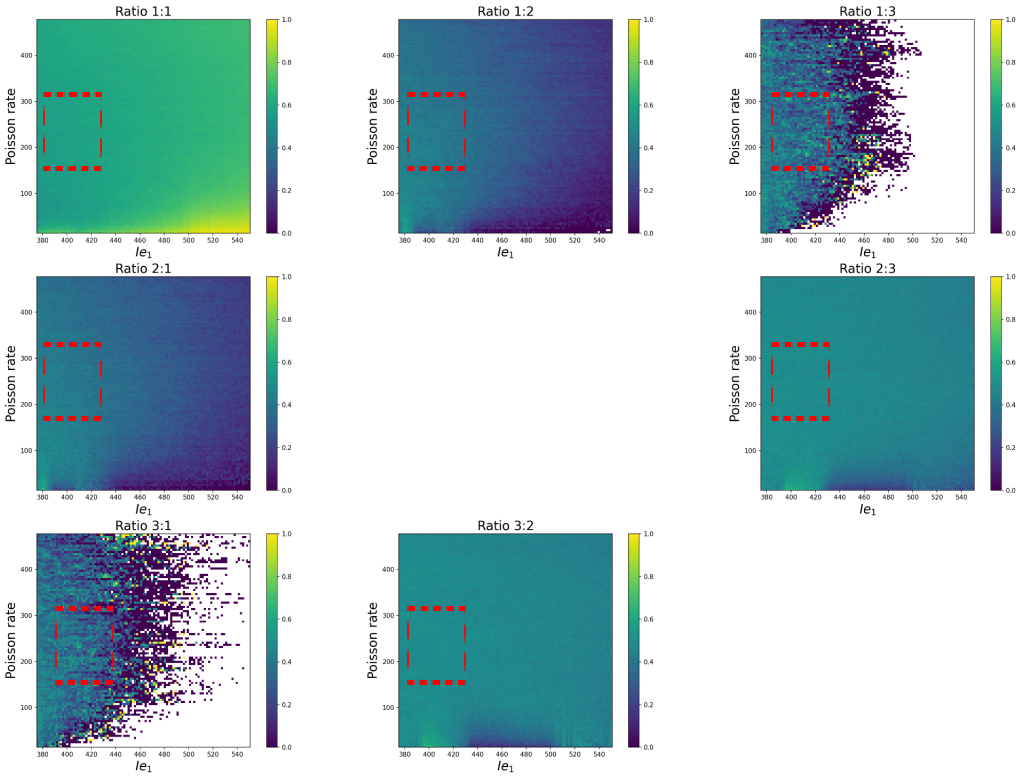


Figure 21: Heatmaps of the fraction of on-integer ratios for a spiking neural network model composed of a single neuron and Poisson generator for varying values of constant input current and Poisson rate. A probability density grid of the red boxes is shown in Figure 23.

However, this view is an artifact of the simulation's spread. The only integer ratio consistently produced by the system is isochrony (1:1). Figure 22 shows a range of KDE probability density functions for a fixed rate of 60 of the Poisson generator, while varying the intrinsic current of the neuron from 376pA to 550pA. This view demonstrates perhaps more clearly what rhythm ratios are actually produced by the model (see Figure 22). Actually, the model only produces one broad cluster of rhythm ratios centered on isochrony, which becomes sharper as the intrinsic current of the neuron increases.

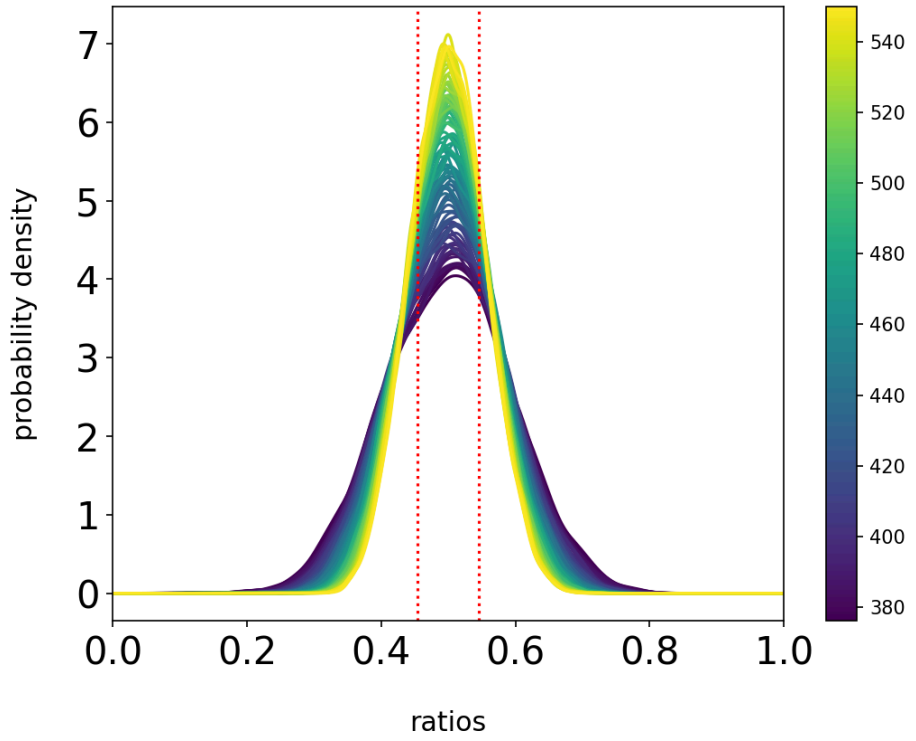


Figure 22: Kernel Density Estimation function of the ratio distribution for varying input current of the neuron [376 -550 ms] and a Poisson generator rate fixed at 235Hz

The presence of other ratios of interest in the results of Figure 21 can be better understood when looking at a grid of density plots for one of subregions (e.g., the red box in Figure 21). A grid of probability density plots corresponding to this area (see Figure 23) reveals that the presence of these ratios are actually due to the broad peak centered on isochrony. The red lines represent, as an example, the position of the 3:2 ratio which was identified as present when looking at the fraction of on-integer graph. This area actually seems to correspond to the side of the main peak of the distribution centered on isochrony (ratio 1:1).

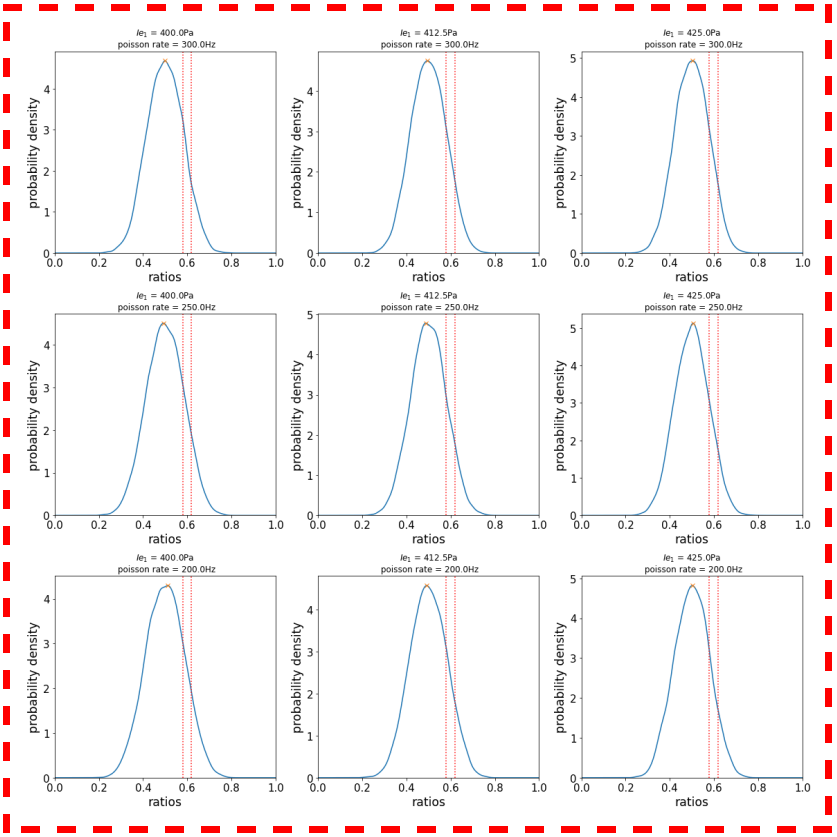


Figure 23: Grid of KDE density plots showing the distribution of ratios for the neuron input current varying in [400pA-425pA] and the Poisson generator rate varying in [200-300Hz]. The red dotted lines represent the on-integer bin the ratio 3:2. The orange crosses in the distribution correspond to the identified peaks.

So this configuration seems to only produce a large cluster around isochrony. This cluster becomes more strongly concentrated around perfect isochrony for a low Poisson rate and an input current from 500pA to 550pA (see fraction of on-integer for the ratio 1:1 in Figure 21). This tendency can be similarly observed in the entropy results. For low Poisson rates, the higher the current becomes, the more organized the behavior of the model (see Figure 24).

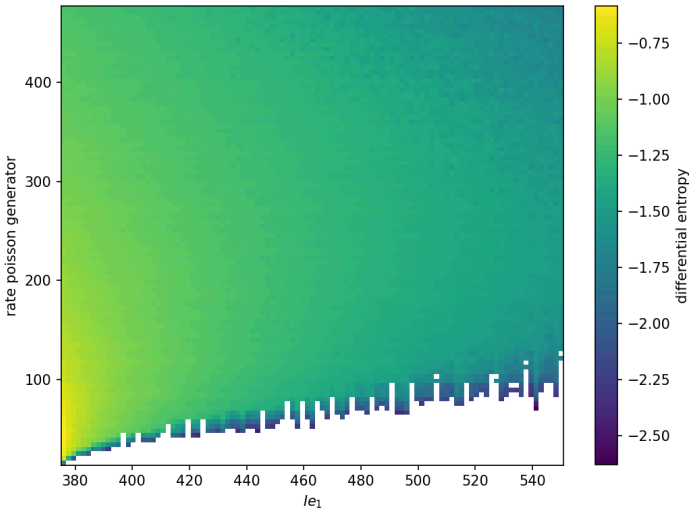


Figure 24: Differential entropy of the rhythm ratios distribution for a neuron and Poisson generator spiking neural network model for a range of constant input current and Poisson rate values.

In the same area where we observe a decrease in entropy, the number of peaks increases, shifting from a single peak distribution to a three-peak distribution (see Figure 25). Examining a grid of density plots for this area (see red box in Figure 25, and the corresponding grid in Figure 26), we can see that this increase of peaks is due to the splitting of the isochronous cluster into multiple peaks, close to but not exactly on the 1:1 isochrony ratio.

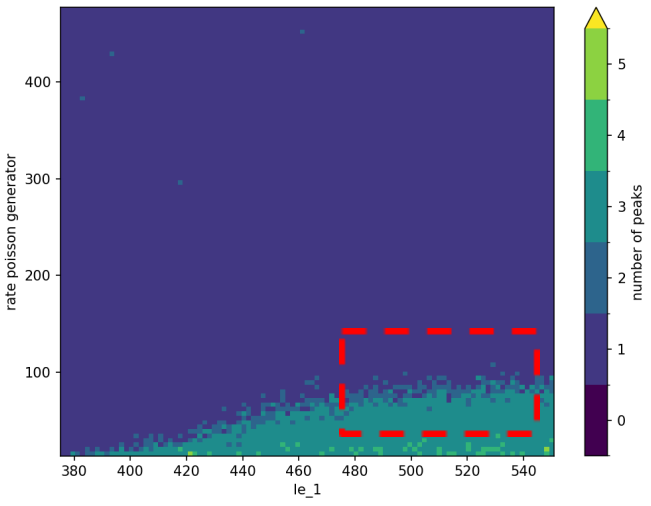


Figure 25: Heatmap of the number of peaks in the rhythm ratios KDE distribution for a neuron and Poisson generator spiking neural network model, for a range of varying constant input current and Poisson rate. A probability density grid of the red box is shown in Figure 26.

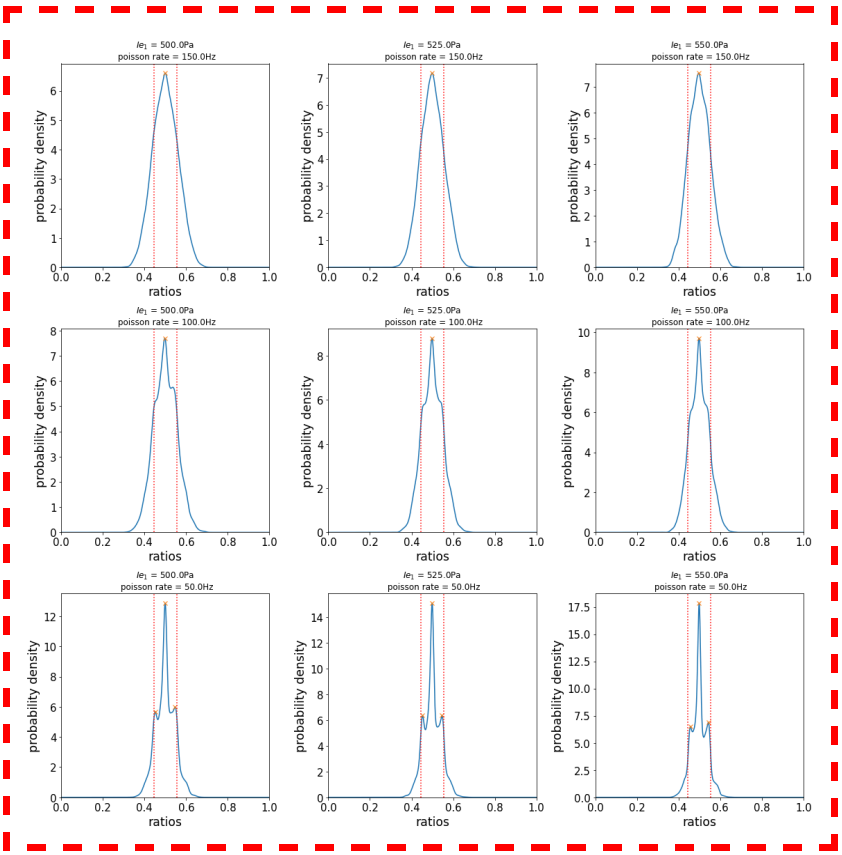


Figure 26: Grid of KDE density plots showing the distribution of ratios for the neuron input current varying in [500pA–550pA] and the Poisson generator rate varying in [50-150Hz]. The red dotted lines represent the on-integer bin for isochrony (1:1). The orange crosses in the distribution correspond to the identified peaks.

- **Conclusion**

The neuron model analyzed in this study exhibits a tendency to transform input with a uniform distribution of ratios into one clustered around isochrony. As the input current increases, this bias becomes more pronounced, with the distribution becoming more sharply concentrated around isochronous values. This clustering around isochrony is especially strong for low Poisson rates, although the cluster splits into several peaks around isochrony. These results suggest that the neuron model inherently favors the emergence of isochronous patterns.

5.6. Cricket model with isochronous stimuli: Relaxation rate and stimulus period

In this simulation, we varied the period of an isochronous stimulus and the relaxation rate of the chirping period in the cricket model.

The analysis of the heatmaps of the fraction of on-integer ratios (Figure 26) reveal that this configuration predominantly produces isochrony. A significant portion of the tested parameters correspond to a fraction of on-integer equal to 1 for the 1:1 ratio. Other ratios of interest are largely absent, resulting in the uncolored areas in the heatmaps. Only two specific regions, where the period of the stimuli is between 0.5-0.75 and 1.35–1.85 and relaxation rate between 0.6–1, produce rhythm ratios other than isochrony.

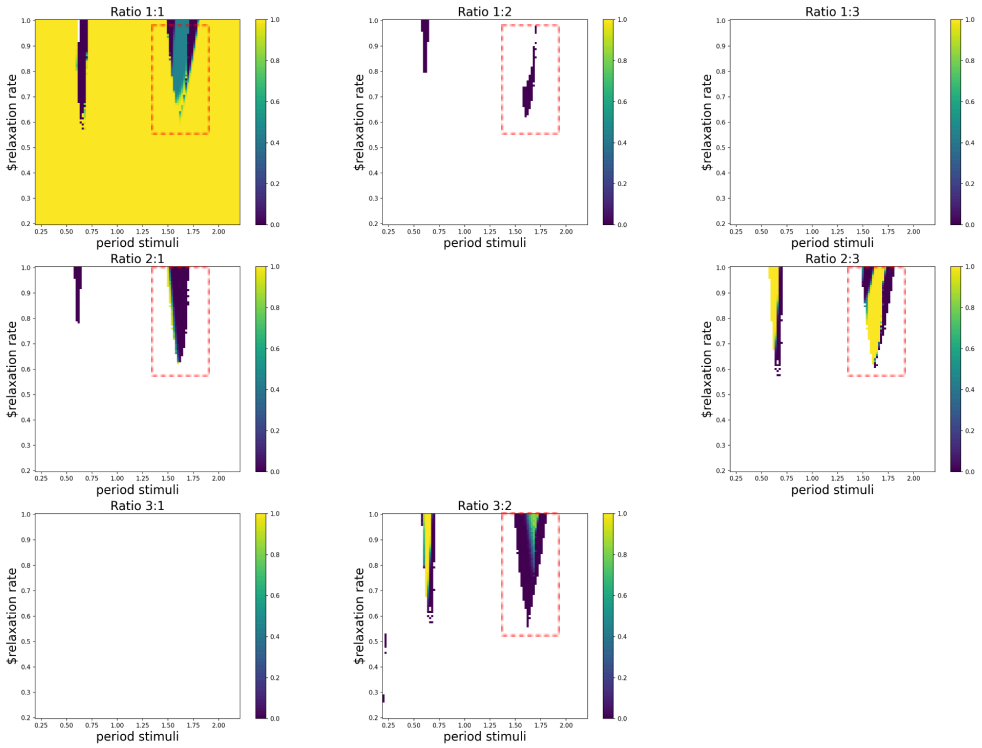


Figure 26: Heatmaps of the fraction of on-integer ratios for a cricket and isochronous stimulus model with varying stimulus period and cricket relaxation.

To better understand what is happening, we zoomed into one of these regions and examined a grid of KDE plots (see red box in Figure 26, and Figure 27). This grid of plots suggests transitional zones where the density distribution shifts from single peaks to multiple peaks. This observation indicates the presence of a bifurcation point in the model. When the parameters values reach a specific threshold the behavior of the cricket changes. It transitions from a simple isochronous rhythm to more varied rhythmic patterns. It produces few sharp clusters of ratios surrounding isochrony. This pattern may indicate that the cricket model alternates between two modes or period lengths, potentially attempting to synchronize optimally with the external stimuli.

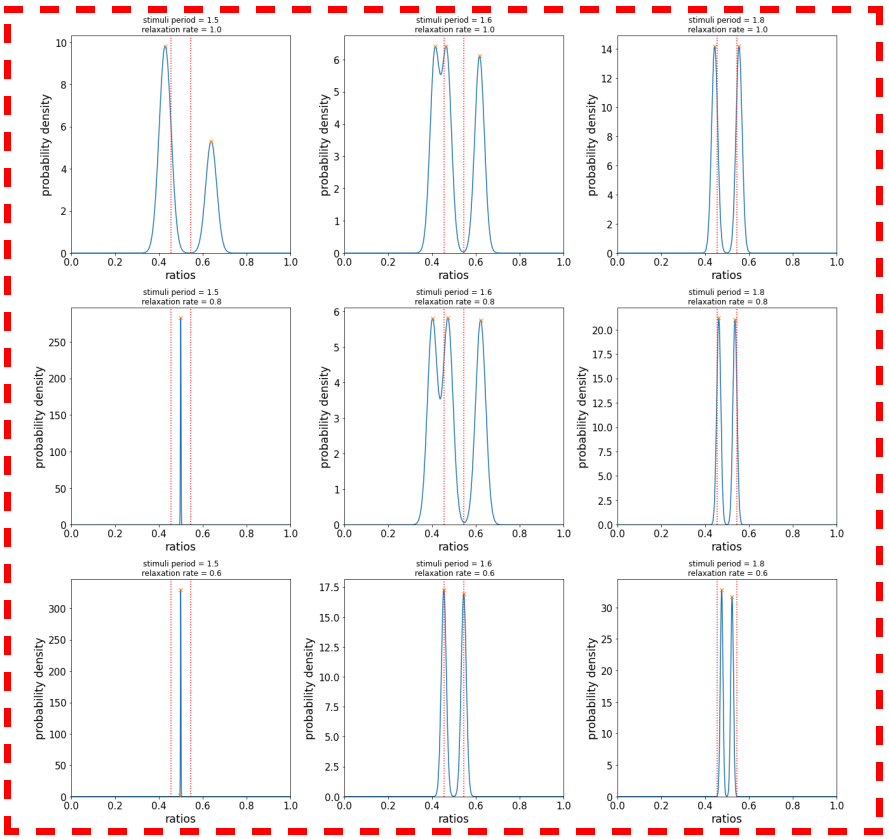


Figure 27: A grid of KDE density plots shows the distribution of ratios for a stimuli period varying from 1.5 to 1.6 and the relaxation time from 0.6 to 1. The red dotted lines represent the on-integer bin for isochrony (1:1) The red dotted lines represent the off-integer bin for isochrony (1:1). The orange crosses in the distribution correspond to the identified peaks.

The differential entropy heatmap (see Figure 28) reveals that the model exhibits higher entropy in the regions where different ratios are produced. As shown by the previous grid of KDE distributions (Figure 27), in this region of the parameter space the cricket oscillates between two types of ratios, both close to but not equal to isochrony, which gradually converge towards isochrony. At a stimulus period of 2, there is a clear demarcation where entropy reaches its minimum. Here, the cricket chirps exactly twice for each stimulus, enabling it to have a stable isochronous chirping behavior. After this precise point, entropy increases again, indicating the initiation of a possible new cycle in the model's behavior.

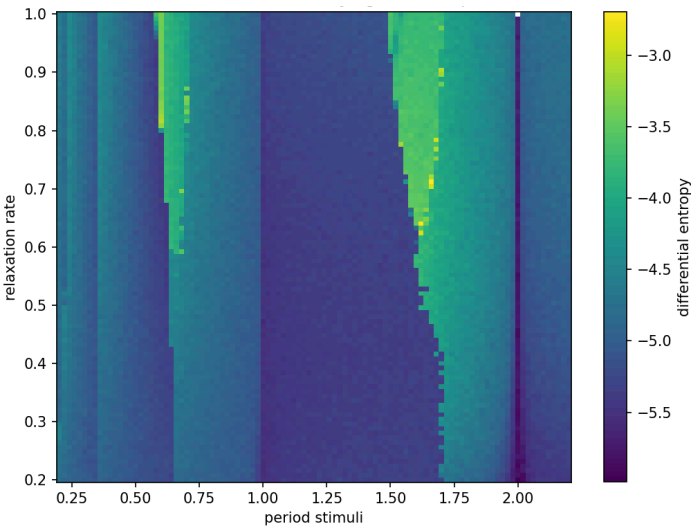


Figure 28 : Heatmap of the differential entropy of rhythm ratio distributions of the chirping in the cricket model, given an isochronous stimulus, over a range of different values for stimulus period and cricket relaxation rate.

To explore this observation further, we repeated the experiment with a stimulus period ranging from 0.2 to 5 (Figure 29) and plotted the same heatmaps of the fraction of on-integer ratios. The previously observed pattern recurs and intensifies across periods, indicating that the model consistently exhibits this behavior under some conditions. The length of the area is increasing because, as the period of the stimuli increases, the relaxation rate needs to be lower to enable the cricket to not reset his period for several unperturbed chirps.

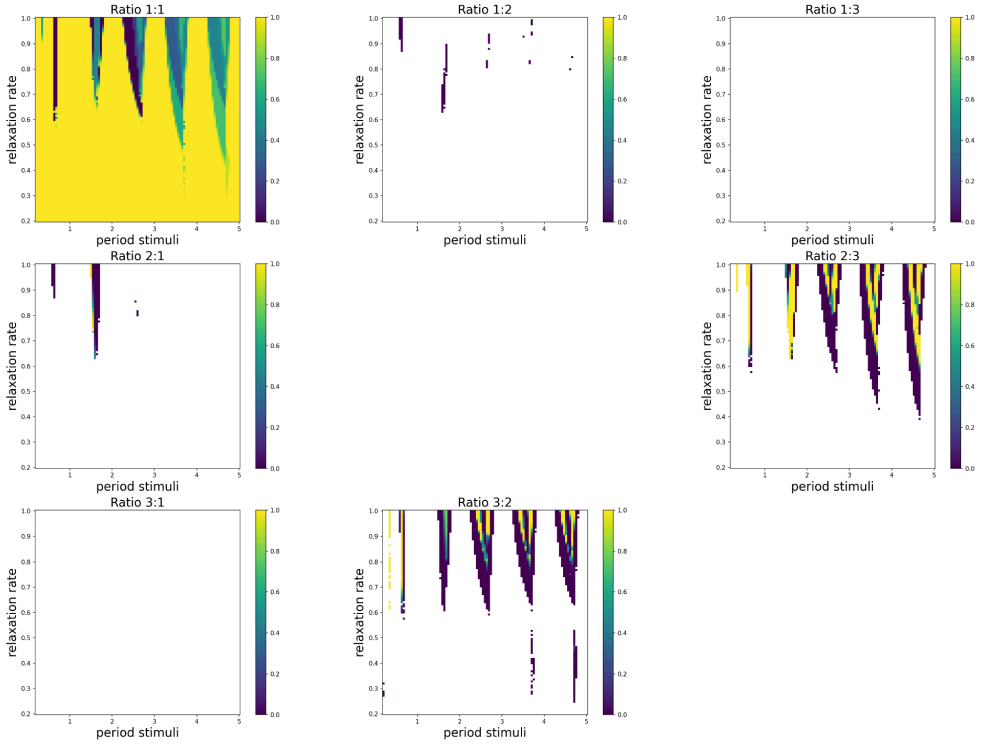


Figure 29: Heatmaps of the fraction of on-integer ratios for an extended period of isochronous stimuli in the cricket model.

- **Conclusion**

In summary, the cricket model demonstrates a tendency towards isochrony. In certain regions of the parameter space, the model generates more diverse kinds of ratios. The recurrence of these patterns across different stimulus periods suggests that the model can alternate between simple and complex rhythms. However, some ratio clusters seem less present (e.g. for the 1:2 and 2:1) than the ones reported by Sismondo (1990) (see Chapter 3, Section 3).

5.7 Cricket model with Poisson stimuli: Relaxation rate and stimulus rate

In this simulation, we varied the relaxation rate of the chirping period and the average period of a Poisson generator. Analogously to section 5.5, this experiment allows us to examine how our model intrinsically transforms input with a uniform rhythm ratio distribution of ratios toward small integer ratios.

In this configuration, the overall output of the model across all tested parameter combinations appears much less clustered around isochrony. We can also observe that the previously identified regions that exhibited distinct ratios of interest with an isochronous stimulus are not

present anymore. The fraction of on-isochrony rhythm ratios (1:1; see Figure 30) is lower and more homogeneous across all tested parameters (around 0.6, compared to 1 for nearly all parameters in the previous simulation). This is logical, as the cricket's input is much more random than the isochronous input in the previous experiment. The ratios 2:3 and 3:2 also show a uniform fraction across all parameters (around 0.4). In contrast, the fraction of on-integer ratios for 1:2 and 2:1 decreases as the cricket's relaxation time increases across all parameter combinations.

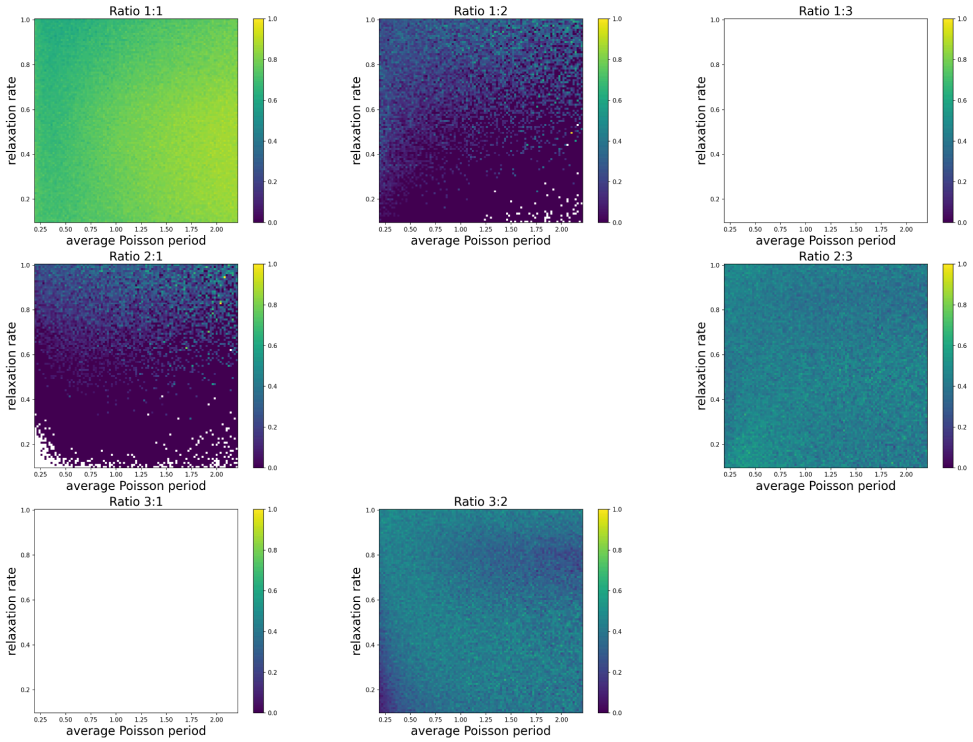


Figure 30: Heatmaps of the fraction of on-integer ratios for a cricket and Poisson generator model with varying average Poisson period and cricket relaxation.

The differential entropy heatmap (Figure 31) reveals that entropy decreases as the average period of the Poisson generator increases. This reduction in differential entropy is especially noticeable when the relaxation time is close to 1. As the cricket reverts more quickly towards its intrinsic period, the random stimuli have a much less pronounced effect on the cricket's intrinsically isochronous output sequences.

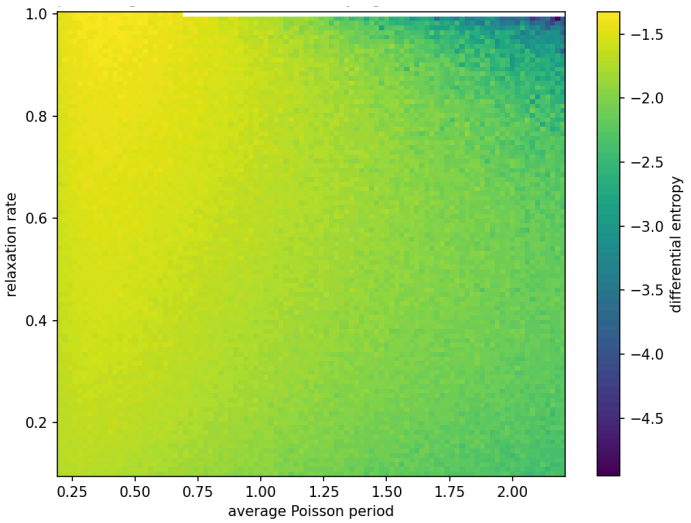


Figure 31: Heatmap of the differential entropy of the rhythm ratios distribution for a cricket and Poisson generator model for a range of parameter values for the average Poisson period and cricket relaxation rate.

Looking at the heatmap of the number of peaks (Figure 32), we observe that this simulation primarily produces a distribution with one peak. However, when the relaxation time falls to 0.4 or below, the distribution begins to show two or even three peaks. This increase in the number of peaks is also observed for long average periods (1.50–2.2) and low relaxation times (0.8–1), and matches the previously observed effect that a higher relaxation rate is less influenced by the random sequence of stimuli.

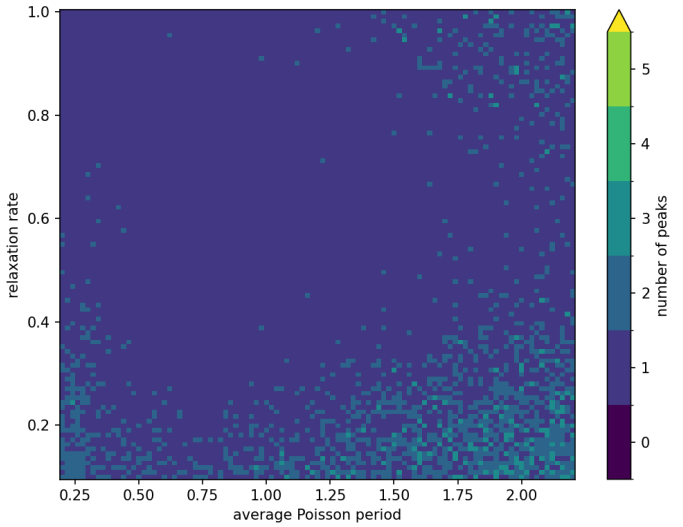


Figure 32: Heatmap of the number of peaks in KDE probability density for the cricket when exposed to stimuli from a Poisson generator. Again, the simulation is run for a range of average Poisson period and cricket relaxation rate values.

A grid of plots across the tested parameters (Figure 33) helps us better understand the different rhythm ratio distribution shapes produced by this simulation. The configuration mainly produces a cluster centered around isochrony. As the average period of the Poisson distribution increases, the cluster sharpens around perfect isochrony. However, for high relaxation rates or when combining long periods with short relaxation rates, two additional peaks emerge around the isochronous cluster. These peaks fall within the isochrony bin or the off-integer bin of isochrony (gray vertical lines). Additionally, the low fraction of on-integer values detected for the 1:2 and 2:1 clusters appears to be caused by the large peaks forming around isochrony, similar to the patterns observed in the neuron-Poisson generator experiment.

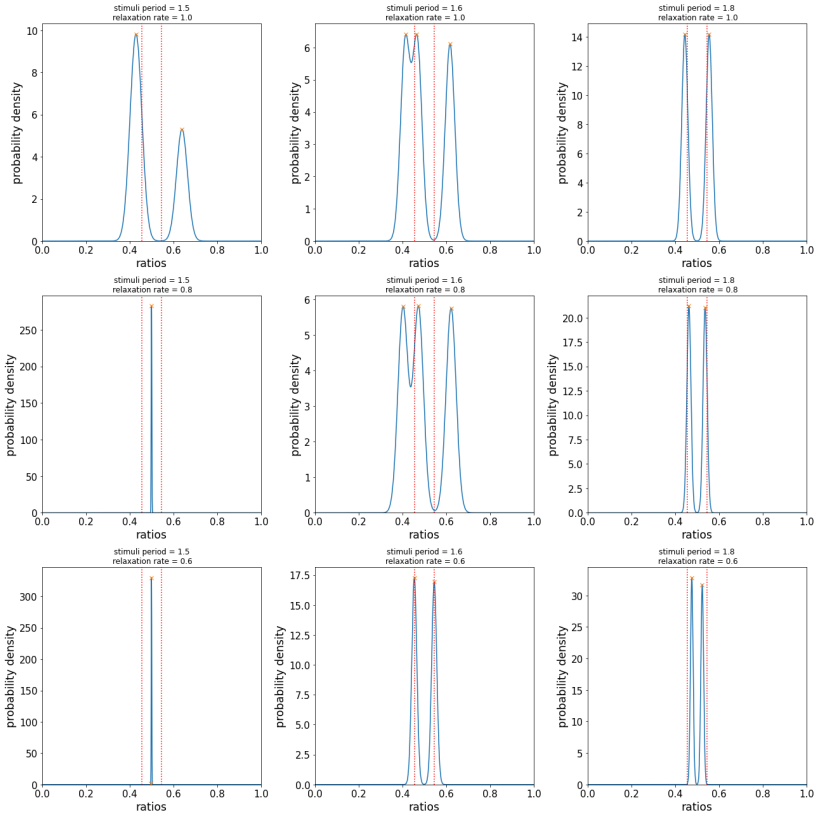


Figure 33: Grid of KDE density plots showing the distribution of ratios for the average period varying in [0.2–2.2] and the relaxation time varying in [0.2–1]. The red dotted lines represent the on-integer bin for isochrony (1:1). The orange crosses in the distribution correspond to the identified peaks.

- **Conclusion**

The results of the cricket model with stimuli from a Poisson generator suggest that the system exhibits a strong tendency to transform an input with a uniform ratio distribution toward isochrony. This effect becomes more pronounced as the Poisson generator's average period increases. The emergence of multiple peaks in the distribution, particularly at shorter relaxation times or in combinations of long periods with short relaxation times, indicates that the system can generate more complex temporal patterns under specific parameter conditions.

Chapter 6: Discussion

6.1 Summary of the Results

Our results demonstrate that the three models — Markov-chain, spiking neural network, and cricket — can produce temporal sequences with inter-onset intervals related to each other by small integer ratios.

6.1.1 Markov-chain model

The Markov-chain model generated the integer ratios 1:1, 1:2, and 2:1. Low transition probabilities between states favored mostly isochronous patterns, while high transition probabilities led to the disappearance of isochrony, producing only 1:2 and 2:1 ratios. Intermediate transition probabilities led to a more diverse production of integer ratio (i.e. 1:1, 1:2, 2:1). Simulations that explored the spread of the IOIs distributions showed that increasing the spread caused distinct clusters of ratios to merge, ultimately forming one broad cluster centered on isochrony. By adjusting the transition probabilities between the states and the spread of IOIs distributions, we can control both the concentration of ratios around integers ratio (i.e., the sharpness of the distribution) and the number of ratios produce each near integer ratio (i.e., the prominence of the distribution peaks). This control allows for the production of rhythmic sequences with precise adjustments to these characteristics, which can enhance the model's accuracy when fitting to specific rhythmic behaviors.

These findings align with our hypothesis that simple stochastic processes can generate structured temporal sequences. In addition, similar to the results observed in the Markov-chain model, developmental changes in children's rhythmic production reflect a process of gradual refinement and convergence toward structured rhythmic patterns, often corresponding to simple integer ratios (Nave et al., 2024). For instance, children's early rhythmic production shows greater variability, but over time, they tend to align more consistently with simpler integer ratios as they develop (McAuley et al., 2006; Monier & Droit-Volet, 2019). This is analogous to the influence of IOI spread in the Markov model, where narrower distributions produce more clearly defined rhythmic categories. Additionally, children tend to first acquire basic rhythmic patterns, while more complex rhythms continue

to develop and fine-tune throughout childhood, potentially influenced by their specific cultural context (Jacoby et al., 2024; Nave et al., 2024). A Markov model could simulate this developmental process by adjusting the transition probabilities between states to favor either simple or more complex rhythmic patterns to emerge, or those typical of a child's culture. Thus, the Markov-chain model provides a useful framework for understanding how basic rhythmic structures emerge and stabilize over time.

Concerning the methodology used by Roeske et al. (2020), it provided an accurate representation of the rhythmic categories related to integer ratios produced by the model. In both experiments, the analysis mirrored the results obtained with our other analytical tools (i.e., differential entropy, number of peaks, and fraction of on-integer ratios) reported above. Moreover, in the second experiment, as the spread of the IOI distributions increased, the number of simulations where secondary clusters (e.g., 1:2 and 2:1) reached statistical significance decreased. However, the main isochronous cluster remained statistically significant in nearly all simulations. This suggests that while increasing the spread of the IOI distributions diminishes the detection of secondary clusters, it has less impact on the main isochronous cluster.

6.1.2 Spiking neural network model

The spiking neural network model produced a greater variety of integer ratios compared to the Markov model. Simulations with a Poisson input revealed that the model naturally transforms random inputs into structured, isochronous rhythms. Key parameters such as input current, connection weight, and synaptic rise time increase the tendency toward isochrony as their values increase. However, the interaction between different parameters of the model allows for the creation of a wide variety of integer ratios (e.g., 1:1, 1:2, 1:3, 2:1, 2:3, 3:1, 3:2). For example, our simulations involving the connection weight and synaptic rise time of the neuron showed that as these two parameters increase concurently, the resulting rhythmic sequences alternate between two behaviors. The model produces distinct categories of ratios dominated by small integer ratios (e.g., 1:1, 1:2, 2:3) alternating with more uniform outcomes, with the formation of one broad cluster centered on isochrony. Additionally, we observed the generation of clusters of ratios beyond our range of integer ratios of interest (1:3-3:1). However, the precise location of these clusters has not been investigated. Overall, this model suggests that a wide variety of rhythmic patterns related to integer ratios can

emerge from the natural properties of neuronal circuits, supporting the neural resonance theory (E. Large & Grondin, 2008).

6.1.3 Cricket model

The cricket model, similarly to the spiking neural network, converts random inputs from a Poisson sequence into structured isochronous outputs. This model also showed a strong tendency toward isochrony when subject to an isochronous stimulus. The production of other integer ratios is constrained to a few specific relaxation rates (i.e., the rate at which the model returns to its fundamental period after a disturbance) and stimulus periods. The model demonstrated a bifurcation point, a critical threshold where the model transit from one type of behavior to another. It shifts from simple isochronous patterns to more diverse rhythmic patterns, alternating between different chirp lengths.

This bifurcation in behavior depends on the period of the stimuli and the relaxation rate. When the stimuli period is longer than the chirp period, unperturbed chirps can occur, and the relaxation rate becomes key. If the relaxation rate is high, in case of unperturbed chirps, the model quickly returns to its fundamental period, this requires a full-readjustment when the next stimulus arrives. This increases variability in chirp length. In contrast, if the relaxation rate is low, the model maintains his adapted period between stimuli even in the case of unperturbed chirps. This results in a more stable pattern, with less need to re-adjustment, leading to more consistent chirp intervals.

These results suggest that basic biological phase-locking mechanisms can generate diverse integer ratios, supporting the hypothesis that rhythmic patterns in species like crickets can emerge from simple feedback loops (Sismondo, 1990). However, this model produced fewer ratios of interests (i.e. 1:2, 2:1) than those described in Sismondo, (1990).

6.2 Comparison between the models

The Markov-chain model offers a relatively simple and easily interpretable mechanism for generating integer ratios. While it efficiently produces common ratios like 1:1, 1:2, and 2:1, it is less effective at generating more complex rhythmic structures. However, the possibility to manipulate transition probabilities and IOIs spread provides fine control over the output characteristics, such as the spread of clusters. This model shows that rhythmic patterns can emerge from stochastic processes. However, in this model, the chain states define the IOIs

that the model can produce. As a result, the ratios that can appear in the distribution are predetermined by the selected IOIs. In our simulation, the IOIs length were set on 1 and 2, meaning, the model should only produce the ratios 1:1, 1:2, 2:1. Therefore, the rhythmic structure in this model is largely determined by the chosen input values (i.e., the length of the IOIs) rather than being generated inherently by the model itself

In contrast, the spiking neural network model is inherently capable of generating integer ratios. Of the three models, it produces the widest range of rhythmic patterns. This model shows an inherent tendency to transform random input into structured, isochronous rhythms. However, the interaction of multiple parameters (e.g., connection weight and input current rise time) allows the model to produce rhythmic patterns with a great variety of ratio clusters, including clusters beyond the expected integer ratios. Its flexibility to generate various rhythmic patterns makes it ideal to study diverse rhythmic behaviors, as it can propose a single mechanism to explain multiple behaviors. However, unlike the Markov chain model, where the behavior is straightforward and predictable, the spiking neural network model is more complex. This complexity makes it difficult to predict how specific parameters and their interactions will affect the model's behavior and the production of integer ratios. This model demonstrated how even complex rhythmic behaviors can emerge from intrinsic neural dynamics (E. Large & Grondin, 2008).

The cricket model offers an interesting middle ground. Like the spiking model, it converts random inputs into structured isochronous outputs, similar to the spiking neural network model. When subjected to an isochronous input, the cricket model can produce both simple and complex rhythmic patterns depending on parameters such as relaxation rate and stimulus period. However, its production of integer ratio and their diversity is reduced compare to the spiking neural network. This difference might be due to its simpler underlying mechanism. Nonenless, the model offers valuable insights on how simble biological feedback mechanisms can generate rhythmic patterns (Sismondo, 1990).

6.3 Theoretical and practical implications

The results of this study have deepen our understanding of the biological mechanism at play behind rhythmic behaviors. Each of our model have demonstrated that rhythmic patterns based on integer ratios can emerge from simple processes, independent of specific cultural or human-cognitive mechanism. In doing so, they are applicable beyond human kind. This

findings also supports the idea that these basic rhythmic patterns observed in humans and animals might stem from shared ancestral mechanisms (Honing et al., 2015).

The cricket model illustrates how basic processes like phase-locking through error correction generate inter-onset intervals related to integer ratios. This simple mechanism, while having a more constrained production of rhythmic patterns compared to others, may have laid the groundwork for mechanisms producing more complex rhythmic patterns in organisms with higher-order cognitive processes. For instance, in humans, synchronization with external rhythms through error correction has been shown to underpin complex musical and speech rhythms, as observed in sensorimotor synchronization tasks (Repp, 2005).

Models based on Neural Resonance Theory often focus on human-specific rhythmic behaviors and its hierarchical perception of rhythmic structure (Grube & Griffiths, 2009) which are supported by a neural architecture believed to be unique to humans (E. Large & Grondin, 2008). In contrast, our simpler neural model demonstrates that fundamental neural properties shared across species can also explain certain basic rhythmic traits, such as temporal sequences with intervals related to integer ratios, observed in both humans and other species (Gregorio et al., 2021; Jacoby et al., 2024; Roeske et al., 2020). These two different approaches suggest that while some core mechanisms underlying rhythmic behavior might be conserved across species, the complexity and specifics of rhythmic behavior can vary significantly. As these basic mechanisms evolve and integrate with additional neural processes, they contribute to the more sophisticated rhythmic behaviors as the one seen in humans. Thus, rhythm may have evolved differently across species with varying neural organization.

Overall, these findings reinforce the hypothesis that rhythm is a multifaceted phenomenon (Honing et al., 2015; Kotz et al., 2018) with both shared and species-specific aspects. The mechanisms underlying rhythmic behavior in humans may not be unique but could instead be part of universal aspects rooted in evolutionary processes. Our results show that basic rhythmic patterns, such as those involving integer ratios, can emerge from simple neural processes or feedback error loops—mechanisms, mechanisms that exist across species. However, more complex aspects of rhythmic behavior like hierarchical rhythmic perception (Grube & Griffiths, 2009) necessitate specific neural architecture, linked with higher order cognitive mechanisms (E. Large & Grondin, 2008). This suggests that rhythmic abilities

likely evolved from common ancestral mechanisms, creating an evolutionary continuum of rhythmic behavior where core components are widespread, while more complex features remain species-specific.

Concerning our test on the method employed by Roeske et al., (2020), understanding how variability in IOIs impacts cluster detection is crucial for setting accurate thresholds in rhythmic pattern analysis. Our results indicate that higher variability in IOIs leads to a more uniform distribution of ratios, as the increased spread of the clusters causes them to merge into each other. This spread of the cluster and resulting uniformity make it more difficult for secondary clusters (e.g. 1:2 and 2:1 in the Markov chain) to reach statistical significance. Variability in rhythmic intervals is omni present in music and animal calls (*). When working on this kind of sequence, the fact that variability in IOIs may obscure secondary clusters should be kept in mind. to address this issue, the use of adaptive statistical threshold could help maintaining an accurate identification of these clusters. Researchers could for example adapt the size of the bin widths based on the spread of the IOIs distribution.

6.4 Strengths and limitation of the study

One of the strength of this study is its comparison of three distinct models. Each of them offering unique insights into how integer ratios can emerge from basic processes. The Markov-chain model highlighted the role of stochastic processes and showed how varying the spread of the IOIs distributions affects the ratio clusters and their statistical analysis and how this could be link with developmental changes in children's rhythmic production. The spiking neural network demonstrated the great variety of rhythmic patterns that can arise from simple neural interactions. The cricket model emphasized how even simple feedback mechanisms found in biological species can generate clusters of ratios under specific conditions.

However, the study also present limitations. The Markov-chain model relied on predefined IOIs. Doing so, it was predisposed to generate ratios cluster around 1:1, 1:2 and 2:1. Futur researches could try to use stochastic model that are not directly tied to integer ratios. For example, increasing the number of states of our Markov chain could already allow for a more natural emergence of integer ratios.

The spiking neural network model was highly sensitive to changes in parameters settings. Further exploration is needed to understand the full range of rhythmic patterns it can produce.

Unidentified clusters outside the expected integer ratios suggest that further analysis is needed to understand how these clusters are generated, and why.

The cricket model, as discussed in Sismondo (1990), did not include detailed modeling of certain chirping adaptations such as, how the cricket reacts when it encounters several stimuli before chirping again. The current model posits that the cricket "forgets" the previous adjustment after hearing a second stimulus. Exploring an alternative, where the cricket adapts based on cumulative adaptation (i.e., the cricket adjusts its chirping based on the current length of its chirp period rather than its fundamental chirp period) could enhance the biological accuracy of the model.

Finally, as explained in Chapter 4, Section 4.2, we can analyze the models by focusing solely on the output events, known as the *output event sequence* (e.g., the cricket's chirps). Alternatively, we can consider the combined events from all the model's components, known as the *merged event sequence*. (e.g., both the stimuli event and the cricket's chirps). In this work, we focused solely on the *output event sequence*, but investigating the *merged event sequence* could help us get further insight into the interaction between the model's components. For the cricket model, this analysis might reveal whether integer ratios emerge more clearly through the interaction between the stimulus and the cricket's chirps, aligning with the findings of Sismondo (1990), rather than from the cricket's adapted chirps alone. We can similarly apply this approach to the spiking neural network model. This would reveal if integer ratios also arise from interactions between the neurons within our simple network.

6.5 Suggestions for future research

Futur research should provide more comparisons between existing models, including the ones that comporte high-order cognitive processes (E. W. Large et al., 2023; Prinz, 2006; Yuan et al., 2024). This comparison with our simple models could help us determine if the added complexity of this model in necessary. For example, we could compare our simple neural model to the neural models from the NRT theory that account for the emergence of integer ratios on the basis of high order cognitive processes (i.e., perception of rhythmic hierarchy)(E. W. Large et al., 2023). This will enable us to determine if the high order cognitive processes are effectively acting on the production of integer ratios or if their production depend on more basic cognitive processus. Comparing models can also lead to the identification of the specific role of each cognitive process. For example, some model use

adaptive oscillators (Yuan et al., 2024). Such a component add a memory and learning power to the model. We could compare this kind of model to our Markov chain model, or spiking neural neetwork model, which don't have memory, to better understand the precise implication of learning and memory in the production on integer ratios.

Additionally, we could compare our model to the behavioral data collected in previous studies (Gregorio et al., 2021; Jacoby et al., 2024; Mehr et al., 2019; Roeske et al., 2020). This would enable us so see which model better explain the rhythmic behavior of which species or across species.

Conclusion

In conclusion, this study demonstrates that simple models, such as a Markov-chain, a spiking neural network, and cricket models can generate temporal sequences with intervals related to integer ratios. Together, they offer valuable insights into rhythmic behaviors. The Markov-chain model shows how stochastic processes can produce integer ratios and how the spread of the IOIs distributions impact the statistical significance of rhythmic clusters. Its behavior could also be used to model developmental changes in children's rhythmic production. The spiking neural network demonstrated how basic neural dynamics can generate rhythmic patterns related to integer ratios. Meanwhile, the cricket model emphasized how basic biological mechanisms, such as phase-locking and feedback loops, can contribute to the formation of ratios clusters on integer ratios.

These findings suggest that some rhythmic features observed both in music and animal vocalization may stem from basic, shared biological mechanisms that do not rely on higher-order cognitive processes, offering insights into the origins of musicality.

Bibliography

Bishop, C. M. (2016). *Pattern Recognition and Machine Learning* (Softcover reprint of the original 1st edition 2006 (corrected at 8th printing 2009)). Springer New York.

Bouwer, F. L., Nityananda, V., Rouse, A. A., & ten Cate, C. (2021). Rhythmic abilities in humans and non-human animals : A review and recommendations from a methodological perspective. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *376*(1835), 20200335. https://doi.org/10.1098/rstb.2020.0335

Cason, N., & Schön, D. (2012). Rhythmic priming enhances the phonological processing of speech. *Neuropsychologia*, *50*(11), 2652‑2658. https://doi.org/10.1016/j.neuropsychologia.2012.07.018

Conard, N. J., Malina, M., & Münzel, S. C. (2009). New flutes document the earliest musical tradition in southwestern Germany. *Nature*, *460*(7256), 737‑740. https://doi.org/10.1038/nature08169

Duignan, B. (2024). Occam’s razor. In *Encyclopedia Britannica*. https://www.britannica.com/topic/Occams-razor

Dutta, S., Schafer, C., Gomez, J., Ni, K., Joshi, S., & Datta, S. (2020). Supervised Learning in All FeFET-Based Spiking Neural Network : Opportunities and Challenges. *Frontiers in Neuroscience*, *14*. https://doi.org/10.3389/fnins.2020.00634

Espinoza Valverde, J. A., Müller, E., Haug, N., Schöfmann, C. M., Linssen, C., Senk, J., Spreizer, S., Trensch, T., Lober, M., Jiang, H., Kurth, A., Acimovic, J., Korcsak-Gorzo, A., Welle Skaar, J.-E., Terhorst, D., Stapmanns, J., Graber, S., de Schepper, R., Eppler, J. M., … Plesser, H. E. (2024). *NEST 3.7* (Version 3.7) [Logiciel]. Zenodo. https://doi.org/10.5281/ZENODO.10834751

Essens, P. J., & Povel, D.-J. (1985). Metrical and nonmetrical representations of temporal

patterns. *Perception & Psychophysics*, *37*(1), 1‑7. https://doi.org/10.3758/BF03207132

Fitch, W. T. (2006). The biology and evolution of music : A comparative perspective. *Cognition*, *100*(1), 173‑215. https://doi.org/10.1016/j.cognition.2005.11.009

Fuhrmann, D., Ravignani, A., Marshall-Pescini, S., & Whiten, A. (2014). Synchrony and motor mimicking in chimpanzee observational learning. *Scientific Reports*, *4*(1), 5283. https://doi.org/10.1038/srep05283

Gregorio, C. D., Valente, D., Raimondi, T., Torti, V., Miaretsoa, L., Friard, O., Giacoma, C., Ravignani, A., & Gamba, M. (2021). Categorical rhythms in a singing primate. *Current Biology*, *31*(20), R1379‑R1380. https://doi.org/10.1016/j.cub.2021.09.032

Grube, M., & Griffiths, T. D. (2009). Metricality-enhanced temporal encoding and the subjective perception of rhythmic sequences. *Cortex: A Journal Devoted to the Study of the Nervous System and Behavior*, *45*(1), 72‑79. https://doi.org/10.1016/j.cortex.2008.01.006

Háden, G. P., Bouwer, F. L., Honing, H., & Winkler, I. (2024). Beat processing in newborn infants cannot be explained by statistical learning based on transition probabilities. *Cognition*, *243*, 105670. https://doi.org/10.1016/j.cognition.2023.105670

Harley, J. (2010). *Xenakis : His Life in Music*. Routledge. https://doi.org/10.4324/9780203342794

Honing, H., ten Cate, C., Peretz, I., & Trehub, S. E. (2015). Without it no music : Cognition, biology and evolution of musicality. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *370*(1664), 20140088. https://doi.org/10.1098/rstb.2014.0088

Horr, N. K., & Di Luca, M. (2015). Timing Rhythms : Perceived Duration Increases with a Predictable Temporal Structure of Short Interval Fillers. *PLoS ONE*, *10*(10),

e0141018. https://doi.org/10.1371/journal.pone.0141018

Huang, J.-K., & Yin, B. (2023). Phylogenic evolution of beat perception and synchronization : A comparative neuroscience perspective. *Frontiers in Systems Neuroscience*, *17*. https://doi.org/10.3389/fnsys.2023.1169918

iaf_psc_alpha – Leaky integrate-and-fire model with alpha-shaped input currents—NEST Simulator Documentation. (s. d.). Consulté 5 septembre 2024, à l'adresse https://nest-simulator.readthedocs.io/en/stable/models/iaf_psc_alpha.html

Jacoby, N., Polak, R., Grahn, J. A., Cameron, D. J., Lee, K. M., Godoy, R., Undurraga, E. A., Huanca, T., Thalwitzer, T., Doumbia, N., Goldberg, D., Margulis, E. H., Wong, P. C. M., Jure, L., Rocamora, M., Fujii, S., Savage, P. E., Ajimi, J., Konno, R., … McDermott, J. H. (2024). Commonality and variation in mental representations of music revealed by a cross-cultural comparison of rhythm priors in 15 countries. *Nature Human Behaviour*, *8*(5), 846‑877. https://doi.org/10.1038/s41562-023-01800-9

Jadoul, Y., Tufarelli, T., Coissac, C., & Ravignani, A. (in prep.). *Mathematical assumptions underlying integer ratio rhythms in bioacoustics and music cognition*.

Janik, V. M., & Slater, P. J. B. (2000). The different roles of social learning in vocal communication. *Animal Behaviour*, *60*(1), 1‑11. https://doi.org/10.1006/anbe.2000.1410

Karbasi, S. M., Haug, H. S., Kvalsund, M.-K., Krzyzaniak, J., & Torresen, J. (2021). *A Generative Model for Creating Musical Rhythms with Deep Reinforcement Learning*.

Killin, A. (2021). Music Archaeology, Signaling Theory, Social Differentiation. In A. Killin & S. Allen-Hermanson (Éds.), *Explorations in Archaeology and Philosophy* (p. 85‑100). Springer International Publishing.

https://doi.org/10.1007/978-3-030-61052-4_6

Kotz, S. A., Ravignani, A., & Fitch, W. T. (2018). The Evolution of Rhythm Processing. *Trends in Cognitive Sciences*, *22*(10), 896‑910. https://doi.org/10.1016/j.tics.2018.08.002

Large, E., & Grondin, S. (2008). Resonating to Musical Rhythm : Theory and Experiment. *Psychol Time*, 189‑232. https://doi.org/10.1016/B978-0-08046-977-5.00006-5

Large, E. W., & Jones, M. R. (1999). The dynamics of attending : How people track time-varying events. *Psychological Review*, *106*(1), 119‑159. https://doi.org/10.1037/0033-295X.106.1.119

Large, E. W., Roman, I., Kim, J. C., Cannon, J., Pazdera, J. K., Trainor, L. J., Rinzel, J., & Bose, A. (2023). Dynamic models for musical rhythm perception and coordination. *Frontiers in Computational Neuroscience*, *17*. https://doi.org/10.3389/fncom.2023.1151895

Large, E. W., & Snyder, J. S. (2009). Pulse and meter as neural resonance. *Annals of the New York Academy of Sciences*, *1169*, 46‑57. https://doi.org/10.1111/j.1749-6632.2009.04550.x

Levitin, D. J. (avec Internet Archive). (2008). *This is your brain on music : Understanding a human obsession*. London : Atlantic. http://archive.org/details/thisisyourbraino0000levi

London, J. (2004). Introduction. In J. London (Éd.), *Hearing in Time : Psychological Aspects of Musical Meter* (p. 0). Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195160819.003.0001

McAuley, J. D., Jones, M. R., Holub, S., Johnston, H. M., & Miller, N. S. (2006). The time of our lives : Life span development of timing and event tracking. *Journal of Experimental Psychology: General*, *135*(3), 348‑367.

https://doi.org/10.1037/0096-3445.135.3.348

Mehr, S. A., Singh, M., Knox, D., Ketter, D. M., Pickens-Jones, D., Atwood, S., Lucas, C., Jacoby, N., Egner, A. A., Hopkins, E. J., Howard, R. M., Hartshorne, J. K., Jennings, M. V., Simson, J., Bainbridge, C. M., Pinker, S., O'Donnell, T. J., Krasnow, M. M., & Glowacki, L. (2019). Universality and diversity in human song. *Science*, *366*(6468), eaax0868. https://doi.org/10.1126/science.aax0868

Merker, B. H., Madison, G. S., & Eckerdal, P. (2009). On the role and origin of isochrony in human rhythmic entrainment. *Cortex; a Journal Devoted to the Study of the Nervous System and Behavior*, *45*(1), 4–17. https://doi.org/10.1016/j.cortex.2008.06.011

Monier, F., & Droit-Volet, S. (2019). Development of sensorimotor synchronization abilities : Motor and cognitive components. *Child Neuropsychology*, *25*(8), 1043–1062. https://doi.org/10.1080/09297049.2019.1569607

Narendra, K. S., & Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, *1*(1), 4–27. IEEE Transactions on Neural Networks. https://doi.org/10.1109/72.80202

Nave, K., Carrillo, C., Jacoby, N., Trainor, L., & Hannon, E. (2024). The development of rhythmic categories as revealed through an iterative production task. *Cognition*, *242*, 105634. https://doi.org/10.1016/j.cognition.2023.105634

Norris, J. R. (1997). *Markov Chains*. Cambridge University Press. https://doi.org/10.1017/CBO9780511810633

Pachet, F. (2010). The Continuator : Musical Interaction With Style. *Journal of New Music Research*, *32*, 333–341. https://doi.org/10.1076/jnmr.32.3.333.16861

Patel, A. D., Iversen, J. R., Chen, Y., & Repp, B. H. (2005). The influence of metricality and modality on synchronization with a beat. *Experimental Brain Research*, *163*(2),

226‑238. https://doi.org/10.1007/s00221-004-2159-8

Payne, R. S., & McVay, S. (1971). Songs of Humpback Whales. *Science*, *173*(3997), 585‑597. https://doi.org/10.1126/science.173.3997.585

Pikovsky, A., Rosenblum, M., & Kurths, J. (2001). *Synchronization : A universal concept in nonlinear sciences*.

Purnell-Webb, P., & Speelman, C. P. (2008). Effects of Music on Memory for Text. *Perceptual and Motor Skills*, *106*(3), 927‑957. https://doi.org/10.2466/pms.106.3.927-957

Raimondi, T., Di Panfilo, G., Pasquali, M., Zarantonello, M., Favaro, L., Savini, T., Gamba, M., & Ravignani, A. (2023). Isochrony and rhythmic interaction in ape duetting. *Proceedings of the Royal Society B: Biological Sciences*, *290*(1990), 20222244. https://doi.org/10.1098/rspb.2022.2244

Ravignani, A., Dalla Bella, S., Falk, S., Kello, C. T., Noriega, F., & Kotz, S. A. (2019). Rhythm in speech and animal vocalizations : A cross-species perspective. *Annals of the New York Academy of Sciences*, *1453*(1), 79‑98. https://doi.org/10.1111/nyas.14166

Repp, B. H. (2005). Sensorimotor synchronization : A review of the tapping literature. *Psychonomic Bulletin & Review*, *12*(6), 969‑992. https://doi.org/10.3758/BF03206433

Roeske, T. C., Tchernichovski, O., Poeppel, D., & Jacoby, N. (2020). Categorical Rhythms Are Shared between Songbirds and Humans. *Current Biology*, *30*(18), 3544-3555.e6. https://doi.org/10.1016/j.cub.2020.06.072

Rohrmeier, M. (s. d.). *TOWARDS A FORMALIZATION OF MUSICAL RHYTHM*.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., & Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, *8*(3), 382‑439.

https://doi.org/10.1016/0010-0285(76)90013-X

Sethares, W. A. (2014). The geometry of musical rhythm : What makes a “good” rhythm good? *Journal of Mathematics and the Arts*, *8*(3‑4), 135‑137. https://doi.org/10.1080/17513472.2014.906116

Sismondo, E. (1990). Synchronous, Alternating, and Phase-Locked Stridulation by a Tropical Katydid. *Science*, *249*(4964), 55‑58.

Sneyd, J., Fewster, R. M., & McGillivray, D. (2022). *Mathematics and Statistics for Science*. Springer International Publishing. https://doi.org/10.1007/978-3-031-05318-4

Susi, G., Garcés, P., Paracone, E., Cristini, A., Salerno, M., Maestú, F., & Pereda, E. (2021). FNS allows efficient event-driven spiking neural network simulations based on a neuron model supporting spike latency. *Scientific Reports*, *11*(1), 12160. https://doi.org/10.1038/s41598-021-91513-8

Temperley, D. (2010). Modeling Common-Practice Rhythm. *Music Perception: An Interdisciplinary Journal*, *27*(5), 355‑376. https://doi.org/10.1525/mp.2010.27.5.355

Todd, N. (1985). A Model of Expressive Timing in Tonal Music. *Music Perception*, *3*(1), 33‑57. https://doi.org/10.2307/40285321

Trehub, S. E. (2003). The developmental origins of musicality. *Nature Neuroscience*, *6*(7), 669‑673. https://doi.org/10.1038/nn1084

Trehub, S. E., Becker, J., & Morley, I. (2015). Cross-cultural perspectives on music and musicality. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *370*(1664), 20140096. https://doi.org/10.1098/rstb.2014.0096

Vernes, S. C., Janik, V. M., Fitch, W. T., & Slater, P. J. B. (2021). Vocal learning in animals and humans. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *376*(1836), 20200234. https://doi.org/10.1098/rstb.2020.0234

Yamazaki, K., Vo-Ho, V.-K., Bulsara, D., & Le, N. (2022). Spiking Neural Networks and Their Applications : A Review. *Brain Sciences*, *12*(7), Article 7. https://doi.org/10.3390/brainsci12070863

Zandberg, L., Lachlan, R. F., Lamoni, L., & Garland, E. C. (2021). Global cultural evolutionary model of humpback whale song. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *376*(1836), 20200242. https://doi.org/10.1098/rstb.2020.0242

Appendix A: Model code

- **Markov chains**

```
import numpy as np
import matplotlib.pyplot as plt
import pydtmc as mc

#FUNCTION SEQUENCE

def sample_next_state(probabilities, rng):
    return rng.choice(len(probabilities), p=probabilities)

def sample_states(nb_step, starting_probabilities, transition_matrix, rng):
    current_state = sample_next_state(starting_probabilities, rng)
    rk = [current_state]
    for i in range(nb_step - 1):
        next_step = sample_next_state(transition_matrix[current_state, :], rng)
        current_state = next_step
        rk.append(current_state)
    return rk


def sample_iois(sampled_states, distributions, rng):
    tk_samples = []
    for state in sampled_states:
        mu, sigma = distributions[state]
        tk_samples.append(rng.normal(mu, sigma))
    return np.array(tk_samples)

#FUNCTION VISUALISATION

def plot_states(sampled_states, state_names):
    plt.figure()
    plt.plot(sampled_states)
    plt.ylabel('States')
    plt.yticks(range(len(state_names)), state_names)


def plot_chain(transition_matrix, states_names):
    mc.plot_graph(mc.MarkovChain(transition_matrix, states_names))

#FUNCTION ANALYSIS OF THE CHAIN

def equilibrium_distribution(p_transition):
    n_states = p_transition.shape[0]
    A = np.append(
        arr=p_transition.T - np.eye(n_states),
        values=np.ones(n_states).reshape(1, -1),
```

```
        axis=0
    )
    # Moore-Penrose pseudoinverse = (A^TA)^{-1}A^T
    pinv = np.linalg.pinv(A)
    # Return last row
    return pinv.T[-1]
```

- **Spiking neural networks**

```
import nest
import numpy as np
import matplotlib.pyplot as plt

#FUNCTION COMPONENTS

def creation_neuron(I_e=0.0, tau_m=10, tau_syn_ex=2.0, tau_syn_in=2.0):
    neuron = nest.Create('iaf_psc_alpha', 1)
    neuron.I_e = I_e  #neuron constant input current
    neuron.tau_m = tau_m  #neuron time membrane
    neuron.tau_syn_ex = tau_syn_ex  #neuron rise time of the excitatory
                                    #synaptic alpha function
    neuron.tau_syn_in = tau_syn_in  #neuron rise time of the inhibitory
                                    #synaptic alpha function

    return neuron


def creation_nest_poisson_generator(rate_ex, rate_in):
    poisson_generator = nest.Create("poisson_generator", 2)
    poisson_generator[0].rate = rate_ex  #excitatory population rate spike / second
    poisson_generator[1].rate = rate_in  #inhibitory population rate spike / second
    return poisson_generator


def creation_poisson_spike_train(rate, nb_spikes):
    spike_times_s = np.cumsum(np.random.exponential(1 / rate, nb_spikes))
    spike_times_ms = 0.1 + np.round(1000 * spike_times_s,
                                    decimals=int(-np.log10(nest.resolution)))
                                    #+0.1 because a neuron cannot fire at time 0
    poisson_generator = nest.Create("spike_generator",
                                    params={"spike_times": spike_times_ms})
    return poisson_generator


def creation_spike_train(time_event_array):
    spike_generator = nest.Create("spike_generator",
                                    params={"spike_times": time_event_array})
    return spike_generator
```

```
def creation_system(input_neuron, neuron, weight, duration_simulation, delay=0.1):
    system = {
        "input_neuron": input_neuron,
        "neuron": neuron,
        "weight": weight,
        "duration_simulation": duration_simulation,
        "delay": delay
    }
    return system

#FUNCTIONS TIME EVENTS SEQUENCES

#System = [neuron, input_neuron, weight[ex,in], duration_simulation, delay]

def get_sequence_spike_time_list (system, plot = False):
    # system: dict with keys
    # "neuron", "input_neuron", "weight[ex,in]", "duration_simulation", "delay"

    neuron = system["neuron"]
    input_neuron = system["input_neuron"]
    weight = np.array(system["weight"])
    duration_simulation = system["duration_simulation"]
    delay = system["delay"]

    sr_neuron = nest.Create("spike_recorder", 1)
    sr_input_neuron = nest.Create("spike_recorder", 1)
    voltmeter = nest.Create('voltmeter')

    nest.Connect(input_neuron,neuron,syn_spec={"weight": [weight], "delay": delay})
    nest.Connect(input_neuron, sr_input_neuron)
    nest.Connect(neuron, sr_neuron)
    nest.Connect(voltmeter, neuron)

    nest.Simulate(duration_simulation)

    spike_recording_neuron = sr_neuron.get("events")
    spikes_time_neuron = spike_recording_neuron.get('times')

    spike_recording_input_neuron = sr_input_neuron.get("events")
    spikes_time_input_neuron = spike_recording_input_neuron.get('times')

    if plot:
        plt.figure(figsize=(20, 4))

        nest.voltage_trace.from_device(voltmeter)

        plt.xlim(0, 400)
        plt.show()
```

```
    return spikes_time_input_neuron, spikes_time_neuron
```

- **Crickets**

```
import itertools
import numpy as np
import matplotlib.pyplot as plt

from utils import ratios


#FUNCTIONS PHASE RESPONSE CURVE

def parameters_prc(discontinuity=0.7, curviness=0.75, m=0.7, M=1.5, noise_mu=0,
noise_sigma=0.003):
    return [discontinuity, curviness, m, M, noise_mu, noise_sigma]


def prc(phi, discontinuity=0.7, curviness=0.75, m=0.7, M=1.5, noise_mu=0,
noise_sigma=0.003):
    if phi <= discontinuity:
        a = curviness
        b = (M - 1) / discontinuity - a * discontinuity
        return 1 + b * phi + a * phi**2 +  np.random.normal(noise_mu,noise_sigma)
    else:
        a = -curviness
        b = (m - 1) / (1 - discontinuity) - a * (1 - discontinuity)
        return ((1 + b * (1 - phi) + a * (1 - phi)**2) +
                np.random.normal(noise_mu,noise_sigma))


# FUNCTIONS STIMULI SEQUENCES

def generate_isochronous_stimuli(period, start_time=1):
    for i in itertools.count():
        yield start_time + i * period

def generate_poisson_spike_train (period, start_time=1):
    time = start_time
    for i in itertools.count():
        time = time + np.random.exponential(period)
        yield time


#FUNCTIONS TIME EVENTS SEQUENCES

def generate_cricket_chirp_events(stimulus_generator, nb_chirps, relaxation=1,
                                  plot=False, **kwargs):
    nb_chirps = nb_chirps + 35
```

```
t = 0
T = 1.0
chirps = []
stimuli = []
count = 0
next_stimulus = next(stimulus_generator, np.inf)
while count < nb_chirps:
    if t <= next_stimulus:
        if len(chirps) > 0:
            T = relaxation + (1 - relaxation) * (t - chirps[-1])
        chirps.append(t)
        count += 1
        t += T
    else:
        prev_t = chirps[-1]
        phi = (next_stimulus - prev_t) / T
        t = prev_t + T * prc(phi, **kwargs)
        stimuli.append(next_stimulus)
        next_stimulus = next(stimulus_generator, np.inf)

if plot:
    plt.figure(figsize=(15, 4))
    plt.xlabel('Time', fontsize=16)
    for chirp in chirps:
        plt.axvline(chirp)
    for stimulus in stimuli:
        plt.axvline(stimulus, color='r')

chirps = np.array(chirps[35:])
stimuli = np.array(stimuli)
return chirps, stimuli
```