

# DESIGN OF NEUROMORPHIC HARDWARE BASED ON ASYNCHRONOUS CELLULAR AUTOMATON

TAKEDA, Kentaro / 武田, 健太郎

---

(開始ページ / Start Page)

1

(終了ページ / End Page)

177

(発行年 / Year)

2022-03-24

(学位授与番号 / Degree Number)

32675甲第540号

(学位授与年月日 / Date of Granted)

2022-03-24

(学位名 / Degree Name)

博士(工学)

(学位授与機関 / Degree Grantor)

法政大学 (Hosei University)

(URL)

<https://doi.org/10.15002/00025233>

DOCTORAL DISSERTATION REVIEWED BY  
HOSEI UNIVERSITY

DESIGN OF NEUROMORPHIC HARDWARE BASED ON  
ASYNCHRONOUS CELLULAR AUTOMATON

KENTARO TAKEDA



# Preface

This thesis summarizes research into efficient designs for neuromorphic hardware and consists of the following chapters, including the introduction and conclusion in Chapter 1 and Chapter 6, respectively.

Chapter 2 focuses on the cochlea, an important part of the auditory sensory system. Previous investigations have found that the cochlea exhibits a variety of nonlinear characteristics and so can be modeled as a nonlinear active filter. On the other hand, current cochlear implants are mainly based on linear systems, which cannot reproduce nonlinearities. It is anticipated that future cochlear implants will incorporate nonlinear signal processing. Therefore, this chapter proposes novel cochlea models, which can be implemented in small and low-power circuits. The proposed models use limit cycle oscillators based on asynchronous cellular automata (ACA), which exhibit nonlinear bandpass filter characteristics and nonlinear distortion characteristics as observed in the cochlea. An iterative map describing the nonlinear behavior of a model is derived, and it is shown theoretically that the model exhibits an Andronov-Hopf bifurcation. The proposed models were compared with a conventional model, the Hopf cochlea model, by implementation of a field-programmable gate array (FPGA), showing that the proposed models employ fewer circuit elements than the conventional model.

Chapter 3 focuses on the central nervous system involved in gait generation. It is thought that biological neural circuits that produce rhythmic motor patterns, known as central pattern generators (CPGs), which are found in the spinal cord. Various animals such as ants, spiders, and snakes can perform locomotory movements utilizing rhythmic patterns produced by CPGs. Future application of such a mechanism to walking robots is expected. In this context, a locomotion controller would need to be small and low-power to enable incorporation into a mobile robot. This chapter proposes novel CPG models, which can be implemented in small and low-power circuits, using coupled limit-cycle oscillators based on ACA and coupled phase oscillators based on ACA. Numerical and theoretical analyses of synchronization phenomena in the proposed models are presented, alongside an investigation of gait generation for snake-like and hexapod robots. An implementation of the proposed models using FPGAs is shown, demonstrating that FPGA-controlled robots can perform multiple gaits. Furthermore, it is shown that the proposed models requires fewer circuit elements than the conventional model.

In Chapter 4, the focus is on the neural integrator (NN) in the central nervous system, such as the oculomotor system. Recently, artificial neural networks (ANNs), such as deep learning, have been actively investigated. ANN models focus on the behavior of neural circuits at a macroscopic scale (i.e., firing frequency). In contrast, spiking neural networks (SNNs) more closely mimic the dynamical system in neural circuits. As SNNs utilize pulse signals (i.e., action potentials) to transmit information between neurons, as in the actual brain, they are expected to consume less power in electronic circuits than

ANNs. However, the computational cost required to reproduce the complex nonlinear dynamics of spiking neurons is an issue. Thus, this chapter proposes a novel neuron model that can be implemented using small low-power circuits. The proposed model consists of an integrate-and-fire-type ACA-based oscillator and has nonlinear response characteristics matching those of a typical neuron. An iterative map describing the nonlinear behavior of the model is derived, and it is shown theoretically that the model exhibits a saddle-node bifurcation. Moreover, a novel SNN model combining the proposed neuron models, which can be implemented in small and low-power circuits, is proposed. To show that this proposed SNN model can be differentiated into neural circuits that play specific roles in the brain, NIs have been reproduced using the proposed SNN model. NIs have been known as neural circuits found in the oculomotor system; in recent years, they are also thought to be involved in cognitive functions such as working memory and decision making. The proposed NI model and a conventional model are implemented on a field-programmable gate array (FPGA), and comparison shows that the proposed NI model employs fewer circuit elements than the conventional model.

In Chapter 5, the design of a neuron model using a quantum-dot cellular automaton (QCA) is discussed. In accordance with Moore's Law, transistor integration capacity has improved exponentially, but in recent years it is thought that the physical limit of miniaturization is approaching due to quantum effects experienced by nanostructures. QCAs are an emerging technology alternative to MOS-FET-based computing architecture. In this chapter, the circuit implementation of a neuron model using quantum-dot cellular automata is discussed. By using a QCA simulator, an integrate-and-fire-type neuron model based on quantum-dot cellular automata is proposed. A spike-phase map of the proposed model is derived, and the characteristics of spike trains generated by the proposed model are analyzed theoretically. Furthermore, the results are used to investigate a heuristic parameter optimization method for ultra-wideband impulse radio communication, which is one of the potential applications of the proposed model.

**Acknowledgement.** This work was carried out under the direction of my supervisor, Prof. Hiroyuki Torikai at Hosei University. I would like to express my gratitude to him for his persistent support of my work over many years. I would like to express my gratitude to my dissertation committee members; Prof. Arata Kawamura at Kyoto Sangyo University, for his essential and insightful comments in the field of signal processing theory, Prof. Toshimichi Saito at Hosei University, for his essential and insightful comments in the field of nonlinear dynamical system theory, and Prof. Akira Yasuda at Hosei University, for his essential and insightful comments in the field of digital circuit theory.

# Contents

<b>Preface</b>	<b>iii</b>
<b>1 General Introduction</b>	<b>1</b>
1.1 Asynchronous cellular automaton . . . . .	1
1.1.1 Four kinds of dynamical systems . . . . .	1
1.1.2 Property of asynchronous cellular automaton . . . . .	3
1.1.3 Advantage of asynchronous cellular automaton . . . . .	4
1.2 Quantum-dot cellular automaton . . . . .	5
Bibliography . . . . .	6
<b>2 Cochlea Model based on Asynchronous Cellular Automaton</b>	<b>7</b>
2.1 Reproduction of nonlinear compression . . . . .	7
2.1.1 Introduction . . . . .	7
2.1.2 Model description . . . . .	8
2.1.3 Theoretical bifurcation analyses . . . . .	10
2.1.4 Reproductions of biological nonlinear frequency tuning curves . . . . .	13
2.1.5 FPGA implementation and comparison . . . . .	16
2.1.6 Conclusions . . . . .	17
2.2 Reproduction of two-tone distortion products . . . . .	17
2.2.1 Introduction . . . . .	17
2.2.2 Model description . . . . .	19
2.2.3 FPGA implementation and comparison . . . . .	24
2.2.4 Conclusions . . . . .	27
Bibliography . . . . .	28
<b>3 CPG Model based on Asynchronous Cellular Automaton</b>	<b>33</b>
3.1 Snake-like Robot Controlled by Coupled Limit Cycle Oscillators . . . . .	33
3.1.1 Introduction . . . . .	33
3.1.2 Model description . . . . .	34
3.1.3 Hardware implementation and comparison . . . . .	41
3.1.4 Conclusions . . . . .	42
3.2 Hexapod Robot Controlled by Coupled Limit Cycle Oscillators . . . . .	43
3.2.1 Introduction . . . . .	43
3.2.2 Asynchronous CA oscillator . . . . .	47
3.2.3 Analyses of small network of CA oscillators as preparations to design CPG . . . . .	56
3.2.4 Design of CPG consisting of network of CA oscillators . . . . .	61

3.2.5	Implementation and comparison . . . . .	70
3.2.6	Conclusions . . . . .	74
3.3	Hexapod Robot Controlled by Coupled Phase Oscillators . . . . .	75
3.3.1	Introduction . . . . .	75
3.3.2	Model description . . . . .	77
3.3.3	Synchronization phenomena in presented CPG model . . . . .	80
3.3.4	FPGA Implementation . . . . .	89
3.3.5	Discussion . . . . .	93
3.3.6	Conclusions . . . . .	98
	Bibliography . . . . .	99
<b>4</b>	<b>Neural Integrator Model based on Asynchronous Cellular Automaton</b>	<b>105</b>
4.1	Spiking Neuron Model . . . . .	105
4.1.1	Introduction . . . . .	105
4.1.2	Model description . . . . .	108
4.1.3	Theoretical bifurcation analyses . . . . .	116
4.1.4	Design of neuron-like current-frequency curves utilizing theoretical bifurcation analyses . . . . .	119
4.1.5	Implementation and Comparison . . . . .	127
4.1.6	Conclusions . . . . .	130
4.2	Spiking Neural Network Model . . . . .	133
4.2.1	Introduction . . . . .	133
4.2.2	Reduced ODE model for NI . . . . .	134
4.2.3	Hardware-oriented model for NI . . . . .	135
4.2.4	Implementation and comparison . . . . .	139
4.2.5	Discussion . . . . .	142
4.2.6	Conclusions . . . . .	143
	Bibliography . . . . .	143
<b>5</b>	<b>Neural Spike-Train Generator Model based on Quantum-dot Cellular Au- tomaton</b>	<b>149</b>
5.1	Neural Spike-train Generator Suitable for UWB-IR Applications . . . . .	149
5.1.1	Introduction . . . . .	149
5.1.2	Model description . . . . .	151
5.1.3	QCA implementation of the proposed spike-train generator . . . . .	156
5.1.4	Generation of various spike-trains and parameter tuning for UWB application . . . . .	162
5.1.5	Conclusions . . . . .	167
	Bibliography . . . . .	168
<b>6</b>	<b>Overall Conclusions and Future Work</b>	<b>171</b>
6.1	Overall conclusions . . . . .	171
6.2	Future work . . . . .	172
	<b>Publication List</b>	<b>174</b>

# Chapter 1

## General Introduction

Neuromorphic engineering is an emerging field which attempts to use electronic circuits to physically mimic biological systems such as the central nervous and sensory system. These electronic circuits, known as neuromorphic hardware, are expected to replace conventional computer architectures. However, biological systems generally exhibit strong nonlinearity, which is challenging to implement in integrated circuits [1]. Therefore, research is needed to investigate circuit implementation methods for designing small and low-power circuits for use in brain-inspired computing devices, bio-inspired robots, and neural prosthetic devices. This thesis aims to investigate efficient designs of neuromorphic hardware to mimic these biological systems, in particular presenting hardware-efficient biological system models based on asynchronous cellular automata (ACA) and quantum-dot cellular automata (QCA). This chapter will now introduce key concepts for the ACA and QCA used throughout this thesis.

### 1.1 Asynchronous cellular automaton

#### 1.1.1 Four kinds of dynamical systems

Many mathematical and electronic circuit models of biological systems have been proposed to date [2–4]. As summarized in Table 1.1.1, such biological system models can be classified into the following four classes based on the continuity and discontinuity of state variables and time.

*Class CTCS.* A nonlinear differential equation model of a biological system that has continuous time and continuous states (CTCS). Such a biological system model can be implemented by an analog nonlinear circuit.

*Class DTCS.* A nonlinear difference equation model of a biological system that has discrete time and continuous states (DTCS). Such a biological system model can be implemented by a switched capacitor circuit.

*Class DTDS.* A numerical integration model of a biological system that has discrete time and discrete states (DTDS). Such biological system models can be implemented by either a digital processor or a sequential logic circuit.

*Class CTDS.* An ACA of a biological system that has continuous time and discrete states (CTDS). Such a biological system model can be implemented by an asynchronous sequential logic circuit.

As a preparation to consider  $n$ -dimensional biological system models, for simplicity, two-dimensional biological system models are explained in the following part. Biological system

Table 1.1.1: Four biological system modeling approaches.

	Continuous States	Discrete States
Continuous Time	<p><i>Class CTCS</i></p> <p><b>Nonlinear Ordinary Differential Equation</b></p> <p>Analog Nonlinear Circuit</p>	<p><i>Class CTDS</i></p> <p><b>Asynchronous Cellular Automaton</b></p> <p>Asynchronous Sequential Logic Circuit</p>
Discrete Time	<p><i>Class DTCS</i></p> <p><b>Nonlinear Difference Equation</b></p> <p>Switched Capacitor Circuit</p>	<p><i>Class DTDS</i></p> <p><b>Numerical Integration /Cellular Automaton</b></p> <p>CPU + Memory /Sequential Logic Circuit</p>

models belonging to class CTCS are described by the following differential equation.

$$\begin{cases} \frac{dX_1(t)}{dt} = F_1(X_1(t), X_2(t)), \\ \frac{dX_2(t)}{dt} = F_2(X_1(t), X_2(t)), \end{cases} \quad (1.1.1)$$

$$X_i \in \mathbb{R}, t \in \mathbb{R}, F_i : \mathbb{R}^2 \rightarrow \mathbb{R}.$$

Here state variables correspond to capacitors and vector field functions are reproduced by the nonlinearity of circuit elements such as MOSFETs. The dynamic system behavior is induced by the characteristics of circuit elements with memory; i.e. capacitors  $Cdv(t)/dt = i(t)$  and inductors  $Ldi/dt = v(t)$ . Biological system models belonging to class DTCS are described by the following difference equation.

$$\begin{cases} X_1(n+1) = X_1(n) + F_1(X_1(n), X_2(n)), \\ X_2(n+1) = X_2(n) + F_2(X_1(n), X_2(n)), \end{cases} \quad (1.1.2)$$

$$X_i \in \mathbb{R}, n \in \mathbb{Z}, F_i : \mathbb{R}^2 \rightarrow \mathbb{R}.$$

Here state variables also correspond to capacitors and vector field functions are again reproduced by the nonlinearity of circuit elements such as MOSFETs. The dynamics system behavior is induced by iterative switching for capacitor charging, following the vector field functions. Biological system models belonging to class DTDS are described by the following difference equation.

$$\begin{cases} X_1(n+1) = X_1(n) + F_1(X_1(n), X_2(n)), \\ X_2(n+1) = X_2(n) + F_2(X_1(n), X_2(n)), \end{cases} \quad (1.1.3)$$

$$X_i \in \mathbb{F} \equiv \{X_{min}, \dots, X_{max}\}, X_{min} \in \mathbb{Z}, X_{max} \in \mathbb{Z}, n \in \mathbb{Z}, F_i : \mathbb{F}^2 \rightarrow \mathbb{F}.$$

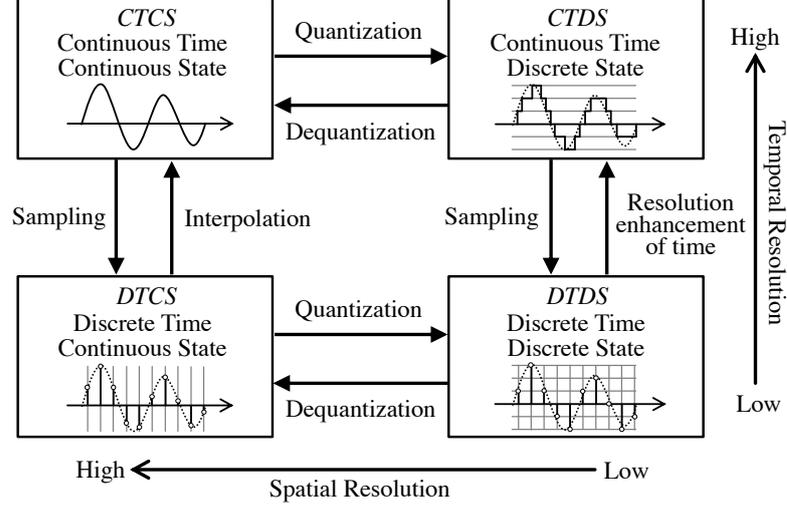


Figure 1.1.1: Relationships between the classes CTCS, DTCS, DTDS, and CTDS.

where  $X_{min}$  and  $X_{max}$  are constants. Here state variables correspond to flip-flops and the vector field functions are reproduced by logic gates or arithmetic processing units. The dynamic system behavior is induced by iterative switching for state transitions according to the vector field functions.

Most conventional biological system models belong to either CTCS, DTCS, or DTDS classes. However, this thesis focuses on the class CTDS described as follows.

$$\begin{cases} X_1((n+1)h_1 + t_1^{(0)}) = X_1(nh_1 + t_1^{(0)}) + F_1(X_1(nh_1 + t_1^{(0)}), X_2(nh_1 + t_1^{(0)})), \\ X_2((n+1)h_2 + t_2^{(0)}) = X_2(nh_2 + t_2^{(0)}) + F_2(X_1(nh_2 + t_2^{(0)}), X_2(nh_2 + t_2^{(0)})), \end{cases} \quad (1.1.4)$$

$$X_i \in \mathbb{F}, h_i \in (0, \infty), t_i^{(0)} \in [0, h_i), n \in \mathbb{Z}, F_i : \mathbb{F}^2 \rightarrow \mathbb{F}.$$

where  $h_i$  and  $t_i^{(0)}$  are constants. Here state variables also correspond to flip-flops and the vector field functions are again reproduced by logic gates or arithmetic processing units. The dynamic system behavior is induced by asynchronous iterative switching for state variable transitions according to the vector field functions. Fig. 1.1.1 shows the relationships between the classes CTCS, DTCS, DTDS, and CTDS. As can be seen from this figure, class CTDS biological system models make more efficient use of temporal axes than DTDS biological system models do.

### 1.1.2 Property of asynchronous cellular automaton

Depending on the parameters  $h_1$  and  $h_2$ , the system in Eq. (1.1.4) is regarded as a hybrid dynamical system with continuous time and discrete states as follows. State transitions of the variables  $X_1$  and  $X_2$  in Eq. (1.1.4) are triggered by the clocks  $C_i \in \{0, 1\}$  with periods

$h_i$ , defined as

$$C_i(t) = \sum_{n=0}^{\infty} \delta(t - nh_i - t_i^{(0)}), \quad (1.1.5)$$

where  $\delta : \mathbb{R} \rightarrow \{1, 0\}$  is a unit impulse function defined by  $\delta(t) = 1$  if  $t = 0$ , and  $\delta(t) = 0$  if  $t \neq 0$ . Let  $\phi_i \in [0, h_i)$  be the following restricted phases of the clocks  $C_i$ ,

$$\phi_i(t) = t + t_i^{(0)} \pmod{h_i}. \quad (1.1.6)$$

Then, using the restricted phases  $\phi_i$ , the definition of the clocks  $C_i$  can be rewritten as

$$C_i(t) = \begin{cases} 1 & \text{if } \phi_i(t) = 0, \\ 0 & \text{otherwise,} \end{cases} \quad (1.1.7)$$

The dynamics of the restricted phase  $\phi_2$  can be described by the following iterative map  $\sigma : [0, h_2) \rightarrow [0, h_2)$

$$\phi_2(t + h_1) = \sigma(\phi_2(t)) \equiv \phi_2(t) + h_1 \pmod{h_2}, \quad (1.1.8)$$

where  $h_1 < h_2$ . Furthermore, using the restricted phase  $\phi_2$ , Eq. (1.1.4) can be rewritten as

$$\begin{cases} X_1(t + h_1) = X_1(t) + F_1(X_1(t), X_2(t + h_1)), \\ X_2(t + h_1) = X_2(t) + F_2(X_1(t), X_2(t))s(\phi_2(t)), \end{cases} \quad (1.1.9)$$

where the function  $s : [0, h_2) \rightarrow \{1, 0\}$  is defined as

$$s(\phi_2) \equiv \begin{cases} 1 & \text{if } \phi_2 \geq h_2 - h_1, \\ 0 & \text{if } \phi_2 < h_2 - h_1. \end{cases} \quad (1.1.10)$$

Note that the system in Eq. (1.1.4) is now represented by Eqs. (1.1.8) and (1.1.9). In addition, the system in Eqs. (1.1.8) and (1.1.9) is equivalent to a Poincaré map [5] of Eq. (1.1.4) with a Poincaré section  $\{(X_1, X_2, \phi_1, \phi_2) \mid X_1 \in \mathbb{Z}, X_2 \in \mathbb{Z}, \phi_1 = 0, \phi_2 \in [0, h_2)\}$ . The system in Eqs. (1.1.8) and (1.1.9) is considered to belong to class DTDS if  $h_1/h_2 \in \mathbb{Q}$  and to class CTDS if  $h_1/h_2 \in \mathbb{R} \setminus \mathbb{Q}$ . In the case of  $h_1/h_2 \in \mathbb{Q}$ , Eq. (1.1.8) is equivalent to a rational rotation [6]. Thus, the system can be regarded as belonging to the DTDS class because  $\phi_2$  can be defined by a finite set. Conversely, in the case of  $h_1/h_2 \in \mathbb{R} \setminus \mathbb{Q}$ , Eq. (1.1.8) is equivalent to an irrational rotation [6]. Then, the system is regarded as belonging to the CTDS class because  $\phi_2$  should be defined by continuous states. Note that  $\phi_2$  implies the restricted phase of the clock  $C_2$ , that is to say, time.

### 1.1.3 Advantage of asynchronous cellular automaton

ACA-based class CTDS biological system models have the advantage that they can produce smoother nonlinear vector fields than class DTDS models can. Fig. 1.1.2 shows the concept and advantage of the ACA-based biological system modeling. Assume that the state variables  $X_1$  and  $X_2$  move one cell ahead when the clock signals  $C_1$  and  $C_2$  with the periods  $h_1$  and  $h_2$  rise, respectively. In Figs. 1.1.2(a) and (b), the period ratios  $h_1/h_2$

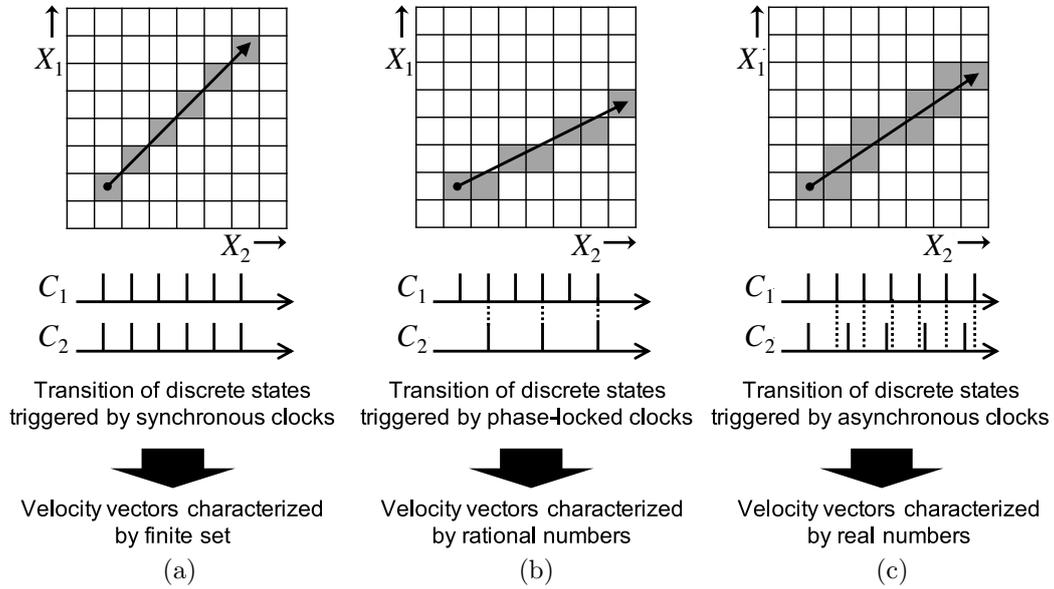


Figure 1.1.2: Concept and advantages of an ACA-based biological system modeling approach.

are 1 and rational, and thus state transitions are triggered synchronously. In these cases, the velocity vectors are characterized by the finite set and rational numbers, respectively. However, in Fig. 1.1.2(c), the period ratio  $h_1/h_2$  is irrational, and thus state transitions are triggered asynchronously. In this case, the velocity vectors are characterized by real numbers. Hence, the asynchronous state transitions make the nonlinear vector fields smooth, unlike the synchronous state transitions in Figs. 1.1.2(a) and (b). Therefore, due to having smooth nonlinear vector fields, ACA-based biological system models can be implemented in digital circuits using fewer circuit elements than are needed for DTDS-class models.

The rest of the thesis will investigate applying ACA-based biological system models to specific areas. Chapter 2 focuses on a cochlea model, Chapter 3 focuses on a central pattern generator model, and Chapter 4 focuses on a neural integrator model.

Alternatively, biological system modeling can be based on quantum-dot cellular automata (QCAs), and this option is also investigated in this thesis as a possible model with low hardware resources requirements for implementation.

## 1.2 Quantum-dot cellular automaton

QCAs are an emerging technology alternative to MOSFET based computing architecture proposed by C. S. Lent [7]. The basic element of the QCA consists of four quantum dots arranged in a square, two of which being charged electrons, as shown in Fig. 1.2.1. Due to Coulomb forces, two electrons can tunnel among the quantum dots when a potential barrier on the dots is low, where diagonally located states (see Fig. 1.2.1) of the two electrons are stable. If the two electrons are placed at the upper right and lower left dot locations, the cell is usually defined as logic “1”, whereas if the two electrons are placed at the upper left

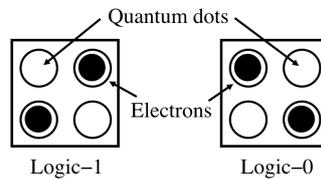


Figure 1.2.1: QCA cell.

and lower right dot locations, the cell is usually defined as logic “0”. Thus, the QCA cell has two stable binary states, can function as memory device, and these binary states can be controlled by neighboring cells via Coulomb forces. Therefore, QCA cells can transmit binary states to an adjacent cell without using the current. Many combinational circuits and sequential circuits using QCA cells have been proposed, [8], but to date no attempts exist to build a biological system based on QCA. Therefore, in Chapter 5, this thesis investigates using QCAs to implement a neural spike-train generator.

## Bibliography

- [1] E. M. Izhikevich, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. Cambridge, MA, USA: MIT Press, 2006.
- [2] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, “A survey of neuromorphic computing and neural networks in hardware,” *arXiv preprint arXiv:1705.06963*, 2017.
- [3] J. Yu, M. Tan, J. Chen, and J. Zhang, “A Survey on CPG-Inspired Control Models and System Implementation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 441–456, 2014.
- [4] R. F. Lyon, *Human and Machine Hearing*. New York, NY, USA: Cambridge University Press, 2017.
- [5] Y. A. Kuznetsov, *Elements of Applied Bifurcation Theory*, 3rd ed., ser. Applied Mathematical Sciences, S. S. Antman, J. E. Marsden, and L. Sirovich, Eds. New York, NY, USA: Springer, 2004, vol. 112.
- [6] R. Devaney, *An Introduction to Chaotic Dynamical Systems*, 2nd ed. Florida, USA: CRC Press, 2003.
- [7] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernstein, “Quantum cellular automata,” *Nanotechnology*, vol. 4, no. 1, pp. 49–57, 1993.
- [8] A. H. Majeed, M. S. Zainal, and E. Alkaldy, “Quantum-dot cellular automata: Review paper,” *International Journal of Integrated Engineering*, vol. 11, no. 8, pp. 143–158, 2019.

# Chapter 2

## Cochlea Model based on Asynchronous Cellular Automaton

### 2.1 Reproduction of nonlinear compression<sup>1</sup>

#### 2.1.1 Introduction

Biological cochleae have a wide variety of responses to sound stimuli [1–3], e.g., Fig. 2.1.1 shows typical response characteristics (called frequency tuning curves) of multiple species [2, 3]. In this figure, the horizontal axis is a frequency of a single tone sound stimulation. The vertical axis is the minimum sound pressure level leading to a pre-determined activity level of the cochlea. The curves have the following features (F1)–(F4): (F1) Some curves have very steep positive slopes and relatively gentle negative slopes as indicated by (i) and (ii); (F2) Some curves have sudden changes of slopes as indicated by (iii); (F3) The curves are relatively symmetric compared to the curves of the mammalian in (a); and (F4) The minimum values of the curves depend on corresponding center frequencies as illustrated by the dotted curve (iv). In order to analyze and reproduce such responses, many mathematical and electronic circuit models of cochleae have been presented [4–10], where applications of such cochlear models and circuits include the cochlear implant. Concerning modeling methods of biological systems, there exist four approaches depending on continuousness of time and state as follows. The first approach is to model a biological system by using a nonlinear ordinary differential equation (ODE), which has a continuous time and continuous states (CTCS). Such a CTCS model can be implemented by a nonlinear electronic circuit. The second approach is to model a biological system by using a nonlinear difference equation, which has a discrete time and continuous states (DTCS). Such a DTCS model can be implemented by a switched capacitor circuit. The third approach is to model a biological system by using a numerical integration in a fixed-point or a floating-point number format or by a cellular automaton, which have discrete times and discrete states (DTDS). Such DTDS models can be implemented by a digital signal processor or a sequential logic. Most biological system modeling approaches are belonging to one of the above three ones. On the other hand, our group has been developing the *fourth missing approach*, i.e., to model a biological system by using an asynchronous cellular automaton, which has a continuous (state transition) time and discrete states (CTDS), e.g., [10–12]. Such a CTDS model can

---

<sup>1</sup>© 2017 IEEE. Reprinted, with permission, from Kentaro Takeda and Hiroyuki Torikai, A Novel Hardware-Efficient Cochlea Model based on Asynchronous Cellular Automaton Dynamics: Theoretical Analysis and FPGA Implementation, IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 9, pp. 1107–1111, Sep. 2017.

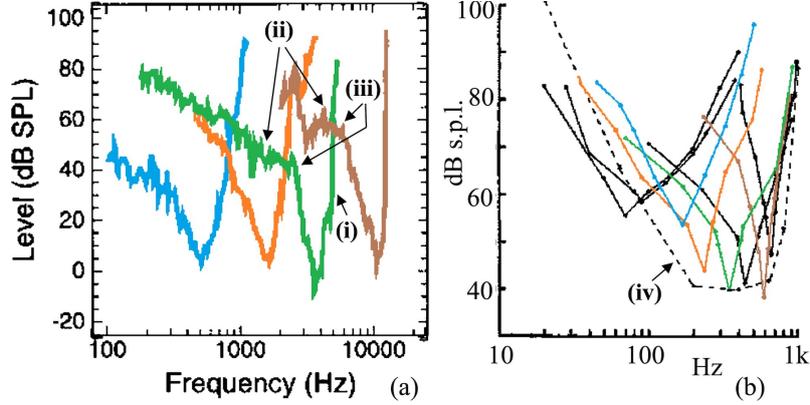


Figure 2.1.1: Frequency tuning curves of biological cochleae. (a) Chinchilla (mammalian) adapted from [2]. (b) Turtle (reptilian) adapted from [3].

be implemented by an asynchronous sequential logic. In this paper, a novel cochlear model based on the asynchronous cellular automaton is presented. Theoretical analyses show that the model can mimic a nonlinear vector field of one of standard ODE cochlea models [4–6]. It is then shown that the model can reproduce typical features of the biologically measured frequency tuning curves. Furthermore, the model is implemented in a field programmable gate array and experiments validate the reproductions of the biological frequency tuning curves. It is also shown that the presented model consumes fewer hardware resources compared to a numerical integration formula of the ODE cochlea model. Novelties of this paper include the following points. (a) The theoretical analyses of the model are presented in this paper for the first time. (b) The frequency tuning curves of the model are analyzed in detailed in this paper for the first time. (c) The experimental tuning curves are measured in this paper for the first time. Very preliminary and limited results of this paper can be found in our conference proceedings [13, 14].

## 2.1.2 Model description

In this section, a novel cochlea model based on an asynchronous cellular automaton dynamics is presented. Let  $t \in \mathbf{R}$  be a continuous time. As shown in Fig. 2.1.2, the model has the following two internal clocks  $C_X(t)$  and  $C_Y(t)$ .

$$C_X(t) = \begin{cases} 1 & \text{if } \Theta(t) = 0, \\ 0 & \text{otherwise,} \end{cases} \quad \Theta(t) = t - \theta \pmod{T_X},$$

$$C_Y(t) = \begin{cases} 1 & \text{if } \Phi(t) = 0, \\ 0 & \text{otherwise,} \end{cases} \quad \Phi(t) = t - \phi \pmod{T_Y},$$

where  $T_X > 0$  and  $T_Y > 0$  are periods,  $\Theta(t) \in [0, T_X)$  and  $\Phi(t) \in [0, T_Y)$  are phases, and  $\theta$  and  $\phi$  are initial phases. In this paper, the clock generators are assumed to be uncoupled, to have different periods, and to be asynchronous. As shown in Fig. 2.1.2, the model has the following four discrete states.

$$\begin{aligned} X &\in \mathbf{Z}_N = \{0, \dots, N-1\}, & Y &\in \mathbf{Z}_N, \\ P &\in \mathbf{Z}_M = \{0, \dots, M-1\}, & Q &\in \mathbf{Z}_M, \end{aligned}$$

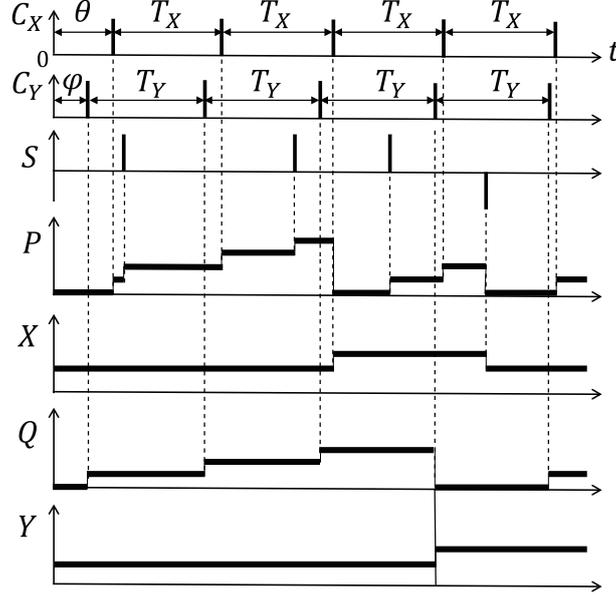


Figure 2.1.2: Timing chart of the proposed model.

where  $N \geq 2$  and  $M \geq 2$  are integers, which determine a resolution of a state space  $\mathbf{Z} = \{(X, Y, P, Q) \mid X \in \mathbf{Z}_N, Y \in \mathbf{Z}_N, P \in \mathbf{Z}_M, Q \in \mathbf{Z}_M\}$  of the model. In order to design a nonlinear vector field, the following two functions  $f_X : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbf{R}$  and  $f_Y : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbf{R}$  are introduced.

$$\begin{aligned} f_X(X, Y) &= \delta l(X - N/2) - \omega l(Y - N/2) - l^3(X - N/2)((X - N/2)^2 + (Y - N/2)^2), \\ f_Y(X, Y) &= \omega l(X - N/2) + \delta l(Y - N/2) - l^3(Y - N/2)((X - N/2)^2 + (Y - N/2)^2), \end{aligned}$$

where  $l \in \mathbf{R}^+ = \{r \in \mathbf{R} \mid r \geq 0\}$ ,  $\delta \in \mathbf{R}$ , and  $\omega \in \mathbf{R}^+$  are parameters. Note that the functions  $f_X$  and  $f_Y$  are not implemented in a circuit but are used to design a nonlinear vector field of the model. Using the functions  $f_X$  and  $f_Y$ , the following discrete functions  $F_X : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbf{Z}_M^\pm = \{-(M-1), \dots, 0, \dots, M-1\}$  and  $F_Y : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbf{Z}_M^\pm$  are designed.

$$\begin{aligned} F_X(X, Y) &= \text{Int}(l/(f_X(X, Y)T_X)), \\ F_Y(X, Y) &= \text{Int}(l/(f_Y(X, Y)T_Y)), \end{aligned}$$

where  $\text{Int}(r)$  represents the integer part of a real number  $r$  and the functions  $F_X$  and  $F_Y$  are saturated at  $M-1$  and  $-(M-1)$ . As shown in Fig. 2.1.2, the asynchronous clocks  $C_X(t)$  and  $C_Y(t)$  trigger the following asynchronous transitions of the discrete states  $(X, Y, P, Q)$ .

If  $C_X(t) = 1$ , then

$$P(t_+) := \begin{cases} P(t) + 1 & \text{if } P(t) < |F_X|, \\ 0 & \text{otherwise,} \end{cases} \quad (2.1.1)$$

$$X(t_+) := X(t) + \text{sgn}(F_X) \quad \text{if } P(t) \geq |F_X|,$$

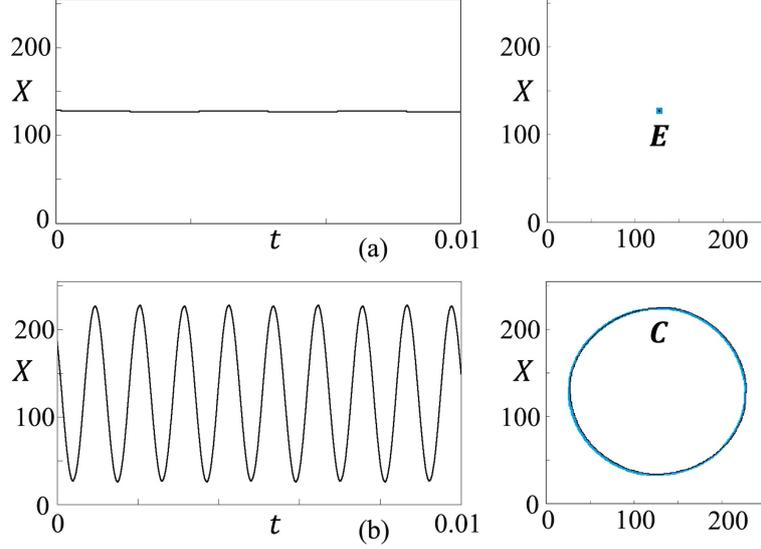


Figure 2.1.3: Typical behaviors of the presented model in steady states.  $(N, M\omega, l, T_X, T_Y) = (256, 256, 2\pi \times 10^3, 0.25, 1 \times 10^{-7}, 1.1 \times 10^{-7})$ . (a)  $\delta = -10$ . (b)  $\delta = 700$ .

If  $C_Y(t) = 1$ , then

$$Q(t_+) := \begin{cases} Q(t) + 1 & \text{if } Q(t) < |F_Y|, \\ 0 & \text{otherwise,} \end{cases} \quad (2.1.2)$$

$$Y(t_+) := Y(t) + \text{sgn}(F_Y) \quad \text{if } Q(t) \geq |F_Y|,$$

where  $\text{sgn}(x) = 1$  if  $x \geq 0$  and  $\text{sgn}(x) = -1$  otherwise and the discrete states  $X$  and  $Y$  ( $P$  and  $Q$ ) are saturated at 0 and  $N - 1$  (0 and  $M - 1$ ). Also, the symbol " $t_+$ " represents " $\lim_{\epsilon \rightarrow +0} t + \epsilon$ " and the symbol " := " represents an "instantaneous state transition" hereafter. Fig. 2.1.3 shows typical behaviors of the model in steady states. In Fig. 2.1.3(a), the state vector  $(X, Y)$  stays in the small blue region  $\mathbf{E}$  around  $(X, Y) \simeq (N/2, N/2)$ . During a transient (not shown in Fig. 2.1.3(a)), the state vector  $(X, Y)$  approaches the region  $\mathbf{E}$ . In Fig. 2.1.3(b), the value of the parameter  $\delta$  is changed from that in Fig. 2.1.3(a). In this case, the state vector  $(X, Y)$  stays in the ring-shaped blue region  $\mathbf{C}$ . During a transient (not shown in Fig. 2.1.3(b)), the state vector  $(X, Y)$  approaches the region  $\mathbf{C}$ . Hence, the model undergoes a *bifurcation phenomenon* when the value of the parameter  $\delta$  is changed. In order to characterize the bifurcation phenomenon, let us introduce a radius  $r = (\max X - \min X)/2$  of the state  $X$  in a steady state. Fig. 2.1.4 shows the characteristics of the radius  $r$  of  $X$  for the parameter  $\delta$ , where the black graph (corresponding to the black orbits in Fig. 2.1.3) is numerically obtained by simulating Eqs. (2.1.1) and (2.1.2). On the other hand, the blue bars in Fig. 2.1.4 (corresponding to the blue regions  $\mathbf{E}$  and  $\mathbf{C}$  in Fig. 2.1.3) are calculated by using theoretical bifurcation analyses in the next section.

### 2.1.3 Theoretical bifurcation analyses

From a view point of dynamical system theory, the presented model has six state variables  $(X, P, \Theta, Y, Q, \Phi)$ . In order to analyze the bifurcation, let us consider the following

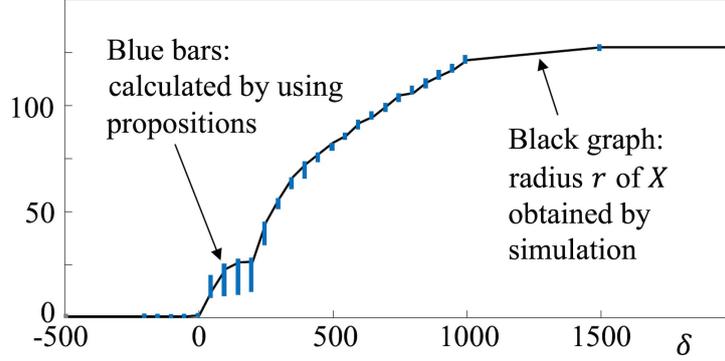


Figure 2.1.4: Bifurcation diagram of the radius  $r$  of  $X$ .  $(N, M, \omega, l, T_X, T_Y) = (256, 256, 2\pi \times 10^3, 0.25, 1 \times 10^{-7}, 1.1 \times 10^{-7})$ .

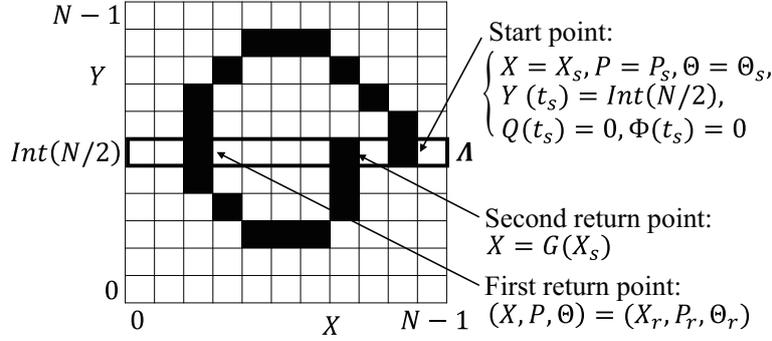


Figure 2.1.5: The region  $\Lambda$  and the map  $G$  used for theoretical analyses.

initial condition of the three states  $(Y, Q, \Phi)$  at  $t = t_s$ .

$$Y(t_s) = \text{Int}(N/2), \quad Q(t_s) = 0, \quad \Phi(t_s) = 0,$$

where note that  $\Phi(t_s) = 0$  implies the clock  $C_Y$  is 1 at  $t = t_s$ . Let the other three states  $(X, P, \Theta)$  at  $t = t_s$  be denoted by

$$X(t_s) = X_s, \quad P(t_s) = P_s, \quad \Theta(t_s) = \Theta_s,$$

where the subscript "s" implies "start point." Now let the state vector  $(X, P, \Theta, Y, Q, \Phi)$  starting from the initial condition  $(X_s, P_s, \Theta_s, \text{Int}(N/2), 0, 0)$  be projected onto the  $(X, Y)$ -plane as shown in Fig. 2.1.5. In the  $(X, Y)$ -plane, let us define a horizontal region  $\Lambda = \{(X, Y) | Y = \text{Int}(N/2)\}$ . Then the trajectory of the state vector  $(X, Y)$  starting from the start point  $(X_s, Y_s)$  in the region  $\Lambda$  has the following four possibilities.

**Case 1.** As shown in Fig. 2.1.5, the state vector  $(X, Y)$  departs from the region  $\Lambda$  and returns into the region  $\Lambda$  at a finite time, say  $t = t_r < \infty$ . In this case, let the three states  $(X, P, \Theta)$  at this return moment  $t_r$  be denoted by  $(X_r, P_r, \Theta_r)$  as shown in Fig. 2.1.5, where the subscript "r" implies "first return point."

**Case 2.** The state vector  $(X, Y)$  stays in the region  $\Lambda$  and transits in the region  $\Lambda$ . In this case, let  $(X_r, P_r, \Theta_r) = (X(t_y), P(t_y), \Theta(t_y))$ , where  $t_y$  is the moment when the clock  $C_Y$

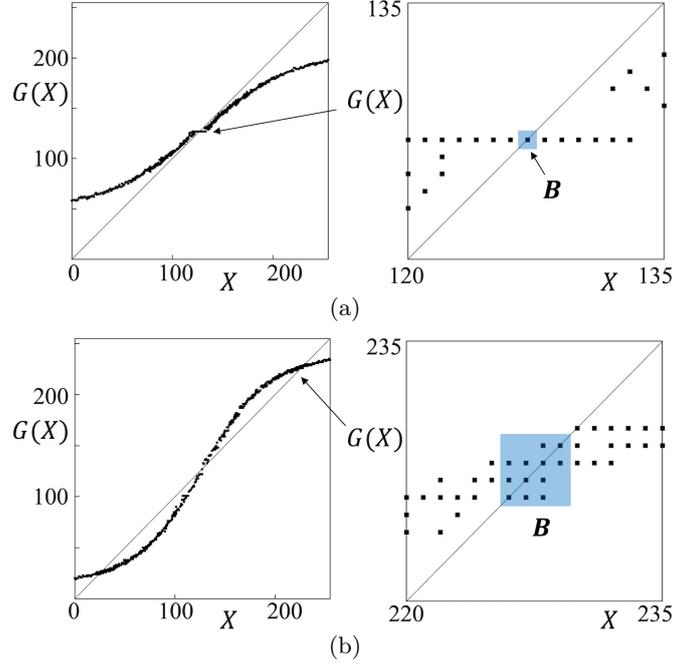


Figure 2.1.6: Tools for theoretical analyses: multi-valued map  $G$  (left), its enlargement (right), and fixed set  $\mathbf{B}$  (blue box).  $(N, M, \omega, l, T_X, T_Y) = (256, 256, 2\pi \times 10^3, 0.25, 1 \times 10^{-7}, 1.1 \times 10^{-7})$ . (a)  $\delta = -10$ . (b)  $\delta = 700$ .

firstly becomes 1 after the transition of  $X$ .

**Case 3.** The state vector  $(X, Y)$  does not transit forever. In this case, let  $(X_r, P_r, \Theta_r) = (X(t_z), P(t_z), \Theta(t_z))$ , where  $t_z > t_s$  is the moment when the clock  $C_Y$  firstly becomes 1.

**Case 4.** The state vector  $(X, Y)$  departs from the region  $\mathbf{\Lambda}$  and never returns into the region  $\mathbf{\Lambda}$ . In this case, let  $(X_r, P_r, \Theta_r) = (\emptyset, \emptyset, \emptyset)$ , where  $\emptyset$  denotes the empty set. Then we define the following maps  $g_X : \mathbf{Z}_N \times \mathbf{Z}_M \times [0, T_X] \rightarrow \mathbf{Z}_N \cup \{\emptyset\}$ ,  $g_P : \mathbf{Z}_N \times \mathbf{Z}_M \times [0, T_X] \rightarrow \mathbf{Z}_M \cup \{\emptyset\}$ , and  $g_\Theta : \mathbf{Z}_N \times \mathbf{Z}_M \times [0, T_X] \rightarrow \times [0, T_X] \cup \{\emptyset\}$ .

$$\begin{cases} g_X(X_s, P_s, \Theta_s) = X_r, \\ g_P(X_s, P_s, \Theta_s) = P_r, \\ g_\Theta(X_s, P_s, \Theta_s) = \Theta_r. \end{cases} \quad (2.1.3)$$

Using the maps  $(g_X, g_P, g_\Theta)$ , the following multi-valued map  $G$  is defined (see Fig. 2.1.5).

$$G(X_s) = g_X(g_X(X_s, P, \Theta), g_P(X_s, P, \Theta), g_\Theta(X_s, P, \Theta)) \\ \text{for all } P \in \mathbf{Z}_N, \Theta \in [0, T_X],$$

where  $G(X_s) = \emptyset$  if  $g_X(X_s, P, \Theta) = \emptyset$ . Figs. 2.1.6(a) and (b) show the map  $G$  corresponding to Figs. 2.1.3(a) and (b), respectively. Let us give the following definition (see Fig. 2.1.6).

**Definition 1:** A subset  $\mathbf{B} = \{X_f, X_f + 1, \dots, X_f + \epsilon\}$  of  $\mathbf{Z}_N$  is said to be a fixed set with size  $\epsilon \in \{1, 2, \dots, N - 1\}$  of the multi-valued map  $G$  if each  $G(X) \in \mathbf{B}$  for all  $X \in \mathbf{B}$ .

In the following analyses, the subset  $\mathbf{B}$  is assumed to have its possible minimum size  $\epsilon$ .

Then we have the following propositions, where their proofs are omitted since they are straightforward from the definitions of the multi-valued map  $G$  and the fixed set  $\mathbf{B}$  (i.e., the definitions of  $G$  and  $\mathbf{B}$  *per se* are important results of this paper).

**Proposition 1** (Equilibrium): Suppose the multi-valued map  $G(X)$  has a fixed set  $\mathbf{B}$  and suppose  $g_X(X, P, \Phi) \in \mathbf{B}$  for all  $X \in \mathbf{B}$ ,  $P \in \mathbf{Z}_N$ , and  $\Phi \in [0, T_X)$ . Then there exists a subset  $\mathbf{E}$  (which is typically small) in the  $(X, Y)$ -plane  $\mathbf{Z}_N \times \mathbf{Z}_N$  from which the state vector  $(X, Y)$  does not escape.

For example, the map  $G$  in Fig. 2.1.6(a) satisfies the Proposition 1 and thus there exists a subset  $\mathbf{E}$  in the  $(X, Y)$ -plane as shown in Fig. 2.1.3(a). Note that the subset  $\mathbf{E}$  corresponds to an equilibrium point of a nonlinear ODE.

**Proposition 2** (Oscillation): Let the map  $G$  has a fixed set  $\mathbf{B}$  and suppose  $g_X(X, P, \Phi) \notin \mathbf{B}$  for all  $X \in \mathbf{B}$ ,  $P \in \mathbf{Z}_N$ , and  $\Phi \in [0, T_X)$ . Then there exists a ring-shaped subset  $\mathbf{C}$  (which is typically thin) in the  $(X, Y)$ -plane  $\mathbf{Z}_N \times \mathbf{Z}_N$  from which the state vector  $(X, Y)$  does not escape.

For example, the map  $G$  in Fig. 2.1.6(b) satisfies the Proposition 2 and thus there exist a ring-shaped subset  $\mathbf{C}$  in the  $(X, Y)$ -plane as shown in Fig. 2.1.3(b). Note that the subset  $\mathbf{C}$  corresponds to an invariant circle of a nonlinear ODE.

**Remark** (Implications of the propositions): In Fig. 2.1.4, the blue bars are calculated by using the fixed sets  $\mathbf{B}$ . It can be seen that the blue bars cover the black graph (the radius  $r$  obtained by the numerical simulation), i.e., the theoretical analyses give a necessary region in which the true bifurcation diagram must exist. Also, the propositions guarantee the following properties of the model: (i) the radius  $r$  of the oscillation is almost 0 for negative  $\delta$ , and (ii) the radius  $r$  is approximately proportional to  $\sqrt{\delta}$  for positive  $\delta$  (with some exceptions). These properties are the same as properties of the normal form of the supercritical Hopf bifurcation [15]. Note that the normal form of the Hopf bifurcation has been utilized as a cochlea model (called Hopf-cochlea) [4–6]. Hence, the theoretical analyses in this section suggest that the presented model may be also useful as a cochlea model. This suggestion is validated in the next section.

## 2.1.4 Reproductions of biological nonlinear frequency tuning curves

Recall that Fig. 2.1.4 shows the characteristics of the radius  $r$  of the state  $X$  for the case where no sound stimulation is applied. From a viewpoint of cochlea model, the parameter  $\delta$  should be set to a negative value since the oscillation of the state  $X$  for a positive  $\delta$  may correspond to an ear ringing (i.e., the model respond to zero sound stimulation). By extensive numerical analyses, we have found that  $\delta = -10$  (i.e., negative  $\delta$  near the bifurcation value 0) is suitable to reproduce the biological frequency tuning curves in Fig. 2.1.1. Now, let us introduce the following tone sound stimulation  $s(t)$ .

$$s(t) = A \sin 2\pi ft.$$

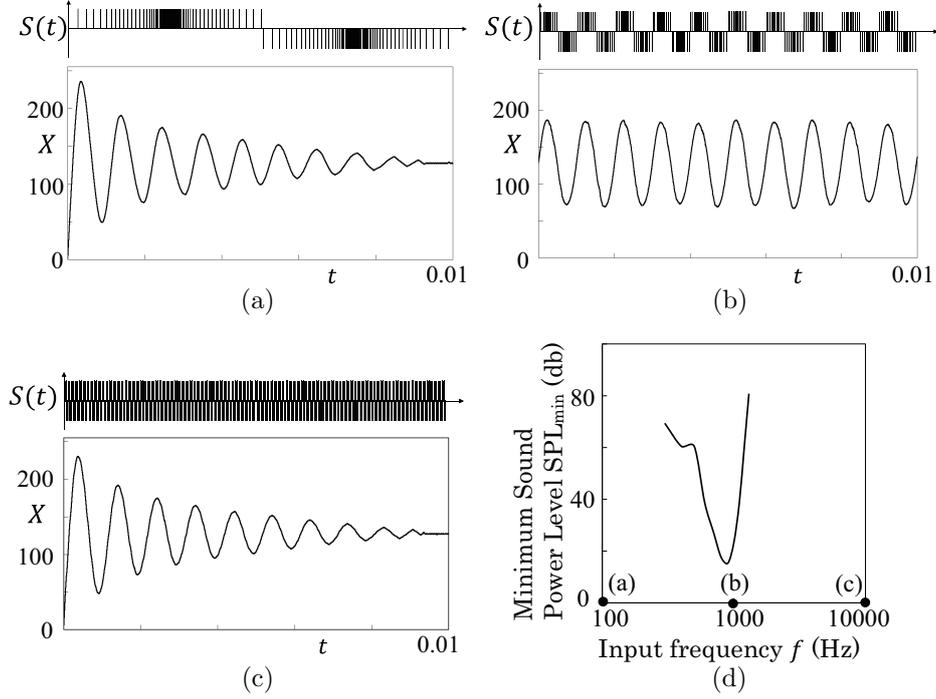


Figure 2.1.7: (a)–(c) Typical responses of the model to the stimulation  $S(t)$ .  $(N, M, \omega, \delta, l, T_X, T_Y) = (256, 256, 2\pi \times 10^3, 0.25, 1 \times 10^{-7}, 1.1 \times 10^{-7})$ .  $A = 127$ . (a)  $f = 100$  [Hz]. (b)  $f = 1$  [kHz]. (c)  $f = 10$  [kHz]. (d) Frequency tuning curve.  $p_0 = 10^6$ ,  $RMS_{th} = 15$ .

Let us also introduce the following spike-density-modulated stimulation  $S(t)$ .

$$S(t) = \begin{cases} 1 & \text{if } t = \tau_p(1), \tau_p(2), \dots, \\ -1 & \text{if } t = \tau_n(1), \tau_n(2), \dots, \\ 0 & \text{otherwise,} \end{cases}$$

where an instantaneous density of the spike positions  $\{\tau_p(1), \tau_p(2), \dots\}$  is proportional to  $s(t)$  for  $s(t) \geq 0$  and is zero for  $s(t) < 0$ ; and the instantaneous density of the spike positions  $\{\tau_n(1), \tau_n(2), \dots\}$  is proportional to  $-s(t)$  for  $s(t) \leq 0$  and is zero for  $s(t) > 0$ . Note that such a density modulator can be easily realized by using a standard density modulator. The stimulation  $S(t)$  triggers the following transitions of the discrete states  $(X, P)$ , which are asynchronous with the transitions in Eqs. (2.1.1) and (2.1.2).

$$\begin{aligned} &\text{If } |S(t)| = 1, \text{ then} \\ &P(t_+) := \begin{cases} P(t) + 1 & \text{if } P(t) < |F_X| \text{ and } S(t) = \text{sgn}(F_X), \\ 0 & \text{if } P(t) \geq |F_X| \text{ and } S(t) = \text{sgn}(F_X), \\ P(t) - 1 & \text{if } P(t) > 0 \text{ and } S(t) \neq \text{sgn}(F_X), \\ |F_X| & \text{if } P(t) \leq 0 \text{ and } S(t) \neq \text{sgn}(F_X), \end{cases} \\ &X(t_+) := \begin{cases} X(t) + \text{sgn}(F_X) & \text{if } P(t) \geq |F_X| \text{ and } S(t) = \text{sgn}(F_X), \\ X(t) - \text{sgn}(F_X) & \text{if } P(t) \leq 0 \text{ and } S(t) \neq \text{sgn}(F_X). \end{cases} \end{aligned} \quad (2.1.4)$$

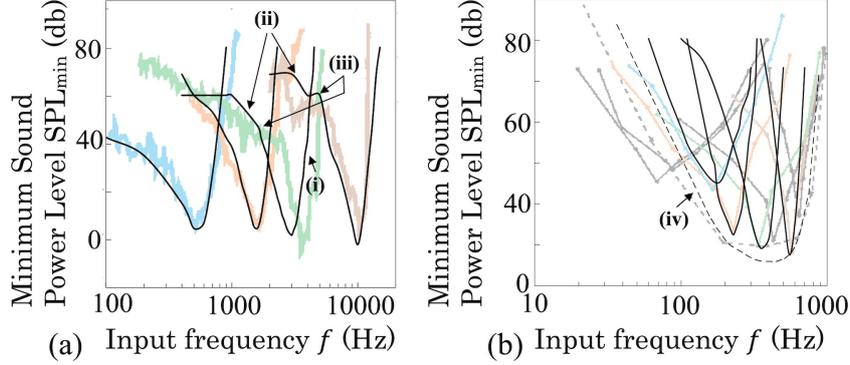


Figure 2.1.8: Reproductions of the biological frequency tuning curves in Figs. 2.1.1(a) and (b).  $(N, M, l, p_0) = (256, 256, 0.25, 10^6)$ . (a) Chinchilla.  $T_X = 10^{-7}$ .  $(\omega/(2\pi), T_Y, RMS_{th}) = (450, 1.5 \times 10^{-7}, 10), (1750, 1.1 \times 10^{-7}, 10), (3630, 1.1 \times 10^{-7}, 10), (2 \times 10^5, 1.1 \times 10^{-7}, 20)$ . (b) Turtle.  $(\omega/(2\pi), T_X, T_Y, RMS_{th}) = (160, 10^{-7}, 3 \times 10^{-7}, 7), (230, 10^{-7}, 1.5 \times 10^{-7}, 8), (350, 3 \times 10^{-7}, 3.10^{-7}, 8), (600, 0.7 \times 10^{-7}, 0.8 \times 10^{-7}, 8)$ .

Figs. 2.1.7(a)–(c) show time waveforms of the model for the stimulation  $S(t)$  with three frequencies. In Fig. 2.1.7(b), the stimulation frequency is  $f = 1$ [kHz] and the amplitude of the state  $X$  is about 180. On the other hand, in Figs. 2.1.7(a) and (c), the stimulation frequencies are  $f = 100$ [Hz] and  $10$ [kHz], and the amplitude of the state  $X$  is about zero. That is, the model has selectivity of the stimulation frequency  $f$ . In order to characterize such frequency selectivity, let us introduce the following indices  $SPL(S)$  and  $RMS(X)$ . The strength of the stimulation  $S(t)$  is characterized by

$$p(S) = \lim_{\tau \rightarrow \infty} \frac{\text{Number of spikes } |S(t)| = 1 \text{ for } t \in [0, \tau]}{\tau}.$$

The strength  $p(S)$  is rescaled to the following stimulation power level  $SPL(S)$ , which corresponds to the sound pressure level in the biological frequency tuning curves in Fig. 2.1.1.

$$SPL(S) = (p(S) - p_0) \times 10^{-5}(\text{dB}),$$

where  $p_0$  is a reference magnitude corresponding to 0 (dB) of the  $SPL$ . In Figs. 2.1.7(a)–(c), the stimulation  $S(t)$  has the same  $SPL(S) = 2.67$ . The strength of the response of the proposed model to the stimulation  $S(t)$  is characterized by the following root mean square of the discrete state  $X$ .

$$RMS(X) = \lim_{T \rightarrow \infty} \sqrt{\frac{l^2}{T} \int_T^{T+l} (X(t) - N/2)^2 dt}.$$

In Figs. 2.1.7(a), (b), and (c), the discrete state  $X$  has  $RMS(X) = 0.19, 10.65,$  and  $0.18,$  respectively. Using the  $SPL(S)$  and the  $RMS(X)$ , let us introduce the following index  $SPL_{min}$ .

**Definition 2:** The minimum stimulation power level  $SPL_{min}$  is the minimum stimulation power level  $SPL(S)$  of the stimulation  $S(t)$  such that the  $RMS(X)$  of the state  $X$  is greater than or equal to a pre-defined threshold value  $RMS_{th}$ . Also, the characteristics of the minimum stimulation power level  $SPL_{min}$  for the frequency  $f$  of the sound stimulation

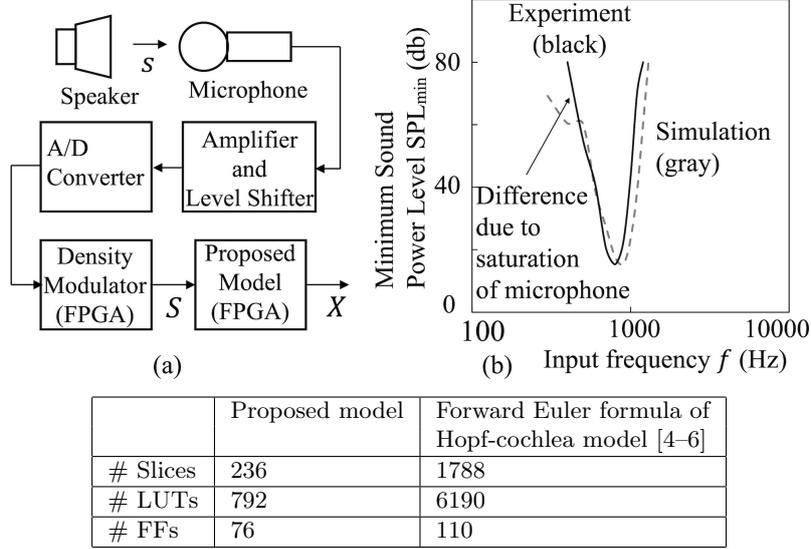


Figure 2.1.9: (a) Experimental setup. (b) Experimental frequency tuning curve and numerical curve. “LUTs” stands for “look-up tables” and “FFs” stands for “flip-flops.”

$S(t)$  is referred to as a frequency tuning curve.

Fig. 2.1.7(d) shows a frequency tuning curve of the presented model, where the dots (a)–(c) correspond to Figs. 2.1.7(a)–(c), respectively. Fig. 2.1.8 shows frequency tuning curves of the presented model, the parameter values of which are tuned to reproduce the biological frequency tuning curves in Fig. 2.1.1. In this figure, the arrows (i)–(iv) correspond to those in Fig. 2.1.1(a) and (b). It can be seen that the presented model can qualitatively reproduce the features (F1)–(F4) of the biological frequency tuning curves in Fig. 2.1.1.

## 2.1.5 FPGA implementation and comparison

Recall that the state transitions of the presented model are described by Eqs. (2.1.1), (2.1.2), and (2.1.4). These equations are rewritten as a VHDL code, which is compiled into a bitstream file, and the resulting file is downloaded to a field programmable gate array (FPGA), Xilinx’s XC7A100T-1CSG324C. Fig. 2.1.9(a) shows an experimental setup consisting of a speaker (Audio-Technica’s AT-SPP30), a microphone (Audio-Technica’s Pro-300), a microphone amplifier (Audio-Technica’s AT-MA2), a level shifter (TI’s TL074), and A/D converter (Analog Devices’ AD7476A). Since the FPGA device and the HDL compiler (Xilinx’s Vivado 2016.02) used in this paper do not support asynchronous triggering, the clocks  $C_X(t)$  and  $C_Y(t)$  and the stimulation  $S(t)$  are generated from a common clock with high frequency (100[MHz]), where, since the frequency of the common clock is much higher than the frequencies of  $C_X(t)$ ,  $C_Y(t)$  and  $S(t)$ , the resulting behavior of the implemented circuit can be regarded to be almost identical with the asynchronous behavior of the presented model. Actually, we have confirmed that the implemented circuit behaves almost identical with the simulation results in Figs. 2.1.3, 2.1.4, and 2.1.7. Fig. 2.1.9(b) shows a frequency tuning curve obtained by FPGA experiments and numerical simulations. It can be seen that the implemented circuit can reproduce the tuning curve, where the difference

from the simulation is not due to a defect in the implementation but is due to saturation of the microphone. In addition, we have confirmed that the implemented circuit can reproduce other tuning curves in Fig. 2.1.8 except for effects of saturation of the microphone. For comparison, the Hopf-cochlea model [4–6] is implemented as a forward Euler formula by using a fixed-point number format in the same FPGA device. The bit-length of the Hopf-cochlea model and the resolution  $(M, N)$  of the presented model are decreased as short as possible under the condition that their bifurcation diagrams qualitatively reproduce the properties (i) and (ii) of the supercritical Hopf bifurcation (see the Remark in Section III), where the resulting bit-length is 26 bit and the resulting resolution is  $M = N = 64$ . The table in Fig. 2.1.9 shows comparison results. It can be seen that our model consumes fewer hardware resources than the Hopf-cochlea model.

### 2.1.6 Conclusions

The presented model was theoretically analyzed by utilizing the multi-valued map  $G$  and the analyses results suggested that the model can mimic the nonlinear vector field of the Hopf-cochlea model. It was then shown that the model can reproduce the biologically measured nonlinear frequency tuning curves of the multiple species. It was also shown by the FPGA experiments that the model can be implemented by using fewer hardware resources compared to the numerical integration formula of the Hopf-cochlea model. Future problems include: (a) realization of other nonlinear responses of biological cochleae such as multi-tone suppression, (b) more detailed comparisons with biological tuning curves, and (c) more intensive hardware experiments and related comparisons.

## 2.2 Reproduction of two-tone distortion products<sup>2</sup>

### 2.2.1 Introduction

A laser Doppler velocimeter performed on live mammalian cochleas has observed that they have an active amplifier and essential nonlinearities [16–18]. For example, Fig. 2.2.1 shows typical nonlinear response characteristics of a mammalian cochlea measured in an anesthetized chinchilla [18]. In this figure, the basilar membrane (BM) is stimulated by two-tone frequencies  $f_1$  and  $f_2$  ( $f_2 > f_1$ ), and it perceives tones with a combination of two frequencies, namely  $n_1 f_1 \pm n_2 f_2$  ( $n_1, n_2 \in \mathbb{Z}$ ), where the characteristic frequency (CF) in the stimulated site of the BM corresponds to  $2f_1 - f_2$ . Such responses to tones with frequencies not included in two-tone stimuli are called two-tone distortion products. Eguiluz *et al.* were the first to introduce a cochlea model based on Hopf oscillators to understand the nonlinearities of hearing [19]. Subsequently, Stoop *et al.* succeeded in demonstrating that the model enables the reproduction of a number of nonlinear hearing phenomena such as nonlinear compression, two-tone suppression, combination tone generation, and first (second) pitch shift [20–25]. Since then, there has been an increasing interest in implementing a cochlea model utilizing nonlinear oscillators on digital and analog electronic circuits [9, 26–44]. Such an electronic circuit can be applied to hearing aids and cochlear implants that incorporate nonlinear signal processing.

---

<sup>2</sup>This section is based on “Two-Tone Distortion Products in Hardware-Efficient Cochlea Model based on Asynchronous Cellular Automaton Oscillator,” by the same author, which appeared in *IEICE Electronics Express*, vol. 18, no. 18, 20210310, 2021, Copyright(C)2021 IEICE.

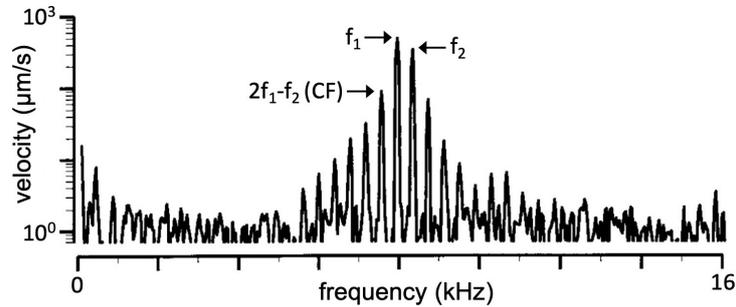


Figure 2.2.1: Spectrum of cochlear basilar membrane (BM) responses to two-tone stimuli with frequencies  $f_1$  and  $f_2$  ( $f_2 > f_1$ ) measured in chinchilla adapted from [18]. The two-tone distortion products are perceived at frequencies not included in the stimulus (e.g.,  $2f_1 - f_2$ ,  $3f_1 - 2f_2$ ,  $2f_2 - f_1$ , and  $3f_2 - 2f_1$ ), where the characteristic frequency (CF) in the stimulated site of the BM corresponds to  $2f_1 - f_2$ .

Traditional methods applicable to modeling and implementing nonlinear biological oscillators, including the cochlear amplifier, can be classified into the following three ways based on the continuousness of time and state.

(i) The first method to model a nonlinear biological oscillator is by using an ordinary differential equation, which has a continuous time and state. Such a model is implemented by a nonlinear electronic analog circuit [26–29].

(ii) The second method to model a nonlinear biological oscillator is by using a difference equation, which has discrete time and continuous states. Such a model is implemented by a switched-capacitor circuit [33–36].

(iii) The third method to model a nonlinear biological oscillator is by using a discrete difference equation or a cellular automaton (CA), which has discrete times and states. Such models are implemented using a digital signal processor (DSP) that performs numerical integration or sequential logic [9, 30–32].

Most nonlinear biological oscillators that have been studied belong to one of the above-mentioned three categories [9, 26–32]. Recently, our group and a few other groups have designed a nonlinear biological oscillator using the below-mentioned fourth method.

(iv) The fourth method to model a nonlinear biological oscillator is by using asynchronous CA, which has a continuous state transition time and discrete states. Such a model is implemented by an asynchronous sequential logic [37–44].

Inspired by the Hopf-type cochlea [19–25], our group has proposed a cochlea model designed by the fourth method and has shown that the model can reproduce the frequency-threshold tuning curves of a mammalian cochlea [41]. However, the reproduction of nonlinear response characteristics to two-tone stimuli remains to be demonstrated.

In this paper, a cochlea model with nonlinear dynamics described by an asynchronous CA was studied. We demonstrated that the proposed cochlea model enables the reproduction of two-tone distortion products [18]. The proposed model was implemented on a field-programmable gate array (FPGA). Subsequently, we demonstrated that the proposed cochlea model can be implemented using fewer hardware resources than a Hopf cochlea-type cochlea model implemented on a DSP performing a numerical integration.

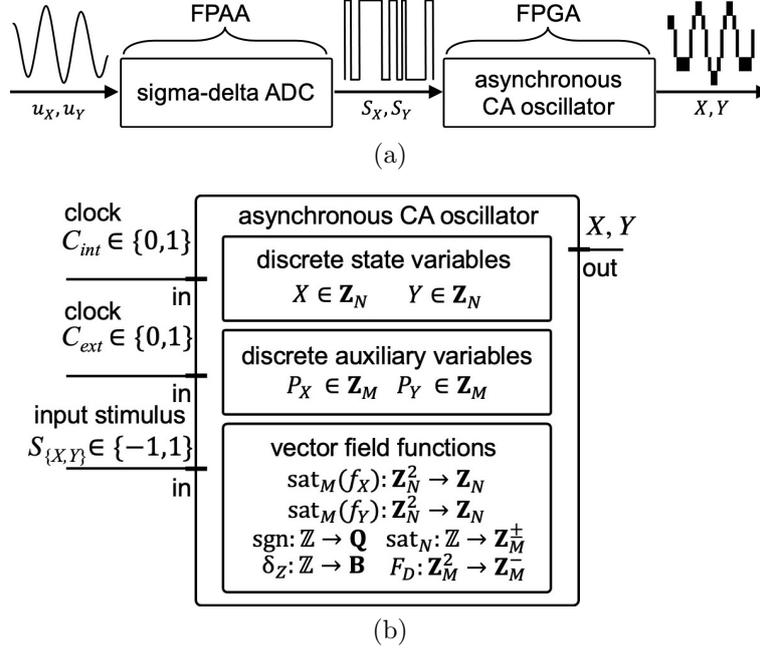


Figure 2.2.2: (a) Hardware configuration of the proposed cochlea model. (b) Schematic of asynchronous cellular automaton (CA) oscillator.

## 2.2.2 Model description

In this section, a hardware-efficient cochlea model based on an asynchronous CA oscillator is proposed. Subsequently, we demonstrated that the proposed cochlea model enables the reproduction of two-tone distortion products. Fig. 2.2.2(a) shows a hardware configuration of the proposed cochlea model. As shown in the figure, the proposed cochlea model comprises a sigma-delta analog-to-digital converter (ADC) and an asynchronous CA oscillator.

### 2.2.2.1 ASYNCHRONOUS CA OSCILLATOR

Fig. 2.2.2(b) shows a schematic of the asynchronous CA oscillator. As shown in the figure, the oscillator accepts the following clock.

$$C_{int}(t) \equiv \sum_{k=0}^{\infty} \delta_R(t - kT_{int}), \quad C_{int} \in \mathbf{B} \equiv \{0, 1\}, \quad (2.2.1)$$

where  $t \in \mathbb{R}$  and  $T_{int} \in \mathbf{R}^+ \equiv \{x \in \mathbb{R} \mid x > 0\}$  represent the continuous time and a period of the clock, respectively, and the function  $\delta_R: \mathbb{R} \rightarrow \mathbf{B}$  represents the unit impulse.

$$\delta_R(x) \equiv \begin{cases} 1 & (x = 0), \\ 0 & (x \neq 0). \end{cases}$$

Furthermore, as shown in Fig. 2.2.2(b), the oscillator has the following two discrete state variables,

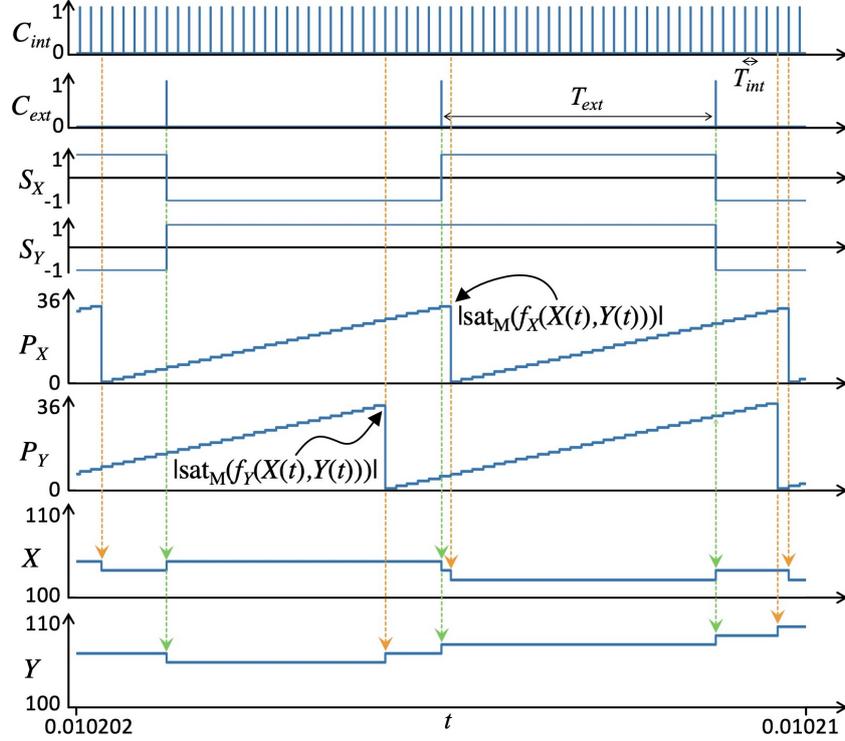


Figure 2.2.3: Timing chart of asynchronous CA oscillator. The parameter values are  $N = 2^7$ ,  $M = 2^8$ ,  $\mu = -10$ ,  $\omega = 2\pi \times 10^3$ ,  $l = 0.25$ ,  $T_{int} = 2^{-23}$ ,  $T_{ext} = 3 \times 10^{-6}$ ,  $\omega_1 = 2\pi \times 10^3$ ,  $\omega_2 = 2\pi \times 200$ ,  $A_1 = 0.125$ ,  $A_2 = 0.125$ , and  $B_0 = 2$ .

$$X \in \mathbf{Z}_N \equiv \{0, \dots, N-1\}, \quad Y \in \mathbf{Z}_N,$$

and the following two discrete auxiliary variables

$$P_X \in \mathbf{Z}_M \equiv \{0, \dots, M-1\}, \quad P_Y \in \mathbf{Z}_M,$$

where  $N \geq 2$  and  $M \geq 2$  are integers determining the resolution of a state space  $\mathbf{Z} \equiv \{(X, Y, P_X, P_Y) \mid X \in \mathbf{Z}_N, Y \in \mathbf{Z}_N, P_X \in \mathbf{Z}_M, P_Y \in \mathbf{Z}_M\}$  of the oscillator. To design a nonlinear vector field, the following two functions  $f_X : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbb{Z}$  and  $f_Y : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbb{Z}$  are introduced.

$$\begin{aligned} f_X(x, y) &\equiv \left\lfloor \frac{1}{T_{int}(\mu(x - \frac{N}{2}) - \omega(y - \frac{N}{2}) - l^2(x - \frac{N}{2})((x - \frac{N}{2})^2 + (y - \frac{N}{2})^2))} \right\rfloor, \\ f_Y(x, y) &\equiv \left\lfloor \frac{1}{T_{int}(\omega(x - \frac{N}{2}) + \mu(y - \frac{N}{2}) - l^2(y - \frac{N}{2})((x - \frac{N}{2})^2 + (y - \frac{N}{2})^2))} \right\rfloor, \end{aligned} \quad (2.2.2)$$

where  $l \in \mathbf{R}^+$ ,  $\mu \in \mathbf{R}$ , and  $\omega \in \mathbf{R}^+$  are the parameters. The derivation of the functions  $f_X$  and  $f_Y$  in Eq. (2.2.2) is presented in Appendix. Furthermore,  $\lfloor \cdot \rfloor$  denotes the following floor function.

$$\lfloor x \rfloor \equiv \max\{l \in \mathbb{Z} \mid l \leq x\}, \quad x \in \mathbf{R}.$$

The functions  $f_X$  and  $f_Y$  are implemented in the lookup tables (LUTs). Fig. 2.2.3 shows the timing chart of the oscillator. As shown in Fig. 2.2.3, the clock  $C_{int}$  triggers the following transitions of the discrete state variables  $X$  and  $Y$ .

$$\begin{aligned} &\text{If } C_{int}(t) = 1, \text{ then} \\ &X(t_+) := \text{sat}_N(X(t) + \delta_Z(P_X(t)) \text{sgn}(\text{sat}_M(f_X(X(t), Y(t))))), \\ &Y(t_+) := \text{sat}_N(Y(t) + \delta_Z(P_Y(t)) \text{sgn}(\text{sat}_M(f_Y(X(t), Y(t))))), \end{aligned} \quad (2.2.3)$$

where the symbol “ $t_+$ ” represents “ $\lim_{\epsilon \rightarrow +0} t + \epsilon$ ”, the symbol “ $:=$ ” represents an “instantaneous state transition,” and  $\text{sat}_N : \mathbb{Z} \rightarrow \mathbf{Z}_N$  and  $\text{sat}_M : \mathbb{Z} \rightarrow \mathbf{Z}_M^\pm \equiv \{-(M-1), \dots, 0, \dots, M-1\}$  denote the following saturation functions.

$$\begin{aligned} \text{sat}_N(x) &\equiv \begin{cases} N-1 & (x > N-1), \\ x & (0 \leq x \leq N-1), \\ 0 & (x < 0). \end{cases} \\ \text{sat}_M(x) &\equiv \begin{cases} M-1 & (x > M-1), \\ x & (0 \leq x \leq M-1), \\ -(M-1) & (x < -(M-1)). \end{cases} \end{aligned}$$

Furthermore,  $\delta_Z : \mathbb{Z} \rightarrow \mathbf{B}$  denotes the following unit impulse function.

$$\delta_Z(x) \equiv \begin{cases} 1 & (x = 0), \\ 0 & (x \neq 0), \end{cases}$$

where  $\text{sgn} : \mathbb{Z} \rightarrow \mathbf{Q} \equiv \{-1, 1\}$  denotes the following signum function.

$$\text{sgn}(x) \equiv \begin{cases} 1 & (x \geq 0), \\ -1 & (x < 0). \end{cases}$$

As shown in Fig. 2.2.3, the clock  $C_{int}$  triggers the following transitions of the discrete auxiliary variables  $P_X$  and  $P_Y$ .

$$\begin{aligned} &\text{If } C_{int}(t) = 1, \text{ then} \\ &P_X(t_+) := P_X(t) + F_D(P_X(t), |\text{sat}_M(f_X(X(t), Y(t)))|), \\ &P_Y(t_+) := P_Y(t) + F_D(P_Y(t), |\text{sat}_M(f_Y(X(t), Y(t)))|), \end{aligned} \quad (2.2.4)$$

where  $F : \mathbf{Z}_M^2 \rightarrow \mathbf{Z}_M^- \equiv \{-(M-1), \dots, 0, 1\}$  denotes the following function.

$$F_D(x, a) \equiv \begin{cases} 1 & (x < a), \\ -x & (x \geq a). \end{cases}$$

Fig. 2.2.4(a) shows a typical time waveform of the discrete state variables  $X$  and  $Y$  without an input stimulus.

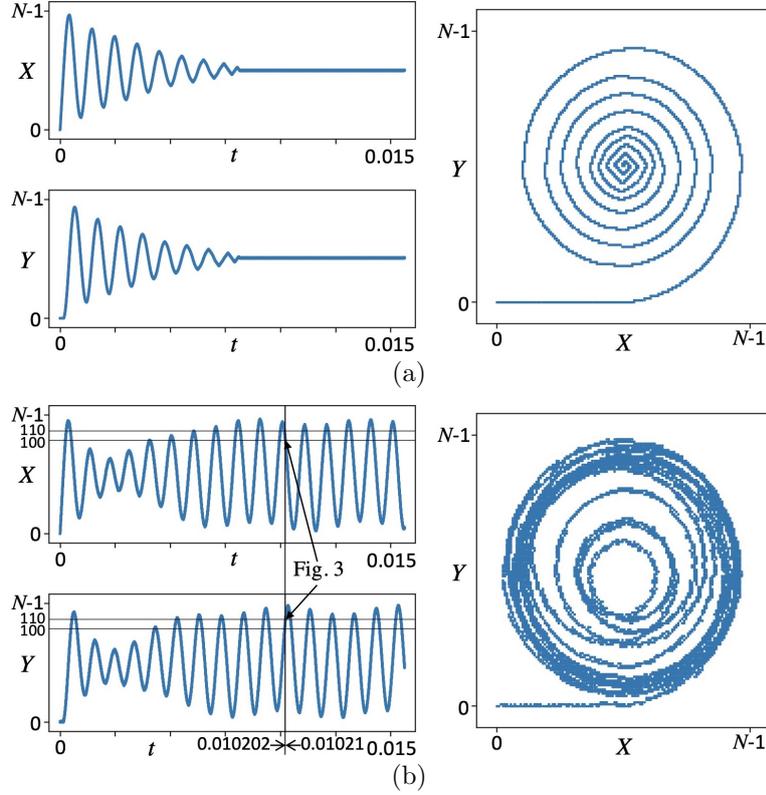


Figure 2.2.4: (a) Typical responses of asynchronous CA oscillator without input stimulus. The parameter values are  $N = 2^7$ ,  $M = 2^8$ ,  $\mu = -10$ ,  $\omega = 2\pi \times 10^3$ ,  $l = 0.25$ , and  $T_{int} = 2^{-23}$ . (b) Typical responses of asynchronous CA oscillator with two-tone input stimulus. The parameter values are the same as those chosen in Fig. 2.2.3.

### 2.2.2.2 INPUT STIMULUS MODULATED BY SIGMA-DELTA ADC

In this study, an input stimulus is assumed to be

$$u(t) \equiv A_1 e^{j\omega_1 t} + A_2 e^{j\omega_2 t}, \quad u \in \mathbb{C}$$

where  $A_1, A_2 \in \mathbf{R}^+$  and  $\omega_1, \omega_2 \in \mathbf{R}^+$  represent the amplitude and angular frequency of the input stimulus, respectively. The input stimulus  $u(t)$  is transformed into a Cartesian coordinate representation for applying to the asynchronous CA oscillator as follows.

$$\begin{aligned} u_X(t) &\equiv A_1 \cos(\omega_1 t) + A_2 \cos(\omega_2 t), \quad u_X \in \mathbb{R}, \\ u_Y(t) &\equiv A_1 \sin(\omega_1 t) + A_2 \sin(\omega_2 t), \quad u_Y \in \mathbb{R}. \end{aligned}$$

The input stimuli  $u_X$  and  $u_Y$  are converted by using a first-order sigma-delta ADC, and a schematic of the ADC is shown in Fig. 2.2.5. As shown in this figure, for sampling an analog signal, the following clock is introduced.

$$C_{ext}(t) \equiv \sum_{k=0}^{\infty} \delta(t - kT_{ext}), \quad C_{ext} \in \mathbf{B}, \quad (2.2.5)$$

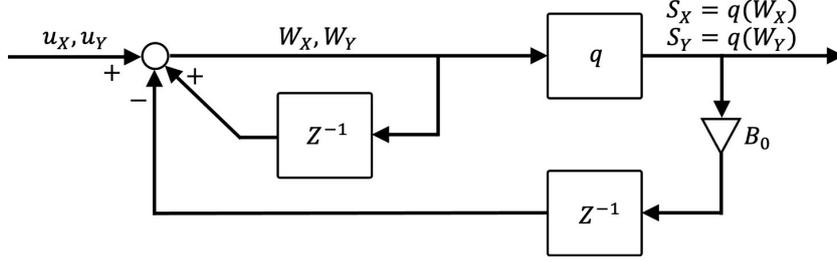


Figure 2.2.5: Schematic of first-order sigma-delta ADC.

where  $T_{ext} \in \mathbf{R}^+$  represents a period of the clock. Furthermore, the input stimuli  $u_X$  and  $u_Y$  are sampled by clock  $C_{ext}$  and integrated by the following signals  $W_X, W_Y \in \mathbf{R}$ .

$$\begin{aligned} \text{If } C_{ext}(t) = 1, \text{ then} \\ W_X(t_+) &:= u_X(t) + W_X(t) - B_0 S_X(t), \\ W_Y(t_+) &:= u_Y(t) + W_Y(t) - B_0 S_Y(t), \end{aligned} \quad (2.2.6)$$

where  $S_X(t) = q(W_X(t))$ ,  $S_Y(t) = q(W_Y(t))$  and  $B_0 \in \mathbf{R}$ , and the function  $Q : \mathbf{R} \rightarrow \mathbf{Q}$  denotes the following quantizer.

$$q(x) \equiv \begin{cases} 1 & (x \geq 0), \\ -1 & (x < 0). \end{cases}$$

Subsequently, the clock  $C_{ext}(t)$  triggers the following transitions of the discrete state variables  $X$  and  $Y$ .

$$\begin{aligned} \text{If } C_{ext}(t) = 1, \text{ then} \\ X(t_+) &:= X(t) + S_X(t), \\ Y(t_+) &:= Y(t) + S_Y(t). \end{aligned} \quad (2.2.7)$$

Fig. 2.2.4(b) shows the typical time waveforms of the asynchronous CA oscillator with a two-tone input stimulus. The transitions of the discrete state variables  $X$  and  $Y$  are triggered by the uncoupled clocks  $C_{int}$  and  $C_{ext}$ , as shown in Fig. 2.2.3, the oscillator can be regarded as an asynchronous CA.

### 2.2.2.3 REPRODUCTIONS OF TWO-TONE DISTORTION PRODUCTS

Fig. 2.2.6(a) shows a frequency spectrum of a time waveform of the proposed cochlea model with two tones modulated by the first-order sigma-delta ADC, where CF corresponds to  $2f_1 - f_2$ . The frequency spectrum herein was obtained via fast Fourier transform (FFT) using a Hanning window with an applied amplitude factor. The proposed cochlea model perceives tones with frequencies not included in the two-tone stimulus, for example,  $2f_1 - f_2$ . Hence, the FFT analysis results shown in Fig. 2.2.6(a) verified that our proposed cochlea model can generate two-tone distortion products (see also Fig. 2.2.1). Fig. 2.2.6(b) shows two-tone distortion products in the proposed cochlea model with input stimuli modulated by the second-order sigma-delta ADC. As shown in Figs. 2.2.6(a) and (b), the order of the sigma-delta ADC has no significant effect on the characteristics of the two-tone distortion products. Hence, in this study, the first-order sigma-delta ADC is used to modulate input

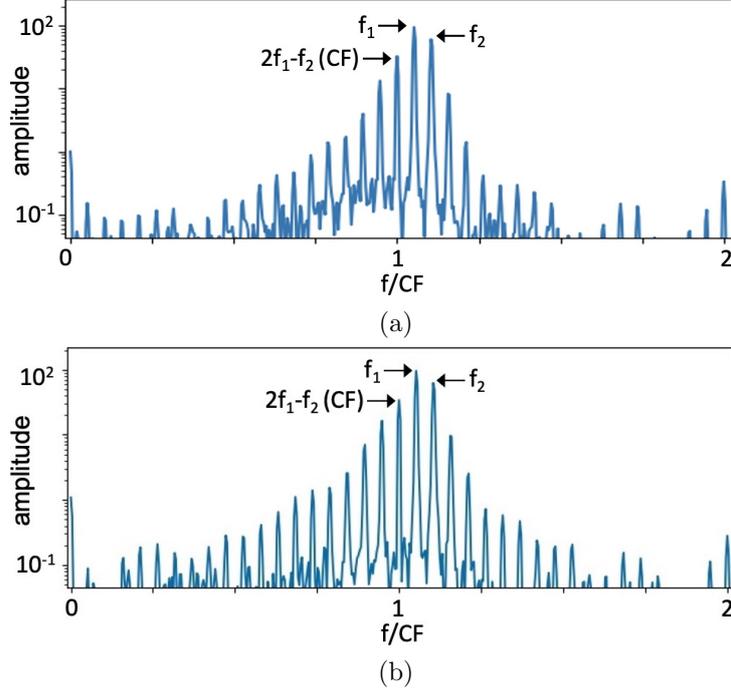


Figure 2.2.6: Two-tone distortion products in the proposed cochlea model (see also Fig. 2.2.1). (a) Case the first-order sigma-delta ADC is used to modulate input stimuli. (b) Case the second-order sigma-delta ADC is used to modulate input stimuli. The parameter values are  $N = 2^7$ ,  $M = 2^8$ ,  $\mu = -10$ ,  $\omega = 2\pi \times CF = 2\pi \times 10^3$ ,  $l = 0.25$ ,  $T_{int} = 2^{-23}$ ,  $T_{ext} = 3 \times 10^{-6}$ ,  $\omega_1 = 2\pi \times 1052.63$ ,  $\omega_2 = 2\omega_1 - (2\pi \times CF)$ ,  $A_1 = A_2 = 0.3$ , and  $B_0 = 2$ .

stimuli by which the proposed cochlea model is stimulated.

## 2.2.3 FPGA implementation and comparison

### 2.2.3.1 FPGA IMPLEMENTATION

The proposed cochlea model is implemented on a field-programmable gate array (FPGA) in a register transfer level (RTL) code using VHDL as follows: The discrete state variables  $X$  and  $Y$  are implemented by registers as  $n$ -bit unsigned integers, where  $n = \lceil \log_2 N \rceil$ . The discrete auxiliary variables  $P_X$  and  $P_Y$  are implemented by using registers as  $m$ -bit unsigned integers, where  $m = \lceil \log_2 M \rceil$ . The functions  $\text{sat}_M(f_X(x, y))$  and  $\text{sat}_M(f_Y(x, y))$  are implemented in lookup tables (LUTs) with an  $n$ -bit unsigned integer input and an  $m+1$ -bit signed integer output in the two's complement format. The dynamic equations in Eqs. (2.2.3), (2.2.4), and (2.2.7) are written by sequential statements triggered by the clocks  $C_{int}$  and  $C_{ext}$  in Eqs. (2.2.1) and (2.2.5), respectively. The RTL code was synthesized by Xilinx Vivado Design Suite v2020.1, and a generated bitstream file was downloaded into Xilinx FPGA Artix-7 XC7A100T-1CSG324C. Fig. 2.2.7(a) shows the resulting RTL schematic of our proposed cochlea model. The sigma-delta ADC is implemented by a switched-capacitor technique on a field-programmable analog array (FPAA) (Anadigm Single Apex 3.3 V Development Kit). The dynamic equation in Eq. (2.2.6) is written using Anadigm Designer 2, where the resulting schematic of the sigma-delta ADC is shown in Fig. 2.2.7(b). Fig. 2.2.8 shows the frequency spectrum of the proposed cochlea model stimulated by two-tone

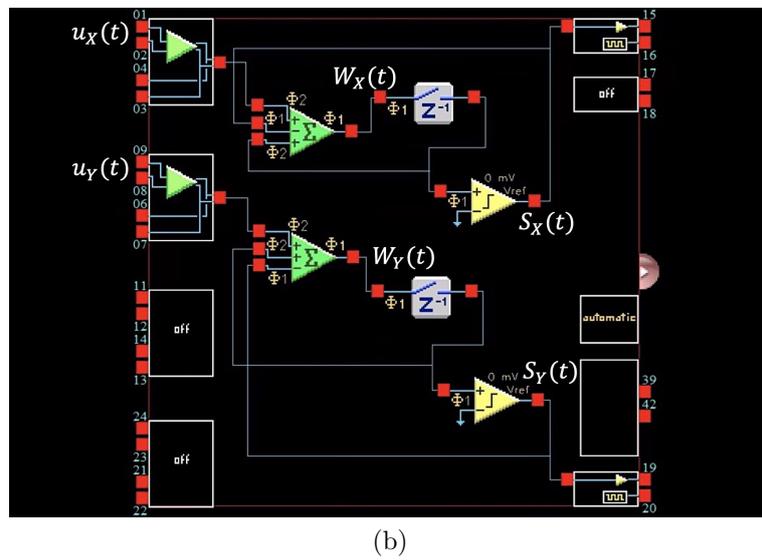
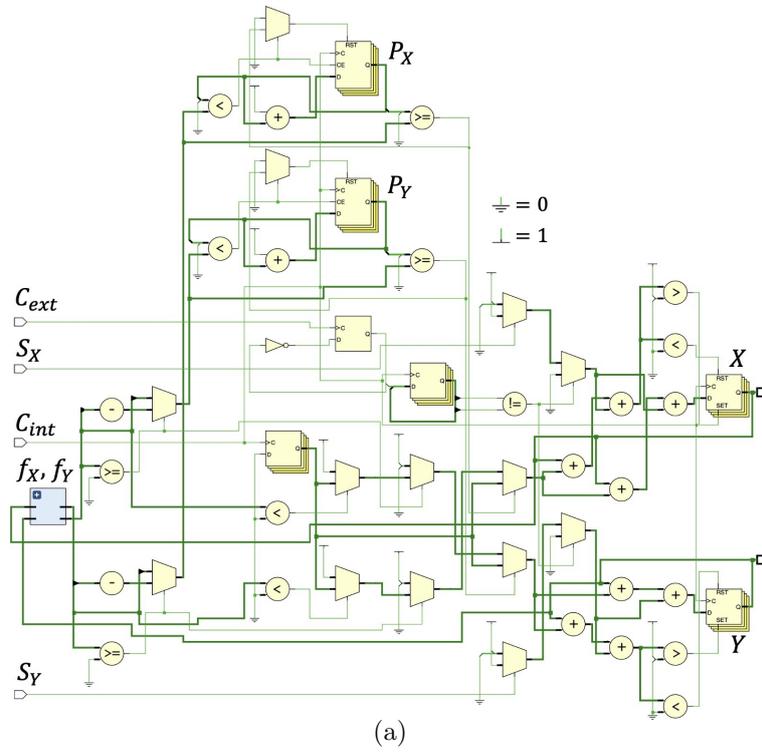


Figure 2.2.7: (a) Schematic of the FPGA for implementing the proposed cochlea model. (b) Schematic of the FPA for implementing the sigma-delta ADC.

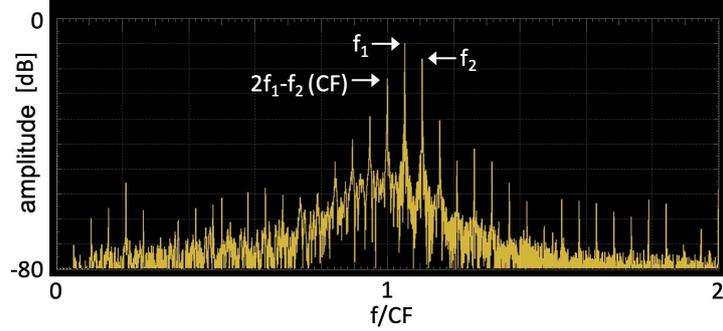


Figure 2.2.8: Two-tone distortion products in the proposed cochlea model implemented on FPGA (see also Figs. 2.2.1 and 2.2.6(a)). The parameter values are the same as those chosen in Fig. 2.2.6.

stimuli implemented on the FPGA. The two-tone distortion product can be confirmed from the frequency spectrum (see also Figs. 2.2.1 and 2.2.6(a)).

### 2.2.3.2 COMPARISON WITH HOPF-TYPE COCHLEA MODEL

For comparison, we implemented a Hopf-type cochlea model [19,20] on the same FPGA, where the hardware configuration is shown in Fig. 2.2.9. The model comprises a sigma-delta ADC<sup>3</sup> and the following Hopf oscillator.

$$\dot{z} = (b + j)\omega_{ch}z - |z|^2z + F(t), \quad z \in \mathbb{C}, \quad (2.2.8)$$

where  $b \in \mathbb{R}$  and  $\omega_{ch} \in \mathbb{R}$  are parameters, and  $F(t) \equiv E_1e^{j\omega_1t} + E_2e^{j\omega_2t}$  is an external input representing a two-tone stimulus. To implement an electronic circuit, the model in Eq. (2.2.8) is transformed into a Cartesian coordinate representation as follows.

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2) + F_1(t), \quad x_1 \in \mathbb{R}, \\ \dot{x}_2 &= f_2(x_1, x_2) + F_2(t), \quad x_2 \in \mathbb{R}, \end{aligned} \quad (2.2.9)$$

where

$$\begin{aligned} f_1(x_1, x_2) &\equiv bx_1 - \omega_{ch}x_2 - x_1(x_1^2 + x_2^2), \\ f_2(x_1, x_2) &\equiv \omega_{ch}x_1 + bx_2 - x_2(x_1^2 + x_2^2), \\ F_1(t) &\equiv E_1 \cos(\omega_1t) + E_2 \cos(\omega_2t), \\ F_2(t) &\equiv E_1 \sin(\omega_1t) + E_2 \sin(\omega_2t). \end{aligned}$$

The Hopf oscillator in Eq. (2.2.9) is discretized using the forward Euler method, which is one of the simplest numerical integration methods as follows.

$$\begin{aligned} x_1(t + \Delta t) &= x_1(t) + \Delta t(f_1(x_1(t), x_2(t)) + F_1(t)), \\ x_2(t + \Delta t) &= x_2(t) + \Delta t(f_2(x_1(t), x_2(t)) + F_2(t)), \end{aligned} \quad (2.2.10)$$

where  $\Delta t \in \mathbb{R}$  denotes a discretized time step. Furthermore,  $F_1(t)$  and  $F_2(t)$  are modulated by the sigma-delta ADC, which is the same as that shown in Fig. 2.2.7(b). As with Section

<sup>3</sup>The sigma-delta modulator is not necessarily chosen for the ADC in the Hopf-type cochlea model.

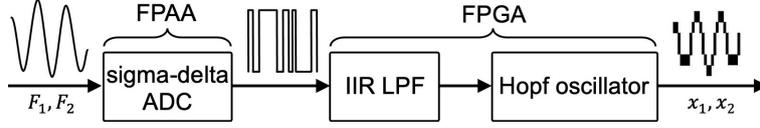


Figure 2.2.9: Hardware configuration of the Hopf-type cochlea model. The parameter values are  $\omega_{ch} = 2\pi \times \text{CF} = 2\pi \times 10^3$ ,  $b = -200$ ,  $\omega_1 = 2\pi \times 1052.63$ ,  $\omega_2 = 2\omega_1 - (2\pi \times \text{CF})$ ,  $L = 33$ , and  $\Delta t = 2^{-17}$ .

Table 2.2.1: Comparisons of Hardware Resource

	asynchronous CA oscillator	Hopf oscillator
# Slices	1031	2237
# LUTs	3713	6666
# FFs	216	161

Note: The hardware resources of the Hopf oscillator do not include those of the IIR LPF. Artix-7 XC7A100T-1CSG324C has 15,850 slices; each slice contains four 6-input lookup tables (LUTs) and eight flip-flops (FFs). DSP slices and block RAM were not used.

3.1, the discretized Hopf oscillator in Eq. (2.2.10) is implemented on the FPGA in an RTL code using VHDL, where the state variables and parameters are represented by fixed-point numbers in the two's complement format. Furthermore, as shown in Fig. 2.2.9, an infinite impulse response (IIR) low-pass filter (LPF) is implemented on the FPGA for demodulation of a sigma-delta ADC signal. Note that this digital filter is not required in the proposed cochlea model. The bit-lengths  $L$  of all variables are reduced to be as short as possible under the condition that the model can generate two-tone distortion products. The RTL code is synthesized using the same development environment as that used in Section 3.1. Table 2.2.1 summarizes the comparison results of hardware resources. As shown in this table, the proposed cochlea model can be implemented using fewer hardware resources than the Hopf oscillator.

## 2.2.4 Conclusions

In this paper, a cochlea model based on an asynchronous CA oscillator is proposed. Our proposed cochlea model enables the reproduction of two-tone distortion products. The proposed cochlea and Hopf-type cochlea models were implemented on the same FPGA. The comparison revealed the following advantages of the proposed cochlea model: (i) the asynchronous CA oscillator can be implemented using fewer hardware resources than the Hopf oscillator and (ii) the proposed cochlea model does not require a digital filter for the demodulation of the sigma-delta ADC signal. Hence, this study contributes to the development of hearing aids and cochlear implants implemented in small-scale circuits. Our future work will include (i) extensive analyses of the two-tone distortion products in the proposed cochlea model and (ii) reproduction of other types of nonlinear response characteristics in a biological cochlea such as two-tone suppression and first (second) pitch shift.

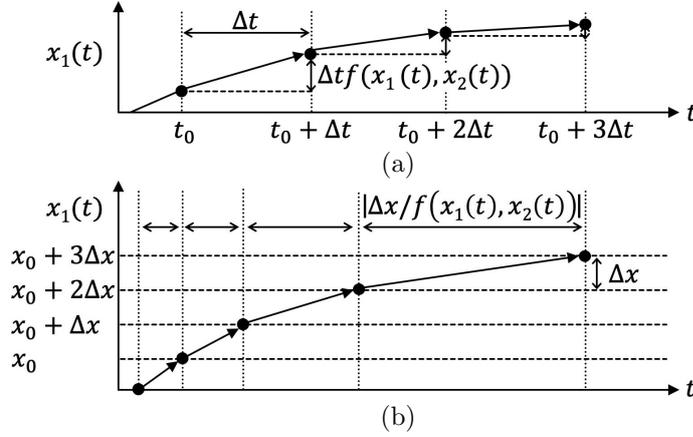


Figure 2.2.10: (a) Forward Euler method. (b) Our method.  $t_0$  and  $x_0$  are constant.

## Appendix: Derivation of functions $f_X$ and $f_Y$

The proposed cochlea model is based on the Hopf-type cochlea model in Eq. (2.2.8). The forward Euler formula for the model in Eq. (2.2.10) without the input term is rewritten as follows.

$$\begin{aligned} x_1 \left( t + \left| \frac{\Delta x}{f_1(x_1(t), x_2(t))} \right| \right) &= x_1(t) + \Delta x \overline{\text{sgn}}(f_1(x_1(t), x_2(t))), \\ x_2 \left( t + \left| \frac{\Delta x}{f_2(x_1(t), x_2(t))} \right| \right) &= x_2(t) + \Delta x \overline{\text{sgn}}(f_2(x_1(t), x_2(t))), \end{aligned} \quad (2.2.11)$$

where  $\Delta x \in \mathbb{R}$  denotes a discretized state step and the function  $\overline{\text{sgn}} : \mathbb{R} \rightarrow \mathbf{Q}$  denotes  $\overline{\text{sgn}}(x) = 1$  for  $x \geq 0$  and  $\overline{\text{sgn}}(x) = -1$  for  $x < 0$ . In the left-hand side of Eq. (2.2.11),  $|\Delta x / f_1(x_1(t), x_2(t))|$  and  $|\Delta x / f_2(x_1(t), x_2(t))|$  represent the amounts of time advance per unit distance  $\Delta x$ . Furthermore, in the right-hand side of Eq. (2.2.11), the amounts of state transitions of  $x_1$  and  $x_2$  are restricted to  $\Delta x$  or  $-\Delta x$ . Fig. 2.2.10 shows the relationship between the forward Euler method and our method. In the proposed cochlea model, the discrete state variables  $\{X, Y\}$  and the functions  $\{f_X, f_Y\}$  correspond to  $\{x_1, x_2\}$  and  $\{\Delta x / f_1(x_1(t), x_2(t)), \Delta x / f_2(x_1(t), x_2(t))\}$  in Eq. (2.2.11), where  $\Delta x = 1$ . Also, the discrete auxiliary variables  $\{P_X, P_Y\}$  whose dynamic equations are defined in Eq. (2.2.4) work as state-dependent frequency dividers using the functions  $\{f_X, f_Y\}$  (see also the bent black arrows in Fig. 2.2.3).

## Bibliography

- [1] J. O. Pickles, *An Introduction to the Physiology of Hearing*, 4th ed. Bingley, U.K.: Emerald Group, 2012.
- [2] J. Wang, N. L. Powers, P. Hofstetter, P. Trautwein, D. Ding, and R. Salvi, "Effects of selective inner hair cell loss on auditory nerve fiber threshold, tuning and spontaneous and driven discharge rate," *Hearing Research*, vol. 107, no. 1-2, pp. 67–82, 1997.

- [3] A. C. Crawford and R. Fettiplace, “The frequency selectivity of auditory nerve fibres and hair cells in the cochlea of the turtle,” *The Journal of Physiology*, vol. 306, no. 1, pp. 79–125, 1980.
- [4] M. Ospeck, V. M. Eguíluz, and M. O. Magnasco, “Evidence of a Hopf Bifurcation in Frog Hair Cells,” *Biophysical Journal*, vol. 80, no. 6, pp. 2597–2607, 2001.
- [5] A. J. Hudspeth, F. Jülicher, and P. Martin, “A Critique of the Critical Cochlea: Hopf—a Bifurcation—Is Better Than None,” *Journal of Neurophysiology*, vol. 104, no. 3, pp. 1219–1229, 2010.
- [6] T. Lorimer, F. Gomez, and R. Stoop, “Mammalian cochlea as a physics guided evolution-optimized hearing sensor,” *Scientific Reports*, vol. 5, no. 1, 12492, 2015.
- [7] A. Mishra and A. E. Hubbard, “A cochlear filter implemented with a field-programmable gate array,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 49, no. 1, pp. 54–60, 2002.
- [8] L. Watts, D. A. Kerns, R. F. Lyon, and C. A. Mead, “Improved implementation of the silicon cochlea,” *IEEE Journal of Solid-State Circuits*, vol. 27, no. 5, pp. 692–700, 1992.
- [9] M. Reit, W. Mathis, and R. Stoop, “Time-Discrete Nonlinear Cochlea Model Implemented on DSP for Auditory Studies,” in *Proc. of the Nonlinear Dynamics of Electronic Systems 2012*, Wolfenbuettel, Germany, 2012, pp. 17–20.
- [10] T. Noguchi and H. Torikai, “Ghost Stochastic Resonance From an Asynchronous Cellular Automaton Neuron Model,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 2, pp. 111–115, 2013.
- [11] K. Isobe and H. Torikai, “A Novel Hardware-Efficient Asynchronous Cellular Automaton Model of Spike-Timing-Dependent Synaptic Plasticity,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 6, pp. 603–607, 2016.
- [12] N. Shimada and H. Torikai, “A Novel Asynchronous Cellular Automaton Multicompartment Neuron Model,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 776–780, 2015.
- [13] M. Izawa and H. Torikai, “A novel hardware-efficient cochlea model based on asynchronous cellular automaton,” in *Proc. of the 2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, 2015, pp. 1–8.
- [14] K. Takeda and H. Torikai, “Reproduction of nonlinear cochlea response by asynchronous bifurcation processor,” in *Proc. of the 2016 International Symposium on Nonlinear Theory and Its Applications (NOLTA)*, Yugawara, Japan, 2016, pp. 279–282.
- [15] Y. A. Kuznetsov, *Elements of Applied Bifurcation Theory*, 3rd ed., ser. Applied Mathematical Sciences, S. S. Antman, J. E. Marsden, and L. Sirovich, Eds. New York: Springer, 2004, vol. 112.

- [16] A. Ruggero, “Responses to sound of the basilar membrane of the mammalian cochlea,” *Current Opinion in Neurobiology*, vol. 2, no. 4, pp. 449–456, 1992.
- [17] M. A. Ruggero, N. C. Rich, A. Recio, S. S. Narayan, and L. Robles, “Basilar-membrane responses to tones at the base of the chinchilla cochlea,” *The Journal of the Acoustical Society of America*, vol. 101, no. 4, pp. 2151–2163, 1997.
- [18] L. Robles, M. A. Ruggero, and N. C. Rich, “Two-Tone Distortion on the Basilar Membrane of the Chinchilla Cochlea,” *Journal of Neurophysiology*, vol. 77, no. 5, pp. 2385–2399, 1997.
- [19] V. M. Eguíluz, M. Ospeck, Y. Choe, A. J. Hudspeth, and M. O. Magnasco, “Essential Nonlinearities in Hearing,” *Physical Review Letters*, vol. 84, no. 22, pp. 5232–5235, 2000.
- [20] A. Kern and R. Stoop, “Essential Role of Couplings between Hearing Nonlinearities,” *Physical Review Letters*, vol. 91, no. 12, 128101, 2003.
- [21] R. Stoop and A. Kern, “Two-Tone Suppression and Combination Tone Generation as Computations Performed by the Hopf Cochlea,” *Physical Review Letters*, vol. 93, no. 26, 268103, 2004.
- [22] R. Stoop, W. H. Steeb, J. C. Gallas, and A. Kern, “Auditory two-tone suppression from a subcritical Hopf cochlea,” *Physica A: Statistical Mechanics and its Applications*, vol. 351, no. 1, pp. 175–183, 2005.
- [23] S. Martignoli and R. Stoop, “Local Cochlear Correlations of Perceived Pitch,” *Physical Review Letters*, vol. 105, no. 4, p. 048101, 2010.
- [24] S. Martignoli, F. Gomez, and R. Campos, “Pitch sensation involves stochastic resonance,” *Scientific Reports*, vol. 3, no. 1, 2676, 2013.
- [25] F. Gomez and R. Stoop, “Mammalian pitch sensation shaped by the cochlear fluid,” *Nature Physics*, vol. 10, no. 7, pp. 530–536, 2014.
- [26] S. Martignoli, J. J. van der Vyver, A. Kern, Y. Uwate, and R. Stoop, “Analog electronic cochlea with mammalian hearing characteristics,” *Applied Physics Letters*, vol. 91, no. 6, 064108, 2007.
- [27] R. Stoop, T. Jasa, Y. Uwate, and S. Martignoli, “From Hearing to Listening: Design and Properties of an Actively Tunable Electronic Hearing Sensor,” *Sensors*, vol. 7, no. 12, pp. 3287–3298, 2007.
- [28] T. J. Hamilton, C. Jin, J. Tapson, and A. van Schaik, “A 2-D Cochlea with Hopf Oscillators,” in *Proc. of the 2007 IEEE Biomedical Circuits and Systems Conference*, Montreal, QC, Canada, 2007, pp. 91–94.
- [29] B. Wen and K. Boahen, “A Silicon Cochlea With Active Coupling,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 3, no. 6, pp. 444–455, 2009.

- [30] M. Nouri, A. Ahmadi, S. Alirezaee, G. Karimi, M. Ahmadi, and D. Abbott, "A Hopf Resonator for 2-D Artificial Cochlea: Piecewise Linear Model and Digital Implementation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 4, pp. 1117–1125, 2015.
- [31] A. Paul and K. P. Menon, "Digital implementation of hyperbolic calculation unit for electronic cochlea on FPGA," in *Proc. of the 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, 2017, pp. 1322–1325.
- [32] C. S. Thakur, T. J. Hamilton, J. Tapson, A. van Schaik, and R. F. Lyon, "FPGA implementation of the CAR Model of the cochlea," in *Proc. of the 2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, Melbourne VIC, Australia, 2014, pp. 1853–1856.
- [33] L. Minati, M. Frasca, N. Yoshimura, and Y. Koike, "Versatile Locomotion Control of a Hexapod Robot Using a Hierarchical Network of Nonlinear Oscillator Circuits," *IEEE Access*, vol. 6, pp. 8042–8065, 2018.
- [34] A. Wagemakers and M. A. F. Sanjuán, "Electronic circuit implementation of the chaotic Rulkov neuron model," *Journal of the Franklin Institute*, vol. 350, no. 10, pp. 2901–2910, 2013.
- [35] F. Folowosele, T. J. Hamilton, and R. Etienne-Cummings, "Silicon Modeling of the Mihalas–Niebur Neuron," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 1915–1927, 2011.
- [36] N. Korkmaz, İ. Öztürk, and R. Kılıç, "The investigation of chemical coupling in a HR neuron model with reconfigurable implementations," *Nonlinear Dynamics*, vol. 86, no. 3, pp. 1841–1854, 2016.
- [37] T. Matsubara and H. Torikai, "Asynchronous Cellular Automaton-Based Neuron: Theoretical Analysis and On-FPGA Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 736–748, 2013.
- [38] M. Gholami and S. Saeedi, "Digital Cellular Implementation of Morris-Lecar Neuron Model," in *Proc. of the 23rd Iranian Conference on Electrical Engineering*, Tehran, Iran, 2015, pp. 1235–1239.
- [39] M. Izawa and H. Torikai, "Asynchronous Cellular Automaton Model of Spiral Ganglion Cell in the Mammalian Cochlea: Theoretical Analyses and FPGA Implementation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E98.A, no. 2, pp. 684–699, 2015.
- [40] H. Ishimoto, M. Izawa, and H. Torikai, "A novel cochlea partition model based on asynchronous bifurcation processor," *Nonlinear Theory and Its Applications, IEICE*, vol. 6, no. 2, pp. 207–225, 2015.

- [41] K. Takeda and H. Torikai, “A Novel Hardware-Efficient Cochlea Model Based on Asynchronous Cellular Automaton Dynamics: Theoretical Analysis and FPGA Implementation,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 9, pp. 1107–1111, 2017.
- [42] T. Naka and H. Torikai, “A Novel Generalized Hardware-Efficient Neuron Model Based on Asynchronous CA Dynamics and Its Biologically Plausible On-FPGA Learnings,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 7, pp. 1247–1251, 2019.
- [43] K. Takeda and H. Torikai, “A Novel Asynchronous CA Neuron Model: Design of Neuron-Like Nonlinear Responses Based on Novel Bifurcation Theory of Asynchronous Sequential Logic Circuit,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 6, pp. 1989–2001, 2020.
- [44] K. Takeda and H. Torikai, “A Novel Hardware-Efficient Central Pattern Generator Model Based on Asynchronous Cellular Automaton Dynamics for Controlling Hexapod Robot,” *IEEE Access*, vol. 8, pp. 139 609–139 624, 2020.

# Chapter 3

## CPG Model based on Asynchronous Cellular Automaton

### 3.1 Snake-like Robot Controlled by Coupled Limit Cycle Oscillators<sup>1</sup>

#### 3.1.1 Introduction

Central pattern generators (CPGs) are intraspinal networks of neural oscillators capable of producing rhythmic output signals to control motor systems in part of animal locomotion [1]. Many mathematical models of CPGs have been proposed and analyzed so far [2–8]. Also, CPG models have been used to control artificial robots of various kinds [3–8] including snake-like robots [5–8] like the one in Fig. 3.1.1. The purpose of this paper is to propose a novel kind of CPG model, which consumes much fewer hardware resources compared to conventional numerical integration models. First, the novel CPG model the dynamics of which is described by an asynchronous cellular automaton is proposed. By intensive numerical analyses, it is shown that the proposed model can generate multi-phase synchronized periodic signals, which are suitable for controlling a serpentine motion of a snake-like robot. The intensive numerical analyses also reveal roles of parameters. Then, the proposed model is implemented on a field programmable gate array (FPGA) and is used to control a snake-like robot. It is shown by experimental validation using a prototype machine that the proposed model can realize rhythmic locomotor activity in snakes. Moreover, it is shown that the proposed model consumes much fewer hardware resources (FPGA slices) than a typical numerical integration CPG model. Novelties and significances of this paper include the following points.

- The dynamics of most conventional CPG models are described by differential equations and are implemented as numerical integrations. On the other hand, the dynamics of the proposed CPG model is described by the asynchronous cellular automaton and is implemented as an asynchronous sequential logic circuit. Although some asynchronous cellular automaton models of biological systems have been proposed so far [9, 10], the asynchronous cellular automaton CPG model is proposed in this paper for the first time. Note that, from a fundamental research view point, such an exploration of designing a bio-inspired engineering system (like the snake-like robot controlled by

---

<sup>1</sup>This section is based on “A Novel Hardware-Efficient CPG Model based on Asynchronous Cellular Automaton,” by the same author, which appeared in *IEICE Electronics Express*, vol. 15, no. 11, 20180387, 2018, Copyright(C)2018 IEICE.

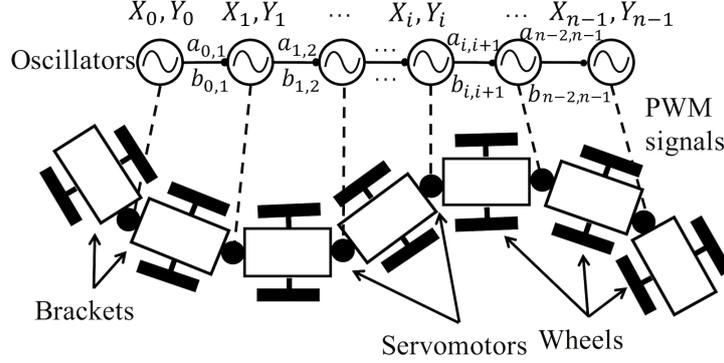


Figure 3.1.1: Snake-like robot controlled by the proposed central pattern generator model (CPG model), which consists of asynchronous cellular automaton oscillators (ACA oscillators).

the CPG model) base on an unconventional method (like the asynchronous cellular automaton modeling method) *per se* is a significant research topic.

- As explained above, roles of parameters of the proposed model are revealed. From a practical view point, this is significant since these roles show how to adjust locomotion of the snake-like robot. Actually, in this paper, parameter setting methods to adjust moving velocity of the snake-like robot are shown based on the revealed roles of the parameters.

### 3.1.2 Model description

In this section, a novel central pattern generator model (CPG model) based on an asynchronous cellular automaton is proposed. Fig. 3.1.1 shows a schematic of a snake-like robot controlled by the proposed CPG model, which consists of asynchronous cellular automaton oscillators (ACA oscillators).

#### 3.1.2.1 ASYNCHRONOUS CA OSCILLATOR

In this subsection, we propose the ACA oscillator, which has discrete states variables  $X_i \in \mathbf{Z}_N = \{0, \dots, N-1\}$  and  $Y_i \in \mathbf{Z}_N$ , and discrete auxiliary variables  $P_i \in \mathbf{Z}_M = \{0, \dots, M-1\}$  and  $Q_i \in \mathbf{Z}_M$ , where  $i$  is an index for the ACA oscillators, and  $N$  and  $M$  are natural numbers that determine resolutions of the discrete variables  $\{X_i, Y_i, P_i, Q_i\}$ . In order to design a vector field of the ACA oscillator, inspired by [7], the following functions  $g_i : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbf{R}$  and  $h_i : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbf{R}$  are prepared:  $g_i(X_i, Y_i) = -\omega_i((X_i - \frac{N}{2})^2 + (Y_i - \frac{N}{2})^2 - \delta_i)(X_i - \frac{N}{2})\delta_i^{-1} - \omega_i(Y_i - \frac{N}{2})$  and  $h_i(X_i, Y_i) = -\omega_i(X_i - \frac{N}{2})$ , where  $\delta_i \in \mathbf{R}$  and  $\omega_i \in \mathbf{R}$  are parameters. Note that the functions  $g_i$  and  $h_i$  are not implemented as hardware but are used to design the following discrete functions  $G_i : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbf{Z}_M^\pm = \{-(M-1), \dots, 0, \dots, M-1\}$  and  $H_i : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbf{Z}_M^\pm$ .

$$G_i(X_i, Y_i) = \left\lfloor \frac{\alpha_x}{g_i(X_i, Y_i)} \right\rfloor, \quad H_i(X_i, Y_i) = \left\lfloor \frac{\alpha_y}{h_i(X_i, Y_i)} \right\rfloor, \quad (3.1.1)$$

where the functions  $G_i$  and  $H_i$  are assumed to be saturated at  $\pm(M-1)$  hereafter, i.e.,  $G_i(X_i, Y_i) = (M-1)$  for  $(M-1) \leq \lfloor \frac{\alpha_x}{g_i(X_i, Y_i)} \rfloor$ ,  $G_i(X_i, Y_i) = -(M-1)$  for  $-(M-1) \geq$

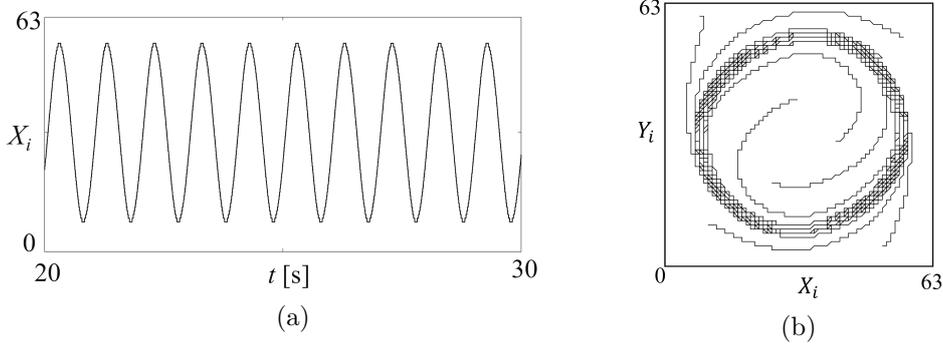


Figure 3.1.2: (a) Typical time waveform of the proposed asynchronous cellular automaton oscillator.  $(N, M, T_X, T_Y, \alpha_x, \alpha_y, \delta_i, \omega_i) = (2^6, 2^{10}, 0.0001, 0.00015\sqrt{1.00020134}, 10000, \frac{100000}{15\sqrt{1.00020134}}, 25^2, 2\pi \times 1.0)$ . (b) Phase plane trajectories for different initial states. The mean radius is  $r_i \simeq 25$ .

$\lfloor \frac{\alpha_x}{g_i(X_i, Y_i)} \rfloor$ ,  $H_i(X_i, Y_i) = (M - 1)$  for  $(M - 1) \leq \lfloor \frac{\alpha_y}{h_i(X_i, Y_i)} \rfloor$ ,  $H_i(X_i, Y_i) = -(M - 1)$  for  $-(M - 1) \geq \lfloor \frac{\alpha_y}{h_i(X_i, Y_i)} \rfloor$ , and  $\lfloor \cdot \rfloor$  denotes the floor function. The ACA oscillator has two internal clocks  $C_X(t)$  and  $C_Y(t)$ , which are described by  $C_X(t) = 1$  if  $t = 0, T_X, 2T_X \dots$  and  $C_X(t) = 0$  otherwise; and  $C_Y(t) = 1$  if  $t = 0, T_Y, 2T_Y \dots$  and  $C_Y(t) = 0$  otherwise, where  $T_X \in \mathbf{R}^+ = \{t | t \geq 0\}$  and  $T_Y \in \mathbf{R}^+$  are periods of  $C_X(t)$  and  $C_Y(t)$ , respectively. In this paper, the clock generators are assumed to be uncoupled and thus the clocks  $C_X(t)$  and  $C_Y(t)$  are assumed to be asynchronous. These asynchronous internal clocks  $C_X(t)$  and  $C_Y(t)$  trigger the following asynchronous transitions of the discrete auxiliary variables  $P_i$  and  $Q_i$ , respectively.

$$\text{If } C_X(t) = 1, \text{ then } P_i(t_+) := \begin{cases} P_i(t) + 1 & \text{if } P_i(t) < |G_i(X_i(t), Y_i(t))|, \\ 0 & \text{if } P_i(t) \geq |G_i(X_i(t), Y_i(t))|, \end{cases} \quad (3.1.2)$$

$$\text{If } C_Y(t) = 1, \text{ then } Q_i(t_+) := \begin{cases} Q_i(t) + 1 & \text{if } Q_i(t) < |H_i(X_i(t), Y_i(t))|, \\ 0 & \text{if } Q_i(t) \geq |H_i(X_i(t), Y_i(t))|, \end{cases} \quad (3.1.3)$$

where the symbol “ $t_+$ ” denotes “ $\lim_{\epsilon \rightarrow +0} t + \epsilon$ ” and the symbol “ $:=$ ” denotes an “instantaneous state transition” hereafter. In addition, the asynchronous internal clocks  $C_X(t)$  and  $C_Y(t)$  trigger the following asynchronous transitions of the discrete state variables  $X_i$  and  $Y_i$ , respectively.

If  $C_X(t) = 1$  and  $P_i(t) \geq |G_i(X_i(t), Y_i(t))|$ , then

$$X_i(t_+) := \begin{cases} X_i(t) + 1 & \text{if } X_i(t) \neq N - 1 \text{ and } G_i(X_i(t), Y_i(t)) \geq 0, \\ X_i(t) - 1 & \text{if } X_i(t) \neq 0 \text{ and } G_i(X_i(t), Y_i(t)) < 0, \end{cases} \quad (3.1.4)$$

If  $C_Y(t) = 1$  and  $Q_i(t) \geq |H_i(X_i(t), Y_i(t))|$ , then

$$Y_i(t_+) := \begin{cases} Y_i(t) + 1 & \text{if } Y_i(t) \neq N - 1 \text{ and } H_i(X_i(t), Y_i(t)) \geq 0, \\ Y_i(t) - 1 & \text{if } Y_i(t) \neq 0 \text{ and } H_i(X_i(t), Y_i(t)) < 0. \end{cases} \quad (3.1.5)$$

Fig. 3.1.2(a) shows a typical time waveform of the discrete state variable  $X_i$  and Fig.

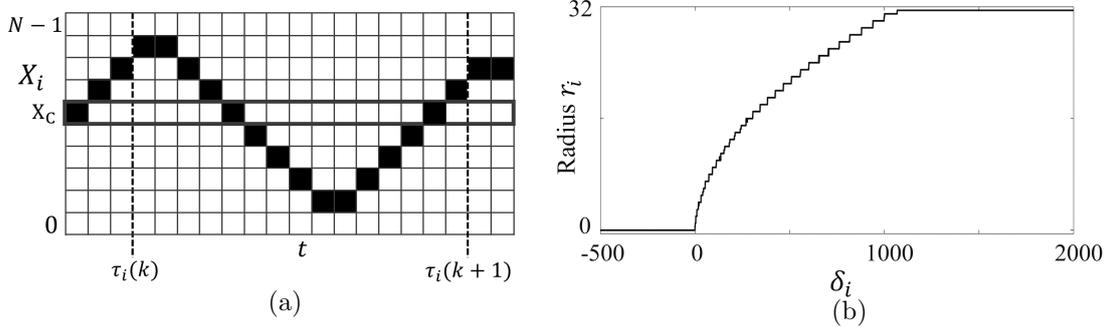


Figure 3.1.3: (a) Definition of  $\tau_i(k)$ , which is the moment when  $X_i$  transits into a local maximum value. (b) Characteristics of the mean radius  $r_i$  of the discrete state variable  $X_i$  with respect to the parameter  $\delta_i$ .  $(N, M, T_X, T_Y, \alpha_x, \alpha_y, \omega_i) = (2^6, 2^{10}, 0.0001, 0.00015\sqrt{1.00020134}, 10000, \frac{100000}{15\sqrt{1.00020134}}, 2\pi \times 1.0)$ .

3.1.2(b) shows phase plane trajectories for different initial states. It can be seen that the ACA oscillator oscillates in a steady state. In order to characterize such an oscillation, let us introduce the following definitions (see also Fig. 3.1.3(a)).

**Definition 1a:** Assume the discrete state variable  $X_i$  is in a steady state. Assume the discrete state variable  $X_i$  ever repeats the following two behaviors: (i)  $X_i$  increases and transits into a certain constant value  $X_C$  and (ii)  $X_i$  decreases and transits from  $X_C$ . Under these assumptions, let  $\tau_i(k)$  denote the moment when the discrete state variable  $X_i$  transits to its maximum value from the moment when (i) occurs to the moment when the next (i) occurs. Then, the discrete state variable  $X_i$  is said to have a *mean radius*  $r_i$  defined by

$$r_i = \frac{1}{2K} \sum_{k=1}^K \max_{\tau_i(k) \leq t < \tau_i(k+1)} \{X_i(t)\} - \min_{\tau_i(k) \leq t < \tau_i(k+1)} \{X_i(t)\},$$

where  $K$  is a sufficiently large positive integer.

**Definition 1b:** Assume the discrete state variable  $X_i$  is in a steady state and ever stays in a certain constant value  $X_C$ . Then, the discrete state variable  $X_i$  is said to have a *mean radius*  $r_i = 0$ .

For example, the discrete state variable  $X_i$  in Fig. 3.1.2 has a mean radius  $r_i \simeq 25.0$ . Fig. 3.1.3(b) shows the characteristics of the mean radius  $r_i$  with respect to the parameter  $\delta_i$ . It can be seen that the ACA oscillator does not oscillate for  $\delta_i < 0$  and oscillates for  $\delta_i > 0$ . Hence the ACA oscillator exhibits a bifurcation near  $\delta_i = 0$ . Since the mean radius  $r_i$  is approximately proportional to  $\sqrt{\delta_i}$  for  $\delta_i > 0$  near  $\delta_i = 0$ , the bifurcation is similar to the supercritical Hopf-bifurcation [11]. Since our goal is to design the CPG model, the ACA oscillator is expected to oscillate and thus the parameter  $\delta_i$  should be set to a positive value. Then we assume the assumptions in Definition 1a. Under these assumptions, let us give the following definition.

**Definition 2:** Assume the assumptions in Definition 1a are satisfied. Then, the discrete

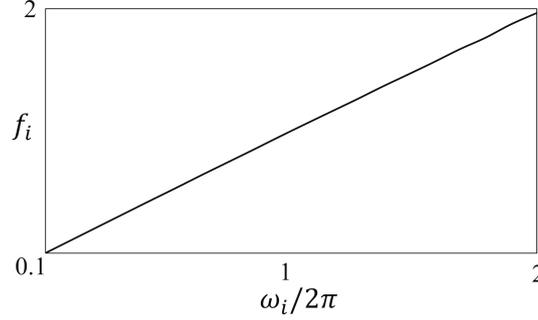


Figure 3.1.4: Characteristics of the mean frequency  $f_i$  of the discrete state variable  $X_i$  with respect to the parameter  $\omega_i$ .  $(N, M, T_X, T_Y, \alpha_x, \alpha_y, \delta_i) = (2^6, 2^{10}, 0.0001, 0.00015\sqrt{1.00020134}, 10000, \frac{100000}{15\sqrt{1.00020134}}, 25^2)$ .

Table 3.1.1: Periods  $(T_X, T_Y)$  of the clocks and the mean frequency  $f_i$  of the discrete state variable  $X_i$ .  $(N, M, \delta_i, \omega_i) = (2^6, 2^{10}, 25^2, 2\pi \times 1.0)$ .

Internal clocks	Period $T_X$	Period $T_Y$	Mean frequency $f_i$
Frequency locked	0.0001	0.00015	1.00004800520048031
Asynchronous	0.0001	$0.00015\sqrt{1.00020134}$	1.000000000000000000

state variable  $X_i$  is said to have a *mean frequency*  $f_i$  defined by

$$f_i = \frac{K}{\sum_{k=1}^K \tau_i(k+1) - \tau_i(k)}.$$

For example, the discrete state variable  $X_i$  in Fig. 2 has a mean frequency  $f_i \simeq 1.0$ . Fig. 3.1.4 shows the characteristics of the mean frequency  $f_i$  with respect to the parameter  $\omega_i$  and Table 3.1.1 shows the mean frequency  $f_i$  for different values of the period  $T_Y$  of the internal clock  $C_Y(t)$ . Based on these characteristics, roles of the parameters can be summarized as follows.

- The mean radius  $r_i$  of the discrete state variable  $X_i$  can be adjusted by the parameter  $\delta_i$  as shown in Fig. 3(b).
- The mean frequency  $f_i$  of the discrete state variable  $X_i$  can be adjusted by the parameter  $\omega_i$  as shown in Fig. 4.
- The mean frequency  $f_i$  of the discrete state variable  $X_i$  can be fine-tuned by the parameter  $T_Y$  as shown in Table 3.1.1.

### 3.1.2.2 CENTRAL PATTERN GENERATOR MODEL

In this subsection, the novel CPG model, which consists of coupled  $n$  ACA oscillators, is proposed. In order to couple the ACA oscillators, the discrete function  $G_i$  in Eq. (3.1.1)

is modified as follows.

$$G_i(X_0, \dots, X_{n-1}, Y_0, \dots, Y_{n-1}) = \left[ \frac{\alpha_x}{g_i(X_i, Y_i) + \sum_{j=0}^{n-1} (a_{i,j} Y_j + b_{i,j} X_j)} \right],$$

where  $a_{i,j}$  and  $b_{i,j}$  are coupling constants between the  $i$ -th and the  $j$ -th ACA oscillators, respectively, and  $i, j \in \{0, 1, \dots, n-1\}$ . In this paper, the following coupling constants are focused on.

$$a_{i,i+1} = k_a \text{ and } b_{i,i+1} = k_b \text{ for } i = 0, 1, \dots, n-2,$$

where other  $a_{i,j}$  and  $b_{i,j}$  are zeros. For example, the CPG model in Fig. 1 is characterized by such coupling constants. Fig. 3.1.5 shows typical time waveforms of the proposed CPG model. It can be seen that the ACA oscillators oscillate with the same mean frequency in Fig. 3.1.5(a) and with different mean frequencies in Fig. 3.1.5(b). In order to characterize such oscillations, let us introduce the following definition.

**Definition 3:** If the ratio  $f_i : f_j$  of the mean frequencies  $f_i$  and  $f_j$  of the  $i$ -th and the  $j$ -th ACA oscillators can be represented by a ratio  $n : m$  of integers  $n$  and  $m$ , then the ACA oscillators are said to exhibit an  $n : m$  *synchronization*. In this case,  $\frac{f_i}{f_j}$  is referred to as a *mean frequency ratio* of the  $i$ -th and the  $j$ -th ACA oscillators.

For example, the ACA oscillators in Fig. 3.1.5(a) exhibit an 1 : 1 synchronization with the mean frequency ratio  $\frac{f_0}{f_1} = 1$  and the ACA oscillators in Fig. 3.1.5(b) exhibit a 2 : 1 synchronization with the mean frequency ratio  $\frac{f_0}{f_1} = 2$ . Fig. 3.1.5(c) shows the characteristics of the mean frequency ratio  $\frac{f_0}{f_1}$  with respect to the parameter  $\omega_0$ . Fig. 3.1.5(d) shows the characteristics of the mean frequency ratio  $\frac{f_0}{f_1}$  with respect to the parameters  $(\omega_0, k_a)$ . In this figure, the parameter region labeled by 1 : 1 leads to the 1 : 1 synchronization. Note that in this paper we design the CPG model, which utilizes the 1 : 1 synchronization. So, the 1 : 1 synchronization is focused on hereafter. Fig. 3.1.6 shows various 1 : 1 synchronizations with different phase differences. In order to characterize such phenomena, let us introduce the following definition (see also Fig. 3.1.7).

**Definition 4:** Assume the assumptions in Definition 1a are satisfied and assume the ACA oscillators in the CPG model exhibit 1 : 1 synchronizations. Then

$$\phi_{i,j} = \frac{1}{K} \sum_{k=1}^K ((\tau_i(k) \pmod{f_i^{-1}}) - (\tau_j(k) \pmod{f_j^{-1}}))$$

is referred to as a phase difference between the  $i$ -th and the  $j$ -th ACA oscillators. Also,

$$\Phi = \frac{1}{n-1} \sum_{i=0}^{n-2} \phi_{i,i+1}$$

is referred to as a *mean phase difference* of the ACA oscillators in the CPG model.

For example, the ACA oscillators in Figs. 3.1.6(a), (b), (c) and (d) have mean phase differences  $\Phi = 1.071, 0.849, 0.599$  and  $-0.472$ , respectively. Fig. 3.1.8 shows the characteristics

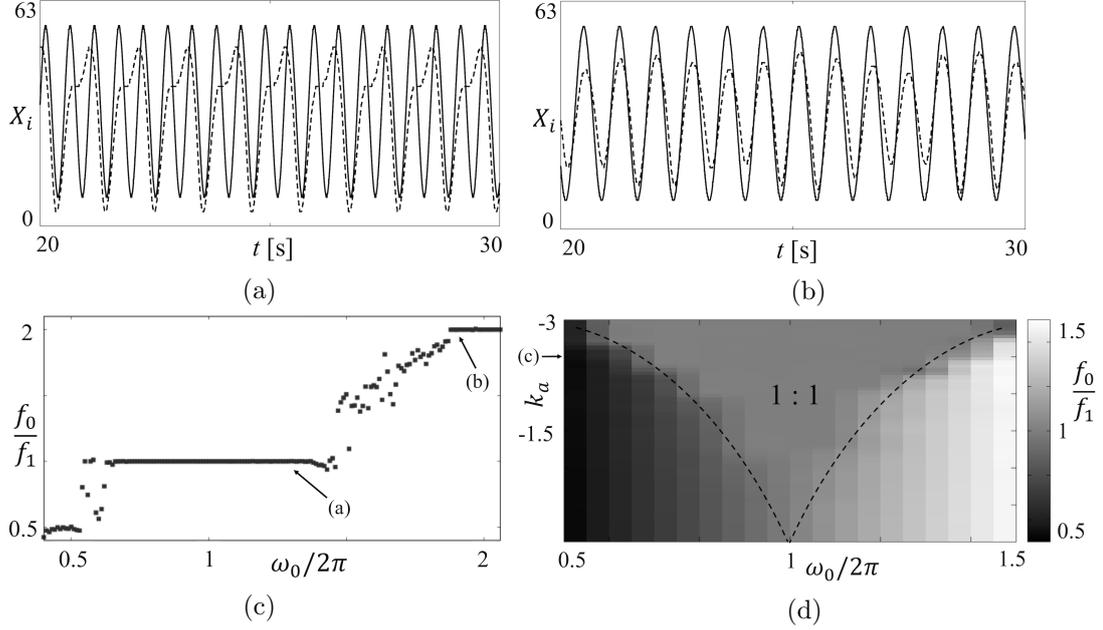


Figure 3.1.5: Typical time waveforms of the CPG model with  $n = 2$  ACA oscillators and their characteristics.  $(N, M, T_X, T_Y, \alpha_x, \alpha_y, \delta_0, \delta_1, \omega_1) = (2^6, 2^{10}, 0.0001, 0.00015\sqrt{1.00020134}, 10000, \frac{100000}{15\sqrt{1.00020134}}, 25^2, 25^2, 2\pi \times 1.0)$ .  $k_b = 0$ . In (a) and (b), the solid curves and the dashed curves show discrete states variables  $X_0$  and  $X_1$ , respectively. (a) 1 : 1 synchronization.  $(\omega_0, k_a) = (2\pi \times 1.3, -2.5)$ . (b) 2 : 1 synchronization.  $(\omega_0, k_a) = (2\pi \times 1.9, -2.5)$ . (c) Characteristic of the mean frequency ratio  $\frac{f_0}{f_1}$  with respect to the parameter  $\omega_0$ .  $k_a = 1.0$ . The arrows (a) and (b) correspond to the figures (a) and (b), respectively. (d) Characteristic of the mean frequency ratio  $\frac{f_0}{f_1}$  with respect to the parameters  $(\omega_0, k_a)$ . The arrow (c) corresponds to the figure (c).

of the mean phase difference  $\Phi$  with respect to the parameter  $k_a$ . Based on the characteristics of the CPG model in Figs. 5 and 8 and the characteristics of the ACA oscillator in Fig. 3(b), Fig. 4, and Table 3.1.1, roles of the parameters from a viewpoint of the CPG model for controlling the snake-like robot in Fig. 1 can be summarized as follows.

- The amplitude of the oscillating rotary motion of the servomotor can be adjusted by the parameter  $\delta_i$ .
- The frequency of the oscillating rotary motion of the servomotor can be adjusted by the parameter  $\omega_i$  and fine-tuned by the parameter  $T_Y$ .
- The phase difference of the oscillating rotary motions of the servomotors can be adjusted by the parameter  $k_a$ . Since the moving direction (forward or backward) of the snake-like robot is determined by the phase difference (see Fig. 3.1.8), it can be adjusted by the parameter  $k_a$ .
- The moving velocity of the snake-like robot can be adjusted by the frequency and the phase differences of the oscillating rotary motions of the servomotors under the condition that the amplitudes of the oscillating rotary motions are appropriately set.

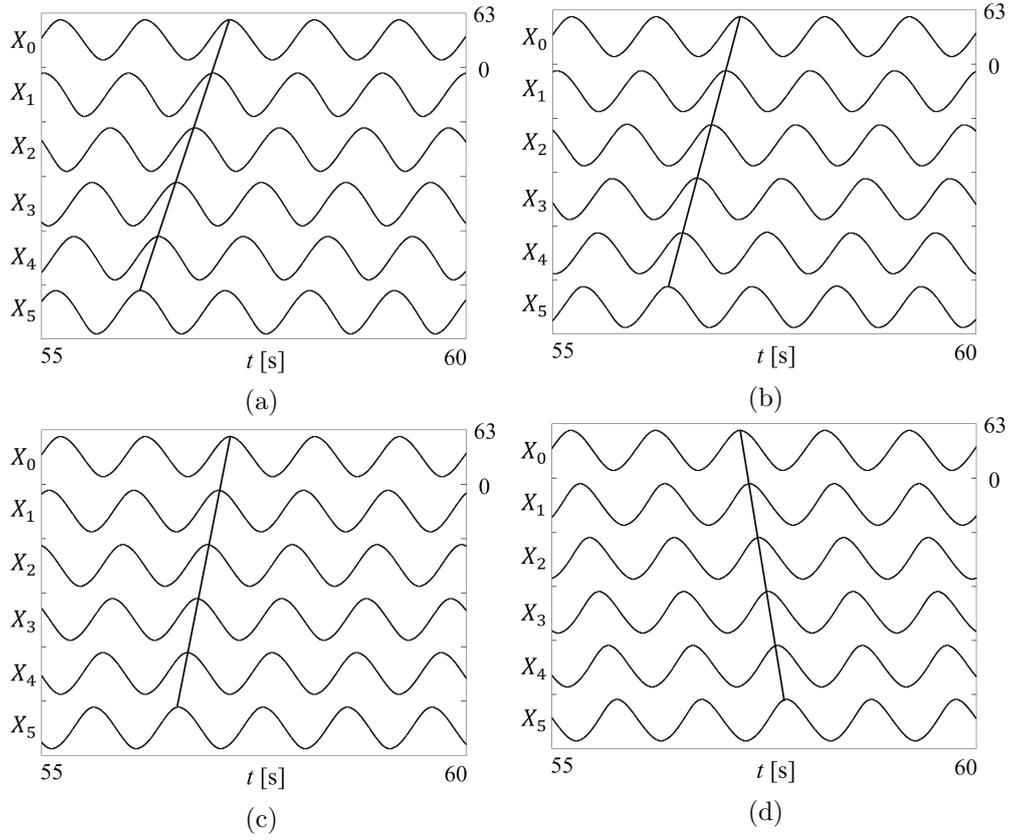


Figure 3.1.6: 1 : 1 synchronizations of the proposed CPG with various mean phase differences  $\Phi$ .  $(N, M, T_X, T_Y, \alpha_x, \alpha_y, \delta_i, \omega_i) = (2^6, 2^{10}, 0.0001, 0.00015\sqrt{1.00020134}, 10000, \frac{100000}{15\sqrt{1.00020134}}, 25^2, 2\pi \times 1.0)$  for all  $i$ .  $(n, k_b) = (6, 0.3)$ . (a)  $k_a = -1.0$ .  $\Phi = 1.071$ . (b)  $k_a = -0.5$ .  $\Phi = 0.849$ . (c)  $k_a = -0.3$ .  $\Phi = 0.599$ . (d)  $k_a = 0.3$ .  $\Phi = -0.472$ .

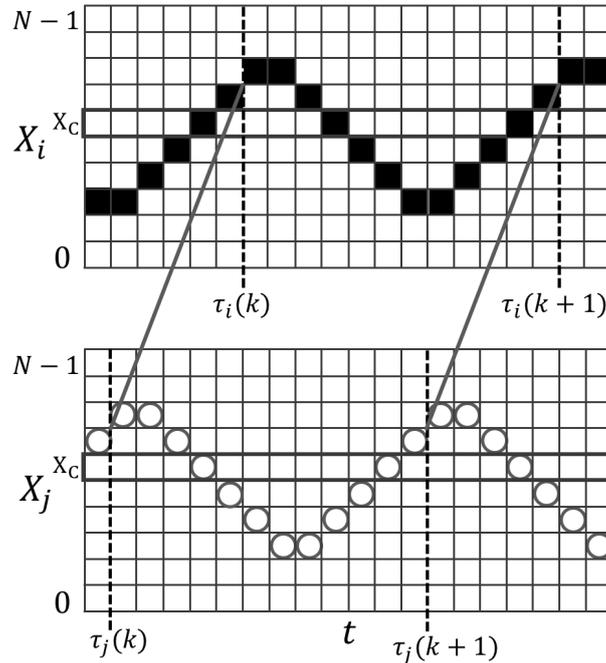


Figure 3.1.7: Definition of the phase difference  $\phi_{i,j}$ . The arrows represent  $\tau_i(k) - \tau_j(k)$  and  $\tau_i(k+1) - \tau_j(k+1)$ .

Therefore, the moving velocity of the snake-like robot can be adjusted by the parameters  $(\omega_i, k_a)$  under an appropriate setting of the value of the parameter  $\delta_i$ .

Hence, in the next section, effects of the parameters  $(\omega, k_a)$  to the moving velocity of the snake-like robot are analyzed.

### 3.1.3 Hardware implementation and comparison

Recall that the dynamics of the proposed CPG model is described by Eqs. (3.1.2)–(3.1.5). These equations are written in a VHDL code, which is compiled into a bitstream file by using Xilinx’s design software environment Vivado 2016. 2. The resulting bitstream file is downloaded to Xilinx’s field programmable gate array (FPGA) XC7K325T-2FFG900C. Fig. 3.1.9(a) shows a picture of the snake-like robot controlled by the FPGA-implemented CPG model. As shown in the figure, the snake-like robot consists of  $n = 6$  sections and an extra bracket-B attached to the head of the snake, and each section consists of two brackets, bracket-A and bracket-B. Each  $i$ -th bracket-A has Tower Pro’s servo motor SG92R, the target angle of which is instructed by the discrete state variable  $X_i$  of the CPG via a pulse width modulation. Figs. 3.1.9(b) and (c) show experimentally measured characteristics of the moving velocity of the robot with respect to the parameters  $k_a$  and  $\omega_i$ , respectively. From these figures, it can be concluded that the moving velocity of the snake-like robot can be controlled by adjusting the parameters  $k_a$  and  $\omega_i$ . For comparison, a CPG model consisting of a simple oscillator [7], the bifurcation mechanism of which is similar to the ACA oscillator, is implemented by the same compiler and the same FPGA device. The oscillator is described by a set of two ordinary differential equations (ODEs), which is implemented

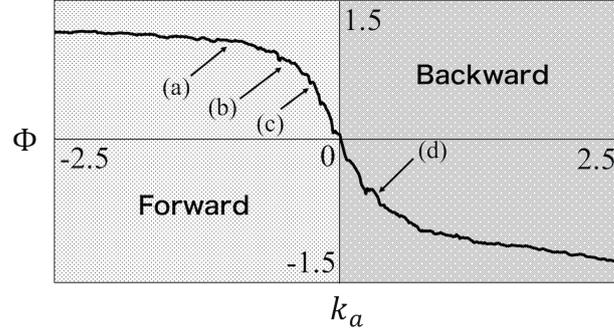


Figure 3.1.8: Characteristics of the mean phase difference  $\Phi$  with respect to the parameter  $k_a$ .  $(N, M, T_X, T_Y, \alpha_x, \alpha_y, \delta_i, \omega_i) = (2^6, 2^{10}, 0.0001, 0.00015\sqrt{1.00020134}, 10000, \frac{100000}{15\sqrt{1.00020134}}, 25^2, 2\pi \times 1.0)$  for all  $i$ .  $(n, k_b) = (6, 0.3)$ . The parameter regions “Forward” and “Backward” lead to forward and backward moving directions of the snake-like robot in Fig. 9. The arrows (a), (b), (c), and (d) correspond to Figs. 3.1.6 (a), (b), (c), and (d), respectively.

Table 3.1.2: Comparison. The symbol “#” denotes “the number of.”

Hardware resources occupied in the FPGA	Proposed CPG model	CPG model consisting of the ODE oscillator [7]
# Look-up-tables	12824	48216
# Flip-flops	1035	816
# Total slices	3633	13519

as a forward Euler formula in a VHDL code. The bit-length of the ODE oscillator is decreased as short as possible under the condition that the resulting characteristics of the moving velocity of the snake-like robot are almost identical with those in Figs. 3.1.9(b) and (c) (see the dashed graphs in these figures). Table 3.1.2 shows comparison of hardware resources used to implement the CPG models in the FPGA device. It can be seen that the proposed CPG model consumes much fewer hardware resources compared to the CPG model consisting of the ODE oscillator.

### 3.1.4 Conclusions

The novel CPG model based on the asynchronous cellular automaton was proposed. The proposed CPG model was implemented on the FPGA and was used to control the snake-like robot. By experimental validation using the prototype machine, it was shown that the proposed model can realize rhythmic locomotor activity in snakes. Moreover, it was shown that the proposed CPG model and the Hopf-oscillator CPG model (which is one of simple CPG models) realize almost identical characteristics of the moving speeds of the snake-like robot but the proposed model consumes much fewer hardware resources. Also, the parameter setting methods to adjust the moving speeds of the snake-like robot were clarified. Future problems are including (a) comparison to other models, (b) more detailed analysis on the hardware cost, and (c) applications to other types robots.

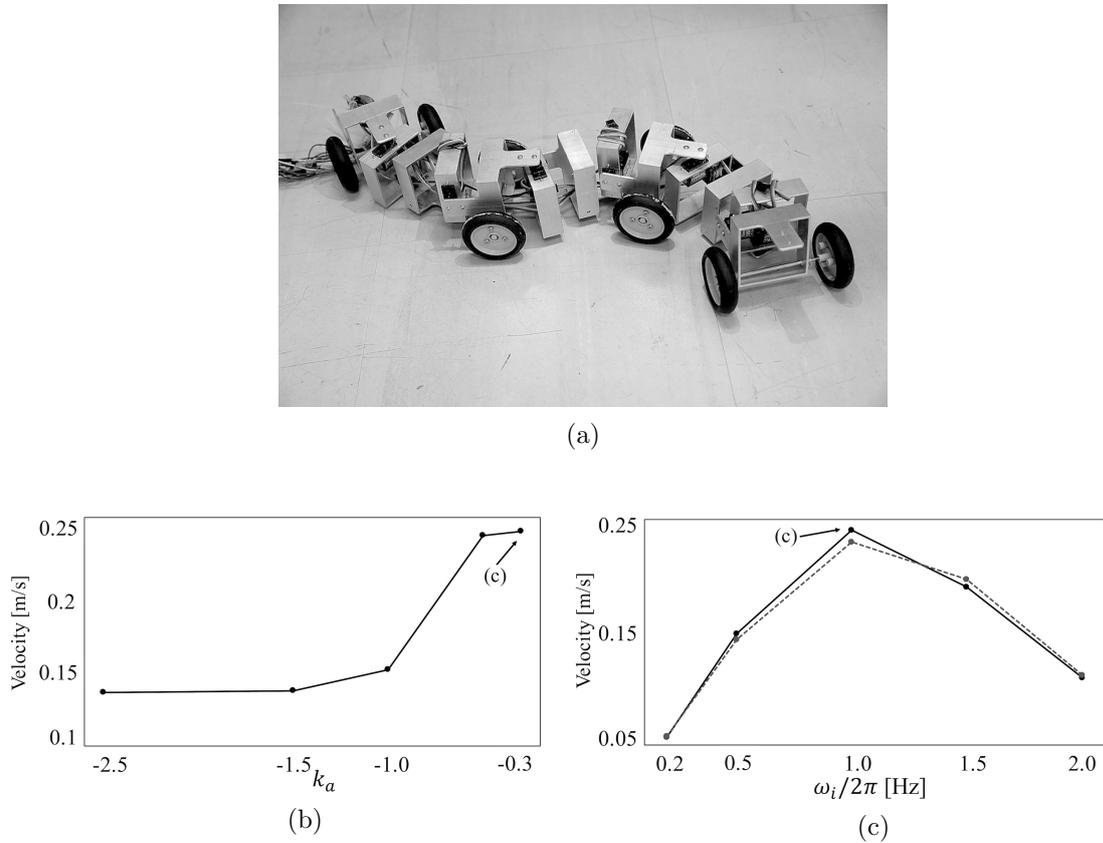


Figure 3.1.9: Snake-like robot controlled by the proposed CPG model.  $(N, M, T_X, T_Y, \alpha_x, \alpha_y, \delta_i) = (2^6, 2^{10}, 0.0001, 0.00015, 10000, 6667, 25^2)$  for all  $i$ .  $(n, k_b) = (6, 0.3)$ . (a) Picture of the implemented snake-like robot. (b) The solid graph shows the characteristics of the moving velocity of the snake-like robot with respect to the parameter  $k_a$ .  $\omega_i = 2\pi \times 1$  for all  $i$ . (c) The solid graph shows the characteristics of the moving velocity of the snake-like robot with respect to the parameter  $\omega_i$ .  $k_a = -0.3$ . In (b) and (c), the dashed graphs show the characteristics of the moving velocity of the snake-like robot controlled by a differential equation CPG model [7].

## 3.2 Hexapod Robot Controlled by Coupled Limit Cycle Oscillators<sup>2</sup>

### 3.2.1 Introduction

Various species of animals, such as ants, spiders, snakes, and fish perform locomotion using flexor and extensor muscles that are driven by multiple signals with rhythmic patterns. Studies in the field of biological motor control system show that such rhythmic patterns are produced by central pattern generators (CPGs) in the central nervous systems [12]. Inspired by this biological principle, many mathematical and electronic circuit models of CPGs have been designed to control artificial robots that are capable of performing useful tasks [3, 13].

<sup>2</sup>© 2020 IEEE. Reprinted, with permission, from Kentaro Takeda and Hiroyuki Torikai, A Novel Hardware-Efficient Central Pattern Generator Model based on Asynchronous Cellular Automaton Dynamics for Controlling Hexapod Robot, IEEE Access, vol. 8, pp. 139609–139624, Jul. 2020.

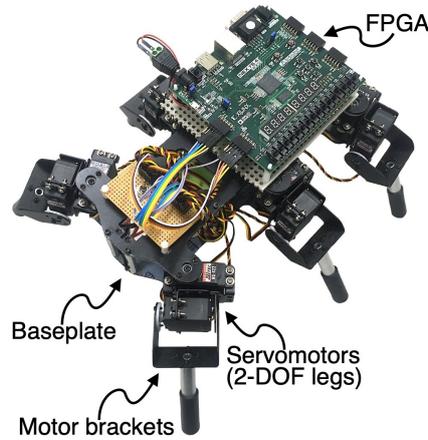


Figure 3.2.1: Hexapod robot [44] mounted with a field programmable gate array (FPGA) with the proposed central pattern generator (CPG) model.

For example, networks of Hopf and Kuramoto oscillators have been used to control hexapod robots, as shown in Fig. 3.2.1. Further, CPG models have been used in the field of medical engineering [14, 15]. For example, a network of integrate-and-fire oscillators were used to control mammalian muscles to assist in damaged nervous systems [15]. Considering the nonlinear circuit and system theory, biomimetic models, including the CPG models, are classified into the following four classes based on continuousness and discontinuousness of state variable and time.

*Class CTCS.* This is a nonlinear differential equation model of a biomimetic system with *a continuous time and continuous states* (CTCS). A class CTCS biomimetic model can be generally implemented in an analog nonlinear circuit, e.g., [15–19].

*Class DTCS.* This is a nonlinear difference equation model of a biomimetic system with *a discrete time and continuous states* (DTCS). A class DTCS biomimetic model can be generally implemented in a switched capacitor circuit, e.g., [20–23].

*Class DTDS.* This is a numerical integration model (in finite binary number representation) of a biomimetic system with *a discrete time and discrete states* (DTDS). A class DTDS biomimetic model can be generally implemented in a digital processor or a biomimetic sequential logic circuit, e.g., [7, 24–28].

*Class CTDS.* This is an asynchronous cellular automaton (CA) model of a biomimetic system with *a continuous (state transition) time and discrete states* (CTDS). A class CTDS biomimetic model can be generally implemented in an asynchronous sequential logic circuit, e.g., [9, 10, 29–43].

Most conventional biomimetic models belong to the CTCS, DTCS, and DTDS classes. The CPG models in [15–18], [20–22], and [7, 24–27] belong to the CTCS, DTCS, and DTDS classes, respectively. We and certain other research groups have been developing various kinds of class CTDS biomimetic models, e.g.,

- Asynchronous CA neuron models [10, 29–31],

- Asynchronous CA cochlea models [9, 32–34],
- Asynchronous CA gene–protein system models [35],
- Asynchronous CA neural network models [36–39],
- Asynchronous CA CPG models [40, 41].

These studies have shown that class CTDS biomimetic models have many advantages, such as the following: (i) The models can be implemented by fewer circuit elements than the numerical integration models employed in digital processors [9, 10, 29–41], and (ii) They consume lower power than numerical integration models employed in digital processors [9, 10, 29–41]. Hence, this study aims at presenting a novel asynchronous CA model of the CPG that consumes fewer circuit elements and lower power than a conventional digital processor CPG model.

Fig. 3.2.2(a) shows a conceptual diagram of the proposed CPG model, where each oscillator is implemented in a sequential logic circuit; thus, its dynamics is described by a CA. In this study, a CA oscillator is introduced in Section II. Detailed analyses of nonlinear dynamics of the CA oscillator reveal an important advantage of the asynchronous nature of the clocks: an asynchronous CA oscillator (i.e., a CA oscillator with multiple asynchronous clocks) can control its oscillation such that the oscillation is suited to control a hexapod robot, while a synchronous CA oscillator (i.e., a CA oscillator with a single clock or multiple synchronous clocks) cannot. As preparations to design the CPG, Section III presents a small network of the CA oscillators. Detailed analyses of the nonlinear dynamics of the network reveal the effects of the parameters on the synchronization phenomena of the network. Section IV presents a novel asynchronous CA CPG model, which consists of a network of CA oscillators. Depending on the asynchronous nature of the clocks *in a single CA oscillator* and *among multiple CA oscillators*, the following four types of CA oscillator networks exist (see also Fig. 3.2.2(b)).

- (a) *Asynchronous network of asynchronous CA oscillators*
- (b) *Synchronous network of asynchronous CA oscillators*
- (c) *Asynchronous network of synchronous CA oscillators*
- (d) *Synchronous network of synchronous CA oscillators*

Detailed analyses of nonlinear dynamics of the four networks reveal that the *asynchronous network of asynchronous CA oscillators* is best suited to be employed as a CPG model. Based on the aforementioned analyses, systematic design procedures of the CPG model are proposed. In Section V, the proposed CPG model is implemented in a field programmable gate array (FPGA). Our experiments indicate that the CPG model implemented in an FPGA can realize a tripod gait of the hardware robot, as shown in Fig. 3.2.1, where the tripod gait is a typical gait observed in six-legged insects [45]. For comparison, a conventional digital processor CPG model is implemented in the same FPGA. It is observed that the proposed CPG model employs fewer circuit elements and lower power than the conventional CPG model. This paper reports the following novelties and significances.

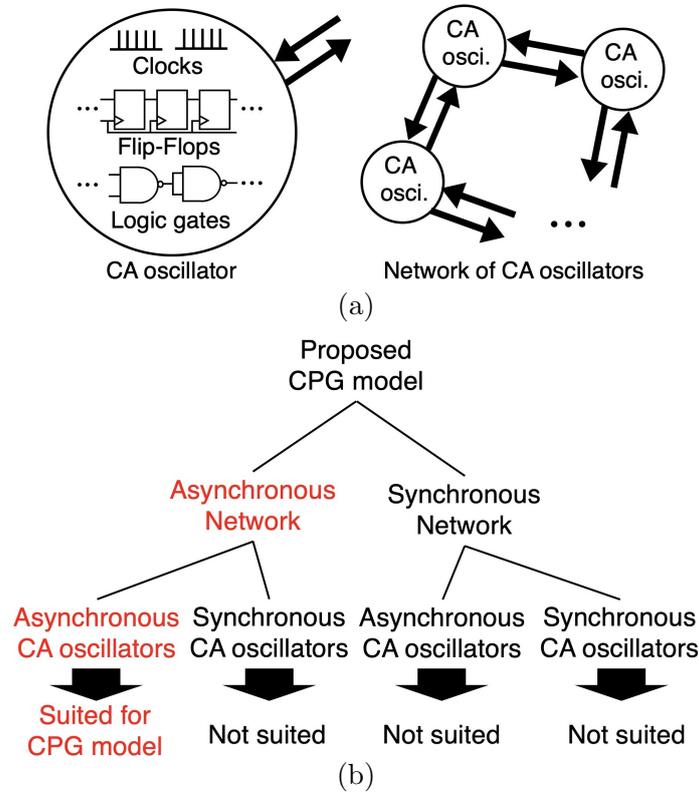


Figure 3.2.2: Cellular automaton oscillator and four types of networks.

- This study proposes a novel CPG model design procedure, which employs few circuit elements and low power. Hence, this study provides fundamental measures to develop small and low-power CPG models, whose potential applications include single-chip low-power controllers for bio-inspired multi-legged robots and implantable muscle controllers to assist in damaged nervous systems.
- This study analyzes the effects of the asynchronous nature of clocks in the CA oscillators in a network. To the best of our knowledge, this analysis has been performed for the first time. Hence, this study can contribute to develop *new nonlinear circuit theories – oscillation and synchronization theories of networks of asynchronous sequential logic circuits*.
- Although preliminary results of this study were presented in an IEEE flagship conference [40], to the best of our knowledge, this paper reports detailed analyses of a CA oscillator and its networks for the first time. These analyses reveal that the network in this study is better suited to be employed as a CPG than the previously reported network.

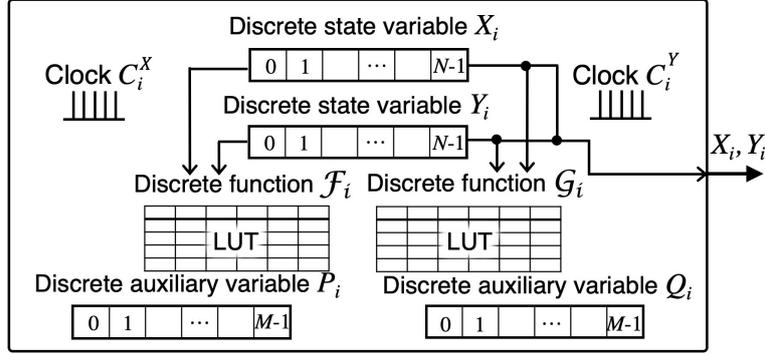


Figure 3.2.3: Schematic diagram of the asynchronous cellular automaton (CA) oscillator. LUT represents look-up table.

## 3.2.2 Asynchronous CA oscillator

### 3.2.2.1 MODEL DESCRIPTION

This subsection describes an asynchronous CA oscillator, which is used as an element of a CPG model, as described in a further section. Fig. 3.2.3 shows a schematic diagram of the CA oscillator. The CA oscillator has two clocks

$$C_i^X(t) = \sum_{k=0}^{\infty} \delta(t - kT_i^X), \quad C_i^Y(t) = \sum_{k=0}^{\infty} \delta(t - kT_i^Y),$$

where  $i$  is an oscillator index,  $t \in \mathbb{R}$  is a continuous time,  $T_i^X \in (0, \infty)$  and  $T_i^Y \in (0, \infty)$  are clock periods, and  $\delta : \mathbb{R} \rightarrow \{0, 1\}$  is the unit impulse function

$$\delta(t) = \begin{cases} 1 & \text{if } t = 0, \\ 0 & \text{if } t \neq 0. \end{cases}$$

Further, as shown in Fig. 3.2.3, the CA oscillator has the following two discrete state variables  $\{X_i, Y_i\}$  and two discrete auxiliary variables  $\{P_i, Q_i\}$ .

$$X_i \in \mathbf{Z}_N \equiv \{0, \dots, N-1\}, \quad Y_i \in \mathbf{Z}_N, \quad (3.2.1)$$

$$P_i \in \mathbf{Z}_M \equiv \{0, \dots, M-1\}, \quad Q_i \in \mathbf{Z}_M, \quad (3.2.2)$$

where  $N$  and  $M$  are positive integers characterizing resolutions of the discrete state variables  $X_i, Y_i, P_i$ , and  $Q_i$ . The two clocks  $C_i^X$  and  $C_i^Y$  trigger transitions of the discrete state variables  $X_i$  and  $Y_i$  as follows (see Fig. 3.2.4(a)).

$$\text{If } C_i^X(t) = 1 \text{ and } P_i(t) \geq |\mathcal{F}_i(X_i(t), Y_i(t))|, \text{ then} \\ X_i(t_+) := \begin{cases} X_i(t) + 1 & \text{if } \mathcal{F}_i(X_i(t), Y_i(t)) \geq 0 \text{ and } X_i(t) < N-1, \\ X_i(t) - 1 & \text{if } \mathcal{F}_i(X_i(t), Y_i(t)) < 0 \text{ and } X_i(t) > 0. \end{cases} \quad (3.2.3)$$

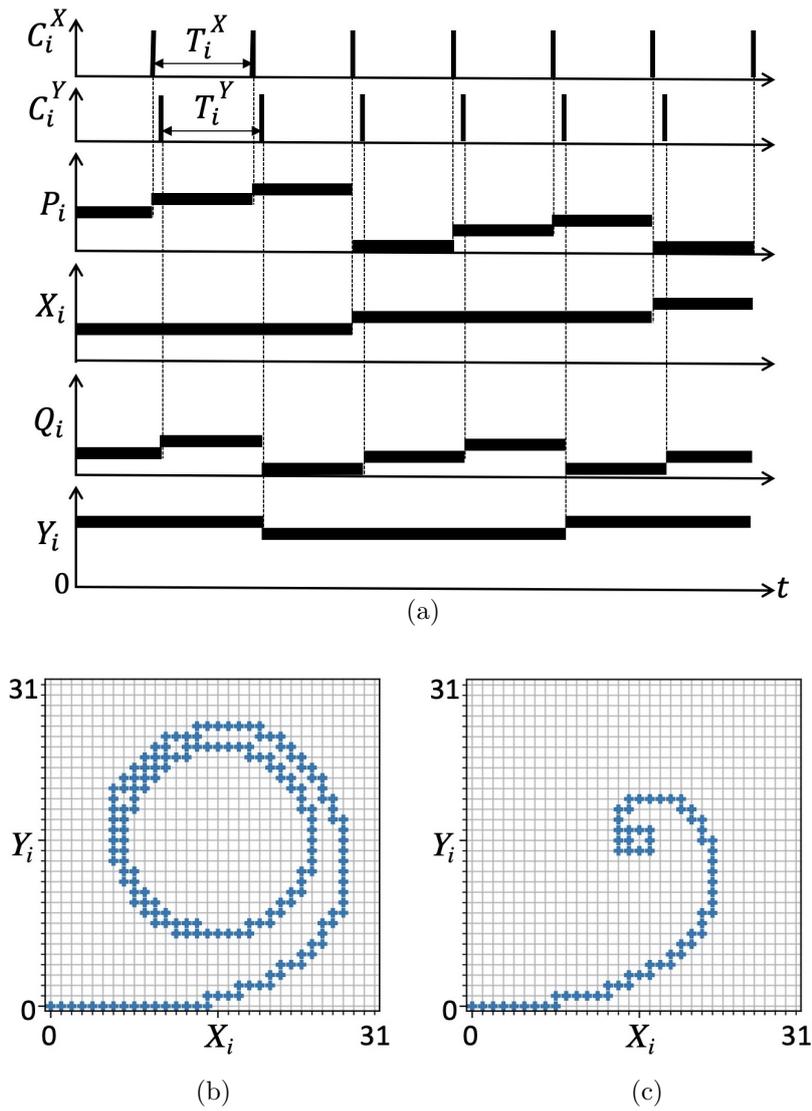


Figure 3.2.4: Nonlinear dynamics of the CA oscillator.  $N = 2^5$ ,  $M = 2^{10}$ ,  $\omega_i = 2\pi/\alpha_i$ ,  $\alpha_i = 0.01$ ,  $T_i^X = 0.001$ , and  $T_i^Y = 0.001014142$ . (a) Timing chart of state transitions. (b) Phase plane trajectory starting from  $X_i = Y_i = P_i = Q_i = 0$ .  $\rho_i = 144$ . (c) Phase plane trajectory starting from  $X_i = Y_i = P_i = Q_i = 0$ .  $\rho_i = -100$ .

If  $C_i^Y(t) = 1$  and  $Q_i(t) \geq |\mathcal{G}_i(X_i(t), Y_i(t))|$ , then

$$Y_i(t_+) := \begin{cases} Y_i(t) + 1 & \text{if } \mathcal{G}_i(X_i(t), Y_i(t)) \geq 0 \text{ and } Y_i(t) < N-1, \\ Y_i(t) - 1 & \text{if } \mathcal{G}_i(X_i(t), Y_i(t)) < 0 \text{ and } Y_i(t) > 0, \end{cases} \quad (3.2.4)$$

where “ $t_+$ ” denotes “ $\lim_{\epsilon \rightarrow 0^+} t + \epsilon$ ” and “ $:=$ ” denotes an “instantaneous state transition.” Moreover,  $\mathcal{F}_i : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbf{Z}_M^\pm \equiv \{-(M-1), \dots, (M-1)\}$  and  $\mathcal{G}_i : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbf{Z}_M^\pm$  denote discrete functions

$$\mathcal{F}_i \equiv F_i \circ f_i, \quad \mathcal{G}_i \equiv G_i \circ g_i, \quad (3.2.5)$$

where  $\mathcal{F}_i$  and  $\mathcal{G}_i$  are implemented in look-up-tables, as shown in Fig. 3.2.3. In this study, we propose to design the functions  $F_i : \mathbb{R} \rightarrow \mathbf{Z}_M^\pm$  and  $G_i : \mathbb{R} \rightarrow \mathbf{Z}_M^\pm$  as follows.

$$F_i(x) = \left\lfloor \frac{1}{\alpha_i T_i^X x} \right\rfloor, \quad G_i(y) = \left\lfloor \frac{1}{\alpha_i T_i^Y y} \right\rfloor,$$

where the functions  $F_i$  and  $G_i$  are assumed to be saturated at  $\pm(M-1)$ ,  $\alpha_i \in (0, \infty)$  is a scaling parameter, and  $\lfloor \cdot \rfloor$  denotes the floor function

$$\lfloor z \rfloor = \max\{n \in \mathbb{Z} \mid n \leq z\}.$$

Further, we propose to design the functions  $f_i : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbb{R}$  and  $g_i : \mathbf{Z}_N \times \mathbf{Z}_N \rightarrow \mathbb{R}$  as follows.

$$\begin{aligned} f_i(x, y) &= \rho_i \left( x - \left\lfloor \frac{N}{2} \right\rfloor \right) - \omega_i \left( y - \left\lfloor \frac{N}{2} \right\rfloor \right) \\ &\quad - \left( x - \left\lfloor \frac{N}{2} \right\rfloor \right) \left( \left( x - \left\lfloor \frac{N}{2} \right\rfloor \right)^2 + \left( y - \left\lfloor \frac{N}{2} \right\rfloor \right)^2 \right), \\ g_i(x, y) &= \omega_i \left( x - \left\lfloor \frac{N}{2} \right\rfloor \right) + \rho_i \left( y - \left\lfloor \frac{N}{2} \right\rfloor \right) \\ &\quad - \left( y - \left\lfloor \frac{N}{2} \right\rfloor \right) \left( \left( x - \left\lfloor \frac{N}{2} \right\rfloor \right)^2 + \left( y - \left\lfloor \frac{N}{2} \right\rfloor \right)^2 \right), \end{aligned}$$

where  $\rho_i \in \mathbb{R}$  and  $\omega_i \in \mathbb{R}$  are parameters. Then, the two clocks  $C_i^X$  and  $C_i^Y$  trigger transitions of the discrete auxiliary variables  $P_i$  and  $Q_i$  as follows (see Fig. 3.2.4(a)).

If  $C_i^X(t) = 1$ , then

$$P_i(t_+) := \begin{cases} P_i(t) + 1 & \text{if } P_i(t) < |\mathcal{F}_i(X_i(t), Y_i(t))|, \\ 0 & \text{if } P_i(t) \geq |\mathcal{F}_i(X_i(t), Y_i(t))|, \end{cases} \quad (3.2.6)$$

If  $C_i^Y(t) = 1$ , then

$$Q_i(t_+) := \begin{cases} Q_i(t) + 1 & \text{if } Q_i(t) < |\mathcal{G}_i(X_i(t), Y_i(t))|, \\ 0 & \text{if } Q_i(t) \geq |\mathcal{G}_i(X_i(t), Y_i(t))|. \end{cases} \quad (3.2.7)$$

As a result, the dynamics of the CA oscillator are described by Eqs. (3.2.3), (3.2.4), (3.2.6), and (3.2.7), and characterized by the parameters

$$N, M, \rho_i, \omega_i, \alpha_i, T_i^X, T_i^Y. \quad (3.2.8)$$

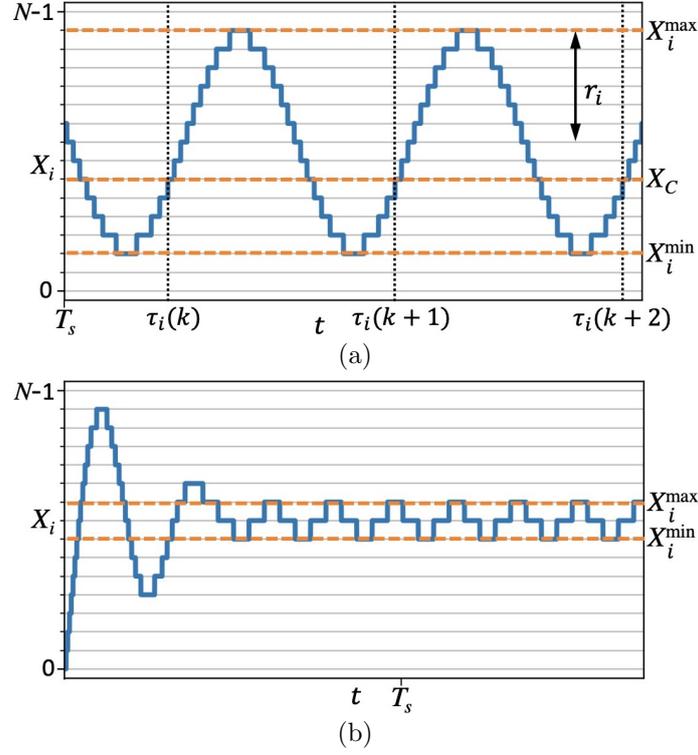


Figure 3.2.5: Definitions of the amplitude  $r_i$  and mean frequency  $f_i$  of the discrete state variable  $X_i$ . (a) Oscillation. The amplitude  $r_i$  is larger than the reference value  $\lambda = 1$ . (b) Convergence. The amplitude  $r_i$  is smaller than the reference value  $\lambda = 1$ .

Figs. 3.2.4 (b) and (c) show phase plane trajectories of the CA oscillator. The CA oscillator exhibits different behaviors, such as oscillation and convergence for different parameter values. The next subsection provides characterizations for such behaviors of the CA oscillator and describes the roles of the parameters.

### 3.2.2.2 ROLES OF PARAMETERS

We assume that the discrete state variable  $X_i$  is in a steady state for  $t > T_s$ . Then, the following is defined.

**Definition 1 (Amplitude):** The discrete state variable  $X_i$  is said to have an amplitude

$$r_i = \frac{X_i^{\max} - X_i^{\min}}{2},$$

where

$$X_i^{\max} = \max\{X_i(t) \in \mathbf{Z}_N \mid t > T_s\},$$

$$X_i^{\min} = \min\{X_i(t) \in \mathbf{Z}_N \mid t > T_s\}$$

are the maximum and minimum values of the discrete state variable  $X_i$  in the steady state, respectively.

For example, in Figs. 3.2.5(a) and (b), the amplitude  $r_i$  of the discrete state variable

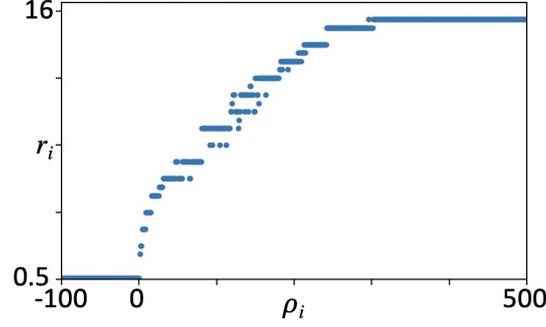


Figure 3.2.6: Characteristics of the amplitude  $r_i$  of the discrete state variable  $X_i$  for the parameter  $\rho_i$ .  $N = 2^5$ ,  $M = 2^6$ ,  $\omega_i = 2\pi/\alpha_i$ ,  $\alpha_i = 0.01$ ,  $T_i^X = 0.0012236$ , and  $T_i^Y = 0.001$ .

Table 3.2.1: Roles of parameters

Param.	Role
$\rho_i$	Adjustment of amplitude $r_i$ of oscillation
$\omega_i$	Adjustment of mean frequency $f_i$ of oscillation
$\alpha_i$	Scaling of mean frequency $f_i$ of oscillation
$T_i^X, T_i^Y$	Adjustment of coexistence of multiple attractors
$w_{i,j}$	Adjustment of $n:m$ phase-locking
	Adjustment of mean phase difference $\Phi_{i,j}$

$X_i$  is 6 and 1, respectively. Fig. 3.2.6 shows the characteristics of the amplitude  $r_i$  for the parameter  $\rho_i$ . Further, it shows the following relations between the parameter  $\rho_i$  and amplitude  $r_i$ .

- Fig. 3.2.6 shows that the amplitude  $r_i$  is approximately 0 for negative  $\rho_i$  and positive for positive  $\rho_i$ .
- Fig. 3.2.6 shows that the amplitude  $r_i$  is approximately proportional to  $\sqrt{\rho_i}$  for relatively small positive  $\rho_i$  and saturates for large positive  $\rho_i$ .

The aforementioned relations reveal that the parameter  $\rho_i$  can adjust the amplitude  $r_i$  (see Table 3.2.1). Using the amplitude  $r_i$ , the following is defined.

**Definition 2 (Oscillation and Convergent):** *The discrete state variable  $X_i$  is said to converge if  $r_i \leq \lambda$  and is said to oscillate if  $r_i > \lambda$ , where  $\lambda$  is an appropriately small positive constant, which is introduced to indicate that “the amplitude  $r_i$  is regarded to be almost 0 if  $r_i \leq \lambda$ .”*

In this study,  $\lambda = 1$  is chosen. In Fig. 3.2.5(a), the amplitude  $r_i$  is  $5 > \lambda$ ; thus, the discrete state variable  $X_i$  oscillates. In Fig. 3.2.5(b), the amplitude  $r_i$  is  $1 \leq \lambda$ ; thus, the discrete state variable  $X_i$  converges. Then, the following is defined.

**Definition 3 (Mean frequency):** *Assume the discrete state variable  $X_i$  oscillates and repeats to transit from a certain constant value  $X_C - 1$  to  $X_C$ . Let  $\tau_i(k)$  be the  $k$ -th moment*

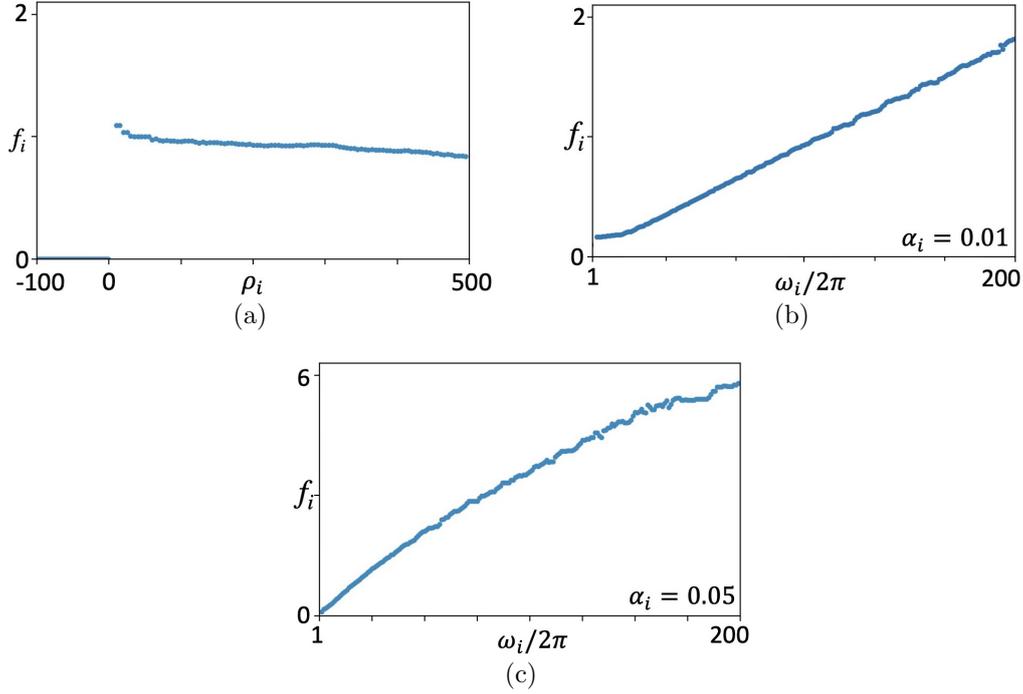


Figure 3.2.7: (a) Characteristics of the mean frequency  $f_i$  for the parameter  $\rho_i$ .  $N = 2^5$ ,  $M = 2^6$ ,  $\omega_i = 2\pi/\alpha_i$ ,  $\alpha_i = 0.01$ ,  $T_i^X = 0.0012236$ , and  $T_i^Y = 0.001$ . (b) Characteristics of the mean frequency  $f_i$  for the parameter  $\omega_i$ .  $N = 2^5$ ,  $M = 2^6$ ,  $\rho_i = 225$ ,  $\alpha_i = 0.01$ ,  $T_i^X = 0.0012236$ , and  $T_i^Y = 0.001$ . (c) Characteristics of the mean frequency  $f_i$  for the parameter  $\omega_i$ .  $\alpha_i = 0.05$  and the remaining parameter values are equal to those in (b).

when the discrete state variable  $X_i$  transits from the constant value  $X_C - 1$  to  $X_C$ . Then, the discrete state variable  $X_i$  is said to have a mean frequency  $f_i$  defined by

$$f_i = \frac{K - 1}{\sum_{k=1}^{K-1} \tau_i(k+1) - \tau_i(k)},$$

where  $K$  is an appropriately large positive integer. If the discrete state variable  $X_i$  converges, the mean frequency is defined as  $f_i = 0$ .

For example, in Figs. 3.2.5(a) and (b), the mean frequency  $f_i$  of the discrete state variable  $X_i$  is positive and 0, respectively. Fig. 3.2.7 shows the characteristics of the mean frequency  $f_i$  for the parameters  $\rho_i$ ,  $\omega_i$ , and  $\alpha_i$ . These figures show the following roles of the parameters.

- Fig. 3.2.7(a) shows that the mean frequency  $f_i$  is almost constant with respect to the parameter  $\rho_i > 0$ .
- Figs. 3.2.7(b) and (c) show that the mean frequency  $f_i$  is almost proportional to the parameter  $\omega_i$ .
- Figs. 3.2.7(b) and (c) show that the parameter  $\alpha_i$  changes the scale of the mean frequency  $f_i$ .

The aforementioned relations show that the parameter  $\omega_i$  can adjust the frequency  $f_i$ , while the parameter  $\rho_i > 0$  does not affect the frequency  $r_i$  significantly (see Table 3.2.1). Further, the parameter  $\alpha_i$  can scale the frequency  $f_i$  (see Table 3.2.1).

### 3.2.2.3 IMPORTANCE OF ASYNCHRONOUS TRANSITIONS

This subsection describes the importance of the asynchronous transitions of the discrete state variables  $X_i, Y_i, P_i$ , and  $Q_i$ .

**Definition 4 (Asynchronous and synchronous CA oscillators):** *The CA oscillator is said to be*

- *asynchronous CA oscillator if  $T_i^X/T_i^Y$  is irrational.*
- *synchronous CA oscillator if  $T_i^X/T_i^Y$  is rational<sup>3</sup>.*

Then, we compare the characteristics of the asynchronous and synchronous CA oscillators.

- (a) Fig. 3.2.8 shows the characteristics of the asynchronous CA oscillator. Fig. 3.2.8(a) shows an orbit of the discrete state vector  $(X_i, Y_i)$  in a steady state. Here, all the trajectories of the discrete state vector  $(X_i, Y_i)$  starting from different initial conditions are attracted into the same orbit in Fig. 3.2.8(a). For certain different parameter values, a small number of orbits coexist in steady states and the asynchronous CA oscillator exhibits one of them depending on the initial condition of the discrete state vector  $(X_i, Y_i)$ . Fig. 3.2.8(b) shows the characteristics of the number of such coexisting orbits in steady states. The asynchronous CA oscillator has a single orbit in a steady state for a wide range of parameter values. Fig. 3.2.8(c) shows the characteristics of the maximum values  $X_i^{\max}$  and minimum values  $X_i^{\min}$  of the coexisting orbits.
- (b) Fig. 3.2.9 shows the characteristics of the synchronous CA oscillator. Fig. 3.2.9(a) shows that the synchronous CA oscillator has many coexisting orbits and the oscillator exhibits one of orbits depending on the initial condition of the discrete state vector  $(X_i, Y_i)$ . Fig. 3.2.9(b) shows the characteristics of the number of such coexisting orbits, and Fig. 3.2.9(c) shows the characteristics of the maximum values  $X_i^{\max}$  and minimum values  $X_i^{\min}$  of the coexisting orbits. These figures show that the synchronous CA oscillator has multiple orbits in steady states for a wide range of parameter values.

The aforementioned characteristics show that the periods  $T_i^X$  and  $T_i^Y$  of the clocks  $C_i^X$ , and  $C_i^Y$  determine the characteristics of the coexisting orbits in steady states (see Table 3.2.1). Further, the aforementioned characteristics show the following significance of the analyses.

**Remark 1 (significance obtained from the analyses of single oscillator):** The CA oscillator is used as an element of a CPG model in Section IV, where the amplitude  $r_i$  and mean frequency  $f_i$  of the discrete state variable  $X_i$  control an amplitude and a frequency of motion of a robot leg, respectively. In this context, the asynchronous CA oscillator is more suited to build the CPG model because the leg motion is expected to be controlled by the

<sup>3</sup>The synchronous CA oscillator for  $T_i^X = T_i^Y$  (or the CA oscillator with a single clock with period  $T_i^X$ ) has simultaneous transitions of the discrete state variables, and the synchronous CA oscillator for  $T_i^X \neq T_i^Y$  has phase-locked transitions of the discrete state variables.

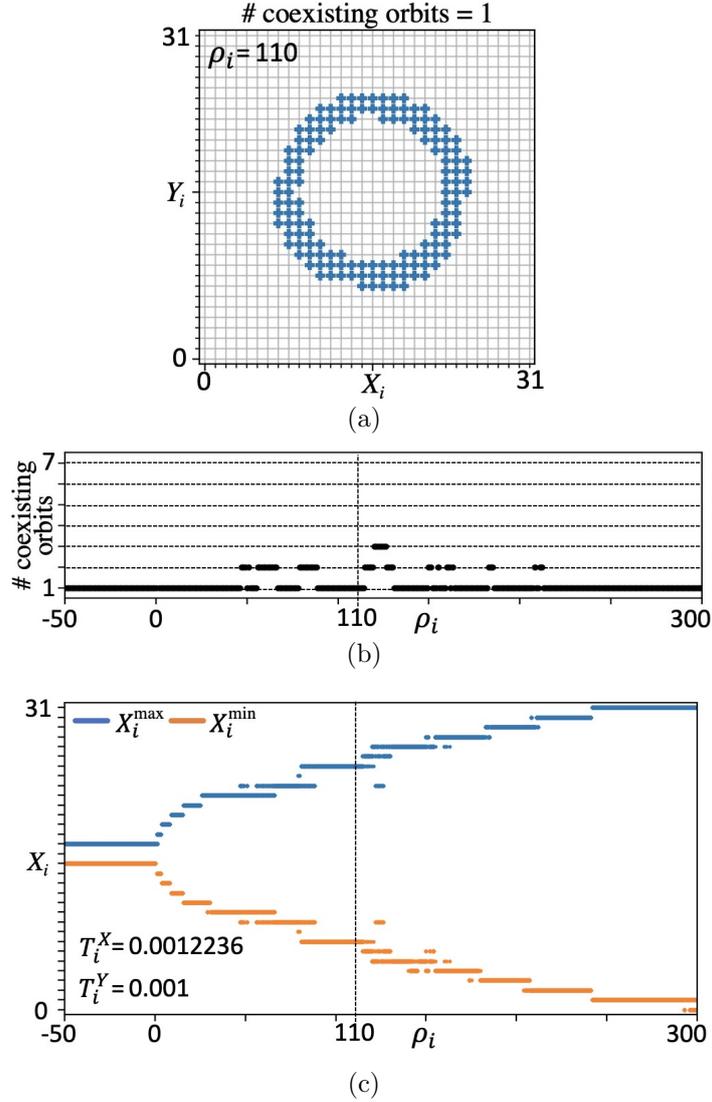


Figure 3.2.8: Characteristics of the asynchronous CA oscillator.  $N = 2^5$ ,  $M = 2^6$ ,  $\omega_i = 2\pi/\alpha_i$ , and  $\alpha_i = 0.01$ .  $T_i^X = 0.0012236$  and  $T_i^Y = 0.001$ , where  $T_i^X/T_i^Y$  can be regarded as almost irrational. (a) Phase plane trajectories starting from the different initial conditions  $X_i(0) \in \mathbf{Z}_N$ ,  $Y_i(0) = \lfloor N/2 \rfloor$ , and  $P_i(0) = Q_i(0) = 0$ .  $\rho_i = 110$ . (b) Characteristics of the number of coexisting orbits. The vertical broken line corresponds to (a). (c) Characteristics of the maximum values  $X_i^{\max}$  and minimum values  $X_i^{\min}$  of the coexisting orbits. The vertical broken line corresponds to (a).

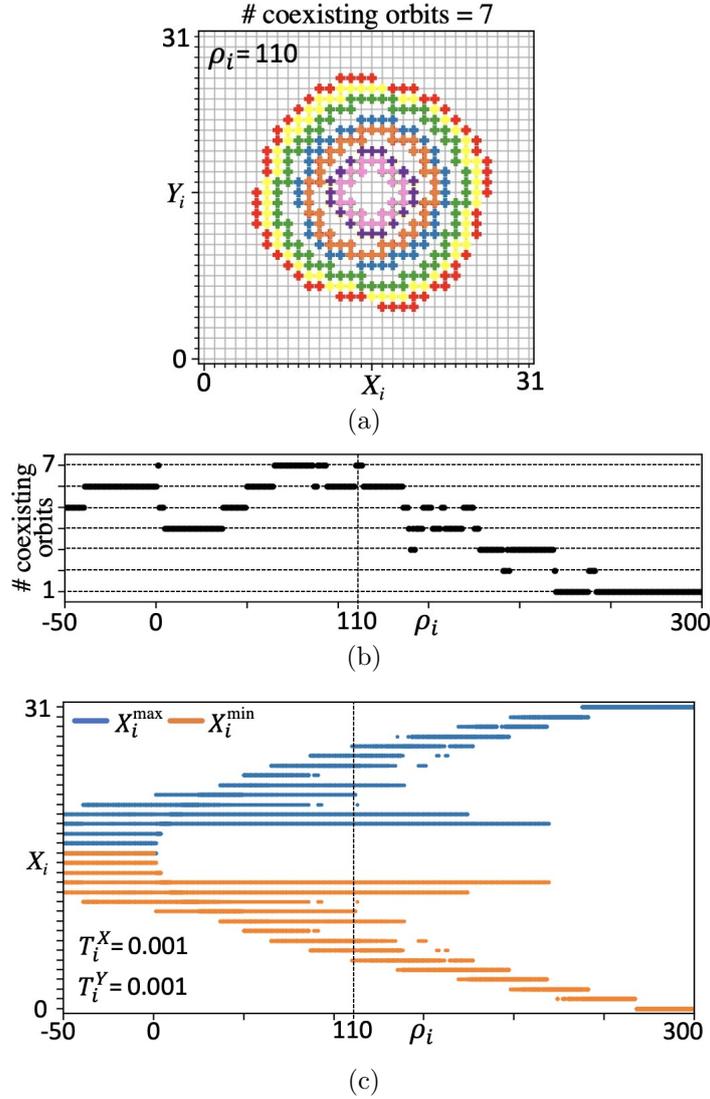


Figure 3.2.9: Characteristics of the synchronous CA oscillator.  $N = 2^5$ ,  $M = 2^6$ ,  $\omega_i = 2\pi/\alpha_i$ , and  $\alpha_i = 0.01$ .  $T_i^X = 0.001$  and  $T_i^Y = 0.001$ , where  $T_i^X/T_i^Y$  is rational. (a) Phase plane trajectories starting from the different initial conditions  $X_i(0) \in \mathbf{Z}_N$ ,  $Y_i(0) = \lfloor N/2 \rfloor$ , and  $P_i(0) = Q_i(0) = 0$ .  $\rho_i = 110$ . (b) Characteristics of the number of coexisting orbits. The vertical broken line corresponds to (a). (c) Characteristics of the maximum values  $X_i^{\max}$  and minimum values  $X_i^{\min}$  of the coexisting orbits. The vertical broken line corresponds to (a).

system parameters as a consequence of the analyses in Subsection II-B and characteristics (a) in Subsection II-C. If the synchronous CA oscillator is used to build the CPG model, the leg motion is not expected to be controlled by the system parameters as a consequence of the characteristics (b) in Subsection II-C. Using these analyses, a CPG model design is proposed in Section IV.

### 3.2.3 Analyses of small network of CA oscillators as preparations to design CPG

In this section, a small network of the CA oscillators is designed and analyzed to design the CPG model for controlling the hexapod robot in Fig. 3.2.1. The following is a procedure to design a modified CA oscillator to build the network.

#### 3.2.3.1 MODIFIED CA OSCILLATOR FOR COUPLING

Fig. 3.2.10(a) shows a schematic diagram of the modified CA oscillator, where the dashed boxes represent modifications of the CA oscillator in Fig. 3.2.3. The modified CA oscillator has a clock

$$C_i^Z(t) = \sum_{k=0}^{\infty} \delta(t - kT_i^Z)$$

for coupling, where  $T_i^Z \in (0, \infty)$  is a clock period. Further, the modified CA oscillator has discrete state variables

$$V_i \in \mathbf{Z}_M, \quad U_i \in \mathbf{Z}_M \quad (3.2.9)$$

for coupling. The clock  $C_i^Z$  triggers transitions of the discrete state variables  $V_i$  and  $U_i$  as follows (see Fig. 3.2.10(b)).

$$\begin{aligned} &\text{If } C_i^Z(t) = 1, \text{ then} \\ V_i(t_+) &:= \begin{cases} V_i(t) + 1 & \text{if } V_i(t) < |\mathcal{H}_i(\mathbf{X}(t))|, \\ 0 & \text{if } V_i(t) \geq |\mathcal{H}_i(\mathbf{X}(t))|, \end{cases} \\ U_i(t_+) &:= \begin{cases} U_i(t) + 1 & \text{if } U_i(t) < |\mathcal{H}_i(\mathbf{Y}(t))|, \\ 0 & \text{if } U_i(t) \geq |\mathcal{H}_i(\mathbf{Y}(t))|, \end{cases} \end{aligned} \quad (3.2.10)$$

where  $\mathbf{X}(t)$  and  $\mathbf{Y}(t)$  denote vector forms

$$\begin{aligned} \mathbf{X}(t) &= (X_0(t), X_1(t), \dots, X_{L-1}(t)), \\ \mathbf{Y}(t) &= (Y_0(t), Y_1(t), \dots, Y_{L-1}(t)), \end{aligned}$$

of the discrete state variables,  $L$  is the number of CA oscillators in the network, and  $\mathcal{H}_i : \mathbf{Z}_N^L \rightarrow \mathbf{Z}_M^\pm$  denotes a discrete function

$$\mathcal{H}_i \equiv H_i \circ h_i, \quad (3.2.11)$$

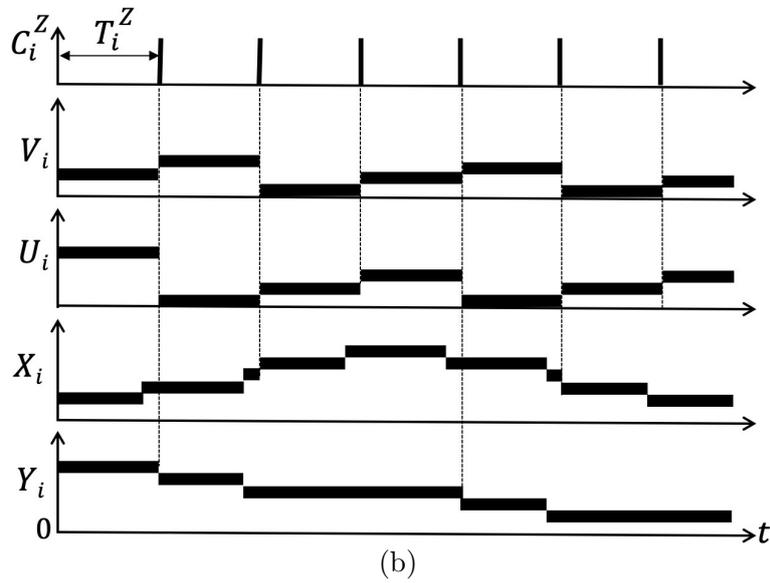
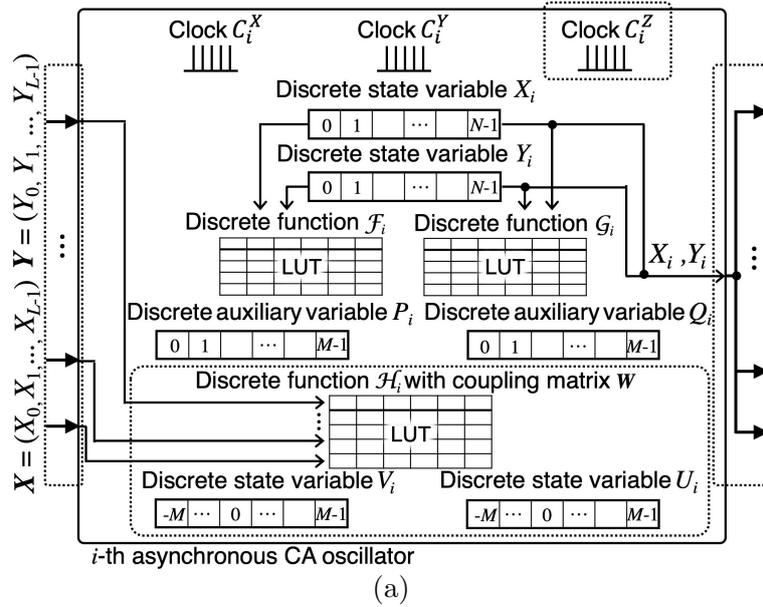


Figure 3.2.10: (a) Schematic diagram of the modified CA oscillator for coupling. (b) Timing chart of state transitions triggered by the clock  $C_i^Z$ .

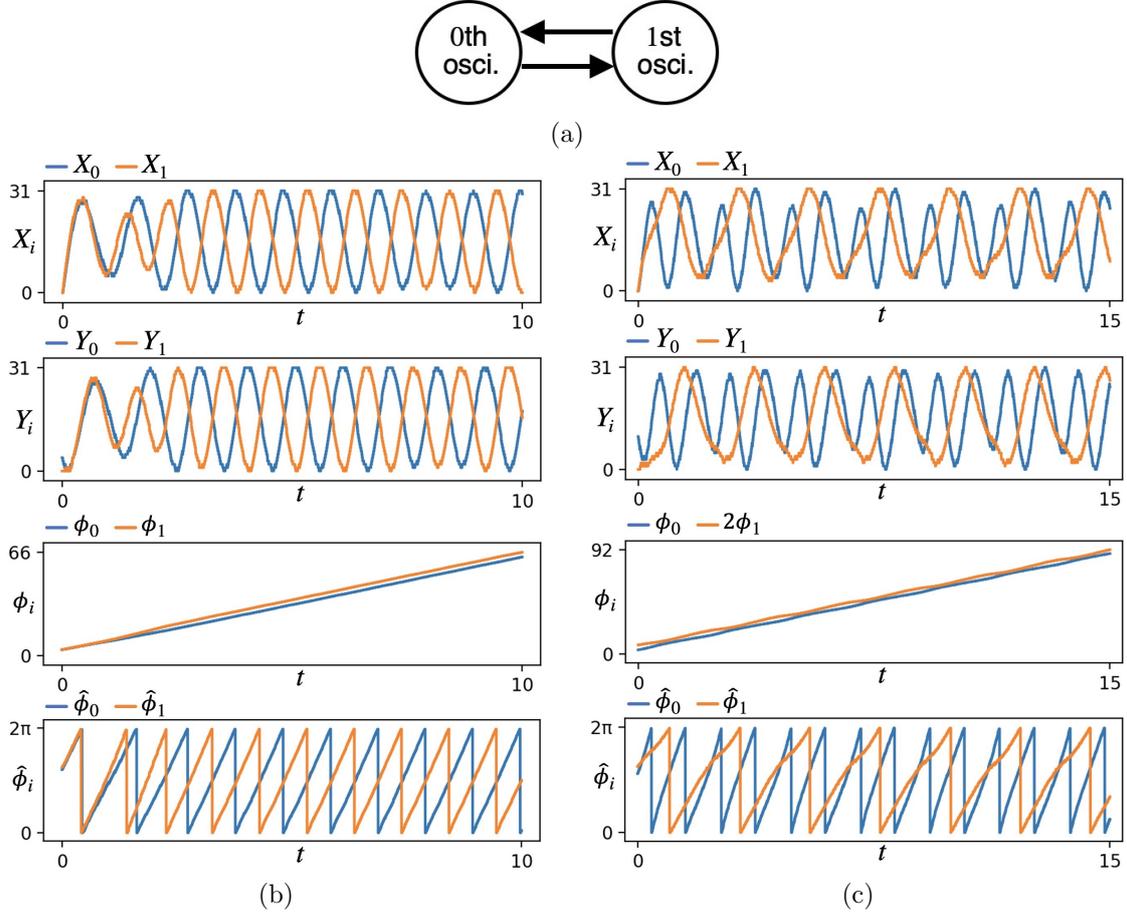


Figure 3.2.11: (a) Network of  $L = 2$  CA oscillators.  $N = 2^5$ ,  $M = 2^6$ ,  $\rho_0 = \rho_1 = 200$ ,  $\alpha_0 = \alpha_1 = 0.01$ ,  $T_0^X = T_1^Y = 0.0012236$ ,  $T_0^Y = T_1^X = 0.001$ ,  $\beta_0 = \beta_1 = 1$ , and  $T_0^Z = T_1^Z = 0.01$ . (b) 1:1 phase-locking.  $\omega_0 = \omega_1 = 2\pi/\alpha$  and  $w_{0,1} = w_{1,0} = -1$ . The mean phase difference is  $\Phi_{0,1} = -3.14$ . (c) 1:2 phase-locking.  $\omega_0 = 2\pi/\alpha$ ,  $\omega_1 = 0.4 \times \omega_0$ ,  $w_{1,2} = w_{2,1} = -1.1$ . The mean phase difference is  $\Phi_{0,1} = -2.87$ .

which is implemented in a look-up-table, as shown in Fig. 3.2.10(a). In this study, we propose to design the function  $H_i(\chi) : \mathbb{R} \rightarrow \mathbf{Z}_M^\pm$  as follows.

$$H_i(\chi) = \left\lfloor \frac{1}{\beta_i T_i^Z \chi} \right\rfloor,$$

where  $\beta_i \in (0, \infty)$  is a scaling parameter and the function  $H_i(\chi)$  is assumed to be saturated at  $\pm(M - 1)$ . We further propose to design the function  $h_i(\boldsymbol{\chi}) : \mathbf{Z}_N^L \rightarrow \mathbb{R}$  as follows.

$$h_i(\boldsymbol{\chi}) = \left\lfloor \sum_{j=0}^{L-1} w_{i,j} \left( \chi_j - \left\lfloor \frac{N}{2} \right\rfloor \right) \right\rfloor,$$

where  $\boldsymbol{\chi} = (\chi_0, \chi_1, \dots, \chi_{L-1})$  and the function  $h_i(\boldsymbol{\chi})$  is assumed to be saturated at  $\pm(M-1)$ . In addition,  $w_{i,j} \in \mathbb{R}$  represents a coupling strength from the  $j$ -th CA oscillator to the  $i$ -th CA oscillator and forms a matrix form

$$\mathbf{W} = \begin{pmatrix} w_{0,0} & \cdots & w_{0,L-1} \\ \vdots & \ddots & \vdots \\ w_{L-1,0} & \cdots & w_{L-1,L-1} \end{pmatrix}.$$

Then, the clock  $C_i^Z$  triggers transitions of the discrete state variables  $X_i$  and  $Y_i$  as follows (see also Fig. 3.2.10(b)).

$$\begin{aligned} &\text{If } C_i^Z(t) = 1 \text{ and } V_i(t) \geq |\mathcal{H}_i(\mathbf{X}(t))|, \text{ then} \\ X_i(t_+) &:= \begin{cases} X_i(t) + 1 & \text{if } \mathcal{H}_i(\mathbf{X}(t)) \geq 0 \text{ and } X_i(t) < N-1, \\ X_i(t) - 1 & \text{if } \mathcal{H}_i(\mathbf{X}(t)) < 0 \text{ and } X_i(t) > 0. \end{cases} \end{aligned} \quad (3.2.12)$$

$$\begin{aligned} &\text{If } C_i^Z(t) = 1 \text{ and } U_i(t) \geq |\mathcal{H}_i(\mathbf{Y}(t))|, \text{ then} \\ Y_i(t_+) &:= \begin{cases} Y_i(t) + 1 & \text{if } \mathcal{H}_i(\mathbf{Y}(t)) \geq 0 \text{ and } Y_i(t) < N-1, \\ Y_i(t) - 1 & \text{if } \mathcal{H}_i(\mathbf{Y}(t)) < 0 \text{ and } Y_i(t) > 0. \end{cases} \end{aligned} \quad (3.2.13)$$

Therefore, the dynamics of the modified CA oscillator are described by Eqs. (3.2.10), (3.2.12), and (3.2.13) in addition to Eqs. (3.2.3), (3.2.4), (3.2.6), and (3.2.7) and have the parameters

$$\beta_i, T_i^Z, w_{i,0}, \dots, w_{i,L} \quad (3.2.14)$$

in addition to the parameters in Eq. (3.2.8).

### 3.2.3.2 ANALYSES OF SMALL NETWORK AS PREPARATIONS TO DESIGN CPG

We analyze a network of two CA oscillators in Fig. 3.2.11(a), which has a matrix form

$$\mathbf{W} = \begin{pmatrix} 0 & w_{0,1} \\ w_{1,0} & 0 \end{pmatrix}$$

of the coupling strength  $w_{i,j}$ . Figs. 3.2.11(b) and (c) show the time waveforms of the discrete state variables  $(X_i, Y_i)$  of the network for different parameter values. To characterize such waveforms, we introduce a phase  $\phi_i(t)$  and restricted phase  $\hat{\phi}_i(t)$  of the state vector  $(X_i(t), Y_i(t))$  as follows.

$$\begin{aligned} \phi_i(t) &= \text{Arg}\left(\left(X_i(t) - \left\lfloor \frac{N}{2} \right\rfloor\right) + j\left(Y_i(t) - \left\lfloor \frac{N}{2} \right\rfloor\right)\right), \\ \hat{\phi}_i(t) &= \phi_i(t) + 2\pi\Omega, \end{aligned}$$

where

$$\text{Arg}(x + jy) = \begin{cases} \arctan(y/x) + \pi & \text{if } x < 0, \\ \arctan(y/x) & \text{if } x > 0 \text{ and } y \geq 0, \\ \arctan(y/x) + 2\pi & \text{if } x > 0 \text{ and } y < 0, \\ \pi/2 & \text{if } x = 0 \text{ and } y > 0, \\ 3\pi/2 & \text{if } x = 0 \text{ and } y < 0, \end{cases}$$

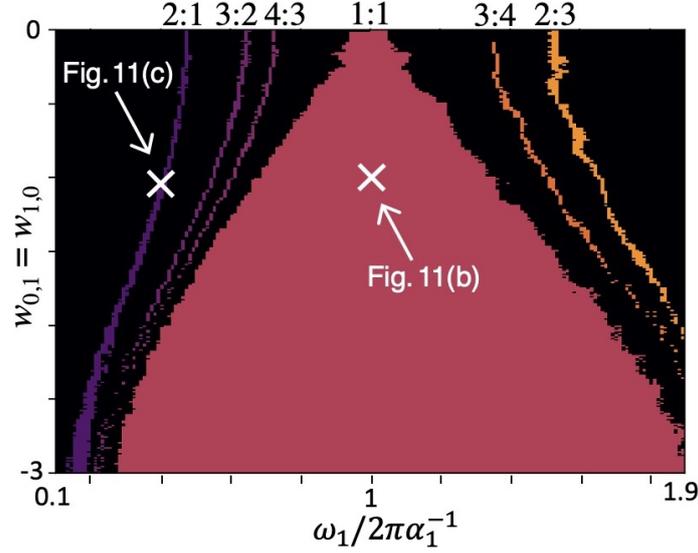


Figure 3.2.12: Parameter regions where  $n:m$  phase-lockings are observed.  $L = 6$ ,  $N = 2^5$ ,  $M = 2^6$ ,  $\rho_0 = \rho_1 = 200$ ,  $\omega_0 = 2\pi/\alpha_0$ ,  $\alpha_0 = \alpha_1 = 0.01$ ,  $T_X^{(0)} = T_X^{(1)} = 0.0012236$ ,  $T_Y^{(0)} = T_Y^{(1)} = 0.001$ ,  $\beta_0 = \beta_1 = 1$ , and  $T_Z^{(0)} = T_Z^{(1)} = 0.01$ .

which is undefined if  $\sqrt{x^2 + y^2} \leq \kappa$  and  $\kappa$  are introduced to indicate that “the radius of  $(x, y)$  is regarded to be almost zero if  $\sqrt{x^2 + y^2} \leq \kappa$ .” Further,  $\Omega$  denotes “the number of times the state vector  $(X_i(t), Y_i(t))$  has passed through a subset

$$\Gamma \equiv \left\{ (x, y) \in \mathbf{Z}_N^2 \mid x > \left\lfloor \frac{N}{2} \right\rfloor, y = \left\lfloor \frac{N}{2} \right\rfloor \right\}$$

of the state space counterclockwise.” Figs. 3.2.11(b) and (c) show the time waveforms of the phases  $(\phi_0(t), \phi_1(t))$  and restricted phases  $(\hat{\phi}_0(t), \hat{\phi}_1(t))$  corresponding to the time waveforms of the discrete state variables  $(X_i(t), Y_i(t))$ . Using the phase  $\phi_i(t)$ , the following is defined.

**Definition 5 (Phase-locking):** *The  $i$ -th and  $j$ -th CA oscillators are said to exhibit  $n:m$  phase-locking if there exists positive constants  $k$  and  $T_p$ , and coprime integers  $n$  and  $m$  such that*

$$|n\phi_i(t) - m\phi_j(t)| < k \text{ for } t > T_p.$$

For example, in Figs. 3.2.11 (b) and (c), the CA oscillator exhibits 1:1 and 1:2 phase-lockings, respectively. Fig. 3.2.12 shows the parameter regions where various  $n:m$  phase-lockings are observed. The figure shows that the coupling strength  $w_{i,j}$  can adjust the  $n:m$  phase-locking (see Table 3.2.1). Then, the following is defined.

**Definition 6 (Mean phase difference):** *Assume the  $i$ -th and  $j$ -th CA oscillators exhibit  $n:m$  phase-locking. Then, the  $i$ -th CA oscillator is said to have a mean phase difference*

$$\Phi_{i,j} = \frac{1}{T - T_p} \int_{T_p}^T n\phi_i(t) - m\phi_j(t) dt$$

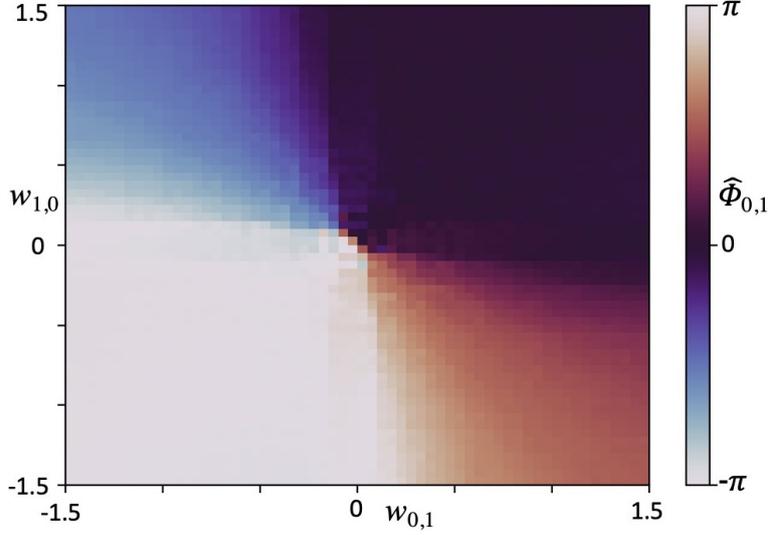


Figure 3.2.13: Characteristics of mean phase difference  $\hat{\Phi}_{i,j}$  of the network of the two CA oscillators in Fig. 3.2.11(a).  $L = 6$ ,  $N = 2^5$ ,  $M = 2^6$ ,  $\rho_0 = \rho_1 = 200$ ,  $\omega_0 = 2\pi/\alpha_0$ ,  $\omega_1 = 2\pi/\alpha_1$ ,  $\alpha_0 = \alpha_1 = 0.01$ ,  $T_X^{(0)} = T_X^{(1)} = 0.0012236$ ,  $T_Y^{(0)} = T_Y^{(1)} = 0.001$ ,  $\beta_0 = \beta_1 = 2$ , and  $T_Z^{(0)} = T_Z^{(1)} = 0.01$ .

with respect to the  $j$ -th CA oscillator, where  $T$  is an appropriately large number.

For example, in Figs. 3.2.11 (b) and (c), the 0-th CA oscillator has mean phase differences  $\hat{\Phi}_{0,1} = -3.14$  and  $-2.87$  with respect to the 1st CA oscillator, respectively. Fig. 3.2.13 shows the characteristics of the mean phase difference  $\hat{\Phi}_{0,1}$  for the coupling strengths  $w_{0,1}$  and  $w_{1,0}$ . The figure shows that the coupling strength  $w_{i,j}$  can adjust the mean phase difference  $\hat{\Phi}_{i,j}$  (see Table 3.2.1).

**Remark 2 (significance obtained from the analyses of small network):** A network of the CA oscillators is used as the CPG model in Section IV, where the ratio  $n : m$  and mean phase difference  $\hat{\Phi}_{i,j}$  of phase locking determine a spatiotemporal pattern of the orbits of the legs of the robot. Here, the analysis results of the  $n : m$  phase locking in Fig. 3.2.12 and that of the mean phase difference  $\hat{\Phi}_{i,j}$  in Fig. 3.2.13 are expected to be useful to design the CPG model. The significances of the analysis results in the CPG design are described in Section IV.

### 3.2.4 Design of CPG consisting of network of CA oscillators

Using the analyses of the single CA oscillator in Section II and those of the small network of the CA oscillators in Section III, in this section, we propose systematic design procedures of a network of the CA oscillators to control the hexapod robot in Fig. 3.2.1. Here, we introduce a bio-inspired target pattern of phase-lockings.

#### 3.2.4.1 TARGET PHASE-LOCKING PATTERN FOR CONTROLLING HEXAPOD ROBOT

Fig. 3.2.14(a) shows an illustration of an insect, where its six legs are labeled as 0–5. Fig. 3.2.14(b) shows a timing chart of a gait, where the horizontal axis represents the time and vertical axis represents movement of each leg relative to the ground. In this chart, the black bar shows the moment when the leg is off the ground and moving forward, and the

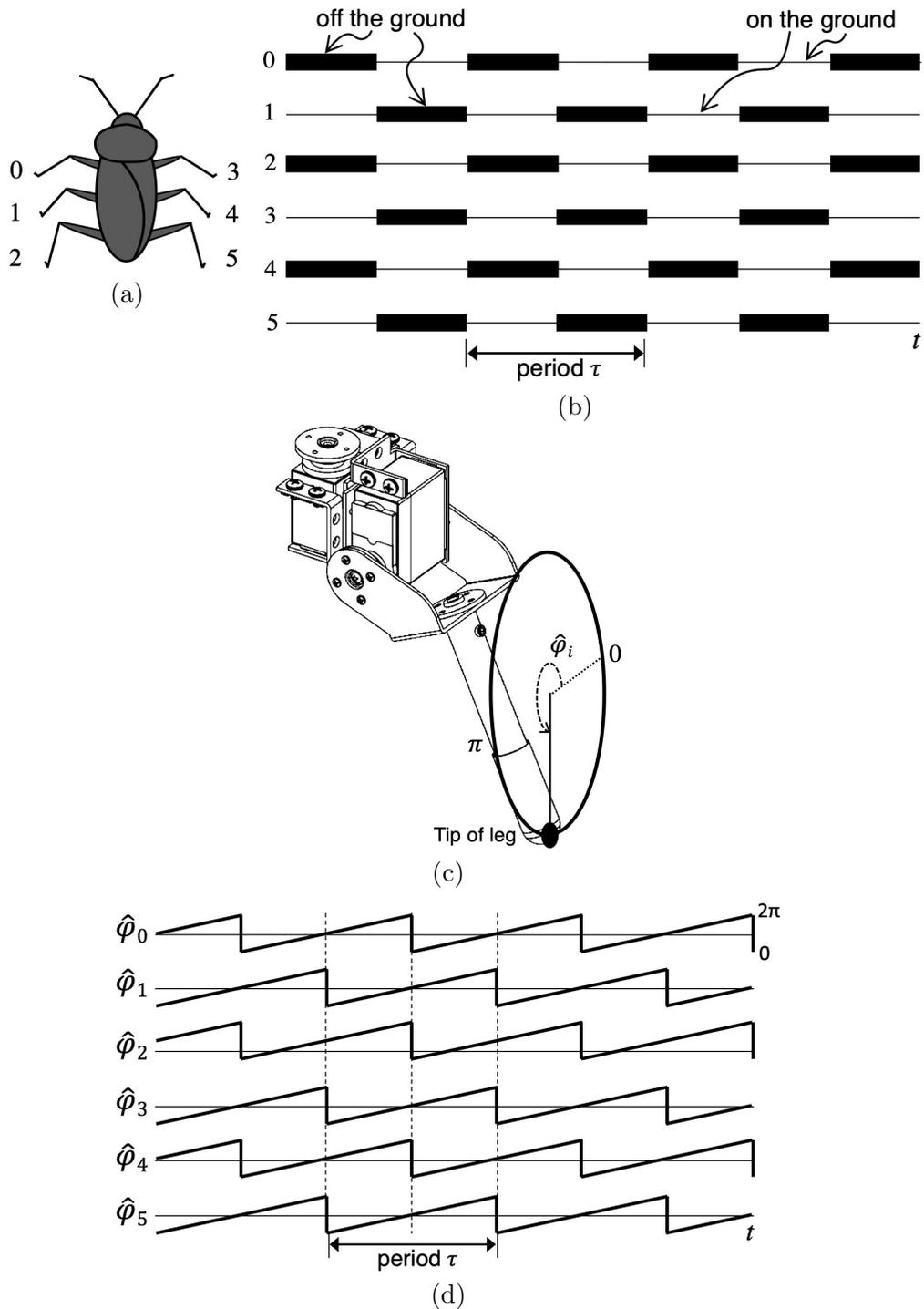


Figure 3.2.14: (a) Illustration of a six-legged insect. (b) Timing chart of a tripod gait [45]. (c)  $i$ -th leg of the hexapod robot in Fig. 3.2.1. The orbit of the tip of the leg is represented by the angle  $\hat{\varphi}_i$ . (d) Example of pattern of time-dependent angles  $\hat{\varphi}_0(t), \dots, \hat{\varphi}_5(t)$ . This pattern is used as a target phase-locking pattern of the CPG.

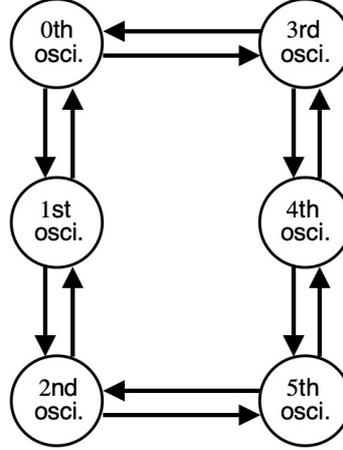


Figure 3.2.15: Network of six CA oscillators used as a CPG to control the hexapod robot in Fig. 3.2.1.

white region indicates the moment when the leg is touching and crawling on the ground. The gait in Fig. 3.2.14(b) is called a tripod gait, which is one of the typical gaits of six-legged insects [45]. As shown in the figure, a pair of black bar and white region form a period  $\tau$  of the tripod gait. Fig. 3.2.14(c) shows the  $i$ -th leg of the hexapod robot in Fig. 3.2.1. The orbit of the tip of the leg is represented by an angle  $\hat{\varphi}_i \in [0, 2\pi)$ . We consider a time-varying angle

$$\hat{\varphi}_i(t) = \frac{2\pi}{\tau}t + \gamma_i \pmod{2\pi},$$

where  $\gamma_i \in [0, 2\pi)$  is an offset parameter. Fig. 3.2.14(d) shows an example of the pattern of six time-varying angles  $(\hat{\varphi}_0(t), \dots, \hat{\varphi}_5(t))$ . Further, we consider a map

$$\sigma(\hat{\varphi}_i) = \begin{cases} \text{“black bar”} & \text{if } \hat{\varphi}_i \geq \pi, \\ \text{“white region”} & \text{if } \hat{\varphi}_i < \pi. \end{cases}$$

Applying the map  $\sigma$  to the six time-varying angles  $(\hat{\varphi}_0(t), \dots, \hat{\varphi}_5(t))$  in Fig. 3.2.14(d), the gait diagram in Fig. 3.2.14(b) is obtained. Hence, here, six CA oscillators are constructed to reproduce the time-varying angles  $(\hat{\varphi}_0(t), \dots, \hat{\varphi}_5(t))$  in Fig. 3.2.14(d). They are expected to exhibit the following pattern of phase-lockings.

**Target phase-locking pattern of six CA oscillators:**

- (i) *Each pair of CA oscillators exhibits 1:1 phase-locking.*
- (ii) *Each mean phase difference  $\Phi_{i,j}$  for  $i \in \{0, 2, 4\}$  and  $j \in \{0, 2, 4\}$  is almost zero.*
- (iii) *Each mean phase difference  $\Phi_{i,j}$  for  $i \in \{1, 3, 5\}$  and  $j \in \{1, 3, 5\}$  is almost zero.*
- (iv) *Each mean phase difference  $\Phi_{i,j}$  for  $i \in \{0, 2, 4\}$  and  $j \in \{1, 3, 5\}$  is almost  $\pi$ .*
- (v) *Each mean phase difference  $\Phi_{i,j}$  for  $i \in \{1, 3, 5\}$  and  $j \in \{0, 2, 4\}$  is almost  $\pi$ .*

The next subsection proposes the systematic design procedures of a network of the CA oscillators, which is used as a CPG to generate the aforementioned target phase-locking pattern.

### 3.2.4.2 DESIGN OF NETWORK OF CA OSCILLATORS USED AS CPG TO GENERATE TARGET PHASE-LOCKING

We propose to use the network of the CA oscillators in Fig. 3.2.15 as a CPG to generate the target phase-locking pattern. The parameters are designed as follows.

**Design procedure 1 (Parameters):** *The parameter vector  $(\rho_i, \omega_i, \alpha_i, \beta_i, T_i^Z)$  of each modified CA oscillator is set to have the same value, and the coupling strengths  $w_{i,j}$  among the oscillators are set to*

$$\mathbf{W} = \begin{pmatrix} 0 & -1 & 0 & -1 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 \end{pmatrix}. \quad (3.2.15)$$

Further, the values of the parameters  $\rho_i$  and  $(\omega_i, \alpha_i)$  are adjusted to realize the desired amplitude and frequency of the motion of the hexapod robot leg.

Here, the significances of the analyses in Section III explained in Remark 2 are as follows.

- Fig. 3.2.12 shows that a pair of the  $i$ -th and  $j$ -th CA oscillators with the same value of the parameter vector  $(\rho_i, \omega_i, \alpha_i, \beta_i, T_i^Z)$  and coupled via the strengths  $w_{i,j} = w_{j,i} = -1$  exhibits a 1:1 phase-locking. Hence, the design procedure 1 is considered suitable to realize the item (i) of the target phase-locking pattern.
- Fig. 3.2.13 shows that a pair of the  $i$ -th and  $j$ -th CA oscillators with the same value of the parameter vector  $(\rho_i, \omega_i, \alpha_i, \beta_i, T_i^Z)$  coupled via the strengths  $w_{i,j} = w_{j,i} = -1$  demonstrate mean phase differences  $|\Phi_{i,j}| \simeq |\Phi_{j,i}| \simeq \pi$ . Then, the network topology in Fig. 3.2.15 is considered suitable to realize the items (ii)–(v) of the target phase-locking pattern.

However, the design procedure 1 is not sufficient. Recall that the Remark 1 in Section II shows that the ratio of the clock periods of the CA oscillator should be tuned to design a CPG model, e.g., an irrational ratio of the clock periods and resulting asynchronous state transitions are preferred. Then, to characterize the networks of the CA oscillators with various ratios of the clock periods, the following is defined.

**Definition 7 (Asynchronous and synchronous networks):** *The network of the CA oscillators is said to be*

- *asynchronous network if  $T_i^X/T_j^X$  and/or  $T_i^Y/T_j^Y$  is irrational for certain  $i \neq j$ .*
- *synchronous network if  $T_i^X/T_j^X$  and  $T_i^Y/T_j^Y$  are rational for all  $i$  and  $j$ .*

Then, the following four types of networks exist.

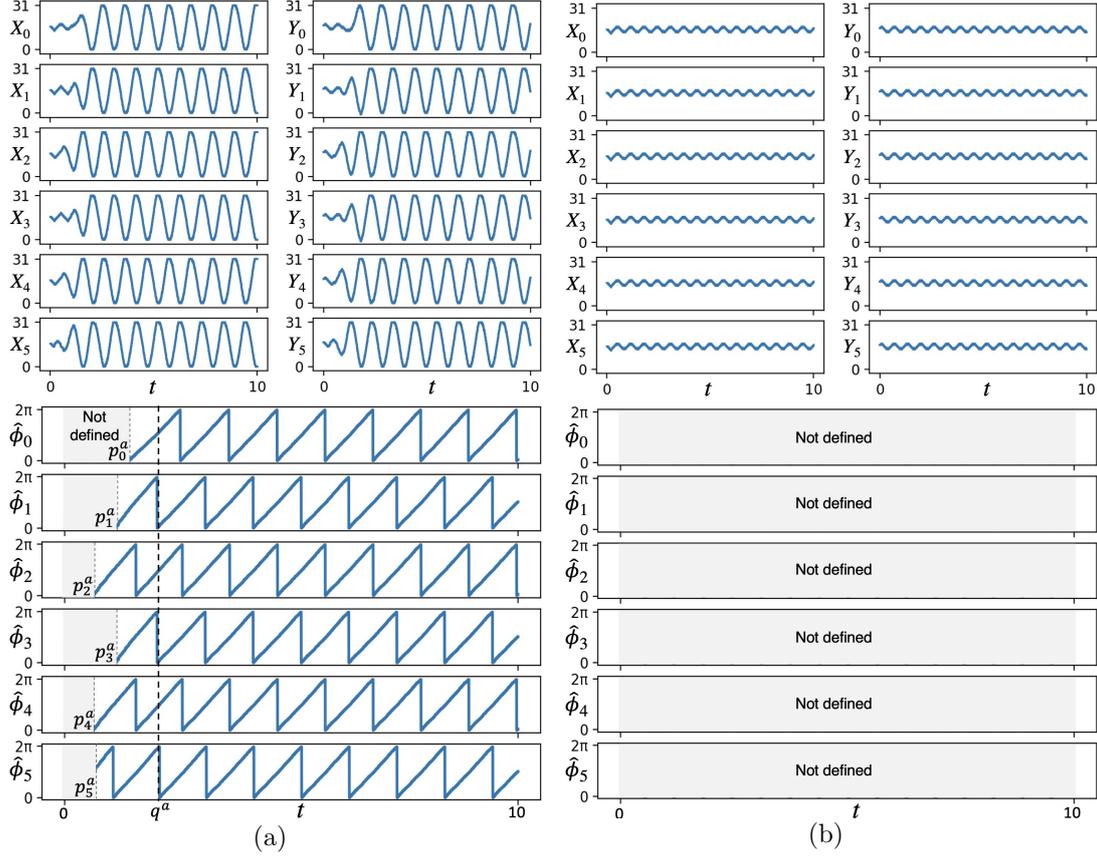


Figure 3.2.16: Time waveforms of the discrete state variables  $(X_i, Y_i)$  and corresponding restricted phases  $\hat{\phi}_i$  of four networks of CA oscillators.  $L = 6$ ,  $N = 2^5$ ,  $M = 2^6$ ,  $\rho_i = 200$ ,  $\omega_i = 2\pi/\alpha_i$ ,  $\alpha_i = 0.01$ ,  $\beta_i = 1$ , and  $T_i^Z = 0.01$  for all  $i$ .  $(X_i(0), Y_i(0), P_i(0), Q_i(0), V_i(0), U_i(0)) = (16, 17, 0, 0, 0, 0)$  for all  $i$ .  $\kappa = 3$ . (a) Asynchronous network of Asynchronous CA oscillators.  $T_i^X = 0.00100031415$  and  $T_i^Y = 0.001$  for  $i = 0, 1, 2, 3$ , and 4.  $T_5^X = 0.001$  and  $T_5^Y = 0.00100031415$ . (b) Synchronous network of Asynchronous CA oscillators.  $T_i^X = 0.00100031415$  and  $T_i^Y = 0.001$  for all  $i$ . (c) Asynchronous network of Synchronous CA oscillators.  $T_i^X = T_i^Y = 0.001$  for  $i = 0, 1, 2, 3$ , and 4.  $T_5^X = T_5^Y = 0.00100031415$ . (d) Synchronous network of Synchronous CA oscillators.  $T_i^X = T_i^Y = 0.001$  for all  $i$ . The ratio  $0.00100031415/0.001$  is regarded as an irrational number owing to the C language implementation of real numbers.

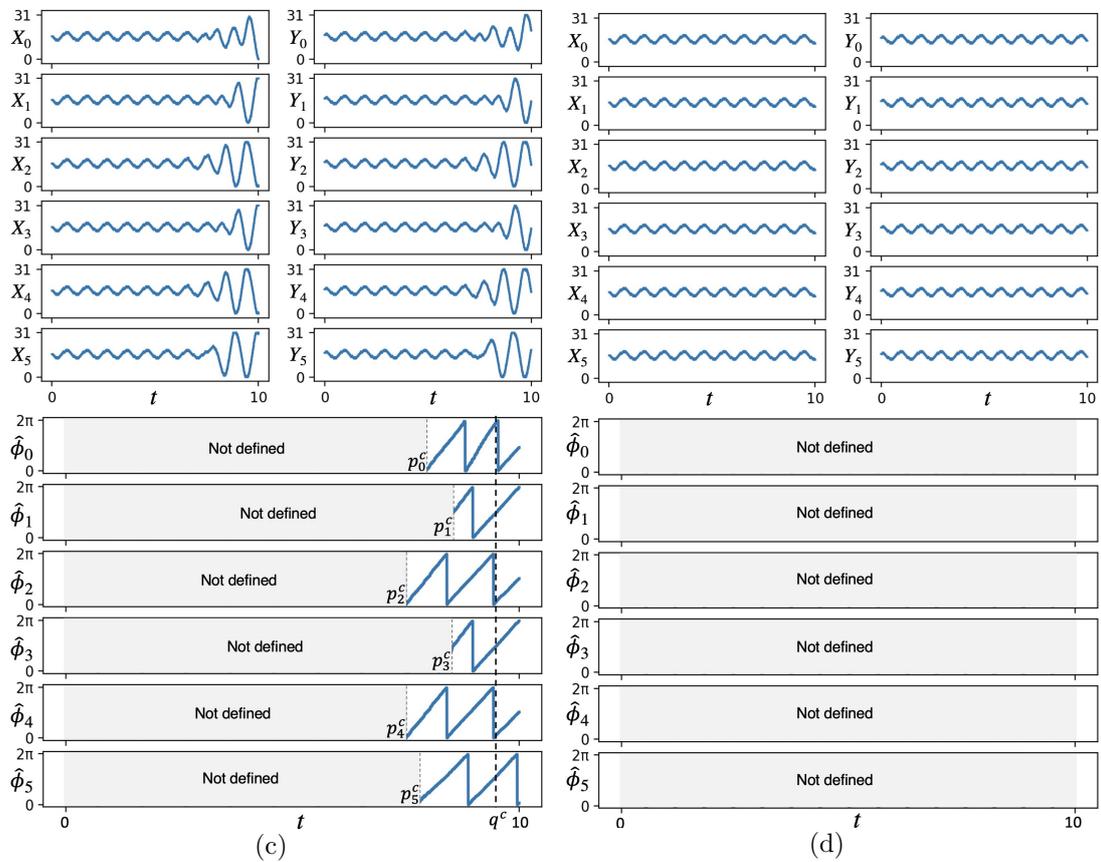


Figure 3.2.16: Continued.

Table 3.2.2: Comparison of four networks

	Asynchronous network of asynchronous CA osci.	Synchronous network of asynchronous CA osci.	Asynchronous network of synchronous CA osci.	Synchronous network of synchronous CA osci.
	Fig. 3.2.16(a)	Fig. 3.2.16(b)	Fig. 3.2.16(c)	Fig. 3.2.16(d)
Waveforms	Defined	Not defined (too small amplitude)	Defined	Not defined (too small amplitude)
Restricted phase $\hat{\phi}_i$	Yes	No	Yes	No
Realization of target phase-locking	Short	-	Long	-
Transient to target phase-locking	Suited	Not suited	Not suited (too long transient)	Not suited
Suitability to be used as CPG				

- (a) *Asynchronous network of Asynchronous CA oscillators.*
- (b) *Synchronous network of Asynchronous CA oscillators.*
- (c) *Asynchronous network of Synchronous CA oscillators.*
- (d) *Synchronous network of Synchronous CA oscillators.*

We compare the characteristics of these networks and their relations with the target phase-locking pattern, where the aforementioned (a)–(d) correspond to the following (a)–(d), respectively.

- (a) Fig. 3.2.16(a) shows the time waveforms of the discrete state variables  $(X_i, Y_i)$  and corresponding restricted phase  $\hat{\phi}_i$  of an *asynchronous network of asynchronous CA oscillators*. Each restricted phase  $\hat{\phi}_i$  is not defined for  $t < p_i^a$  as the radius  $\sqrt{(X_i - \lfloor N/2 \rfloor)^2 + (Y_i - \lfloor N/2 \rfloor)^2}$  is too small (i.e., smaller than  $\kappa$ ). Further, the network exhibits the target phase-locking pattern for  $t > q^a$  (see Table 3.2.2).
- (b) Fig. 3.2.16(b) shows the time waveforms of a *synchronous network of asynchronous CA oscillators*. Each restricted phase  $\hat{\phi}_i$  is not defined for  $t > 0$  as the radius  $\sqrt{(X_i - \lfloor N/2 \rfloor)^2 + (Y_i - \lfloor N/2 \rfloor)^2}$  is too small. Hence, the synchronous network of the asynchronous CA oscillators cannot realize the target phase-locking pattern (see Table 3.2.2).
- (c) Fig. 3.2.16(c) shows the time waveforms of an *asynchronous network of the synchronous CA oscillators*. The time interval  $[0, p_i^c]$  where each restricted phase  $\hat{\phi}_i$  is not defined is much longer than that of the asynchronous network of the asynchronous CA oscillators. The transient time interval  $[0, q^c]$  to achieve the target synchronization is much longer than that of the asynchronous network of the asynchronous CA oscillators. Hence, the asynchronous network of the synchronous CA oscillators is not suitable to be used as a CPG when compared with the asynchronous network of the asynchronous CA oscillators (see Table 3.2.2).
- (d) Fig. 3.2.16(d) shows the time waveforms of a *synchronous network of the synchronous CA oscillators*. The figure shows that each restricted phase  $\hat{\phi}_i$  is not defined for  $t > 0$  as the radius  $\sqrt{(X_i - \lfloor N/2 \rfloor)^2 + (Y_i - \lfloor N/2 \rfloor)^2}$  is too small. Hence, the synchronous network of asynchronous CA oscillators cannot realize the target phase-locking pattern (see Table 3.2.2).

The aforementioned four characteristics reveal the following consequence.

**Remark 3 (significance obtained from analyses of CPG network):** The asynchronous network of asynchronous CA oscillators is best suitable to be used as the CPG as summarized in Table 3.2.2.

Thus, we propose the following design procedure.

**Design procedure 2 (Asynchronous clocks):** *Each CA oscillator is set to have the clocks  $C_i^X$  and  $C_i^Y$  that have an irrational ratio  $T_i^X/T_i^Y$  of the periods. Further, at least one pair of the CA oscillators in the network is set to have an irrational ratio  $T_i^X/T_j^X$  or  $T_i^Y/T_j^Y$  of the periods of the clocks.*

Using the design procedures 1 and 2, the asynchronous network of the asynchronous CA

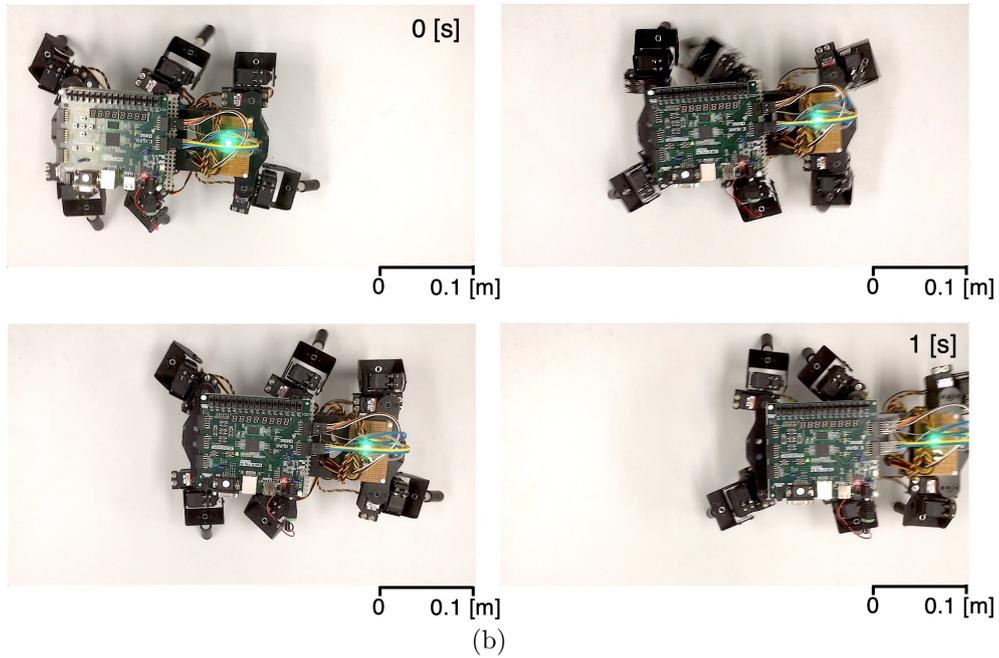
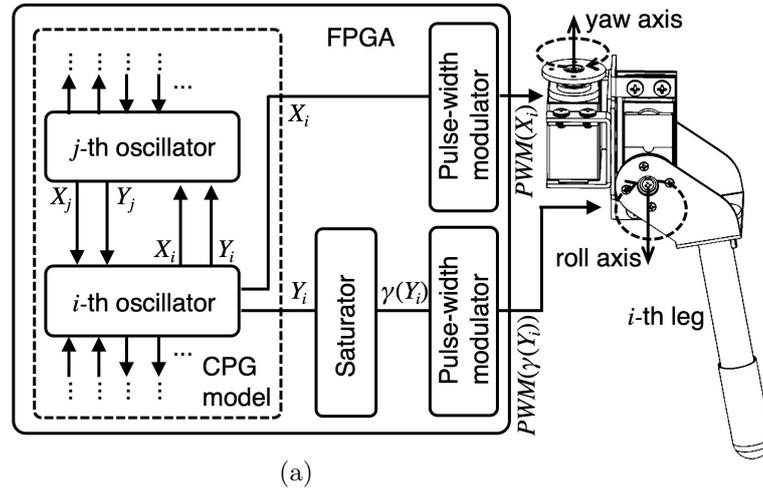


Figure 3.2.17: (a) Structure of each leg. Each leg has Hitec's servomotors HS-422 (yaw axis) and HS-645MG (roll axis). (b) Tripod gait of the robot.  $N = 2^5$ ,  $M = 2^6$ ,  $\rho_i = 200$ ,  $\omega_i = 2\pi/\alpha_i$ ,  $\alpha_i = 0.01$ ,  $L = 6$ ,  $\beta_i = 1$ , and  $T_i^Z = 0.01$  for all  $i$ .  $T_i^X = 0.0012236$  and  $T_i^Y = 0.001$  for  $i = 0, 1, 2, 3$ , and 4.  $T_5^X = 0.001$  and  $T_5^Y = 0.0012236$ .

oscillators can be designed, which can realize the target phase-locking pattern with a short transient period, as shown in Fig. 3.2.16(a).

## 3.2.5 Implementation and comparison

### 3.2.5.1 IMPLEMENTATION

This subsection shows that the proposed CPG model designed by the procedures proposed in the previous section can realize the tripod gait of the hexapod robot in Fig. 3.2.1. The dynamics of the CA oscillator are described as a register transfer level (RTL) VHDL code as follows. The discrete state variables  $(X_i, Y_i)$  are described by unsigned integers to reflect Eq. (3.2.1) and implemented by  $n$ -bit registers, where  $n = \lceil \log_2 N \rceil$ . The discrete auxiliary variables  $(P_i, Q_i)$  are described by unsigned integers to reflect Eq. (3.2.2) and implemented by  $m$ -bit registers, where  $m = \lceil \log_2 M \rceil$ . The functions  $\mathcal{F}_i$  and  $\mathcal{G}_i$  are described by two's complement signed integers to reflect Eq. (3.2.5) and implemented by look-up-tables with  $n$ -bit unsigned inputs and an  $(m+1)$ -bit signed output. Then, the state transitions in Eqs. (3.2.3), (3.2.4), (3.2.6), and (3.2.7) are described by sequential statements triggered by the clocks  $C_i^X$  and  $C_i^Y$ . Using the aforementioned CA oscillator as a component, the dynamics of the proposed CPG model are described as an RTL VHDL code as follows. The discrete state variables  $(V_i, U_i)$  for the coupling are represented by unsigned integers to reflect Eq. (3.2.9) and implemented by  $m$ -bit registers. The function  $\mathcal{H}_i$  for the coupling is represented by two's complement signed integers to reflect Eq. (3.2.11) and implemented by adders and look-up-tables with an  $(n+1)$ -bit signed input and  $(m+1)$ -bit signed output. The state transitions in Eqs. (3.2.10), (3.2.12), and (3.2.13) realize the coupling of the asynchronous CA oscillators. Then, these state transitions are described by sequential statements triggered by the clock  $C_i^Z$ . Fig. 3.2.17(a) shows a structure of the hexapod robot leg. The discrete state variable  $X_i$  of the  $i$ -th CA oscillator is transformed into a pulse-width modulated (PWM) signal  $PWM(X_i)$  by a pulse-width modulator, PWM signal  $PWM(X_i)$  instructs the angle of the servomotor, and then, the servomotor determines the angle in the yaw axis of the  $i$ -th leg of the hexapod robot. Further, the discrete state variable  $Y_i$  of the  $i$ -th CA oscillator is transformed into a saturated signal  $\gamma(Y_i)$  by a saturator

$$\gamma(Y_i) = \begin{cases} Y_i & \text{if } Y_i \geq N/2, \\ 0 & \text{otherwise,} \end{cases} \quad (3.2.16)$$

the saturated signal  $\gamma(Y_i)$  is transformed into a PWM signal  $PWM(\gamma(Y_i))$  by the pulse-width modulator, the PWM signal  $PWM(\gamma(Y_i))$  instructs the angle of the servomotor, and then, the servomotor determines the angle in the roll axis of the  $i$ -th leg. The set of VHDL codes describing the CPG model, pulse-width modulators, and saturators are compiled by Xilinx's design software environment Vivado 2018.2 and a resulting bitstream file is downloaded to Xilinx's FPGA Artix-7 XC7A100T-1CSG324C [46] mounted on Digilent's Nexys 4 DDR evaluation platform [47]. Because the FPGA and design software environment used in this study do not support asynchronous triggering, the clocks  $C_i^X$ ,  $C_i^Y$ , and  $C_i^Z$  are generated from a common clock with a high frequency (100[MHz]) such that the least common multiple of the periods of the clocks is much longer than the periods of oscillations of the discrete state variables  $X_i$  and  $Y_i$ . Thus, the clocks can be regarded to be asynchronous in practice. Fig. 3.2.17(b) shows snapshots of the hexapod robot controlled by the proposed CPG model. It can be observed that the robot can realize the tripod gait. Features of the

Table 3.2.3: Comparisons

	Proposed CPG model implemented by asynchronous sequential logic $n = 5$ bit, $m = 6$ bit	Hopf CPG model [3] implemented by customized hardware DSP 14-bit fixed point	Proposed CPG model implemented as software running on customized CPU 32-bit integer	Previous CPG model [40] implemented by asynchronous sequential logic $n = 5$ bit, $m = 10$ bit
Bit length				
Number of slices	892	3079	1458	1113
Number of LUTs	3192	9763	3548	3816
Number of FFs	497	370	3535	652
Number of BRAMs	-	-	52	-
On-Chip Power	0.218 W	0.421 W	0.313 W	0.248 W
Oscillation frequency determining gate velocity	1 Hz	1 Hz	1 Hz	1 Hz

Notes: LUT represents look-up-table, FF represents flip-flop, and BRAM represents block random access memory. For comparison, each model is implemented by the same design software environment and same FPGA device used to implement the proposed model. Each on-chip power is estimated by the same method used to estimate the power of the proposed model.

proposed CPG model implemented by FPGA are summarized in Table 3.2.3. The bit lengths  $n$  and  $m$  are sufficiently shortened under the condition that the hexapod robot realizes an appropriate tripod gait, where the resulting bit lengths are  $n = 5$  and  $m = 6$ . The on-chip power is the total of static power consumption (e.g., device static power) and dynamic power consumption (e.g., powers of clock, signal, logic, and BRAM). It is estimated by the design software environment at the post-routing stage, which provides the most accurate power estimation when compared with other design stages, such as post-synthesize and post-implementation stages.

### 3.2.5.2 COMPARISONS

This subsection compares the proposed CPG model with other CPG models.

*Hopf CPG model implemented by customized DSP* For comparison, we introduce the following Hopf CPG model as a conventional model [3].

$$\begin{aligned}
 x_i(t+h) &= x_i(t) + h(F_x(x_i(t), y_i(t)) + \sum_{j=0}^5 w_{i,j}x_j(t)), \\
 y_i(t+h) &= y_i(t) + h(F_y(x_i(t), y_i(t)) + \sum_{j=0}^5 w_{i,j}y_j(t)), \\
 F_x(x_i, y_i) &= (\mu_i^2 - (x_i^2 + y_i^2))x_i - \xi_i y_i, \\
 F_y(x_i, y_i) &= (\mu_i^2 - (x_i^2 + y_i^2))y_i + \xi_i x_i,
 \end{aligned} \tag{3.2.17}$$

where  $h$  is a stepsize of the time  $t$ ,  $x_i \in \mathbb{R}$  and  $y_i \in \mathbb{R}$  are continuous state variables, and  $w_{i,j}$  is the coupling strength defined in Eq. (3.2.15). Further,  $\mu_i \in \mathbb{R}$  and  $\xi_i \in \mathbb{R}$  are parameters characterizing an amplitude and intrinsic oscillation frequency of the state variables  $(x_i, y_i)$ , respectively. Fig. 3.2.18 shows time waveforms of the Hopf CPG model. It can be observed that the Hopf CPG model realizes the target phase-locking pattern. The dynamics of the Hopf CPG model in Eq. (3.2.17) are described as an RTL VHDL code as follows. The discrete state variables  $(x_i, y_i)$  and parameters  $(\mu_i, \xi_i)$  are described by signed fixed point numbers with 8-bit integer parts and 5-bit decimal parts, and implemented by 14-bit registers. The coupling terms are described by summations and implemented by adders. Then, the state transitions in Eq. (3.2.17) are described by sequential statements that are triggered by a single clock. The state variable  $x_i$  is transformed into a PWM signal  $PWM(ax_i + b)$  by the pulse-width modulator to determine the angle of the yaw axis of the  $i$ -th leg of the hexapod robot. Further, the state variable  $y_i$  is transformed into a saturated signal  $\gamma(ay_i + b)$  by the saturator, and the saturated signal  $\gamma(ay_i + b)$  is transformed into a PWM signal  $PWM(\gamma(ay_i + b))$  by the pulse-width modulator to determine the angle of the roll axis of the  $i$ -th leg. The set of VHDL codes describing the Hopf CPG model, pulse-width modulators, and saturator is compiled by the same design software environment that was used to compile the proposed CPG model. It is implemented by the same FPGA device that was used to implement the proposed CPG model. The bit length of the Hopf CPG model is shortened based on the same criteria as that of the proposed CPG model. It is then confirmed that the Hopf CPG model realizes the target phase-locking pattern. Features of the Hopf CPG model are summarized in Table 3.2.3. Note that the Hopf CPG model that is implemented in the FPGA can be regarded as a hardware digital signal processor (DSP),

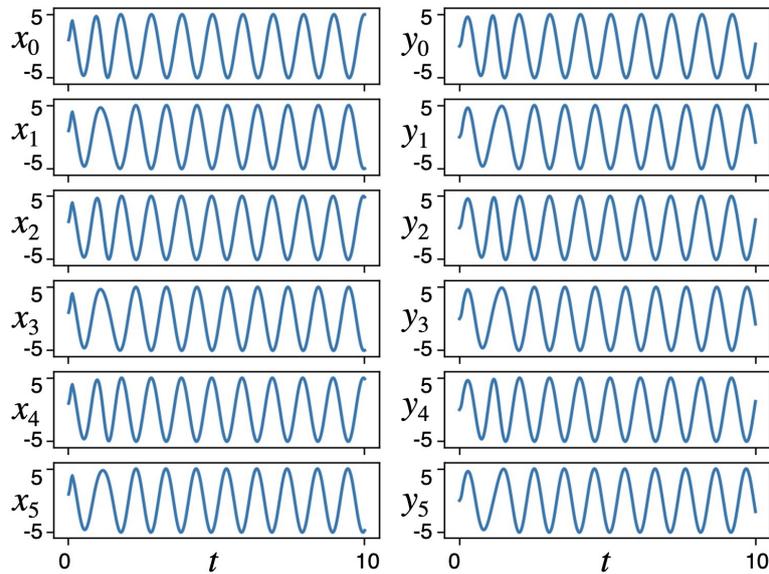


Figure 3.2.18: Time waveforms of the Hopf CPG model [3].  $\mu_i = 5^2$  and  $\xi_i = 2\pi$  for all  $i$ .  $h = 2^{-5}$ .

which is customized to execute the dynamic equation in Eq. (3.2.17).

*Proposed CPG model implemented as software running on customized CPU* For comparison, the proposed CPG model is implemented as a software running on a customized CPU as follows. Using the same design software environment used to design the proposed CPG model, a soft-core CPU called Microblaze [48] is custom designed to execute the dynamic equations in Eqs. (3.2.3), (3.2.4), (3.2.6), (3.2.7), (3.2.10), (3.2.12), and (3.2.13) of the presented CPG model. In this design, unnecessary units (e.g., floating point unit and integer multiplier) to execute the dynamic equations are not included in the CPU and the predefined configuration is selected as “application preset,” which is among the most standard predefined configurations. The resulting customized CPU is implemented by the same FPGA device used to implement the proposed CPG model. Further, the dynamic equations of the proposed CPG model are written in C-language code, the code is compiled by a compiler embedded in the design software environment used to compile the presented CPG model, and the resulting executable file is downloaded to the customized CPU with the FPGA. It is then confirmed that the CPU-based CPG model realizes the target phase-locking pattern. Features of the CPU-based CPG model are summarized in Table 3.2.3.

*Our previously reported CPG model* Our previously reported CPG model [40] is designed by the same design software environment used to design the proposed CPG model and is implemented by the same FPGA device used to implement the proposed CPG model. The differences between the previously reported and proposed CPG model are (a) the previous model has more clocks and flip-flops that do not play any important roles to realize the target phase-locking pattern, and (b) the previous model has more complicated network topology. Features of the previous model with the FPGA are summarized in Table 3.2.3.

### 3.2.5.3 DISCUSSIONS

- Table 3.2.3 shows that the proposed CPG model employs the least number of circuit elements and the lowest power compared to the other models. However, the number of flip-flops of the proposed model is slightly higher than that of the Hopf CPG model. This exception can be ignored as the number of circuit elements required to implement a flip-flop is much smaller than the number of circuit elements required to implement a look-up-table.
- The number of circuit elements employed depends on the optimization algorithm of the design software while the power consumption depends on the process rule of the target device. In this study, the proposed model and other models are designed and compiled by the same design software and implemented by the same FPGA device. Hence, it can be said that the comparisons were appropriate. This is an advantage of the FPGA-based prototype design [49,50]. In addition, the capability to analyze the specifications of customized processors (e.g., the customized DSP and CPU analyzed in this study) is another advantage of the FPGA-based prototype design.
- The FPGA is further advantageous in designing an advanced CPG model. An advanced hexapod robot requires to change its gate patterns dynamically, and thus, an advanced CPG model requires to change its coupling pattern dynamically to change the phase-locking patterns. A dynamically reconfigurable FPGA is suitable to implement such dynamically reconfigurable neuromorphic hardware [30]. However, such an advanced function of the CPG model is out of the scope of this study as we focused on fundamental studies, e.g., the detailed analyses of the nonlinear dynamics of the CPG model (Sections II and III), the detailed analyses of roles of the asynchronous clocks (Sections II and IV), development of the systematic design method of the CPG model based on the analyses results (Section IV), and implementation of the prototype and comparisons with other models (Section V). Design of a *dynamically reconfigurable CPG model based on the dynamically reconfigurable FPGA* is an important future challenge.

### 3.2.6 Conclusions

This paper presented a novel CPG model consisting of a network of CA oscillators. The detailed analyses showed the effects of the parameters on the nonlinear characteristics of the CA oscillator and its network, such as amplitude of oscillation, frequency of oscillation, phase-locking between the CA oscillators, mean phase difference between the phase-locked CA oscillators, and transient period to the phase-locking. Moreover, the detailed analyses confirmed: *the asynchronous network of asynchronous CA oscillators* is best suited to be used as the CPG to realize the bio-inspired tripod gait of the hexapod robot when compared with the other three types of networks. Using these results, we proposed the systematic design procedures of the proposed CPG model to realize the tripod gait. Then, the CPG model designed by the proposed procedures was implemented in the FPGA, and its operation was verified through experiments. It was shown that the proposed CPG model employs much fewer circuit elements and lower power than the conventional CPG model. Future scope for research is as follows: (a) detailed analyses of various nonlinear dynamics of the proposed CPG model, (b) realization of other gaits of the hexapod robot based on

the proposed CPG model, (c) development of further hardware-efficient CPG model based on dimension reduction techniques, and (d) development of a dynamically reconfigurable CPG model based on the dynamically reconfigurable FPGA.

## 3.3 Hexapod Robot Controlled by Coupled Phase Oscillators<sup>4</sup>

### 3.3.1 Introduction

Legged animals perform complex yet stable locomotion and even adapt their rhythmic patterns for gait appropriate to their walking speed and certain types of terrain. It has been thought that such rhythmic patterns are produced by central pattern generators (CPGs), biological neural circuits found in the spinal cords, without sensory feedback [1]. The CPGs are mostly modeled by networks composed of nonlinear oscillators (e.g., Van der Pol oscillators [51], Hopf oscillators [52], Kuramoto oscillators [53], and spiking neurons [27]), which employ their synchronization properties to produce the rhythmic patterns. These models have been successfully utilized in gait generations for various types of legged robots [13]. Further, in recent years, CPG models have been applied in the field of medical engineering, e.g., non-invasive and invasive prostheses [14,15]. For such practical applications, the CPG models must utilize fewer hardware resources and consume less power. However, bio-inspired models, including the CPG models, tend to increase the hardware resources required for circuit implementation due to their nonlinearities. Considering the nonlinear circuit and system theory, the bio-inspired models are classified into the following four classes based on continuousness and discontinuousness of state variables and time.

*Class CTCS.* This is a nonlinear differential equation model of a bio-inspired system with *a continuous time and continuous states* (CTCS). A class CTCS bio-inspired model can be generally implemented in an analog nonlinear circuit, e.g., [15,17,18,54–57].

*Class DTCS.* This is a nonlinear difference equation model of a bio-inspired system with *a discrete time and continuous states* (DTCS). A class DTCS bio-inspired model can be generally implemented in a switched capacitor circuit, e.g., [20–22].

*Class DTDS.* This is a numerical integration model (in finite binary number representation) of a bio-inspired system with *a discrete time and discrete states* (DTDS). A class DTDS bio-inspired model can be generally implemented in a digital processor or a sequential logic circuit, e.g., [5,14,27,51–53,58].

*Class CTDS.* This is an asynchronous cellular automaton (CA) model of a bio-inspired system with *a continuous (state transition) time and discrete states* (CTDS). A class CTDS bio-inspired model can be generally implemented in an asynchronous sequential logic circuit, e.g., [40,59–61].

---

<sup>4</sup>This section is based on “Smooth Gait Transition in Hardware-Efficient CPG Model based on Asynchronous Coupling of Cellular Automaton Phase Oscillators,” by the same author, which appeared in *Nonlinear Theory and Its Applications*, IEICE, vol. 12, no. 3, pp. 336–356, 2021, Copyright(C)2021 IEICE.

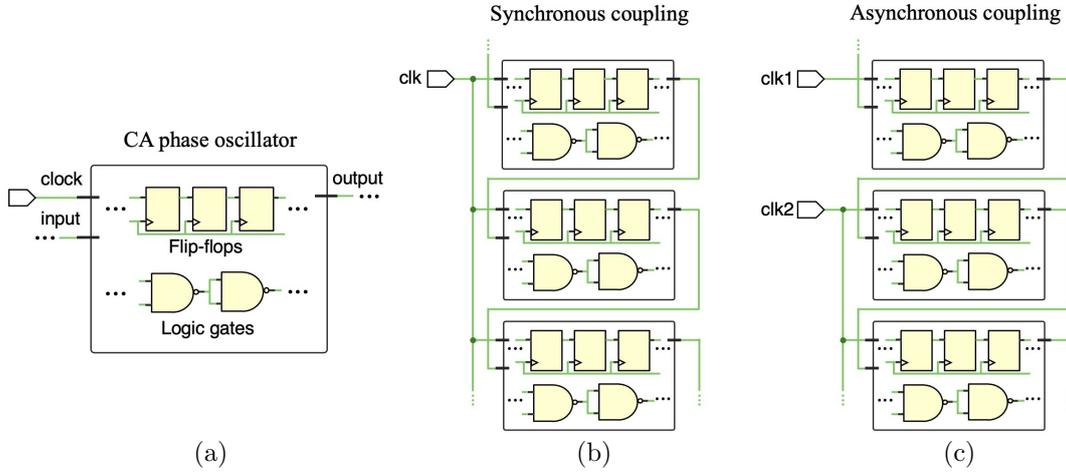


Figure 3.3.1: (a) Cellular automaton (CA) phase oscillator. (b) Synchronous coupling of CA phase oscillators. (c) Asynchronous coupling of CA phase oscillators.

Our group has been developing CPG models for hexapod robots belonging to the class CTDS system, which utilize fewer hardware resources for circuit implementation than models belonging to the class DTDS system [40, 59–61]. Fig. 3.3.1 shows conceptual diagrams of our CPG model studied in this paper. As shown in Fig. 3.3.1(a), our CPG model is composed of CA phase oscillators implemented in a sequential logic circuit, which have flip-flops storing discrete state variables and logic gates realizing discrete nonlinear coupling functions. In particular, the state space of our CPG model is roughly discretized, i.e., the required bit length for the state space is very small. Accordingly, the logic gates for the nonlinear coupling function can also be designed to be small. However, in this study, we found that, due to the rough discretization, transitions of discrete variables do not work properly if they are triggered by a single clock (synchronous coupling of the CA phase oscillators as shown in Fig. 3.3.1(b)). In order to overcome this difficulty, we employed the asynchronous nature of the clocks, that is, the state transitions of the discrete variables are triggered by multiple asynchronous clocks (asynchronous coupling of the CA phase oscillators as shown in Fig. 3.3.1(c)). We also clarified that the CPG model based on the asynchronous coupling of CA phase oscillators can perform smooth gait transition for the hexapod robot while the CPG model based on the synchronous coupling of CA phase oscillators cannot. It should be noted that a CPG model belonging to the class DTDS system means that the oscillators composing a network are synchronously coupled, i.e., their state transitions are triggered by a single clock. Preliminary results can be found in our conference proceedings, which reported that our CPG model can produce some types of gait patterns and can be implemented in a field programmable gate array (FPGA) utilizing fewer hardware resources than a CPG model belonging to the class DTDS system [60, 61]. That is to say this paper studied the remaining important issues about the smooth transition between the gait patterns in our CPG model and comparisons of asynchronously coupled and synchronously coupled CA phase oscillators.

The significance of this paper is as follows.

**Significance:** In this study, we firstly demonstrate that roughly discretizing a state space

of a CPG model in order to reduce hardware resources causes gait transitions failures. Then, this paper presents a CPG model based on asynchronous coupling of oscillators and demonstrates that the model can realize smooth transition between different gait patterns. This advantage of the presented model suggests that this study contributes to design future applications such as the non-invasive and invasive prosthetic devices [14, 15], and the bio-inspired robots [13], which can be implemented as a small-scale circuit and has low power consumption.

### 3.3.2 Model description

This section presents a central pattern generator (CPG) model based on asynchronous coupling of cellular automaton (CA) phase oscillators for a hexapod robot shown in Fig. 3.3.2(a). The presented CPG model consists of the CA phase oscillators, where a schematic diagram of each CA phase oscillator is shown in Fig. 3.3.2(b). As shown in this figure, each CA phase oscillator has the following discrete phase variable and discrete auxiliary variable.

**Discrete phase variable:**

$$\Phi_i \in \mathbf{Z}_N^+ \equiv \{0, \dots, N-1\}, \quad (3.3.1)$$

**Discrete auxiliary variable:**

$$P_i \in \mathbf{Z}_M^+ \equiv \{0, \dots, M-1\}, \quad (3.3.2)$$

where  $N$  and  $M$  are positive integers. Further,  $i \in \{1, \dots, n\}$  represents an index for the oscillators with periodic boundary conditions (e.g.,  $\Phi_{n+1} = \Phi_1$  and  $P_{n+1} = P_1$ ), where  $n$  is a positive integer representing the number of oscillators. State transitions of the discrete variables  $\Phi_i$  and  $P_i$  are triggered by the following internal clock.

**Internal clock:**

$$Clk_i(t) \equiv \sum_{j=0}^{\infty} \delta(t - jT_i), \quad Clk_i \in \{0, 1\}, \quad (3.3.3)$$

where  $t \in \mathbb{R}$  represents a continuous time,  $T_i \in (0, \infty)$  represents a period of the  $i$ -th internal clock  $Clk_i$ , and  $\delta$  represents the following unit impulse.

$$\delta(x) \equiv \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{if } x \neq 0, \end{cases} \quad \delta : \mathbb{R} \rightarrow \{0, 1\}. \quad (3.3.4)$$

Then, as shown in Fig. 3.3.3, the internal clock  $Clk_i$  triggers the following state transitions of the discrete phase variable  $\Phi_i$ .

$$\begin{aligned} & \text{If } Clk_i(t) = 1 \text{ and } P_i(t) \geq |H(\Delta\Phi_i^+)|, \text{ then} \\ \Phi_i(t_+) & := \begin{cases} \Phi_i(t) + 1 & \text{if } H(\Delta\Phi_i^+) \geq 0 \text{ and } \Phi_i(t) < N - 1, \\ 0 & \text{if } H(\Delta\Phi_i^+) \geq 0 \text{ and } \Phi_i(t) = N - 1, \\ \Phi_i(t) - 1 & \text{if } H(\Delta\Phi_i^+) < 0 \text{ and } \Phi_i(t) > 0, \\ N - 1 & \text{if } H(\Delta\Phi_i^+) < 0 \text{ and } \Phi_i(t) = 0, \end{cases} \end{aligned} \quad (3.3.5)$$

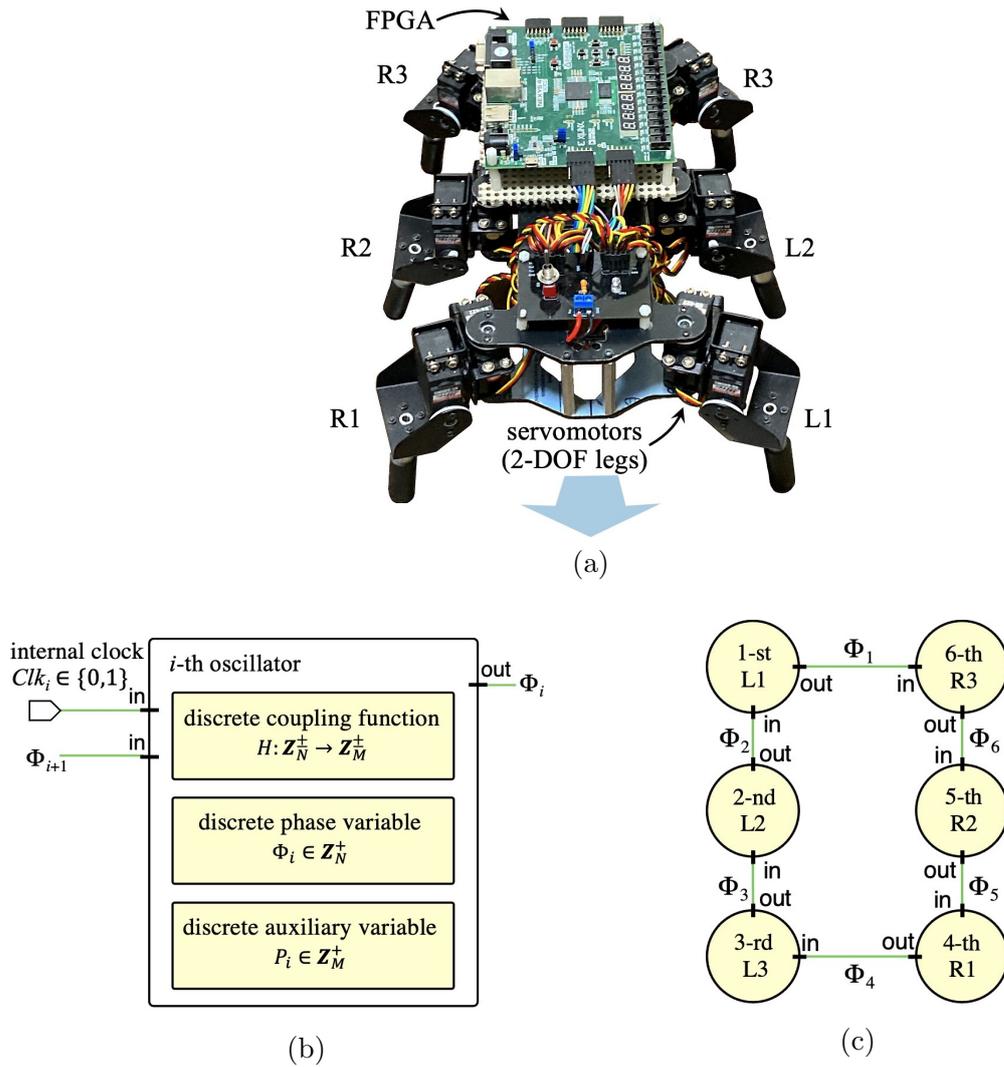


Figure 3.3.2: (a) Hexapod robot mounted with a field programmable gate array (FPGA), in which the presented CPG model is implemented. (b) Schematic diagram of the CA phase oscillator. (c) Network topology of the CPG presented model.

where the symbol “ $t_+$ ” denotes “ $\lim_{\varepsilon \rightarrow +0} t + \varepsilon$ ” and the symbol “ $:=$ ” denotes an “instantaneous state transition” throughout the paper. Further, the discrete function  $H$ , which works as a coupling function as shown in Fig. 3.3.2(b), is defined as follows.

**Discrete coupling function :**

$$H(\Delta\Phi^+) \equiv \begin{cases} M-1 & \text{if } h(\Delta\Phi^+) = 0 \text{ or } \lfloor h(\Delta\Phi^+)^{-1} \rfloor > M-1, \\ -(M-1) & \text{if } \lfloor h(\Delta\Phi^+)^{-1} \rfloor < -(M-1), \\ \lfloor h(\Delta\Phi^+)^{-1} \rfloor & \text{otherwise,} \end{cases} \quad (3.3.6)$$

$$H : \mathbf{Z}_N^\pm \equiv \{0, \dots, \pm 2(N-1)\} \rightarrow \mathbf{Z}_M^\pm \equiv \{0, \dots, \pm(M-1)\},$$

where  $\lfloor \cdot \rfloor$  denotes the following floor function.

$$\lfloor x \rfloor \equiv \max\{l \in \mathbb{Z} \mid l \leq x\}, \quad x \in \mathbb{R}. \quad (3.3.7)$$

The function  $h$  is defined as

$$h(\Delta\Phi^+) \equiv \Gamma N \sin(2\pi\Delta\Phi^+/N), \quad h : \mathbf{Z}_N^\pm \rightarrow \mathbb{R}, \quad (3.3.8)$$

where  $\Gamma \in \mathbf{R}$  represents a coupling constant.  $\Delta\Phi_i^+$  denotes

$$\Delta\Phi_i^+ \equiv \Phi_{i+1}(t) - \Phi_i(t) + \lfloor N\phi/2\pi \rfloor \bmod N, \quad (3.3.9)$$

where  $\phi \in [0, 2\pi)$  is a parameter representing the phase difference of the discrete phase variables  $\Phi_i$  and  $\Phi_{i+1}$ . Hence, the CA phase oscillators are unidirectionally coupled in a loop arrangement, as shown in Fig. 3.3.2(c). It should be noted that if  $N$  and  $M$  are not so large, the discrete coupling function  $H$  can be implemented in look-up-tables (LUTs), which consume less hardware resources than arithmetic logic units. Further, as shown in Fig. 3.3.3, the internal clock  $Clk_i$  triggers the following state transition of the discrete auxiliary variable  $P_i$ .

$$\text{If } Clk_i(t) = 1, \text{ then } P_i(t_+) := \begin{cases} P_i(t) + 1 & \text{if } P_i(t) < |H(\Delta\Phi_i^+)|, \\ 0 & \text{if } P_i(t) \geq |H(\Delta\Phi_i^+)|. \end{cases} \quad (3.3.10)$$

Figs. 3.3.4(a) and (b) show the examples of time waveforms of the CA phase oscillators. Further, Figs. 3.3.4(c) and (d) show the Cartesian coordinate representations of the discrete phase variables  $\Phi_i$  on the unit circles for  $t = 0$  and  $t = 3$ , where the  $i$ -th oscillator is plotted at

$$x = \cos(2\pi\Phi_i/N), \quad y = \sin(2\pi\Phi_i/N), \quad (3.3.11)$$

as the black circle with the index. These figures show that the coupled CA phase oscillators exhibit in-phase synchronization. Depending on the clock periods  $T_i$ , the following definition is introduced.

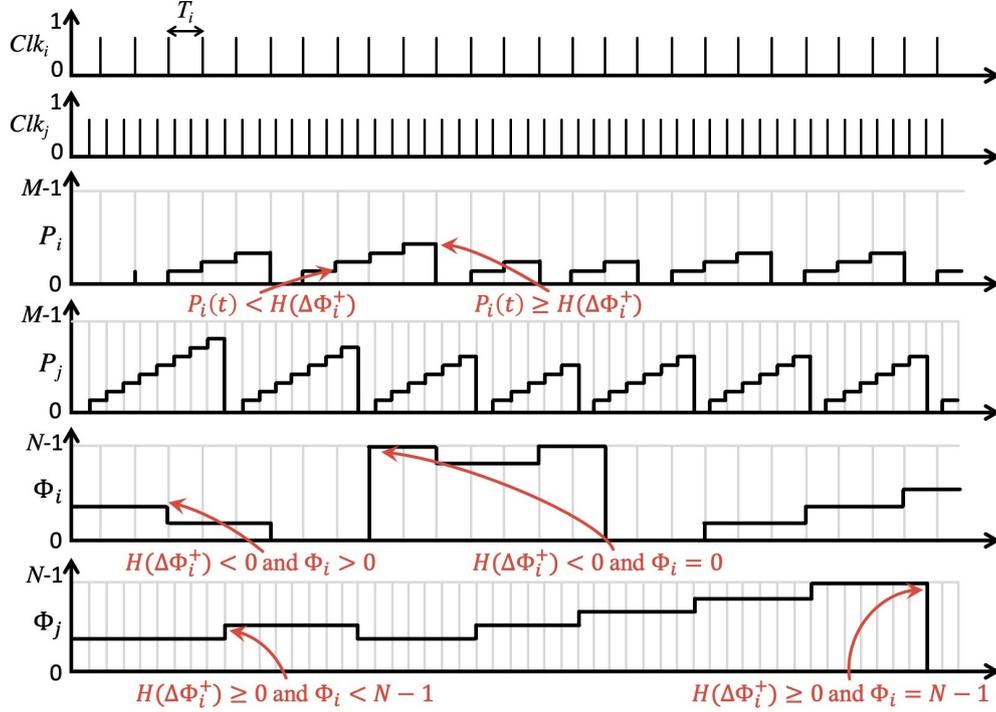


Figure 3.3.3: State transitions of discrete phase variables  $\Phi_i$  and discrete auxiliary variables  $P_i$  triggered by internal clocks  $Clk_i$  defined by Eqs. (3.3.5) and (3.3.10).

*Definition.* The CA phase oscillators are said to be

$$\begin{cases} \text{synchronously coupled} & \text{if } T_i/T_j = \text{rational number} \text{ for all } i \text{ and } j, \\ \text{asynchronously coupled} & \text{if } T_i/T_j = \text{irrational number} \text{ for certain } i \neq j. \end{cases} \quad (3.3.12)$$

In the case of Fig. 3.3.4, the CA phase oscillators are synchronously coupled. The difference in behaviors between the synchronously and the asynchronously coupled CA phase oscillators is discussed in Subsection 3.2.

### 3.3.3 Synchronization phenomena in presented CPG model

This section shows that the presented CPG model can imitate typical gait patterns for the hexapod robot. By analyses using the evaluation function to quantify synchronization states, an advantage of the asynchronously coupled CA phase oscillators is clarified. Let us begin by introducing target gait patterns in the following subsection.

#### 3.3.3.1 TARGET GAIT PATTERNS

Figs. 3.3.5(a) and (b) show gait diagrams [45] of a six-legged insect, where the labels L1–L3 and R1–R3 correspond to those shown in Figs. 3.3.2(a) and (c). These gait patterns, called tripod gait (fast) and wave gait (slow), are used as the target gait patterns to be imitated in this study. In the diagrams, the blue rectangle shows the moment when the leg is off the ground and moving forward, and the orange rectangle shows the moment when the leg is touching and crawling the ground. A pair of the blue rectangle and the orange

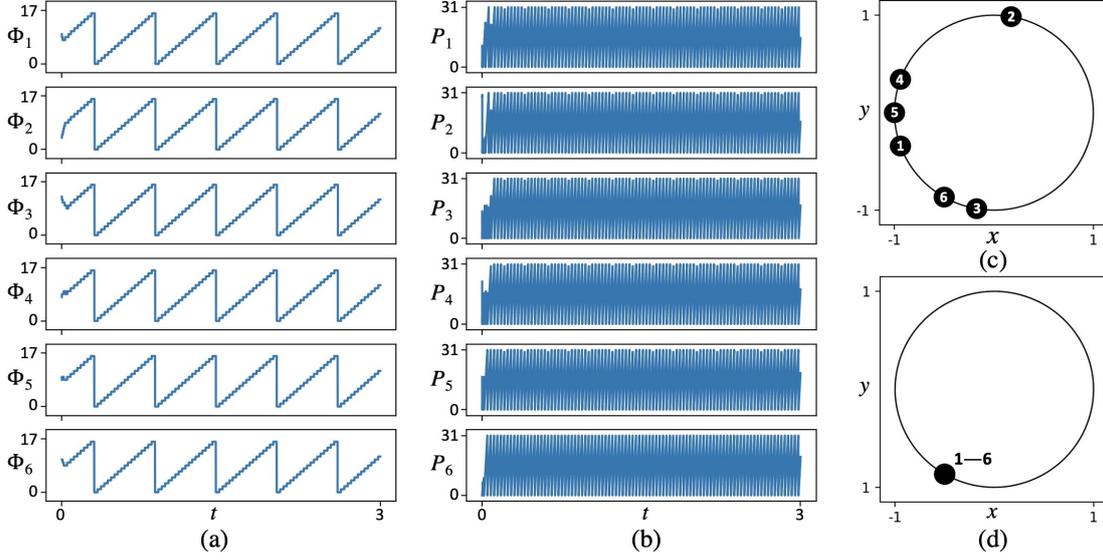


Figure 3.3.4: (a) Example of time waveforms of discrete phase variables  $\Phi_i$ . (b) Example of time waveforms of discrete auxiliary variables  $P_i$ . (c) Cartesian coordinate representation on unit circle for  $t = 0$ . The CA phase oscillator are plotted based on Eq. (3.3.11). (d) Cartesian coordinate representation on unit circle for  $t = 3$ . The parameter values are fixed as follows:  $N = 18$ ,  $M = 32$ ,  $\Gamma = 10^{-2}$ ,  $\phi = 0$ , and  $T_i = 10^3$  for all  $i$ .

rectangle forms a period  $\tau$  as indicated by the black arrow in Fig. 3.3.5(a). In order to realize synchronization patterns for the coupled CA phase oscillators from the above target gait patterns, let us consider the following time-varying phases as shown in Figs. 3.3.5(c) and (d),

$$\varphi_i(t) \equiv 2\pi t/\tau + \psi_i \pmod{2\pi}, \quad \varphi_i \in [0, 2\pi), \quad (3.3.13)$$

where  $\psi_i \in [0, 2\pi)$  represents an initial phase as indicated by the black arrow in Fig. 3.3.5 (c). Let us also consider the following map,

$$\sigma(\varphi) \equiv \begin{cases} \text{“blue rectangle”} & \text{for } \varphi < \theta, \\ \text{“orange rectangle”} & \text{for } \varphi \geq \theta, \end{cases} \quad (3.3.14)$$

$$\sigma : [0, 2\pi) \rightarrow \{\text{“blue rectangle”, “orange rectangle”}\},$$

where  $\theta \in [0, 2\pi)$  represents a threshold to determine a boundary between the “blue rectangle” and the “orange rectangle” as indicated by the black arrow in Fig. 3.3.5 (c). Applying the map  $\sigma$  to the six time-varying phases  $(\varphi_1(t), \dots, \varphi_6(t))$ , a gait diagram can be obtained. For example, the diagram of the tripod gait shown in Fig. 3.3.5(a) can be obtained by applying the maps  $\sigma$  to the phases  $\varphi_i$  for a parameter case

$$(\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6, \theta) = (0, \pi, 0, \pi, 0, \pi, 0, \pi). \quad (\textit{tripod gait}) \quad (3.3.15)$$

Fig. 3.3.5(c) shows the time waveforms of the phases  $\varphi_i$  under the above parameter case, where the shadow rectangle shows the region when  $\varphi_i(t) < \theta$ , which corresponds to the

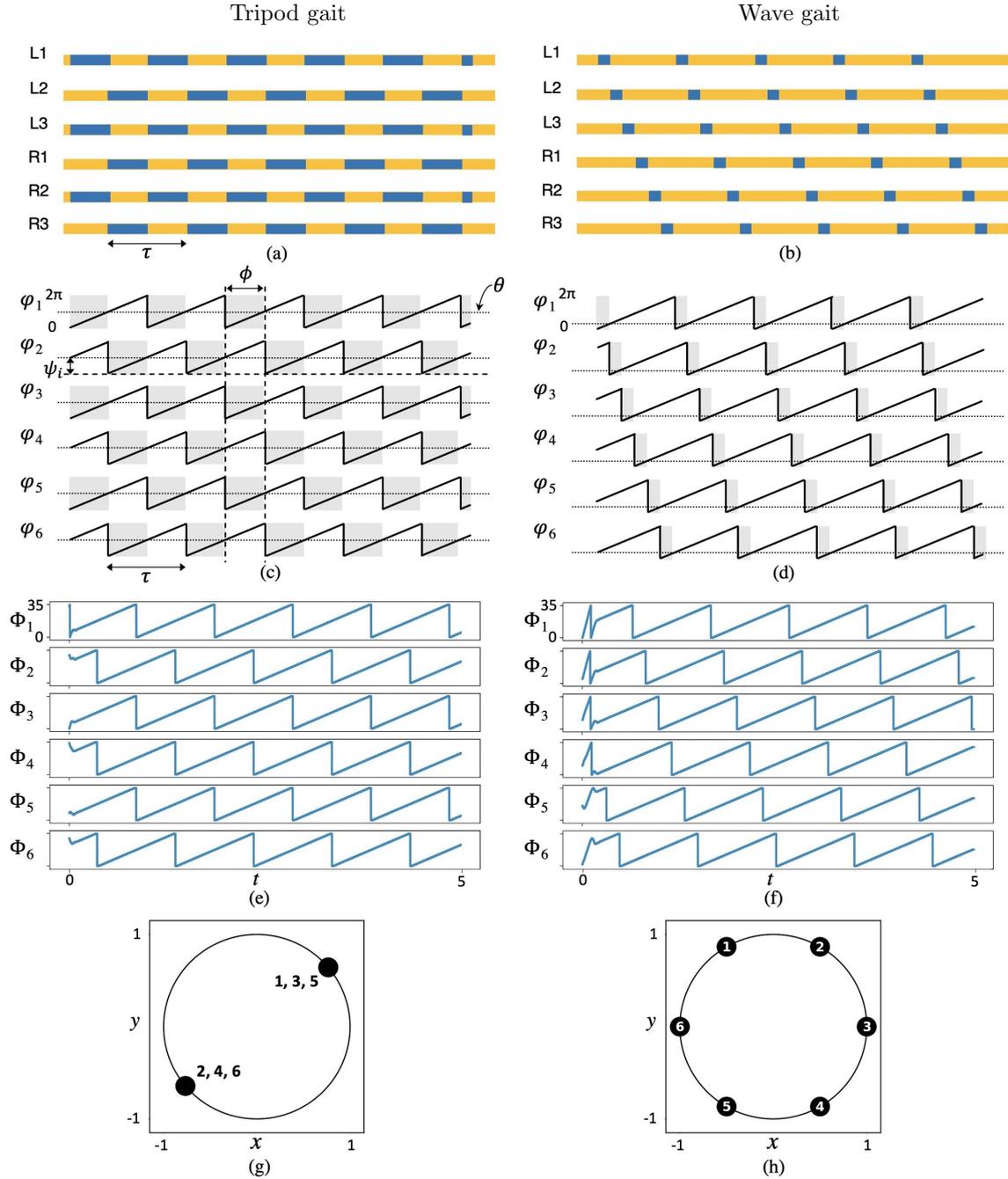


Figure 3.3.5: Two typical gait diagrams [45] and their target synchronization patterns. (a) Gait diagram of tripod gait (fast). (b) Gait diagram of wave gait (slow). (c) Target synchronization pattern for tripod gait defined by Eq. (3.3.13). The parameter values are described in Eq. (3.3.15). (d) Target synchronization pattern for wave gait defined by Eq. (3.3.13). The parameter values are described in Eq. (3.3.16). (e) Time waveforms of coupled CA phase oscillators for tripod gait. The parameter values are described in Eq. (3.3.17). The others are fixed as follows:  $N = 36$ ,  $M = 64$ ,  $\Gamma = 4.347 \times 10^{-3}$ , and  $T_i = 4.347 \times 10^{-4}$  for all  $i$ . (f) Time waveforms of coupled CA phase oscillators for wave gait. The parameter values are described in Eq. (3.3.18). The others are fixed as the same values chosen in (e). (g) Cartesian coordinate representation on unit circle for  $t = 5$  of time waveforms in (e). (h) Cartesian coordinate representation on unit circle for  $t = 5$  of time waveforms in (f).

“blue rectangle” in the diagram in Fig. 3.3.5(a). Further, the diagram of the wave gait shown in Fig. 3.3.5(b) can be obtained by applying the maps  $\sigma(\varphi_i)$  to the phases  $\varphi_i$  for a parameter case

$$(\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6, \theta) = (0, 5\pi/3, 4\pi/3, \pi, 2\pi/3, \pi/3, \pi/3). \quad (\text{wave gait}) \quad (3.3.16)$$

Fig. 3.3.5(d) shows the time waveforms of the phases  $\varphi_i$  under the above parameter case. Hence, the synchronization pattern of the time-varying phases  $\varphi_i$  shown in Figs. 3.3.5(b) and (c) should be generated by the presented CPG model for imitating the tripod gait and the wave gait. In the presented CPG model, the phase differences of the oscillators (e.g., the black arrow indicated in Fig. 3.3.5(c)) coincide with the system parameter  $\phi$ . Fig. 3.3.5(e) shows the time waveforms of the tripod gait generated by the presented CPG model, where the values of the parameter  $\phi$  is

$$\phi = \pi. \quad (\text{tripod gait}) \quad (3.3.17)$$

Fig. 3.3.5(g) shows Cartesian coordinate representations of the discrete phase variables  $\Phi_i$  corresponding to Fig. 3.3.5(e) in a steady state. Further, Fig. 3.3.5(f) shows the time waveforms of the wave gait generated by the presented CPG model, where the values of the parameter  $\phi$  is

$$\phi = \pi/3. \quad (\text{wave gait}) \quad (3.3.18)$$

Fig. 3.3.5(h) shows Cartesian coordinate representations of the discrete phase variables  $\Phi_i$  corresponding to Fig. 3.3.5(e) in a steady state. The next subsection investigates differences between the synchronously and asynchronously coupled CA phase oscillators for the cases in Eqs. (3.3.17) and (3.3.18).

### 3.3.3.2 SYNCHRONIZATION ANALYSIS OF ASYNCHRONOUSLY AND SYNCHRONOUSLY COUPLED CA PHASE OSCILLATORS

In order to quantify the synchronization patterns of the coupled CA phase oscillators, the following instantaneous evaluation function for the tripod gait is introduced.

$$r_{\text{tripod}}(t) \equiv \frac{1}{n} \left| \sum_{i=1}^{n/2} \left( e^{j2\pi \frac{\Phi_{2i}(t)}{N}} + e^{j \left( 2\pi \frac{\Phi_{2i+1}(t)}{N} + \pi \right)} \right) \right|, \quad r_{\text{tripod}} \in [0, 1], \quad (3.3.19)$$

where  $j = \sqrt{-1}$ . The instantaneous evaluation function  $r_{\text{tripod}}$  closes to 1 means that the coupled CA phase oscillators achieve the target synchronization pattern for the tripod gait. It should be noted that  $r_{\text{tripod}}$  only evaluates phase relationship of the oscillators, which does not necessarily mean that they are oscillating properly even if  $r_{\text{tripod}} \approx 1$ . Accordingly, the following instantaneous average velocity of the coupled CA phase oscillators is introduced.

$$v_{\text{ave}}(t) \equiv \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^{K_i(t)} \frac{\Phi_i(t_{i,k}) - \Phi_i(t_{i,k-1})}{N(t_{i,k} - t_{i,k-1})} I_{i,k}(t), \quad v_{\text{ave}} \in \mathbb{R}, \quad (3.3.20)$$

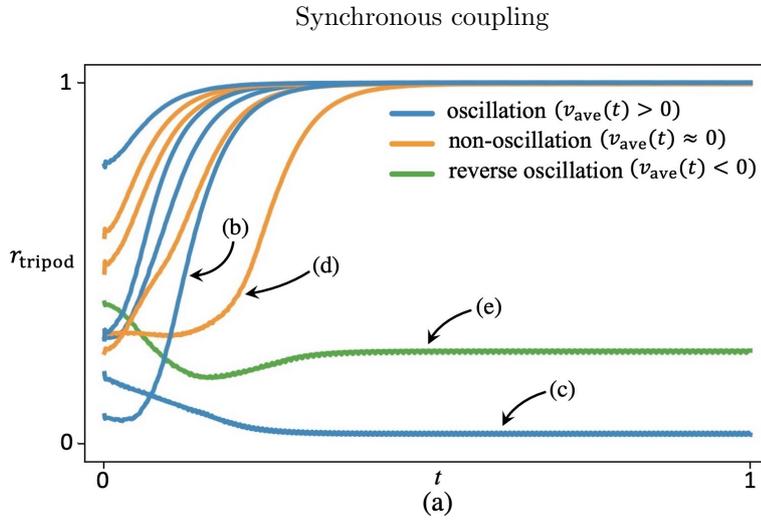


Figure 3.3.6: (a) Instantaneous evaluation function  $r_{\text{tripod}}$  defined by Eq. (3.3.19) for ten simulation trials of synchronously coupled CA phase oscillators from randomly chosen initial values. The parameter values are fixed as follows:  $N = 36$ ,  $M = 64$ ,  $\Gamma = 4.347 \times 10^{-3}$ , and  $T_i = 4.347 \times 10^{-4}$  for all  $i$ . The others are described in Eq. (3.3.17). (b) Oscillation with target synchronization. The time waveforms and Cartesian coordinate representations on the unit circle correspond to the blue lines approaching 1 in (a). (c) Oscillation with non-target synchronization. The time waveforms and Cartesian coordinate representations on the unit circle correspond to the blue line not approaching 1 in (a). (d) Non-oscillation. The time waveforms and Cartesian coordinate representations on the unit circle correspond to the orange line in (a). (e) Reverse oscillation. The time waveforms and Cartesian coordinate representations on the unit circle correspond to the green line in (a).

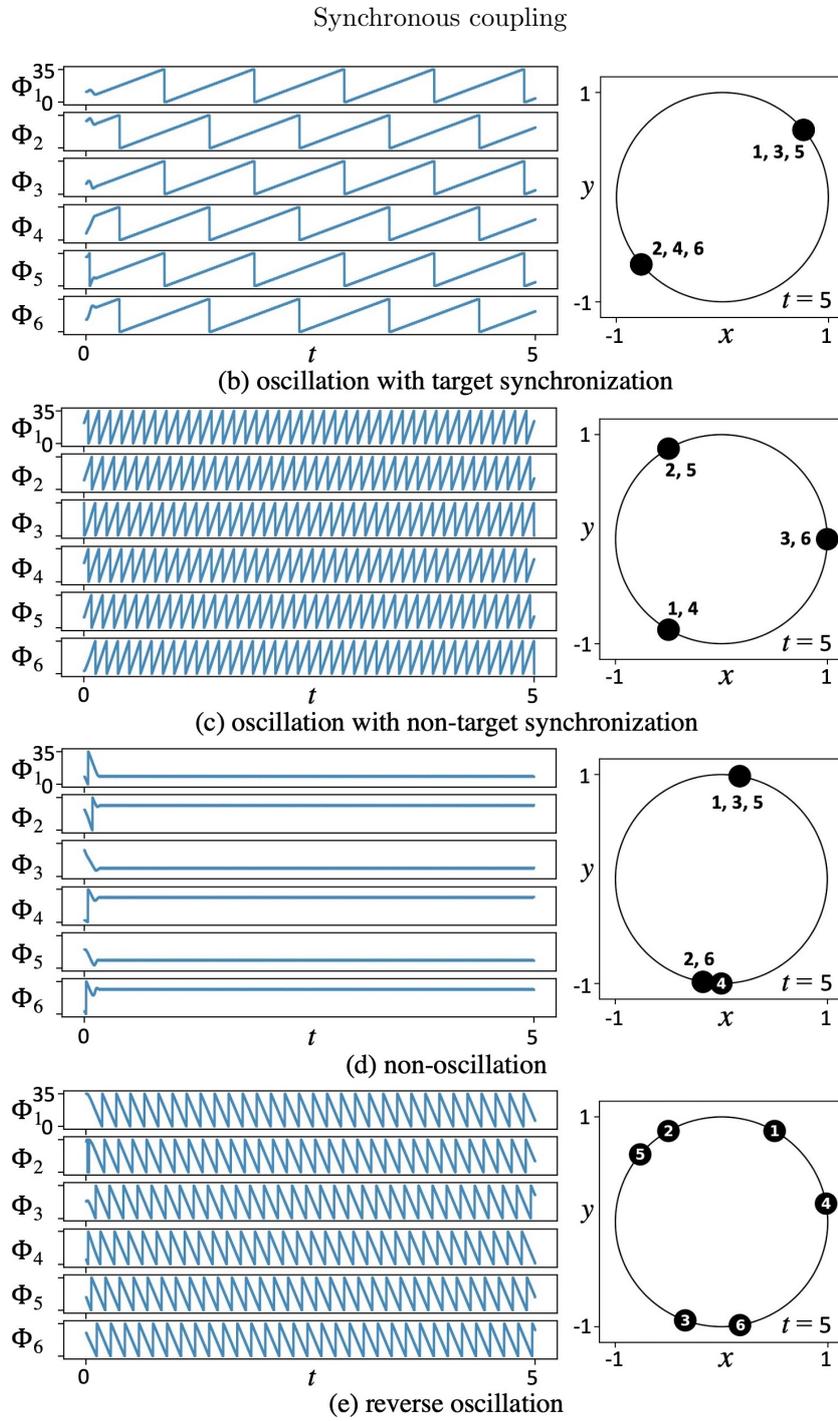


Figure 3.3.6: Continued.

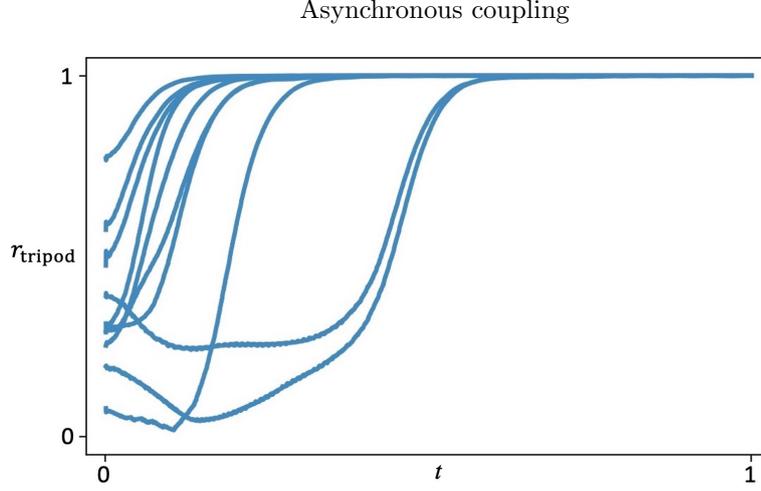


Figure 3.3.7: Instantaneous evaluation function  $r_{\text{tripod}}$  defined by Eq. (3.3.19) for ten simulation trials of asynchronously coupled CA phase oscillators from the same initial values as those chosen in Fig. 3.3.6. The parameter values are fixed as follows:  $N = 36$ ,  $M = 64$ ,  $\Gamma = 4.347 \times 10^{-3}$ ,  $T_i = 4.347 \times 10^{-4}$  for  $1 \leq i \leq 5$ , and  $T_6 = 1.872\pi \times 10^{-4}$ . The others are described in Eq. (3.3.17).

where  $K_i(t) \equiv \max\{k \in \mathbb{Z} \mid t_{i,k} \leq t\}$ ,  $\Phi_i(t_{i,k}) - \Phi_i(t_{i,k-1})$  is calculated on the circle,  $t_{i,k}$  represents the time when the  $k$ -th state transition of the discrete phase variable  $\Phi_i$  is occurred, and  $I_{i,k}$  is the following indicator function.

$$I_{i,k}(t) \equiv \begin{cases} 1 & \text{if } t \in [t_{i,k}, t_{i,k+1}), \\ 0 & \text{if } t \notin [t_{i,k}, t_{i,k+1}), \end{cases} \quad I_{i,k} : \mathbb{R} \rightarrow \{0, 1\}. \quad (3.3.21)$$

Fig. 3.3.6(a) shows the instantaneous evaluation functions  $r_{\text{tripod}}$  and the instantaneous average velocity  $v_{\text{ave}}$  for ten simulation trials of the synchronously coupled CA phase oscillators, where the initial values are randomly chosen. In these trials, the four types of behaviors are observed as follows.

- (i) *Oscillation with target synchronization pattern.* In Fig. 3.3.6(a), the values of the instantaneous evaluation functions  $r_{\text{tripod}}$  for the positive  $v_{\text{ave}}$  in the steady states are drawn by the blue lines. Among them, the blue lines approaching 1 in steady states mean that the coupled CA phase oscillators exhibit the synchronization pattern for the tripod gait properly. For example, Fig. 3.3.6(b) shows the time waveforms and Cartesian coordinate representations of the discrete phase variables  $\Phi_i$  in the trial indicated by the corresponding arrow in Fig. 3.3.6(a); the features of them are consistent with those shown in Figs. 3.3.5(d)–(h).
- (ii) *Oscillation with non-target synchronization pattern.* In Fig. 3.3.6(a), the values of the instantaneous evaluation function  $r_{\text{tripod}}$  drawn by the blue line that does not approach 1 in the steady state mean that the coupled CA phase oscillators do not exhibit the synchronization pattern for the tripod gait. For example, Fig. 3.3.6(c) shows the time waveforms and Cartesian coordinate representations of the discrete phase variables  $\Phi_i$  in the trial indicated by the corresponding arrow in Fig. 3.3.6(a).

As shown in this figure on the right, the CA phase oscillators exhibit three-phase synchronization, which is not the target synchronization pattern for the tripod gait.

- (iii) *Non-oscillation.* In Fig. 3.3.6(a), the values of the instantaneous evaluation functions  $r_{\text{tripod}}$  for  $v_{\text{ave}} \approx 0$  in the steady states are drawn by the orange lines. The orange lines approach 1 in the steady states; however, all the CA phase oscillators stop oscillating. In this case, the coupled CA phase oscillators do not achieve the synchronization pattern for the tripod gait. For example, Fig. 3.3.6(d) shows the time waveforms and the Cartesian coordinate representations of the discrete phase variables  $\Phi_i$  in the trial indicated by the corresponding arrow in Fig. 3.3.6(a). As shown in this figure, the whole CA phase oscillators do not oscillate in the steady state (left side).
- (iv) *Reverse oscillation.* In Fig. 3.3.6(a), the values of the instantaneous evaluation function  $r_{\text{tripod}}$  for the negative  $v_{\text{ave}}$  in the steady states are drawn by the green line. The green line does not approach 1 in the steady state; besides, all the CA phase oscillators are reversely oscillating. In this case, the coupled CA phase oscillators do not achieve the synchronization pattern for the tripod gait. For example, Fig. 3.3.6(e) shows the time waveforms and Cartesian coordinate representations of the discrete phase variables  $\Phi_i$  in the trial indicated by the corresponding arrow in Fig. 3.3.6(a). As shown on the left side in this figure, all the CA phase oscillators evolve with negative slopes in the steady state that means the CA phase oscillators are reversely oscillating.

**Remark 1 (advantage of asynchronous coupling in gait):** Fig. 3.3.7 shows the instantaneous evaluation functions  $r_{\text{tripod}}$  for ten simulation trials of the asynchronously coupled CA phase oscillators, where the parameter values (except for the parameter  $T_5$  for the period of the clock  $Clk_5$ ) and the initial values are the same as those chosen in Fig. 3.3.6(a). For all the trials, the coupled CA phase oscillators exhibit the synchronization pattern for the tripod gait where all the behaviors correspond to (i).

The characteristic above that the synchronously coupled CA phase oscillators may fail to synchronize is also observed in gait transitions; it is not observed in the asynchronously coupled CA phase oscillators. The result of the comparison on the gait transitions is shown below. For the wave gait, the following another instantaneous evaluation function to quantify the synchronization patterns of the coupled CA phase oscillators is introduced.

$$r_{\text{wave}}(t) \equiv \frac{1}{n} \left| \sum_{i=1}^n e^{j\left(2\pi\frac{\Phi_i(t)}{N} + \frac{\pi}{3}(i-1)\right)} \right|, \quad r_{\text{wave}} \in [0, 1]. \quad (3.3.22)$$

As with the one for the tripod gait in Eq. (3.3.19), the instantaneous evaluation function  $r_{\text{wave}}$  closes to 1 means that the coupled CA phase oscillators achieve the target synchronization pattern for the wave gait. Fig. 3.3.8(a) shows the time waveforms of the synchronously coupled CA phase oscillators, where the values of the parameter  $\phi$  is time variant as follows.

$$\begin{aligned} \phi = \pi/3 & \text{ for } 0 \leq t < 1.25 \text{ and } 2.5 \leq t < 3.75, & (\text{wave gait}) \\ \phi = \pi & \text{ for } 1.25 \leq t < 2.5 \text{ and } 3.75 \leq t \leq 5. & (\text{tripod gait}) \end{aligned} \quad (3.3.23)$$

Fig. 3.3.8(b) shows the instantaneous evaluation functions  $r_{\text{tripod}}$  (solid line) and  $r_{\text{wave}}$  (dashed line), and the instantaneous average velocity  $v_{\text{ave}}$  corresponding to Fig. 3.3.8(a).

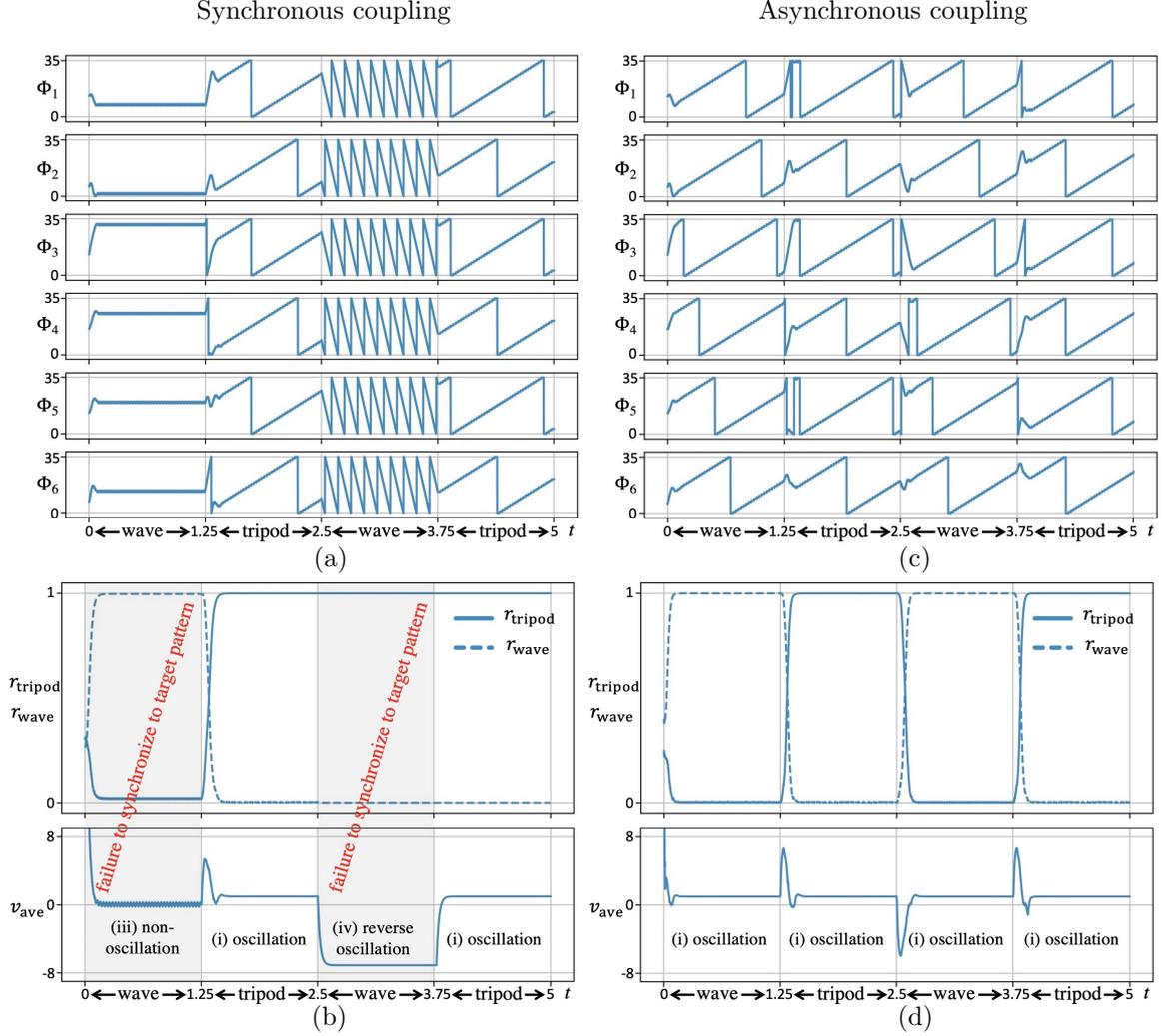


Figure 3.3.8: Gait transitions between wave gait and tripod gait. (a) Time waveforms of the synchronously coupled CA phase oscillators. The values of the parameter  $\phi$  is time variant as described in Eq. (3.3.23). The other parameter values are the same as those chosen in Fig. 3.3.6. (b) Instantaneous evaluation functions  $r_{\text{tripod}}$  (solid line) and  $r_{\text{wave}}$  (dashed line) defined by Eq. (3.3.38), and instantaneous average velocity  $v_{\text{ave}}$  defined by Eq. (3.3.20) of the synchronously coupled CA phase oscillators. (c) Time waveforms of the asynchronously coupled CA phase oscillators. The values of the parameter  $\phi_i$  is switched as described in Eq. (3.3.23). The other parameter values are the same as those chosen in Fig. 3.3.7. (d) Instantaneous evaluation functions  $r_{\text{tripod}}$  (solid line) and  $r_{\text{wave}}$  (dashed line), and instantaneous average velocity  $v_{\text{ave}}$  of the asynchronously coupled CA phase oscillators.

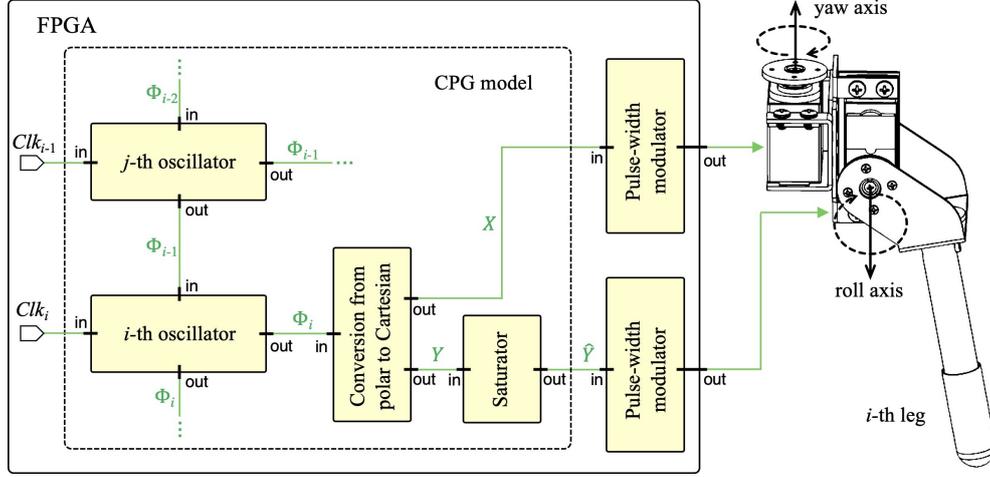


Figure 3.3.9: Schematic diagram of control strategy for  $i$ -th leg's servomotor. The blocks of  $i$ -th and  $j$ -th oscillators are identical with the circuit in Fig. 3.3.2(b).

For  $0 \leq t < 1.25$ , the synchronously coupled CA phase oscillators fail to synchronize to the target synchronization pattern, where the behavior is corresponding to (iii). For  $1.25 \leq t < 2.5$ , the synchronously coupled CA phase oscillators achieve the target synchronization pattern, where the behavior is corresponding to (i). For  $2.5 \leq t < 3.75$ , the synchronously coupled CA phase oscillators fail to synchronize to the target synchronization pattern, where the behavior is corresponding to (iv). For  $3.75 \leq t \leq 5$ , the synchronously coupled CA phase oscillators achieve the target synchronization pattern, where the behavior is corresponding to (i). On the other hand, Figs. 3.3.8(c) and (d) show the time waveforms, the instantaneous evaluation functions  $r_{\text{tripod}}$  and  $r_{\text{wave}}$ , and the instantaneous average velocity  $v_{\text{ave}}$  of the asynchronously coupled CA phase oscillators, where the parameter values (except for the parameter  $T_5$  for the period of the clock  $Clk_5$ ) and the initial values are the same as those chosen in Figs. 3.3.8(a) and (b). As shown in these figures, the asynchronously coupled oscillators achieve all the gait transitions between the tripod gait and the wave gait.

**Remark 2 (advantage of asynchronous coupling in gait transition).** Our extensive simulations reveal that the synchronously coupled CA phase oscillators often fail to realize the proper gait transitions. On the other hand, the asynchronously coupled CA phase oscillators mostly realize the proper gait transitions. Hence, the analyses in this study clarified that the asynchronously coupled CA phase oscillators are suitable to perform the smooth gait transition for the hexapod robot compared to the synchronously coupled CA phase oscillators. It should be noted that this study does not guarantee that the asynchronously coupled CA phase oscillator can always realize the target synchronization patterns. Hence, future work is needed to theoretically analyze the synchronization phenomena of the coupled CA phase oscillators.

### 3.3.4 FPGA Implementation

This section shows the hexapod robot mounted with an FPGA, in which the presented CPG model is implemented, can realize the wave gait, the tripod gait, and their transitions.

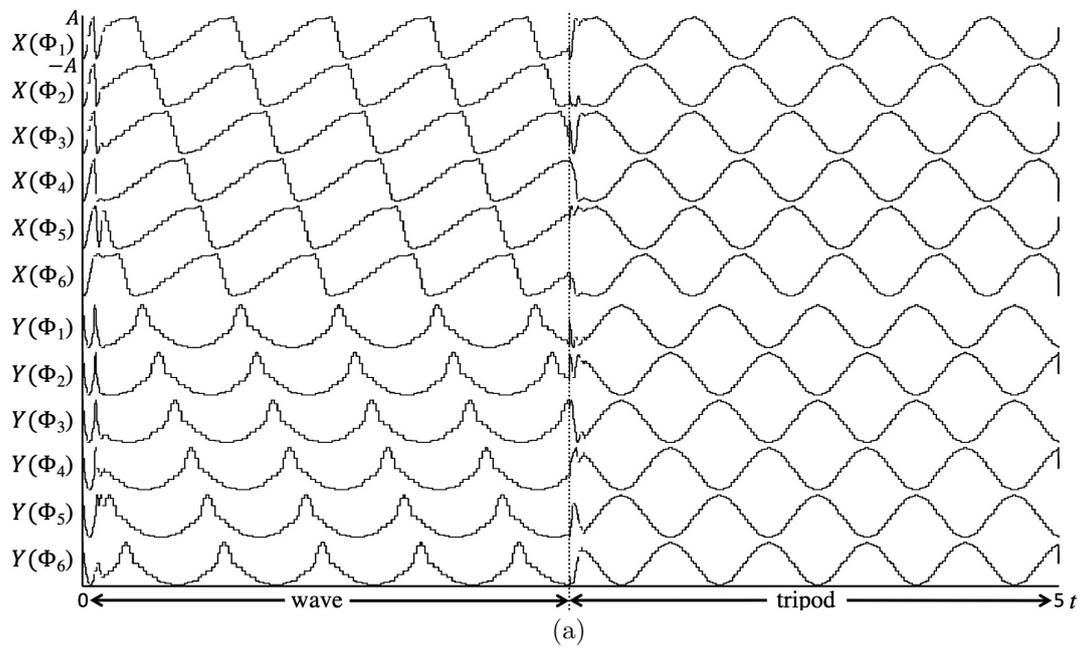
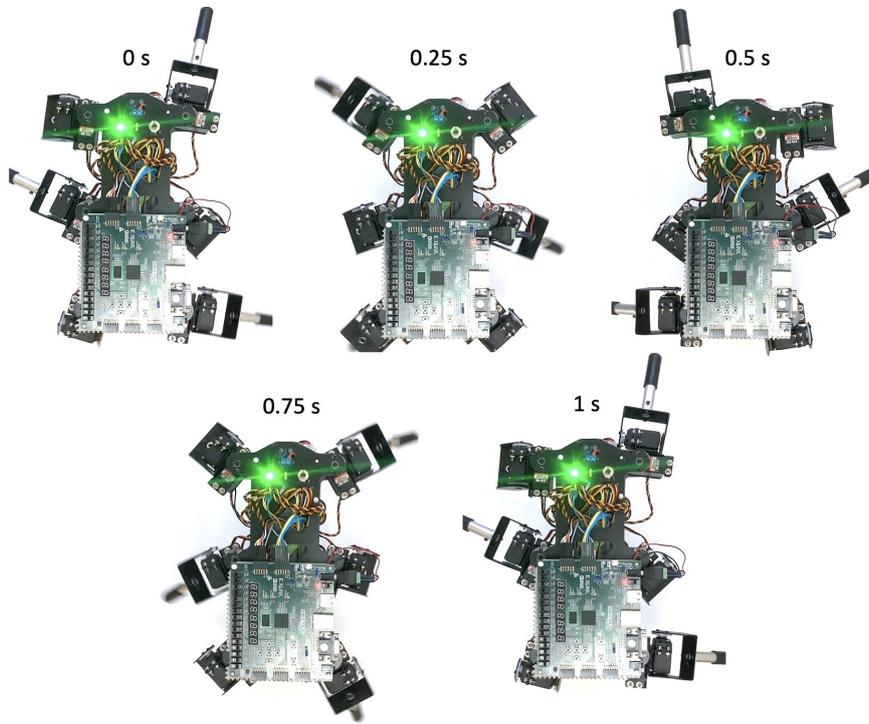
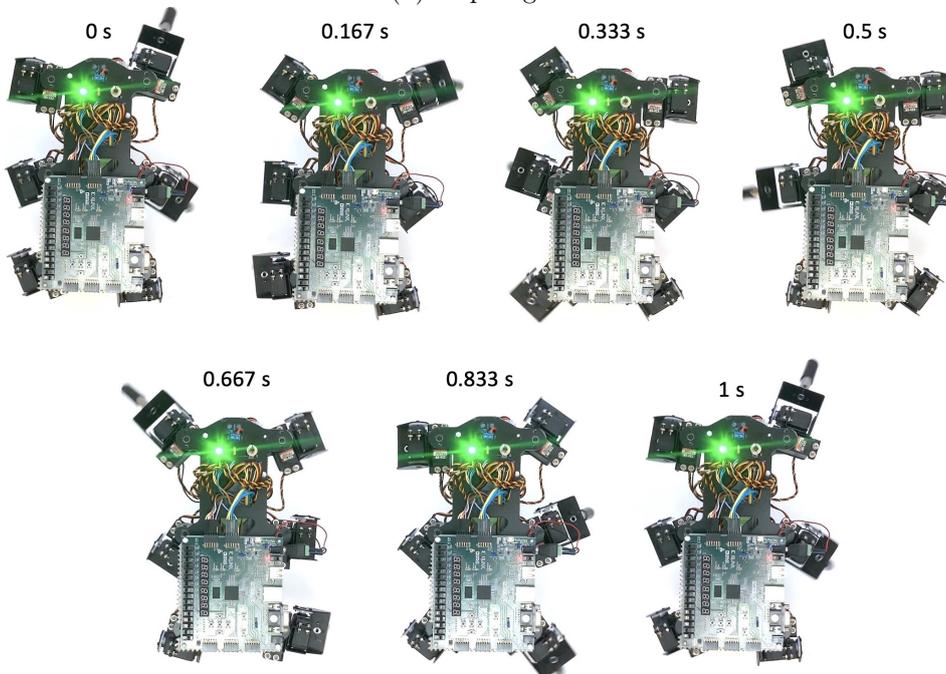


Figure 3.3.10: (a) Waveforms of signals  $X$  and  $Y$  simulated by Xilinx Vivado Design Suite v2018.2. (b) Snapshots of tripod gait. (c) Snapshots of wave gait. The parameter values are the same as those chosen in Fig. 3.3.7, where  $T_5$  is approximated as  $5.88 \times 10^{-4}$ . Also,  $\theta = \pi/3$  for the wave gait and  $\theta = \pi$  for the tripod gait.



(b) Tripod gait



(c) Wave gait

Figure 3.3.10: Continued.

Fig. 3.3.9 shows the block diagram of a control system for the  $i$ -th leg, where each leg has the two servomotors for the 2-degrees-of-freedom. As shown in this figure, the discrete phase variables  $\Phi_i$  are converted to Cartesian coordinate representations from polar coordinate representations as follows.

**Conversion from polar to Cartesian:**

$$\begin{aligned}
 X(\Phi) &\equiv \begin{cases} \lfloor A \cos(2\pi(\pi\Phi/\theta)/N) \rfloor & \text{if } \Phi < N\theta/2\pi, \\ \lfloor A \cos\left(2\pi\left(\frac{\Phi - N\theta/2\pi}{2 - \theta/\pi} + \frac{N}{2}\right)/N\right) \rfloor & \text{if } \Phi \geq N\theta/2\pi, \end{cases} & X : \mathbf{Z}_N^+ \rightarrow \mathbb{Z}, \\
 Y(\Phi) &\equiv \begin{cases} \lfloor A \sin(2\pi(\pi\Phi/\theta)/N) \rfloor & \text{if } \Phi < N\theta/2\pi, \\ \lfloor A \sin\left(2\pi\left(\frac{\Phi - N\theta/2\pi}{2 - \theta/\pi} + \frac{N}{2}\right)/N\right) \rfloor & \text{if } \Phi \geq N\theta/2\pi, \end{cases} & Y : \mathbf{Z}_N^+ \rightarrow \mathbb{Z},
 \end{aligned} \tag{3.3.24}$$

where  $A \in \mathbb{R}$  is a scaling parameter for a pulse-width modulation and these functions are implemented in LUTs. Fig. 3.3.10(a) shows the example of the signals  $X(\Phi_i(t))$  and  $Y(\Phi_i(t))$  simulated by Xilinx Vivado Design Suite v2018.2, an integrated design environment for synthesis and analysis of hardware description language (HDL) designs, where the detail of the design is described later. As shown in Fig. 3.3.9, the servomotors corresponding to yaw axes are controlled by pulse width-modulated signals of the signals  $X$ . Also, the servomotors corresponding to roll axes are controlled by pulse width-modulated signals  $\hat{Y}$  of the signals through the following a saturator function.

**Saturator:**

$$\hat{Y}(Y) \equiv \begin{cases} B & \text{if } Y \geq 0, \\ -B & \text{if } Y < 0, \end{cases} \quad \hat{Y} : \mathbb{Z} \rightarrow \{-B, B \mid B \in \mathbb{Z}\}, \tag{3.3.25}$$

where  $B \in \mathbb{Z}$  is a scaling parameter for the pulse-width modulation. The dynamics of the presented model are written as a register transfer level (RTL) code using VHDL as follows. The discrete variables  $\Phi_i$  and  $P_i$  are implemented by registers as  $N$ -bit and  $M$ -bit unsigned integers, where  $N = \lceil \log_2 N \rceil$  and  $M = \lceil \log_2 M \rceil$ , respectively. The function  $H$  is implemented in LUTs having a  $2(N+1)$ -bit signed integer input and a  $(M+1)$ -bit signed integer output in the two's complement format. The functions  $X$  and  $Y$  are respectively implemented in LUTs having a  $N$ -bit unsigned integer input and a  $L$ -bit unsigned integer output, where  $L$  is depending on the resolution of the pulse-width modulator. The state transitions in Eqs. (3.3.5) and (3.3.10) are written by sequential statements triggered by the clocks  $Clk_i$ . The VHDL code is synthesized by Xilinx Vivado Design Suite v2018.2 and a resulting bitstream file is downloaded into the Xilinx's FPGA, Artix-7 XC7A100T-1CSG324C [46], mounted on Digilent's Nexys 4 DDR evaluation platform [47]. Since the FPGA device does not support asynchronous triggering, the internal clocks  $Clk_i$  are generated by frequency-dividing a common clock with a high frequency (100 [MHz]). The FPGA device, in which the presented CPG model is implemented, is mounted on the hexapod robot, the Lynxmotion's MH2 hexapod robot [44], as shown in Fig. 3.3.2(a). Figs. 3.3.10(b) and (c) show snapshots of the hexapod robot controlled by the presented CPG model while performing the tripod gait and the wave gait. The laboratory experiments verified that the hexapod

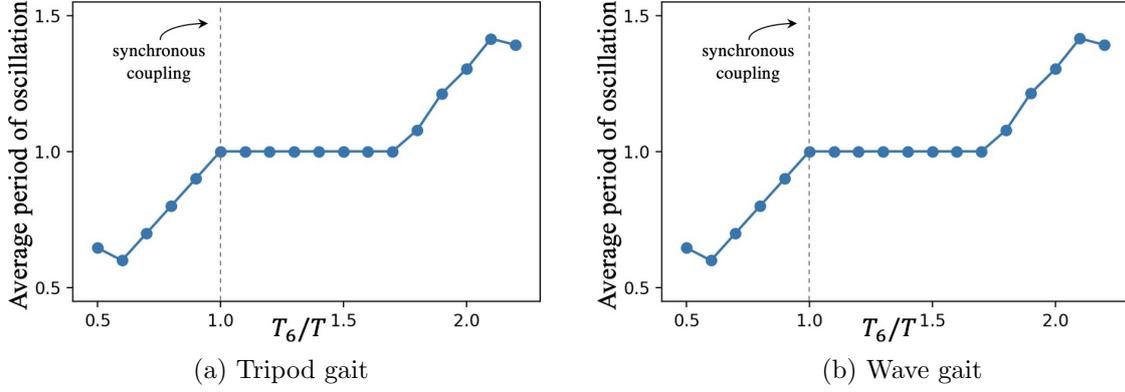


Figure 3.3.11: Oscillation periods of asynchronously coupled CA phase oscillators in steady states. (a) Tripod gait ( $\phi = \pi$ ). (b) Wave gait ( $\phi_i = \pi/3$ ). The other parameter values are the same as those chosen in Fig. 3.3.7, where  $T_i = T = 4.347 \times 10^{-4}$  for  $1 \leq i \leq 5$ .

robot can perform the gait transition between the wave gait, the tripod gait, and their transitions.

### 3.3.5 Discussion

#### 3.3.5.1 OSCILLATION PERIODS

In the case of the synchronously coupled CA phase oscillators, e.g.,  $T_i = T$  for all  $i$ , the oscillation period of each oscillator can be easily delivered. Assume

$$\begin{aligned} \Phi(0) &\in \Phi^* \equiv \{(\Phi_1, \dots, \Phi_n) \mid \Phi_i = \Phi_{i+1} + \lfloor N\phi/2\pi \rfloor \text{ for all } i\}, \\ \mathbf{P}(0) &\in \mathbf{P}^* \equiv \{(P_1, \dots, P_n) \mid P_i = P_j \text{ for all } i \text{ and } j\}, \end{aligned} \quad (3.3.26)$$

where  $\Phi(t) \equiv (\Phi_1(t), \dots, \Phi_n(t))$  and  $\mathbf{P}(t) \equiv (P_1(t), \dots, P_n(t))$ . This assumption means that the CA phase oscillators are synchronized to a target pattern determined by the parameter  $\phi$ . Under the assumption, from Eq. (3.3.8), return values of the function  $h$  are always  $h(\Delta\Phi_i^+) = 0$  for all  $i$ . Then, from Eq. (6), return values of the coupling function  $H$  are always  $H(\Delta\Phi_i^+) = M - 1$  for all  $i$ . Therefore, the oscillation period of each phase variable  $\Phi_i$  is obtained by

$$\text{Oscillation period} = TMN. \quad (3.3.27)$$

In the case of the asynchronously coupled CA phase oscillators, the oscillation period may change from  $TMN$  depending on the clock periods  $T_6$ , where  $T_i = T$  for  $1 \leq i \leq 5$ . Fig. 3.3.11 shows average periods of oscillators for the case of the asynchronously coupled CA phase oscillators. As shown in the figures (a) and (b), the average periods of oscillators are almost the same for the tripod gait and the wave gait.

#### 3.3.5.2 APPROPRIATE PARAMETERS OF $N$ AND $M$ FOR PRACTICAL USE

By choosing a desired oscillation period and a clock period  $T$ , the parameters  $N$  and  $M$  can be determined as  $MN = \text{Oscillation period}/T$ . Fig. 3.3.12 shows rates at which the orbits of the oscillators starting from randomly chosen initial values synchronize to the

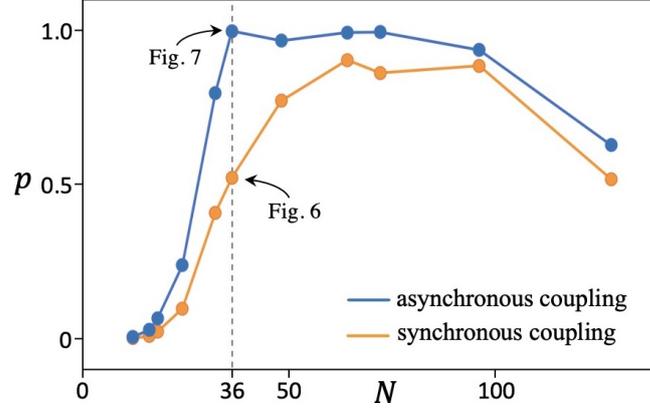


Figure 3.3.12: Rates at which orbits of oscillators starting from randomly chosen initial values synchronize to tripod gait pattern for  $N$ . Total number of trials = 1000 and  $M = \text{Oscillation period}/TN$ , where the oscillation period and  $T$  are chosen as 1.0015488 and  $4.347 \times 10^{-4}$ , respectively. The other parameter values are the same as those chosen in Fig. 3.3.6 and Fig. 3.3.7.

tripod gait pattern for  $N$ , where

$$p = \frac{\text{Number of trials synchronized to tripod gait pattern}}{\text{Total number of trials}}, \quad (3.3.28)$$

and  $N$  and  $M$  are assumed to be integers satisfying Eq. (3.3.27). A large value of the rate  $p$  correspond to a large attraction basin of the target gait pattern. Therefore, for practical use, the values of the parameters  $N$  and  $M$  should be chosen such that  $p \simeq 1$  (e.g.,  $N = 36$  and  $M = 64$  as shown in Fig. 3.3.12).

### 3.3.5.3 REALIZABLE PHASE LOCKED PATTERNS

Further, under the assumption that  $\mathbf{P} \in \mathbf{P}^*$ , realizable phase locked patterns in the synchronously coupled CA phase oscillators ( $T_i = T_j$  for all  $i$  and  $j$ ) can be analyzed in a similar way in [62]. From Eq. (3.3.8), the phases  $\Phi_i$  are locked if  $\mathbf{P} \in \mathbf{P}^*$  and

$$\sin(2\pi\Delta\Phi_i^+/N) = \sin(2\pi\Delta\Phi_{i+1}^+/N) \quad \text{for all } i, \quad (3.3.29)$$

where  $\phi = 0$ . Since  $\Delta\Phi_i^+$  is defined on  $\{0, \dots, N-1\}$ , Eq. (3.3.29) can be written in the form

$$(\Delta\Phi_i^+ - \Delta\Phi_{i+1}^+)(\Delta\Phi_i^+ - N/2 + \Delta\Phi_{i+1}^+) = 0 \quad \text{for all } i. \quad (3.3.30)$$

Every combination of  $\Delta\Phi_i^+$  and  $\Delta\Phi_{i+1}^+$  that makes either the first or the second factor equal to zero corresponds to a realizable phase locked pattern. If  $\Delta\Phi_i^+$  in Eq. (3.3.30) is assumed to be  $\lambda \in \mathbf{Z}_N^+$ , then  $\Delta\Phi_{i+1}^+$  in Eq. (3.3.30) should be  $\lambda$  or  $N/2 - \lambda$ . On the other hand, for all the phase differences  $\Delta\Phi_i^+$ , the following equation always holds.

$$\sum_i^N \Delta\Phi_i^+ = Nk, \quad (3.3.31)$$

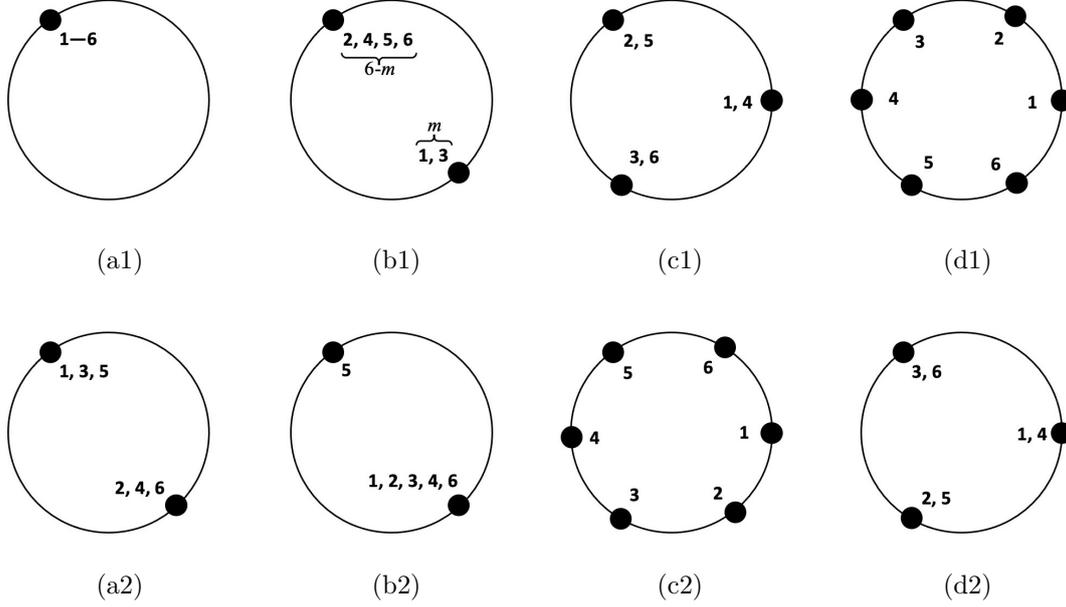


Figure 3.3.13: (a1) In-phase synchronized solution. (b1) Elementary solution. (c1) and (d1) Traveling wave solution.  $\phi = 0$ . (a2) Configuration of (a1) transformed by Eq. (3.3.41). (b2) Configuration of (b1) transformed by Eq. (3.3.41). (c2), (d2) Configurations of (c2) and (d2) transformed by Eq. (3.3.41).

where  $k \in \{0, \dots, n-1\}$ . Therefore, we can take  $m\lambda$  and  $(n-m)(N/2 - \lambda)$  such that Eq. (3.3.31) is satisfied as follows.

$$m\lambda + (n-m)(N/2 - \lambda) = Nk, \quad (3.3.32)$$

where  $m \leq n$ . It follows that every vector  $\Delta\Phi^+ \equiv (\Delta\Phi_1^+, \dots, \Delta\Phi_n^+)$  that is a permutation of the following vector corresponds to a realizable phase locked pattern.

$$\Delta\Phi^+ = (\lambda_1, \dots, \lambda_m, N/2 - \lambda_{m+1}, \dots, N/2 - \lambda_n), \quad (3.3.33)$$

where  $\lambda_i = \lambda_j$  for all  $i$  and  $j$ . A typical realizable phase locked pattern, described by the following vector, is called an *in-phase synchronized solution*.

$$\Delta\Phi^+ = (0, \dots, 0), \quad (3.3.34)$$

where an example configuration of the CA phase oscillators for the in-phase synchronized solution is shown in Fig. 3.3.13(a1). If  $N = \text{even number}$ , then the following phase locked pattern called an *elementary solution* [62] is realizable.

$$\Delta\Phi^+ = (\underbrace{0, \dots, 0}_m, \underbrace{N/2, \dots, N/2}_{n-m}), \quad (3.3.35)$$

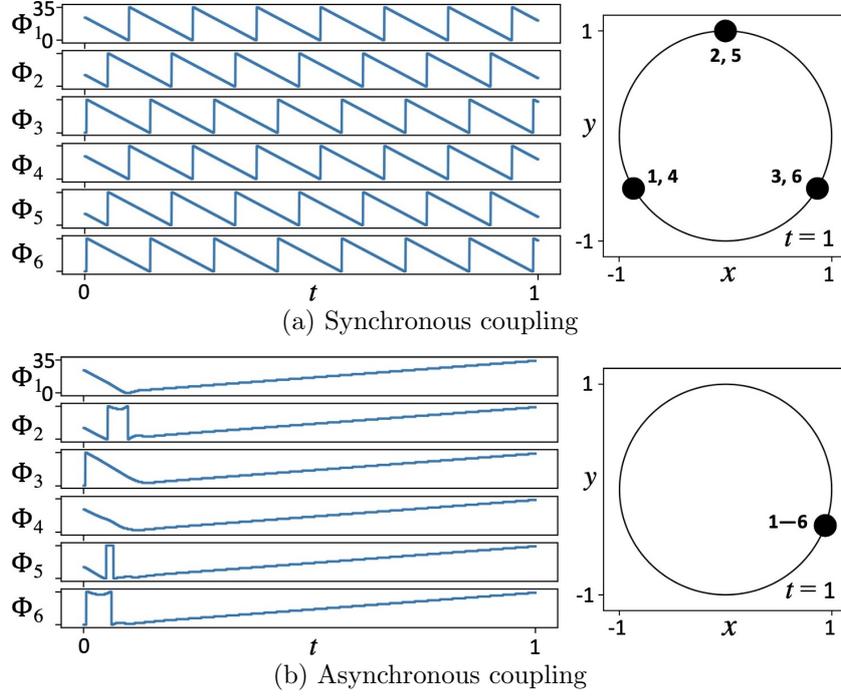


Figure 3.3.14: Steady-state solutions for synchronously and asynchronously coupled CA phase oscillators. The initial values of both the oscillators are  $\Phi(0) = (24, 12, 0, 24, 12, 0)$ ,  $\mathbf{P}(0) = (0, \dots, 0)$ . (a) Traveling wave solution with reverse oscillation for synchronously coupled CA phase oscillators.  $\phi = 0$ . The other parameter values are the same as those chosen in Fig. 3.3.6. (b) In-phase synchronized solution for asynchronously coupled CA phase oscillators.  $\phi_i = 0$ . The other parameter values are the same as those chosen in Fig. 3.3.7.

where  $m \neq n$ , and an example configuration of the CA phase oscillators for the elementary solution is shown in Fig. 3.3.13(b1). Note that the every vector  $\Delta\Phi^+$  that is a permutation of the vector Eq. (3.3.35) is the elementary solution. Further, if  $N \bmod n = 0$ , then the following phase locked pattern called a *traveling wave solution* [62] is realizable.

$$\Delta\Phi^+ = (Nk/n, \dots, Nk/n), \quad (3.3.36)$$

where  $k \in \{1, \dots, n-1\}$ , and example configurations of the phase CA oscillators for the elementary solution are shown in Figs. 3.3.13(c1) and (d1).

However, as demonstrated in Section 3.2, the synchronously coupled CA phase oscillators may not oscillate properly even if they are synchronized. For example, when  $\Delta\Phi^+$  satisfies

$$\Delta\Phi^+ = (\lambda_-, \dots, \lambda_-) \text{ such that } H(\lambda_-) < 0, \quad (3.3.37)$$

the synchronously coupled CA phase oscillators oscillate reversely in the phase locked pattern, where  $\mathbf{P} \in \mathbf{P}^*$ . In this case, the oscillation period is determined by  $T(1 + H(\lambda_-))N$ . Fig. 3.3.14(a) shows an example of the reverse oscillation for the synchronously coupled CA phase oscillators, where  $\Delta\Phi^+ = (24, \dots, 24)$ ,  $T = 4.347 \times 10^{-4}$ ,  $H(24) = -8$ , and  $N = 36$ , and thus the oscillation period is 0.1408428. On the other hand, Fig. 3.3.14(b) shows the

in-phase synchronized solution for the asynchronously coupled CA phase oscillators, where the initial values are the same as those chosen in Fig. 3.3.14(a). As shown in this figure, the asynchronously coupled CA phase oscillators do not have the traveling wave solution  $\Delta\Phi^+ = (24, \dots, 24)$ . Our extensive analyses have revealed that the asynchronously coupled CA phase oscillators rarely have the reverse oscillation. Analysis of detailed occurrence mechanism of the reverse oscillation is an important future problem.

#### 3.3.5.4 STABILITY AGAINST PERTURBATIONS

In order to quantify the in-phase synchronized solution, the following instantaneous evaluation function is introduced.

$$r_{\text{inphase}}(t) \equiv \frac{1}{n} \left| \sum_{i=1}^n e^{j(2\pi \frac{\Phi_i(t)}{N})} \right|, \quad r_{\text{inphase}} \in [0, 1]. \quad (3.3.38)$$

The instantaneous evaluation function  $r_{\text{inphase}} = 1$  means the coupled CA oscillators in the in-phase synchronized solution shown in Fig. 3.3.13(a1). Fig. 3.3.15(a) shows the instantaneous evaluation functions  $r_{\text{inphase}}$  for the synchronously and asynchronously coupled CA phase oscillators for

$$\phi = 0, \quad (3.3.39)$$

where the initial values are set on the in-phase synchronized solution, the elementary solution, or the traveling wave solution shown in Figs. 3.3.13(a1)–(d1). As shown in this figure, for the asynchronously coupled CA phase oscillators, the in-phase synchronized solution is dominant.

Let us consider the equivalence of the dynamics of the coupled CA phase oscillators for the in-phase synchronized solution and the tripod gait pattern. Note that the coupling function defined by Eq. (6) is basically characterized by Eqs. (3.3.8) and (3.3.9). Eq. (3.3.9) for the in-phase synchronized solution (see also Eq. (3.3.39)) is as follows.

$$\Delta\Phi_i^+|_{\phi=0} = \Phi_{i+1}(t) - \Phi_i(t) \pmod{N}. \quad (3.3.40)$$

Let us introduce the following transformation of the phase variable (see also Figs. 3.3.13(a2)–(d2)).

$$\Phi'_i \equiv \begin{cases} \Phi_i & \text{for } i = 1, 3, 5, \\ \Phi_i - \lfloor N/2 \rfloor \pmod{N} & \text{for } i = 2, 4, 6. \end{cases} \quad (3.3.41)$$

Eq. (3.3.9) for the tripod gait pattern (see also Eq. (3.3.17)) can be transformed by Eq. (3.3.41) into

$$\Delta\Phi_i^+|_{\phi=\pi} = \Phi'_{i+1}(t) - \Phi'_i(t) \pmod{N}. \quad (3.3.42)$$

Therefore, the tripod gait pattern can be considered as the special case of the in-phase synchronized solution via the change of variable. In fact, Fig. 3.3.15(a) and (b) shows examples of this equivalence. Fig. 3.3.15(b) shows the instantaneous evaluation functions  $r_{\text{tripod}}$  defined by Eq. (3.3.19) for the synchronously and asynchronously coupled CA phase oscillators for  $\phi = \pi$ , where the initial values in Fig. 3.3.15(a) are transformed by Eq. (3.3.41) for the initial values in Fig. 3.3.15(b). As shown in these figures, the coupled CA phase oscillators has the same structure of attraction basins.

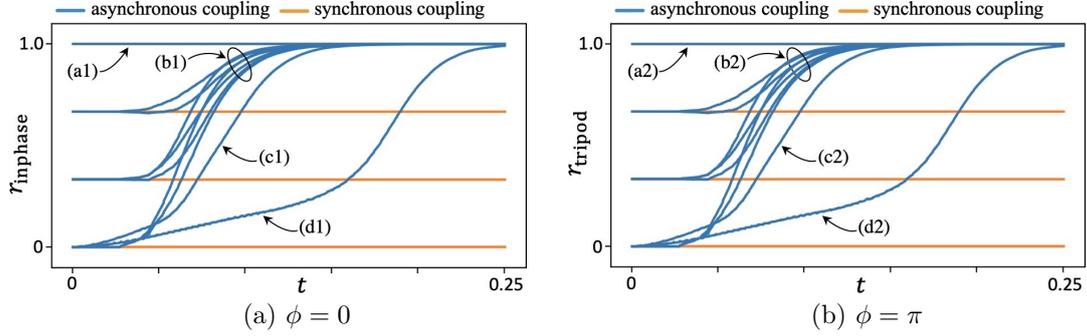


Figure 3.3.15: (a) Instantaneous evaluation functions  $r_{\text{inphase}}$  for synchronously and asynchronously coupled CA phase oscillators. The initial values are set on the in-phase synchronized solution, the elementary solution, or the traveling wave solution, where (a1)–(d1) in this figure correspond to (a1)–(d1) in Fig. 3.3.13 as follows:  $\Phi(0) = (0, 0, 0, 0, 0, 0)$ ,  $(0, 0, 0, 0, 0, 18)$ ,  $(0, 0, 0, 0, 18, 18)$ ,  $(0, 0, 0, 18, 18, 18)$ ,  $(0, 0, 18, 18, 18, 18)$ ,  $(0, 0, 18, 0, 0, 18)$ ,  $(0, 0, 18, 18, 0, 18)$ ,  $(0, 18, 18, 18, 18, 18)$ ,  $(0, 6, 12, 18, 24, 30)$ ,  $(24, 12, 0, 24, 12, 0)$ , and  $P(0) \in P^*$ .  $\phi = 0$  and the other parameter values are the same as those chosen in Figs. 3.3.6 and 3.3.7. (b) Instantaneous evaluation functions  $r_{\text{tripod}}$  for synchronously and asynchronously coupled CA phase oscillators. The initial values  $\Phi(0)$  in (a) transformed by Eq. (3.3.41) and the resulting values  $(0, 18, 0, 18, 0, 18)$ ,  $(0, 18, 0, 18, 0, 0)$ ,  $(0, 18, 0, 18, 18, 0)$ ,  $(0, 18, 0, 0, 18, 0)$ ,  $(0, 18, 18, 0, 18, 0)$ ,  $(0, 18, 18, 18, 0, 0)$ ,  $(0, 18, 18, 0, 0, 0)$ ,  $(0, 0, 18, 0, 18, 0)$ ,  $(0, 24, 12, 0, 24, 12)$ ,  $(24, 30, 0, 6, 12, 18)$  are used as the initial values.  $P(0) \in P^*$ .  $\phi = \pi$  and the other parameter values are the same as those chosen in Figs. 3.3.6 and 3.3.7.

Further, Fig. 3.3.16 shows the instantaneous evaluation functions  $r_{\text{inphase}}$  for the synchronously and asynchronously coupled CA phase oscillators with random perturbations. As shown in Fig. 3.3.16 (a), for the synchronously coupled CA phase oscillators, the in-phase synchronized solution has stability against small perturbations. On the other hand, as shown in Fig. 3.3.16 (b), for the asynchronously coupled CA phase oscillators, the in-phase synchronized solution has stability against small and large perturbations.

### 3.3.5.5 FUTURE DEVELOPMENT OF IMPLEMENTATION METHOD FOR THE COUPLING FUNCTIONS

In the implementation of the presented CPG model in Section 4, the coupling function  $H$  is directly implemented in LUTs. For further reduction of hardware resources, the COordinate Rotational DIgital Computer (CORDIC) algorithm [63] might be applied to implement the coupling function. In addition, if the coupling function  $H$  is simplified to three-valued function as proposed in [58], the presented CPG model is considered to be implemented by fewer hardware resources. However, in this case, careful consideration should be given to synchronous speed for practical use.

## 3.3.6 Conclusions

In this paper, the CPG model based on the asynchronous coupling of CA phase oscillators for the hexapod robot was presented. Analyses using the evaluation functions for the target gait patterns of the hexapod robot clarified that the asynchronously coupled CA phase oscillators are more suitable to perform gait transitions than the synchronously

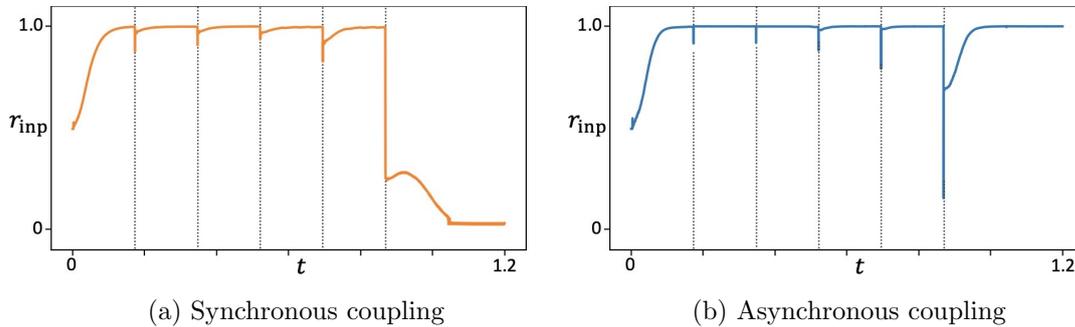


Figure 3.3.16: Instantaneous evaluation functions  $r_{\text{inphase}}$  for synchronously and asynchronously coupled CA phase oscillators with random small perturbations and large perturbations to the phases  $\Phi_i$ . (a) Synchronous coupling.  $\phi = 0$  and the other parameter values are the same as those chosen in Fig. 3.3.6. (b) Asynchronous coupling.  $\phi = 0$  and the other parameter values are the same as those chosen in Fig. 3.3.7.

coupled CA phase oscillators. For example, as described in Remarks 1 and 2, the asynchronously coupled CA phase oscillators mostly realize the proper gaits and their transitions while the synchronously coupled CA phase oscillators often fail to realize them. Then, the presented CPG model was implemented in the FPGA. It was verified that the hexapod robot mounted with the FPGA can perform the smooth gait transitions between target gait patterns. Our future work includes: (i) Systematic analyses (e.g., group theoretic approach) of various phase locked patterns [64, 65] and (ii) Investigation of how the pattern transitions will be affected by incorporating external perturbations into the proposed CPG model such as friction and floor reaction forces [66].

## Bibliography

- [1] E. Kandel, J. Schwartz, and T. Jessell, *Principles of Neural Science*, 4th ed. New York: MacGraw-Hill, 2000.
- [2] X. Li and L. Li, “Efficient implementation of FPGA based central pattern generator using distributed arithmetic,” *IEICE Electronics Express*, vol. 8, no. 21, pp. 1848–1854, 2011.
- [3] J. Yu, M. Tan, J. Chen, and J. Zhang, “A Survey on CPG-Inspired Control Models and System Implementation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 441–456, 2014.
- [4] L. Righetti and A. J. Ijspeert, “Pattern generators with sensory feedback for the control of quadruped locomotion,” in *Proc. of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 2008, pp. 819–824.
- [5] J. H. Barron-Zambrano and C. Torres-Huitzil, “FPGA implementation of a configurable neuromorphic CPG-based locomotion controller,” *Neural Networks*, vol. 45, pp. 50–61, 2013.

- [6] Z. Wang, Q. Gao, and H. Zhao, “CPG-Inspired Locomotion Control for a Snake Robot Basing on Nonlinear Oscillators,” *Journal of Intelligent & Robotic Systems*, vol. 85, no. 2, pp. 209–227, 2017.
- [7] A. Crespi, A. Badertscher, A. Guignard, and A. J. Ijspeert, “AmphiBot I: An amphibious snake-like robot,” *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 163–175, 2005.
- [8] T. Sato, W. Watanabe, and A. Ishiguro, “An adaptive decentralized control of a serpentine robot based on the discrepancy between body, brain and environment,” in *Proc. of the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010)*, Anchorage, AK, 2010, pp. 709–714.
- [9] K. Takeda and H. Torikai, “A Novel Hardware-Efficient Cochlea Model Based on Asynchronous Cellular Automaton Dynamics: Theoretical Analysis and FPGA Implementation,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 9, pp. 1107–1111, 2017.
- [10] T. Matsubara and H. Torikai, “Asynchronous Cellular Automaton-Based Neuron: Theoretical Analysis and On-FPGA Learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 736–748, 2013.
- [11] Y. A. Kuznetsov, *Elements of Applied Bifurcation Theory*, 3rd ed., ser. Applied Mathematical Sciences, S. S. Antman, J. E. Marsden, and L. Sirovich, Eds. New York: Springer, 2004, vol. 112.
- [12] S. Grillner, “Neurobiological Bases of Rhythmic Motor Acts in Vertebrates,” *Science*, vol. 228, no. 4696, pp. 143–149, 1985.
- [13] A. J. Ijspeert, “Central pattern generators for locomotion control in animals and robots: A review,” *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.
- [14] R. R. Torrealba, J. Cappelletto, L. Fermín, G. Fernández-López, and J. C. Grieco, “Cybernetic knee prosthesis: Application of an adaptive central pattern generator,” *Kybernetes*, vol. 41, no. 1/2, pp. 192–205, 2012.
- [15] R. J. Vogelstein, F. V. G. Tenore, L. Guevremont, R. Etienne-Cummings, and V. K. Mushahwar, “A Silicon Central Pattern Generator Controls Locomotion in Vivo,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 2, no. 3, pp. 212–222, 2008.
- [16] N. Sasagawa, K. Tani, T. Imamura, and Y. Maeda, “Quadruped locomotion patterns generated by desymmetrization of symmetric central pattern generator hardware network,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E101.A, no. 10, pp. 1658–1667, 2018.
- [17] I. Köymen and E. M. Drakakis, “Current-input current-output analog half center oscillator and central pattern generator circuits with memristors: Analog HCO and CPG circuits with memristors,” *International Journal of Circuit Theory and Applications*, vol. 46, no. 7, pp. 1294–1310, 2018.

- [18] F. Li, A. Basu, C. H. Chang, and A. H. Cohen, "Dynamical Systems Guided Design and Analysis of Silicon Oscillators for Central Pattern Generators," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 12, pp. 3046–3059, 2012.
- [19] H. Soleimani, A. Ahmadi, M. Bavandpour, and O. Sharifipour, "A generalized analog implementation of piecewise linear neuron models using CCII building blocks," *Neural Networks*, vol. 51, pp. 26–38, 2014.
- [20] L. Minati, M. Frasca, N. Yoshimura, and Y. Koike, "Versatile Locomotion Control of a Hexapod Robot Using a Hierarchical Network of Nonlinear Oscillator Circuits," *IEEE Access*, vol. 6, pp. 8042–8065, 2018.
- [21] N. Korkmaz, İ. Öztürk, and R. Kılıç, "The investigation of chemical coupling in a HR neuron model with reconfigurable implementations," *Nonlinear Dynamics*, vol. 86, no. 3, pp. 1841–1854, 2016.
- [22] Y. Maeda, T. Hiramatsu, S. Miyoshi, and H. Hikawa, "Pulse coupled oscillator with learning capability using simultaneous perturbation and its FPAA implementation," in *Proc. of the ICROS-SICE International Joint Conference 2009*, Fukuoka, Japan, 2009, pp. 3142–3145.
- [23] C. Mayr, J. Partzsch, M. Noack, S. Hanzsche, S. Scholze, S. Hoppner, G. Ellguth, and R. Schuffny, "A Biological-Realtime Neuromorphic System in 28 nm CMOS Using Low-Leakage Switched Capacitor Circuits," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 1, pp. 243–254, 2016.
- [24] X. Li, H. Liu, X. Wu, R. Li, and X. Wang, "Improved CPG Model Based on Hopf Oscillator for Gait Design of a New Type of Hexapod Robot," in *Intelligent Robotics and Applications*, H. Yu, J. Liu, L. Liu, Z. Ju, Y. Liu, and D. Zhou, Eds. Cham: Springer International Publishing, 2019, vol. 11741, pp. 72–83.
- [25] B. Xu, W. Li, Y. Ni, H. Chi, and W. Ouyang, "Adaptive Locomotion Generation for a Bionic Hexapod Robot," in *Proc. of the 45th Annual Conference of the IEEE Industrial Electronics Society*, Lisbon, Portugal, 2019, pp. 5173–5178.
- [26] B. Zhong, S. Zhang, Mv Xu, Y. Zhou, T. Fang, and W. Li, "On a CPG-Based Hexapod Robot: AmphiHex-II With Variable Stiffness Legs," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 542–551, 2018.
- [27] E. I. Guerra-Hernandez, A. Espinal, P. Batres-Mendoza, C. H. Garcia-Capulin, R. De J. Romero-Troncoso, and H. Rostro-Gonzalez, "A FPGA-Based Neuromorphic Locomotion System for Multi-Legged Robots," *IEEE Access*, vol. 5, pp. 8301–8312, 2017.
- [28] A. Zahedi, S. Haghiri, and M. Hayati, "Multiplierless Digital Implementation of Time-Varying FitzHugh–Nagumo Model," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 7, pp. 2662–2670, 2019.
- [29] K Takeda and H. Torikai, "A Novel Asynchronous CA Neuron Model: Design of Neuron-Like Nonlinear Responses Based on Novel Bifurcation Theory of Asynchronous

- Sequential Logic Circuit,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 6, pp. 1989–2001, 2020.
- [30] T. Matsubara, H. Torikai, and T. Hishiki, “A Generalized Rotate-and-Fire Digital Spiking Neuron Model and Its On-FPGA Learning,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 10, pp. 677–681, 2011.
- [31] T. Hishiki and H. Torikai, “A Novel Rotate-and-Fire Digital Spiking Neuron and its Neuron-Like Bifurcations and Responses,” *IEEE Transactions on Neural Networks*, vol. 22, no. 5, pp. 752–767, 2011.
- [32] H. Ishimoto, M. Izawa, and H. Torikai, “A novel cochlea partition model based on asynchronous bifurcation processor,” *Nonlinear Theory and Its Applications, IEICE*, vol. 6, no. 2, pp. 207–225, 2015.
- [33] M. Izawa and H. Torikai, “Asynchronous Cellular Automaton Model of Spiral Ganglion Cell in the Mammalian Cochlea: Theoretical Analyses and FPGA Implementation,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E98.A, no. 2, pp. 684–699, 2015.
- [34] T. Noguchi and H. Torikai, “Ghost Stochastic Resonance From an Asynchronous Cellular Automaton Neuron Model,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 2, pp. 111–115, 2013.
- [35] T. Yoshimoto and H. Torikai, “A novel hardware-efficient gene network model based on asynchronous cellular automaton dynamics,” *Nonlinear Theory and Its Applications, IEICE*, vol. 8, no. 4, pp. 302–318, 2017.
- [36] T. Naka and H. Torikai, “A Novel Generalized Hardware-Efficient Neuron Model Based on Asynchronous CA Dynamics and Its Biologically Plausible On-FPGA Learnings,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 7, pp. 1247–1251, 2019.
- [37] T. Matsubara and H. Torikai, “An Asynchronous Recurrent Network of Cellular Automaton-Based Neurons and Its Reproduction of Spiking Neural Network Activities,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 836–852, 2016.
- [38] K. Isobe and H. Torikai, “A Novel Hardware-Efficient Asynchronous Cellular Automaton Model of Spike-Timing-Dependent Synaptic Plasticity,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 6, pp. 603–607, 2016.
- [39] N. Shimada and H. Torikai, “A Novel Asynchronous Cellular Automaton Multicompartment Neuron Model,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 776–780, 2015.
- [40] K. Takeda and H. Torikai, “A novel hardware-efficient CPG model for a hexapod robot based on nonlinear dynamics of coupled asynchronous cellular automaton oscillators,” in *Proc. of the 2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, 2019, pp. 1–8.

- [41] ———, “A novel hardware-efficient CPG model based on asynchronous cellular automaton,” *IEICE Electronics Express*, vol. 15, no. 11, pp. 1–11, 2018.
- [42] M. Bavandpour, H. Soleimani, S. Bagheri-Shouraki, A. Ahmadi, D. Abbott, and L. O. Chua, “Cellular Memristive Dynamical Systems (CMDs),” *International Journal of Bifurcation and Chaos*, vol. 24, no. 05, p. 1430016, 2014.
- [43] M. Gholami and S. Saeedi, “Digital Cellular Implementation of Morris-Lecar Neuron Model,” in *Proc. of the 23rd Iranian Conference on Electrical Engineering*, Tehran, Iran, 2015, pp. 1235–1239.
- [44] Lynxmotion, *Mini-Hex Combo Kit for FlowBotics Studio (MH2F)*. Accessed: Jul. 30, 2020. [Online]. Available: <http://www.lynxmotion.com/>.
- [45] D. M. Wilson, “Insect walking,” *Annual Review of Entomology*, vol. 11, no. 1, pp. 103–122, 1966.
- [46] Xilinx Inc., *Xilinx’s FPGA Artix-7 XC7A100T-1CSG324C*. Accessed: Jul. 30, 2020. [Online]. Available: <http://www.xilinx.com/>.
- [47] Digilent Inc., *Digilent’s Nexys 4 DDR Evaluation Platform*. Accessed: Jul. 30, 2020. [Online]. Available: <http://www.digilentinc.com/>.
- [48] P. B. Minev and V. S. Kukenska, “Implementation of soft-core processors in FPGAs,” in *Proc. of the International Scientific Conferences*, Gabrovo, Bulgaria, 2007, pp. 1–4.
- [49] R. Bani-Hani, S. Harb, K. Mhaidat, and E. Taqieddin, “High-Throughput and Area-Efficient FPGA Implementations of Data Encryption Standard (DES),” *Circuits and Systems*, vol. 05, no. 03, pp. 45–56, 2014.
- [50] M. A. Mohsin and D. G. Perera, “An FPGA-Based Hardware Accelerator for K-Nearest Neighbor Classification for Machine Learning on Mobile Devices,” in *Proc. of the 9th International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies*, Toronto, Canada, 2018, pp. 1–7.
- [51] H. Yu, H. Gao, L. Ding, M. Li, Z. Deng, and G. Liu, “Gait Generation With Smooth Transition Using CPG-Based Locomotion Control for Hexapod Walking Robot,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 9, pp. 5488–5500, 2016.
- [52] W. Chen, G. Ren, J. Zhang, and J. Wang, “Smooth transition between different gaits of a hexapod robot via a central pattern generators algorithm,” *Journal of Intelligent & Robotic Systems*, vol. 67, no. 3-4, pp. 255–270, 2012.
- [53] A. Crespi and A. J. Ijspeert, “AmphiBot II: An Amphibious Snake Robot that Crawls and Swims using a Central Pattern Generator,” in *Proc. of the 9th International Conference on Climbing and Walking Robots*, Brussels, Belgium, 2006, pp. 19–27.
- [54] K. Nakada, T. Asai, and Y. Amemiya, “An analog CMOS central pattern generator for interlimb coordination in quadruped locomotion,” *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1356–1365, 2003.

- [55] K. Saito, A. Matsuda, K. Saeki, F. Uchikoba, and Y. Sekine, “Synchronization of Coupled Pulse-Type Hardware Neuron Models for CPG Model,” in *The Relevance of the Time Domain to Neural Network Models*, A. R. Rao and G. A. Cecchi, Eds. Boston, MA: Springer US, 2012, pp. 117–133.
- [56] K. Matsuzaka, T. Tohara, K. Nakada, and T. Morie, “Analog CMOS circuit implementation of a pulse-coupled phase oscillator system and observation of synchronization phenomena,” *Nonlinear Theory and Its Applications, IEICE*, vol. 3, no. 2, pp. 180–190, 2012.
- [57] H. Kojima, Y. Maeda, and T. Nomura, “Reproduction of Four-Leg Animal Gaits Using a Coupled System of Simple Hardware CPG Models,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E98.A, no. 2, pp. 508–509, 2015.
- [58] Y. Suedomi, H. Tamukoh, K. Matsuzaka, M. Tanaka, and T. Morie, “Parameterized digital hardware design of pulse-coupled phase oscillator networks,” *Neurocomputing*, vol. 165, pp. 54–62, 2015.
- [59] K. Takeda and H. Torikai, “A Novel Hardware-Efficient Central Pattern Generator Model Based on Asynchronous Cellular Automaton Dynamics for Controlling Hexapod Robot,” *IEEE Access*, vol. 8, pp. 139 609–139 624, 2020.
- [60] —, “A novel hardware-efficient CPG model based on asynchronous coupling of cellular automaton phase oscillators for a hexapod robot,” in *Proc. of the 2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, United Kingdom, 2020, pp. 1–8.
- [61] —, “Asynchronous CA model of central pattern generator,” in *Proc. of the 2020 International Symposium on Nonlinear Theory and Its Applications (NOLTA)*, Online, 2020, pp. 303–306.
- [62] J. A. Rogge and D. Aeyels, “Stability of phase locking in a ring of unidirectionally coupled oscillators,” *Journal of Physics A: Mathematical and General*, vol. 37, no. 46, pp. 11 135–11 148, 2004.
- [63] R. Andraka, “A survey of CORDIC algorithms for FPGA based computers,” in *Proc. of the 1998 ACM/SIGDA Sixth International Symposium*, Monterey, California, United States, 1998, pp. 191–200.
- [64] P. C. Bressloff and S. Coombes, “Traveling Waves in a Chain of Pulse-Coupled Oscillators,” *Physical Review Letters*, vol. 80, no. 21, pp. 4815–4818, 1998.
- [65] P. C. Bressloff, S. Coombes, and B. de Souza, “Dynamics of a Ring of Pulse-Coupled Oscillators: Group-Theoretic Approach,” *Physical Review Letters*, vol. 79, no. 15, pp. 2791–2794, 1997.
- [66] D. Owaki and A. Ishiguro, “A Quadruped Robot Exhibiting Spontaneous Gait Transitions from Walking to Trotting to Galloping,” *Scientific Reports*, vol. 7, p. 277, 2017.

# Chapter 4

## Neural Integrator Model based on Asynchronous Cellular Automaton

### 4.1 Spiking Neuron Model<sup>1</sup>

#### 4.1.1 Introduction

A wide variety of neuromorphic electronic circuits have been developed in circuits and systems research societies, where two of their major applications are neural prosthetic device and neurocomputing device. Various kinds of neural prosthetic devices have been developed so far, such as cochlear implants [1,2], retinal implants [3,4], and brain implants [5–7]. Brain implants for various species have been developed year by year, such as brain implants for mice [5] and brain implants for monkeys [6]. In addition, the state of the art of the neural prosthetic device has reached a brain implant for human [7]. From the stand point of the fundamental circuit theory, one of the most important problems to design a high performance brain implant is how to design a biologically plausible electrical circuit model of the major building block of the brain – the neuron. Biological neurons have lots of nonlinear ionic channels, which form various nonlinear vector fields. Due to such nonlinearities, biological neurons exhibit various bifurcation phenomena and related various nonlinear input-output characteristics. Fig. 4.1.1 illustrates typical nonlinear input-output characteristics of a biologically plausible Hodgkin-Huxley-type mathematical neuron model [8]. In each graph, the horizontal axis is a stimulation strength (i.e., input DC current) and the vertical axis is a firing frequency (i.e., average frequency of output spikes), and so the graph is called a current-frequency curve. The differences of the shapes of the nonlinear current-frequency curves are caused by differences of nonlinear dynamics of the neuron such as underlying bifurcation mechanisms as follows [9].

- In Fig. 4.1.1(a), as the stimulation strength is gradually increased, the firing frequency continuously increases. In such a case, the neuron is said to exhibit class 1 excitability. As the stimulation strength is gradually decreased, the firing frequency continuously decreases. In such a case, the neuron is said to exhibit class 1 spiking. Also, the current-frequency curve in Fig. 4.1.1(a) has no hysteresis loop. As a result, the neuron is said to exhibit "class 1 excitability and class 1 spiking without hysteresis." This nonlinear current-frequency curve is caused by a saddle-node on invariant circle

---

<sup>1</sup>© 2020 IEEE. Reprinted, with permission, from Kentaro Takeda and Hiroyuki Torikai, A Novel Asynchronous CA Neuron Model: Design of Neuron-like Nonlinear Responses based on Novel Bifurcation Theory of Asynchronous Sequential Logic Circuit, IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 6, pp. 1989–2001, Jun. 2020.

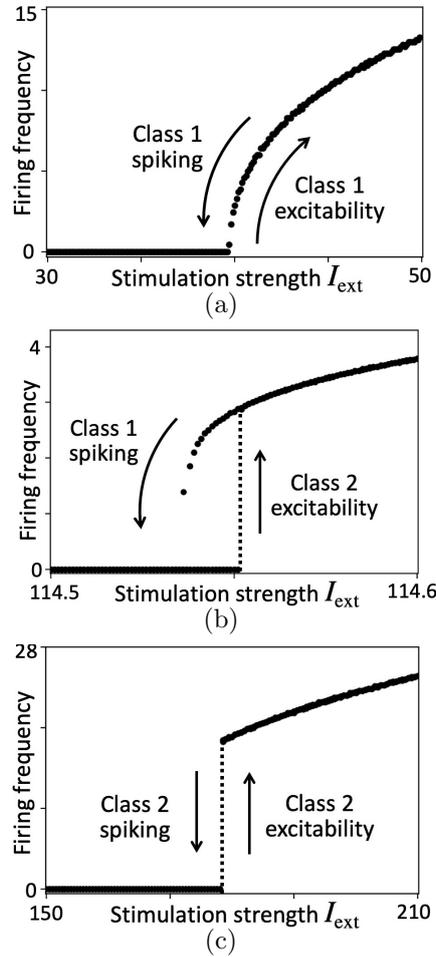


Figure 4.1.1: Three typical current-frequency curves of a biologically plausible Hodgkin-Huxley-type neuron model [8]. (a) Class 1 excitability and class 1 spiking without hysteresis. (b) Class 2 excitability and class 1 spiking with hysteresis. (c) Class 2 excitability and class 2 spiking without hysteresis.

bifurcation.

- In Fig. 4.1.1(b), as the stimulation strength is gradually increased, the firing frequency discontinuously increases. In such a case, the neuron is said to exhibit class 2 excitability. As the stimulation strength is gradually decreased, the neuron exhibits the class 1 spiking and the current-frequency curve has a hysteresis loop. As a result, the neuron is said to exhibit "class 2 excitability and class 1 spiking with hysteresis." This nonlinear current-frequency curve is caused by combination of a saddle-node bifurcation and a saddle-homoclinic bifurcation.
- In Fig. 4.1.1(c), the neuron exhibits "class 2 excitability and class 2 spiking without hysteresis." This nonlinear current-frequency curve is caused by a supercritical Hopf bifurcation.

Various biologically plausible analog and digital circuits models of neurons have been developed so far. From the stand point of the fundamental circuit theory, these circuits can be classified into four classes based on continuousness of time and state variables as follows.

*Class CTCS.* A nonlinear differential equation model of a neuron having *a continuous time and continuous states* (CTCS). Such a class CTCS system can be typically implemented by an analog nonlinear circuit, e.g., [10–13].

*Class DTCS.* A nonlinear difference equation model of a neuron having *a discrete time and continuous states* (DTCS). Such a class DTCS system can be typically implemented by a switched capacitor circuit, e.g., [14–17].

*Class DTDS.* A numerical integration model of a neuron having *a discrete time and discrete states* (DTDS). Such a class DTDS system can be typically implemented by a digital signal processor or a sequential logic circuit, e.g., [18–21].

*Class CTDS.* An asynchronous cellular automaton model of a neuron having *a continuous time (state transition times defined on a real time axis) and discrete states* (CTDS). Such a class CTDS system can be typically implemented by an asynchronous sequential logic circuit, e.g., [22–28].

Almost neuromorphic electrical circuits belong to the classes CTCS, DTCS, and DTDS. On the other hand, our group and some other groups have been developing neuromorphic electrical circuits belonging to the class CTDS. It has been shown that class CTDS neuromorphic electrical circuits can realize various nonlinear vector fields and consume fewer circuit elements compared to class DTDS neuromorphic electrical circuits [22–26]. These results motivate us to design a class CTDS electrical circuit neuron model suited to be used as a building block of a brain implant device. Then the purposes of this paper are (a) to present a novel class CTDS neuron model capable of realizing various nonlinear dynamics; (b) to present systematic design methods of the presented model so that the model can reproduce not only nonlinear current-frequency curves of the biologically plausible neuron model but also their underlying bifurcation mechanisms; and (c) to show the presented model can be implemented by fewer circuit elements and consumes less power compared to some typical conventional neuron models.

In Section II, a circuit diagram and dynamic equations of the neuron model are presented. It is shown the nonlinear dynamics of the presented model can be reduced into a continuous-discrete hybrid Poincare map without any approximation. It is then shown the presented model can reproduce fundamental properties of Poincare maps of the biologically plausible neuron model [8]. In Section III, using the hybrid Poincare map, novel theoretical results on stable and unstable fixed points of the presented model are provided. It is then shown the presented model can reproduce fundamental bifurcation phenomena of the biologically plausible neuron model for different parameter values of ionic channels. In Section IV, using the theoretical bifurcation analyses in the previous section, step-by-step systematic design methods of the presented model are proposed. It is shown the systematic design methods enables the presented model to reproduce the three typical neuron-like current-frequency curves in Fig. 4.1.1 as well as their underlying bifurcation mechanisms qualitatively. In Section V, the presented model is implemented in a field programmable gate array (FPGA) and experiments validate the FPGA-implemented presented model can

exhibit the three typical neuron-like current-frequency curves. For comparison, some typical conventional neuron models are implemented in the same FPGA device. It is shown the presented model consumes fewer circuit elements and lower power compared to the conventional neuron models. The significances and the novelties of this paper include the following points.

(i) This paper provides the theoretical analysis tools for several bifurcations (e.g., saddle-node bifurcation, saddle-node on invariant circle bifurcation, saddle-homoclinic bifurcation, and border collision bifurcation) in an asynchronous sequential logic circuit for the first time. In other words, this paper provides a new kind of nonlinear circuit theory – *the bifurcation theory of asynchronous sequential logic circuit*.

(ii) The theories in this paper are not useless theories working only on papers but are utilized as rigid tools to design the presented model. Recall the presented model can reproduce the nonlinear dynamics of the biologically plausible neuron model, can be implemented by few circuit elements, and consumes low power. Hence the presented model is a *”theoretically designable, biologically plausible, and hardware-efficient neuron model,”* which will be a strong candidate to be used as a building block in a neural prosthetic device.

(iii) Several class CTDS neuron models capable of exhibiting various firings (e.g., tonic spiking, bursting, quasi-periodic firing, chaos-like firing, and intermittent firing) have been presented [24–27], whereas their systematic design methods have not been provided. On the other hand, this paper provides the theoretical-analysis-based systematic design methods of the class CTDS neuron model for the first time.

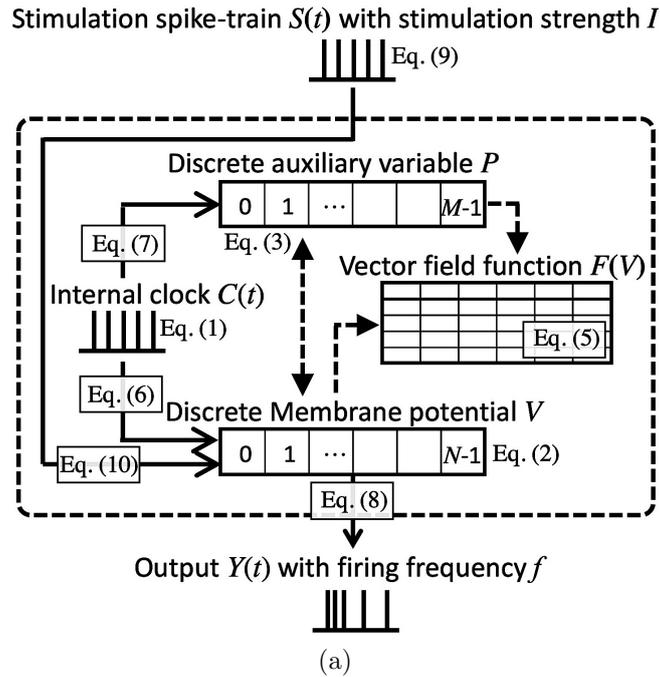
(iv) Several class CTDS cochlear models have been presented, where their nonlinear phenomena (e.g., bifurcations and stochastic resonances) have been utilized to reproduce nonlinear signal processing functions of cochleae [22,28]. Hence the analysis and design methods of the nonlinear phenomena of the class CTDS neuron model provided in this paper will contribute to design class CTDS cochlear models as well as class CTDS nonlinear signal processors.

(v) Various neuron models, spiking neural networks, and FPGA-based dynamical systems have been developed and their engineering applications such as image cryptography have been also developed [29–34]. It has been shown that different classes of current-frequency curves of neuron models can realize different functions of spiking neural networks, e.g., [9, 35, 36]. Hence the design methods of the various current-frequency curves of the FPGA-based neuron model provided in this paper will contribute to design various functions of FPGA-based neuromorphic and other systems.

## 4.1.2 Model description

### 4.1.2.1 CIRCUIT DIAGRAM AND DYNAMIC EQUATIONS

In this subsection, a novel neuron model, the dynamics of which is described by an asynchronous cellular automaton, is presented. Fig. 4.1.2(a) shows a circuit diagram of the presented model. Fig. 4.1.2(b) shows meanings of elements of the model in a context of neuron model and their relations to equations in this section. As shown in the diagram, the model has an internal clock  $C(t)$ , which is generated as follows.



Elements of presented model	Meanings as neuron model
Register storing $V$ in Eq. (4.1.2)	Membrane potential
LUT storing $F(V)$ in Eq. (4.1.4)	Nonlinear vector field
Transitions of $V$ in Eqs. (4.1.6) and (4.1.10)	Dynamic equation of membrane potential
Register storing $P$ in Eq. (4.1.3)	Auxiliary variable modulating membrane potential
Transition of $P$ in Eq. (4.1.7)	Dynamic equation of auxiliary variable
Output $Y$ in Eq. (4.1.8)	Action potential

(b)

Figure 4.1.2: Presented neuron model. (a) Circuit diagram. (b) Meanings of elements in a context of neuron model and their relations to the equations. LUT stands for look-up-table. (c) Typical time waveforms.

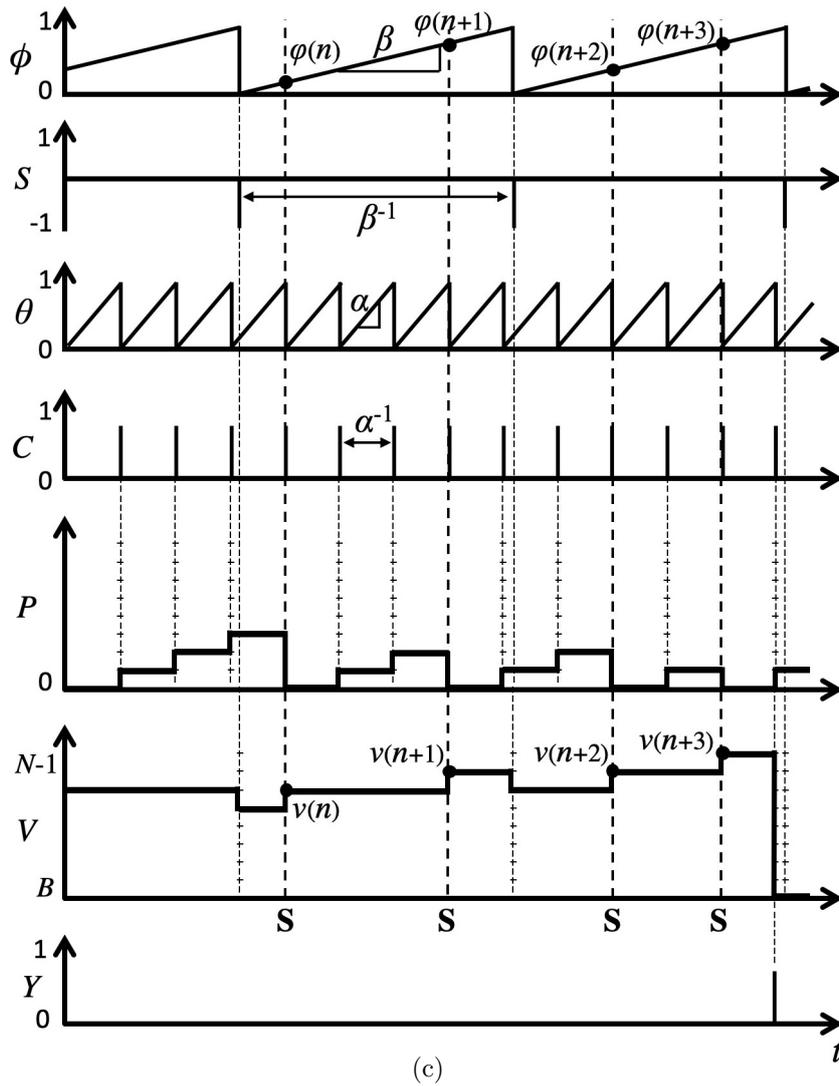


Figure 4.1.2: Continued.

**Internal clock:**

$$C(t) = \begin{cases} 1 & \text{if } \Theta(t) = 0 \pmod{1}, \\ 0 & \text{otherwise,} \end{cases} \quad \frac{d\Theta(t)}{dt} = \alpha, \quad (4.1.1)$$

where  $t \in \mathbf{R}$  is a continuous time,  $\Theta(t) \in \mathbf{R}$  is a phase,  $\Theta(0) \in [0, 1)$  is an initial phase, and  $\alpha \in (0, \infty)$  is a parameter. Fig. 4.1.2(c) shows a typical time waveform of the internal clock  $C(t)$  with the following restricted phase:

$$\theta = \Theta \pmod{1}.$$

As shown in the figure, the slope of the restricted phase  $\theta$  is  $\alpha$  and thus the frequency of the internal clock  $C(t)$  becomes  $\alpha$ . As shown in Fig. 4.1.2(a), the presented model has two registers, which store the following two discrete state variables.

**Discrete membrane potential:**

$$V \in \mathbf{Z}_N = \{0, 1, \dots, N-1\}, \quad (4.1.2)$$

**Discrete auxiliary variable:**

$$P \in \mathbf{Z}_M = \{0, 1, \dots, M-1\}. \quad (4.1.3)$$

Also, the presented model has logic gates and/or look-up-tables, which work as the following discrete function.

**Discrete vector field function:**

$$F(V) : \mathbf{Z}_N \rightarrow \mathbf{Z}_M^\pm = \{-(M-1), \dots, 0, \dots, M-1\}. \quad (4.1.4)$$

Adjusting the function  $F$  (or parameters of  $F$ ), the presented model can realize various nonlinear vector fields. In order to realize the three typical neuron-like current-frequency curves in Fig. 4.1.1, this paper proposes to use the following discrete vector field function  $F(V)$ :

$$F(V) = \left\lfloor \frac{1}{a(V-b)^2 + c} \right\rfloor, \quad (4.1.5)$$

where  $\lfloor \cdot \rfloor$  is the floor function;  $F(V)$  is saturated at  $M-1$ ;  $F(V)$  is  $-(M-1)$  if  $a(V-b)^2 + c = 0$ ; and  $a \in [0, \infty)$  and  $b \in \mathbf{R}$  are parameters. Depending on the discrete vector field function  $F(V)$ , the internal clock  $C(t)$  triggers transitions of the discrete states  $V$  and  $P$  as follows.

**State transitions triggered by internal clock:**

$$\begin{aligned} & \text{If } C(t) = 1 \text{ and } P(t) \geq |F(V)|, \text{ then} \\ V(t_+) = & \begin{cases} V(t) + 1 & \text{if } F(V) \geq 0 \text{ and } V(t) < N-1, \\ V(t) - 1 & \text{if } F(V) < 0 \text{ and } V(t) > 0, \\ B & \text{if } F(V) > 0 \text{ and } V(t) = N-1, \end{cases} \end{aligned} \quad (4.1.6)$$

If  $C(t) = 1$ , then

$$P(t_+) = \begin{cases} P(t) + 1 & \text{if } P(t) < |F(V)|, \\ 0 & \text{if } P(t) \geq |F(V)|. \end{cases} \quad (4.1.7)$$

where  $t_+ = \lim_{\epsilon \rightarrow 0} t + \epsilon$  and  $\epsilon > 0$ . Typical state transitions triggered by the internal clock  $C(t)$  are shown in Fig. 4.1.2(c). As shown in Fig. 4.1.2(a), the presented model generates an output  $Y(t)$  as follows.

**Output:**

$$Y(t) = \begin{cases} 1 & \text{if } C(t) = 1 \text{ and } V(t) = N - 1, \\ 0 & \text{otherwise.} \end{cases} \quad (4.1.8)$$

A typical output  $Y(t)$  is shown in Fig. 4.1.2(c). The presented model is said to exhibit firing if a spike  $Y = 1$  is generated and non-firing otherwise. As shown in Fig. 4.1.2(a), the presented model accepts a stimulation spike-train, which is described as follows.

**Stimulation spike-train:**

$$S(t) = \begin{cases} 1 & \text{if } t \in \{t_1^+, t_2^+, \dots\}, \\ -1 & \text{if } t \in \{t_1^-, t_2^-, \dots\}, \\ 0 & \text{otherwise,} \end{cases} \quad (4.1.9)$$

where  $t_n^+ \in \mathbf{R}$  and  $t_n^- \in \mathbf{R}$ ,  $t_n^+ \neq t_m^-$  for all  $n$  and  $m$ , are spike positions. The stimulation spike-train  $S(t)$  triggers the following transition of the discrete membrane potential  $V$ .

**State transition triggered by stimulation spike-train:**

$$\begin{aligned} &\text{If } |S(t)| = 1 \text{ and } 0 < V(t) < N - 1, \text{ then} \\ &V(t_+) = V(t) + S(t). \end{aligned} \quad (4.1.10)$$

Typical state transitions triggered by the stimulation spike-train  $S(t)$  are shown in Fig. 4.1.2(c). Note that the presented model has the discrete state variables  $V$  and  $P$  whose transitions are triggered by the asynchronous signals  $C(t)$  and  $S(t)$ . So, the presented model can be regarded as a special kind of asynchronous cellular automaton.

#### 4.1.2.2 NEURON-LIKE MAPS

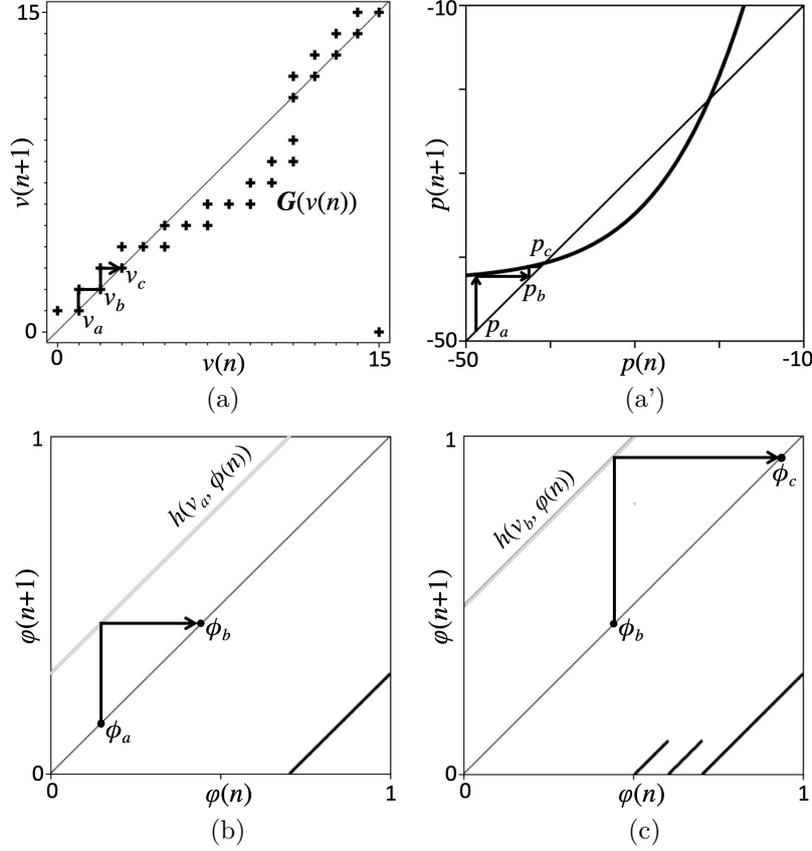
In order to analyze bifurcations of the presented model, the following periodic stimulation spike-train  $S(t)$  (corresponding to a DC stimulation current) is focused on hereafter:

$$S(t) = \begin{cases} \sigma & \text{if } \Phi(t) = 0 \pmod{1}, \\ 0 & \text{otherwise,} \end{cases} \quad \frac{d\Phi(t)}{dt} = \beta, \quad (4.1.11)$$

where  $\sigma \in \{-1, 1\}$  is a parameter determining the sign of the stimulation,  $\Phi(t) \in \mathbf{R}$  is a phase,  $\Phi(0) \in [0, 1)$  is an initial phase, and  $\beta \in [0, \infty)$  is a parameter. Fig. 4.1.2(c) shows a typical time waveform of the periodic stimulation spike-train  $S(t)$  with the following restricted phase  $\phi$ :

$$\phi = \Phi \pmod{1}.$$

As shown in Fig. 4.1.2(c), the frequency of the periodic stimulation spike-train  $S(t)$  becomes  $\beta$ . Then the periodic stimulation spike-train  $S(t)$  is characterized by the following scalar value.



Maps and variables	Meanings
Poincaré map $(g, h)$ in Eq. (4.1.12)	Dynamic equations (4.1.6), (4.1.7), (4.1.10), and (4.1.11) are reduced into $(g, h)$
State $v(n)$ of $(g, h)$	Sample of membrane potential $V$
Projected map $\mathbf{G}$ in Eq. (4.1.13)	One-dimensional approximation of $(g, h)$ describing major dynamics of $v(n)$
State $\phi(n)$ of $(g, h)$	Noise to projected map $\mathbf{G}$
Map $h$	Ergodic map [37] producing noise $\phi(n)$

(d)

Figure 4.1.3: (a)–(d) are figures of the presented model. (a) Projected map  $\mathbf{G}$  of the Poincaré map  $(g, h)$  of the presented model.  $N = 4, M = 4, a = 0.01, b = 7, c = 0, \alpha = 0.05^{-1}, \beta = 2, \gamma = 5^{-1}, B = 0, \sigma = -1$  and  $I = -0.4$ . (b) and (c) show the map  $h$  on the  $(\phi(n), \phi(n+1))$ -plane. The map  $h$  depends on  $v(n)$ . (d) Meanings of maps and variables. (a') Poincaré map  $\mathbf{G}$  of the biologically plausible Hodgkin-Huxley-type neuron model [8], where its Poincaré section is a linear approximation of a center manifold. The parameter values are  $C_M = 20, \bar{g}_{Ca} = 4, \bar{g}_K = 8, g_L = 2, V_{Ca} = 120, V_K = -80, V_L = -60, V_1 = -1.2, V_2 = 18, V_3 = 12, V_4 = 17.4, \phi = \frac{1}{15}, I_{\text{ext}} = 30$ , and  $V_E = 0$ .

**Stimulation strength:**

$$I = \sigma\beta\gamma \in \mathbf{R},$$

which corresponds to a DC stimulation current input to a neuron and thus is referred to as a stimulation strength, where  $\gamma \in [0, \infty)$  is a scaling parameter. Since the time development of the presented model is determined by the four state variables  $V$ ,  $P$ ,  $\theta$ , and  $\phi$ , their vector  $(V, P, \theta, \phi)$  can be treated as a state vector of the presented model. Hence the presented model has the following whole state space:

$$\mathbf{S}_0 = \{(V, P, \theta, \phi) \mid V \in \mathbf{Z}_N, P \in \mathbf{Z}_M, \theta \in [0, 1), \phi \in [0, 1)\}.$$

The following subset  $\mathbf{S}$  of  $\mathbf{S}_0$  has the same meaning as a Poincaré section [38].

**Poincaré section:**

$$\mathbf{S} = \{(V, P, \theta, \phi) \mid P = 0, \theta = 0\} \subset \mathbf{S}_0.$$

Fig. 4.1.2(c) shows a Poincaré section  $\mathbf{S}$ . As shown in the figure, the state vector  $(V, P, \theta, \phi)$  enters into the Poincaré section  $\mathbf{S}$  repeatedly. Hence the following state variables sampled on the Poincaré section  $\mathbf{S}$  can be defined as shown in Fig. 4.1.2(c).

$$v(n) = V|_{P=0, \theta=0}, \quad \varphi(n) = \phi|_{P=0, \theta=0},$$

where  $n = 1, 2, 3, \dots$  represents the  $n$ -th event when the state vector  $(V, P, \theta, \phi)$  enters into the Poincaré section  $\mathbf{S}$ . Since the time waveform of the state vector  $(V, P, \theta, \phi)$  starting from a sampled point  $(v(n), 0, 0, \phi(n)) \in \mathbf{S}$  is uniquely determined, the next sampled point  $(v(n+1), 0, 0, \phi(n+1)) \in \mathbf{S}$  is also uniquely determined. Hence, the following iterative map can be uniquely determined, which has the same meaning as a Poincaré map [38].

**Poincaré map:**

$$\begin{aligned} v(n+1) &= g(v(n), \varphi(n)), \\ \varphi(n+1) &= h(v(n), \varphi(n)), \end{aligned} \tag{4.1.12}$$

where  $g : \mathbf{Z}_N \times [0, 1) \rightarrow \mathbf{Z}_N$  and  $h : \mathbf{Z}_N \times [0, 1) \rightarrow [0, 1)$ . The following projected map  $\mathbf{G}(v(n))$  of the Poincaré map  $(g, h)$  onto the  $(v(n), v(n+1))$ -plane is also determined.

**Projected map:**

$$\mathbf{G}(v(n)) = \{v \mid v = g(v(n), \varphi), \varphi \in [0, 1)\}. \tag{4.1.13}$$

Fig. 4.1.3(d) summarizes meanings of the maps  $(g, h)$  and  $\mathbf{G}$  as explained below.

**Remarks on maps:**

(i) In the context of a neurons model, the spike-train  $S(t)$  is treated as a stimulation input applied to the presented model. On the other hand, in the context of the bifurcation analysis, the phase  $\Phi$  of  $S(t)$  in Eq. (4.1.11) is used as the state variable of the Poincaré map.

(ii) The dynamics of the presented model with the stimulation spike-train  $S(t)$  corresponding to the DC stimulation strength  $I$  is described by Eqs. (4.1.6), (4.1.7), (4.1.10), and (4.1.11). These equations are reduced into the two-dimensional Poincaré map  $(g, h)$  described by Eq. (4.1.12) without any approximation. The projected map  $\mathbf{G}$  can be regarded as a one-dimensional approximation of the Poincaré map  $(g, h)$ .

(iii) The projected map  $\mathbf{G}$  is multi-valued for some  $v(n)$ . Since the difference of  $v(n+1)$

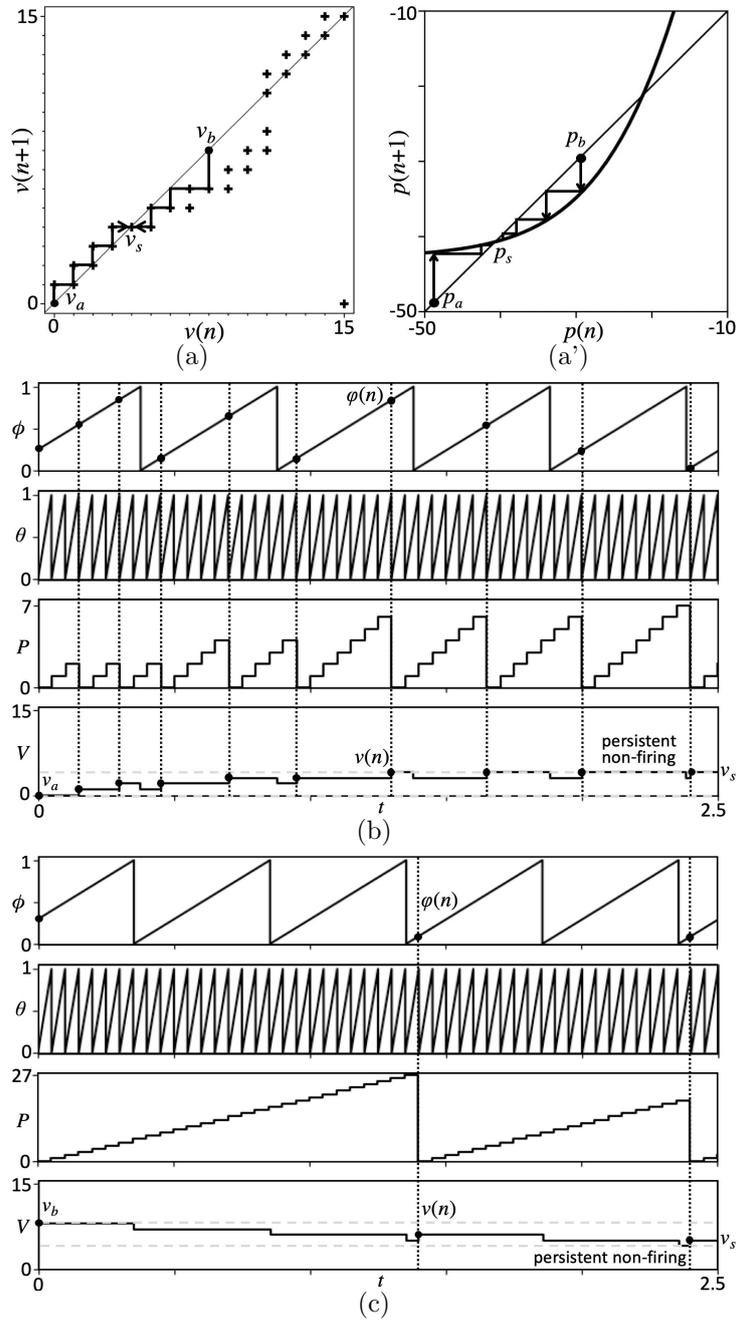


Figure 4.1.4: (a)–(c) are figures of the presented model. The parameter values are the same as those in Fig. 4.1.3(a). (a) Projected map  $G$  and its stable fixed point  $v_s$ . The trajectories of  $v(n)$  starting from  $v_a$  and  $v_b$  enters into the stable fixed point  $v_s$  and then ever stays at  $v_s$  as guaranteed by Proposition 1. (b) and (c) show time waveforms corresponding to the trajectories of  $v(n)$  in (a) starting from  $v_a$  and  $v_b$ , respectively. The time waveform of the membrane potential  $V$  passing through the stable fixed point  $v_s$  on the Poincaré section  $S$  corresponds to a persistent non-firing behavior (so-called resting state) of a neuron. (a') Poincaré map  $\mathcal{G}$  of the biologically plausible Hodgkin-Huxley-type neuron model [8], where its Poincaré section is a linear approximation of a center manifold. The parameter values are the same as those in Fig. 4.1.3(a').  $p_s$  is a stable fixed point.

for a  $v(n)$  is small, the projected map  $\mathbf{G}$  can be regarded to capture the major dynamics of the membrane potential  $V$ . The precise value of  $v(n+1)$  is determined by the values of  $v(n)$  and  $\varphi(n)$ , and thus  $\varphi(n)$  is regarded to give a small change in  $v(n+1)$ .

(iv) The dynamics of  $\varphi(n)$  is described by the map  $h$ , where  $\partial h/\partial\varphi(n) = 1$  for all  $\varphi(n)$  and  $h$  has a finite number of discontinuities. This guarantees that the trajectory of  $\varphi(n)$  becomes ergodic generically [37]. Hence,  $\varphi(n)$  can be regarded as a weak noise (i.e., ergodic random trajectory) applied to the projected map  $\mathbf{G}$ .

(v) Fig. 4.1.3(a) shows a projected map  $\mathbf{G}$  of the presented model and Fig. 4.1.3(a') shows a Poincaré map  $\mathcal{G}$  of a biologically plausible Hodgkin-Huxley-type neuron model [8]. These maps have qualitatively similar shapes. The similarity suggests that the projected map  $\mathbf{G}$  can reproduce properties of the Poincaré map  $\mathcal{G}$  and thus the presented model can reproduce the nonlinear dynamics of the biologically plausible neuron model. This suggestion is discussed in detailed in Section III.

(vi) In Fig. 4.1.3, the resolutions of the state variables  $(V, P)$  are  $(N, M) = (16, 16)$ , which lead to a visible figure of the projected map  $\mathbf{G}$  and thus are suitable for understanding theories of the map  $\mathbf{G}$ . On the other hand, resolutions of the state variables suitable for practical applications are discussed in Section V.

### 4.1.3 Theoretical bifurcation analyses

Since the projected map  $\mathbf{G}$  is different from traditional one-dimensional Poincaré maps, in order to analyze the presented model, it is necessary to introduce new mathematical tools (e.g., definitions and propositions) specialized for  $\mathbf{G}$ .

#### 4.1.3.1 STABLE FIXED POINT

A stable fixed point of  $\mathbf{G}$  is defined as follows.

**Definition 1:** A point  $v_s \in \mathbf{Z}_N$  is referred to as a stable fixed point if  $\mathbf{G}(v_s) = \{v_s\}$ .

An example of the stable fixed point  $v_s$  is shown in Fig. 4.1.4. Then the following proposition is given.

**Proposition 1:** Once a trajectory of the state  $v(n)$  of the Poincaré map  $(g, h)$  enters into a stable fixed point  $v_s$ , the state  $v(n)$  ever stays at  $v_s$  for any value of  $\varphi(n)$ .

*Proof:* Assume  $v(n) = v_s$ , where  $v_s$  is a stable fixed point.  $\mathbf{G}(v_s) = \{v_s\}$  in Definition 1 guarantees  $v(n+1) = v_s$  and thus  $v(n+m) = v_s$  for all  $m \geq 0$ . Hence Proposition 1 is proven.

Fig. 4.1.4(a) shows an example of Proposition 1, i.e., the trajectories of  $v(n)$  starting from  $v_a$  and  $v_b$  enter into  $v_s$  and then ever stay at  $v_s$ . Figs. 4.1.4(b) and (c) show time waveforms of the state variables  $(V, P, \theta, \phi)$  corresponding to the two trajectories of  $v(n)$  in Fig. 4.1.4(a). In these figures, the time waveforms of the discrete membrane potential  $V$  starting from  $v_a$  and  $v_b$  on the Poincaré section  $\mathcal{S}$  enter into the stable fixed point  $v_s$  on  $\mathcal{S}$  and then ever repeat to visit  $v_s$  on  $\mathcal{S}$ . Hence, in a context of neuron model, Proposition 1 guarantees existence of a persistent non-firing behavior (so-called resting state). Fig. 4.1.4(a') shows a Poincaré map  $\mathcal{G}$  of the biologically plausible Hodgkin-Huxley-type neuron model [8]. The map  $\mathcal{G}$  has a stable fixed point  $p_s$ , which has similar dynamic properties to the stable fixed point  $v_s$  of the projected map  $\mathbf{G}$ . Hence, the presented model is regarded to reproduce the

stable fixed point of the biologically plausible neuron model.

#### 4.1.3.2 UNSTABLE FIXED POINT

An unstable fixed point of  $\mathbf{G}$  is defined as follows.

**Definition 2:** A point  $v_u \in \mathbf{Z}_N$  is referred to as an unstable fixed point if either (i) or (ii) is satisfied: (i)  $v_u \notin \mathbf{G}(v_u)$  and there exist points  $w_+ \in \mathbf{G}(v_u)$  and  $w_- \in \mathbf{G}(v_u)$  such that  $w_+ > v_u$  and  $w_- < v_u$ ; and (ii)  $v_u \in \mathbf{G}(v_u)$ ; there exist points  $w_+ \in \mathbf{G}(v_u)$  and  $w_- \in \mathbf{G}(v_u)$  such that  $w_+ > v_u$  and  $w_- < v_u$ ; and  $h(v_u, \phi) < \phi$  and  $g(v_u, \phi) = v_u$  for all  $\phi$  or  $h(v_u, \phi) > \phi$  and  $g(v_u, \phi) = v_u$  for all  $\phi$ .

An example of the unstable fixed point  $v_u$  is shown in Fig. 4.1.5. Then the following proposition is given.

**Proposition 2:** Any trajectory of the state  $v(n)$  of the Poincaré map  $(g, h)$  starting from an unstable fixed point  $v_u$  escapes from  $v_u$  for any value of  $\varphi(n)$ .

*Proof:* Assume  $v(n) = v_u$ , where  $v_u$  is an unstable fixed point. First, assume (i) in Definition 2. Then  $v_u \notin \mathbf{G}(v_u)$  guarantees  $v(n+1) \neq v(u)$ , and the existences of  $w_+ \in \mathbf{G}(v_u)$  and  $w_- \in \mathbf{G}(v_u)$  guarantee existence of  $v(n+1) \in \{w_+, w_-\}$ . Thus,  $v(n+1) \neq v_u$ . Next, assume (ii) in Definition 2 under the condition that  $h(v_u, \varphi) < \varphi$  and  $g(v_u, \varphi) = v_u$  for all  $\varphi$ . For contradiction, assume  $v(n+m) = v_u$  for all  $m \geq 1$ . Then,  $h(v_u, \phi) < \phi$  guarantees  $h(v_u, \varphi(n)) < \varphi(n+m)$  for all  $m \geq 1$  meaning  $\varphi(n+m)$  ever increases. However, this contradicts to the definition of  $\phi = \Phi \pmod{1}$ . Thus,  $v(n+m) \neq v_u$  for some  $m \geq 1$ . For the other case of the assumption (ii) in Definition 2, the similar consideration leads to  $v(n+m) \neq v_u$  for some  $m \geq 1$ . Hence Proposition 2 is proven.

Figs. 4.1.5(a) shows an example of the unstable fixed point, i.e., the state  $v(n)$  starting from  $v_u$  escapes from  $v_u$ , where directions of the escape depend on the value of the phase  $\varphi(n)$ . Figs. 4.1.5(b) and (c) show time waveforms of the state variables  $(V, P, \theta, \phi)$  corresponding to the two trajectories of  $v(n)$  in Fig. 4.1.5(a). In these figures, the time waveforms of the discrete membrane potential  $V$  starting from  $v_u$  on the Poincaré section  $\mathbf{S}$  escape from  $v_u$ , where the escaped membrane potential  $V$  in Fig. 4.1.5(b) leads to the firing but the escaped membrane potential  $V$  in Fig. 4.1.5(c) leads to the non-firing behavior. Hence, in the context of a neuron model, Proposition 2 guarantees existence of a boundary (so-called separatrix) of the membrane potential  $V$  between the firing and the non-firing. Fig. 4.1.5(a') shows a Poincaré map  $\mathcal{G}$  of the biologically plausible Hodgkin-Huxley-type neuron model [8]. The map  $\mathcal{G}$  has an unstable fixed point  $p_u$ , which has similar dynamic properties to the unstable fixed point  $v_u$  of the projected map  $\mathbf{G}$ . Hence, the presented model is regarded to reproduce the unstable fixed point of the biologically plausible neuron model.

#### 4.1.3.3 SADDLE-NODE BIFURCATION

Fig. 4.1.6(a) shows a theoretically obtained bifurcation diagram of the presented model and Figs. 4.1.6(b)–(d) show corresponding projected maps  $\mathbf{G}$ . As shown in these figures, as the stimulation strength  $I$  increases, a stable fixed point  $v_s$  and an unstable fixed point  $v_u$  approach each other, touch, and disappear. This change of phenomena corresponds to a saddle-node bifurcation observed in nonlinear dynamical systems [38]. Effects of parameters to the saddle-node bifurcation are summarized in Table 4.1.1. Fig. 4.1.6(a') shows a bifurcation diagram of the biologically plausible Hodgkin-Huxley-type neuron model [8] and Figs.

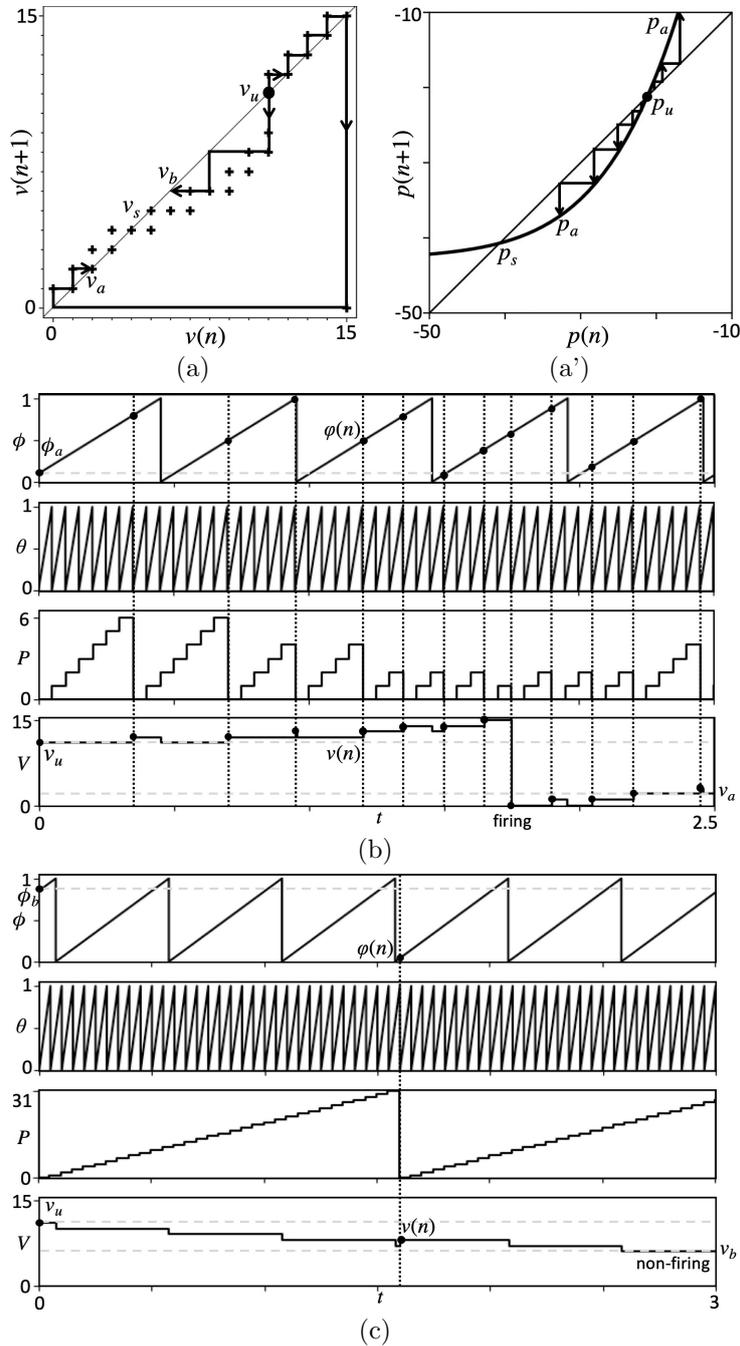


Figure 4.1.5: (a)–(c) are figures of the presented model. The parameter values are the same as those in Fig. 4.1.3(a). (a) Projected map  $G$  and its unstable fixed point  $v_u$ . The trajectories of  $v(n)$  starting from the unstable fixed point  $v_u$  escape from  $v_u$  as guaranteed by Proposition 2. The direction of the escape of  $v(n)$  depends on the value of the phase  $\varphi(n)$ . (b) and (c) show time waveforms corresponding to the trajectories of  $v(n)$  escaped to  $v_a$  and  $v_b$ , respectively. The unstable fixed point  $v_u$  corresponds to a boundary of the membrane potential  $V$  between the firing behavior in (b) and the non-firing behavior in (c). (a') Poincaré map  $\mathcal{G}$  of the biologically plausible Hodgkin-Huxley-type neuron model [8], where its Poincaré section is a linear approximation of a center manifold. The parameter values are the same as those in Fig. 4.1.3(a').  $p_u$  is an unstable fixed point.

Table 4.1.1: Effects of parameters to the bifurcations.

Param.	Effect to saddle-node bif.	Effect to border-collision bif.
$a$	$a > 0$ leads to saddle-node bifurcation	$a = 0$ leads to border-collision bifurcation
$b$	Horizontal position of bifurcation diagram	No effect when $a = 0$
$c$	Vertical position of bifurcation diagram	Vertical position of bifurcation diagram

4.1.6(b')–(d') show corresponding Poincaré maps  $\mathcal{G}$ . Comparisons between 4.1.6(a)–(d) and Figs. 4.1.6(a')–(d') suggest that the presented model can reproduce the occurrence mechanism of the saddle-node bifurcation of the biologically plausible neuron model qualitatively.

#### 4.1.3.4 BORDER-COLLISION BIFURCATION

Fig. 4.1.7(a) shows a theoretically obtained bifurcation diagram of the presented model and Figs. 4.1.7(b)–(d) show corresponding projected maps  $\mathbf{G}$ . In Fig. 4.1.7(b), the projected map  $\mathbf{G}$  has the following two regions: the flat shape of the projected map  $\mathbf{G}$  in the region (i), which is caused by saturation of the membrane potential  $V$  at 0; and the non-flat shape of the projected map  $\mathbf{G}$  in the region (ii), which is caused by the asynchronous state transitions in Eqs. (4.1.6), (4.1.7), and (4.1.10). As the stimulation strength  $I$  increases, the border between the regions (i) and (ii) approaches a stable fixed point  $v_s$ , the border touches the stable fixed point  $v_s$ , and the stable fixed point  $v_s$  disappears. This change of phenomena corresponds to a border-collision bifurcation observed in nonlinear dynamical systems [39]. Effects of parameters to the border-collision bifurcation are summarized in Table 4.1.1. Fig. 4.1.7(a') shows a bifurcation diagram of the biologically plausible Hodgkin-Huxley-type neuron model [8] and Figs. 4.1.7(b')–(d') show corresponding Poincaré maps  $\mathcal{H}$ . At Fig. 4.1.7(b'), the biologically plausible neuron model exhibits a Hopf bifurcation. Comparisons between Figs. 4.1.7(a)–(d) and Figs. 4.1.7(a')–(d') suggest that the presented model can reproduce the occurrence mechanism of the Hopf bifurcation of the biologically plausible neuron model by the border-collision bifurcation qualitatively. In fact, at the bifurcation points in Figs. 4.1.7(c) and (c'), the maps  $\mathbf{G}$  and  $\mathcal{H}$  of the presented model and the biologically plausible neuron model have almost diagonal shapes near the stable fixed points  $v_s$  and  $q_s$ , respectively.

### 4.1.4 Design of neuron-like current-frequency curves utilizing theoretical bifurcation analyses

Based on the theoretical bifurcation analyses in Section III, parameter setting methods to realize the three typical neuron-like current-frequency curves in Fig. 4.1.1 are proposed. The firing frequency of the proposed model is defined by

$$f = \frac{\text{Number of firing spikes } Y(t) = 1 \text{ during } t \in [0, \tau]}{\tau}, \quad (4.1.14)$$

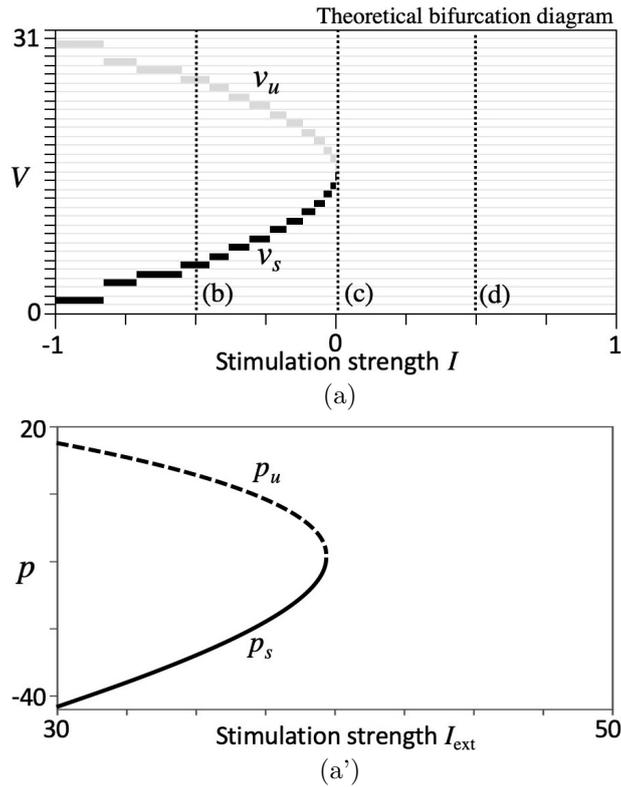


Figure 4.1.6: Saddle-node bifurcation. (a)–(d) are figures of the presented model. (a) Theoretically obtained bifurcation diagram of the presented model. The parameter values are  $N = 5$ ,  $M = 10$ ,  $a = 0.001$ ,  $b = 15$ ,  $c = 0$ ,  $\alpha = 0.010014015^{-1}$ ,  $B = 1$ , and  $\gamma = 20^{-1}$ . The projected maps  $\mathcal{G}$  in (b)–(d) correspond to (b)–(d) in (a). The presented neuron model exhibits a saddle-node bifurcation at (c). (a')–(d') are figures of the biologically plausible Hodgkin-Huxley-type neuron model [8]. (a') Bifurcation diagram. The parameter values are the same as those in Fig. 4.1.3(a'). The Poincaré maps  $\mathcal{G}$  in (b')–(d') correspond to (b')–(d') in (a'), where their Poincaré sections are linear approximations of center manifolds. The biologically plausible neuron model exhibits a saddle-node bifurcation at (c').

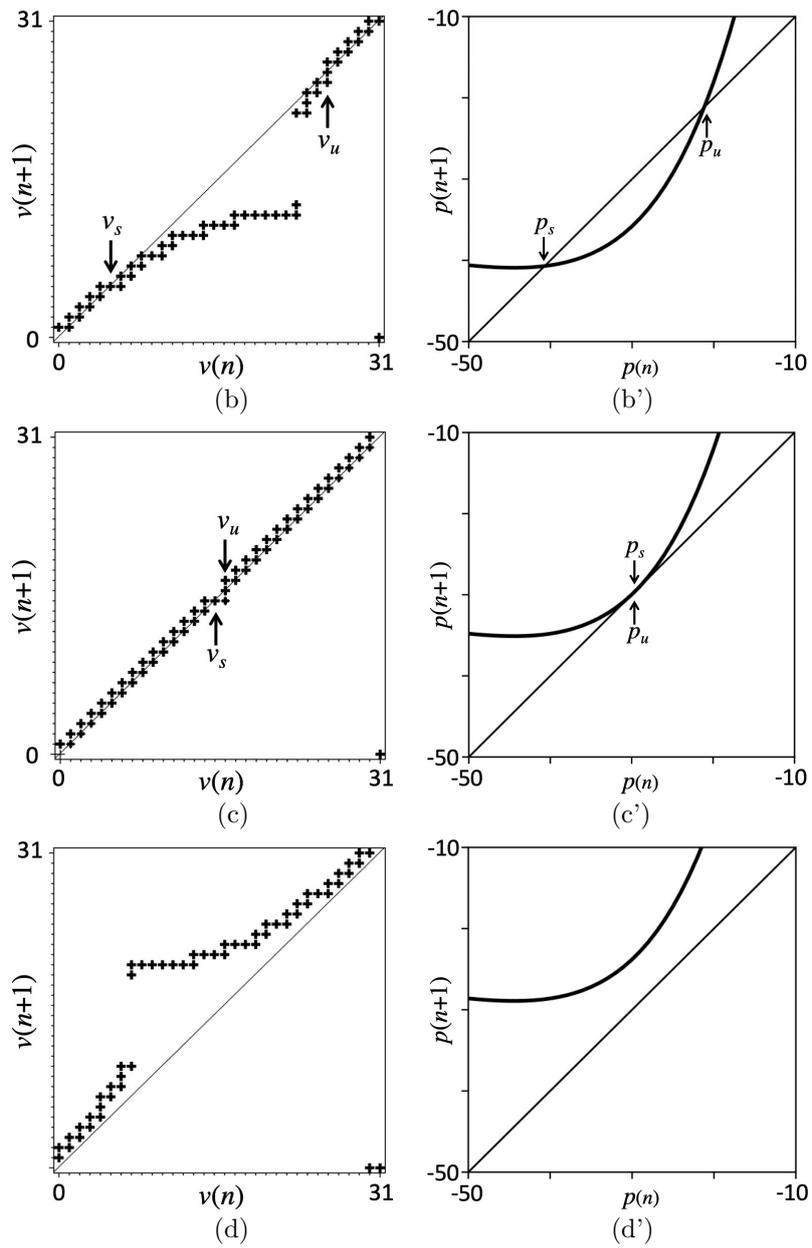


Figure 4.1.6: Continued.

where  $\tau$  is an appropriately large positive number.

#### 4.1.4.1 CLASS 1 EXCITABILITY AND CLASS 1 SPIKING WITHOUT HYSTERESIS

A step-by-step systematic design method of the presented model so that it exhibits class 1 excitability and class 1 spiking without hysteresis like Fig. 4.1.1(a) is proposed as follows.

*Step 1:* Set the values of the parameters  $a > 0$ ,  $b$ , and  $c$  so that the presented model exhibits a saddle-node bifurcation like Fig. 4.1.6(a), where effects of the parameters are in Table 4.1.1.

*Step 2:* Set the value of the parameter  $B$  on a stable fixed point  $v_s$ , which can be theoretically obtained by using Definition 1.

*Step 3:* Use the parameters  $\alpha$  and  $(\beta, \gamma)$  to adjust the scales of the firing frequency  $f$  and the stimulation strength  $I$  of the current-frequency curve, respectively.

Fig. 4.1.8(a2) shows a current-frequency curve of the presented model designed by the above method and Fig. 4.1.8(a1) shows a corresponding bifurcation diagram obtained by the theoretical bifurcation analysis. At the arrow (i) in Figs. 4.1.8(a1) and (a2), there exists a stable fixed point  $v_s$ , which corresponds to persistent non-firing. At the arrow (ii) in Figs. 4.1.8(a1) and (a2), the stable fixed point  $v_s$  and an unstable fixed point  $v_u$  touch. In this case, the membrane potential  $V$  starting from the unstable fixed point  $v_u$  increases, reaches  $N - 1$ , is reset to  $B$ , increases, and enters into the stable fixed point  $v_s$ . This situation corresponds to a saddle-node on invariant circle bifurcation [9]. At the arrow (iii) in Figs. 4.1.8(a1) and (a2), the stable fixed point  $v_s$  and the unstable fixed point  $v_u$  vanish. In this case, the presented model exhibits repeatedly firing, where the membrane potential  $V$  passes through a region in which  $V$  increases very slowly. This slow transition leads to the class 1 excitability and the class 1 spiking as shown in Fig. 4.1.8(a2).

#### 4.1.4.2 CLASS 2 EXCITABILITY AND CLASS 1 SPIKING WITH HYSTERESIS

A step-by-step systematic design method of the presented model so that it exhibits class 2 excitability and class 1 spiking with hysteresis like Fig. 4.1.1(b) is proposed as follows.

*Step 1:* Set the values of the parameters  $a > 0$ ,  $b$ , and  $c$  so that the presented model exhibits the saddle-node bifurcation like Fig. 4.1.6(a), where effects of the parameters are in Table 4.1.1.

*Step 2:* Set the value of the parameter  $B$  on an unstable fixed point  $v_u$ , which can be theoretically obtained by using Definition 2.

*Step 3:* Use the parameters  $\alpha$  and  $(\beta, \gamma)$  to adjust the scales of the firing frequency  $f$  and the stimulation strength  $I$  of the current-frequency curve, respectively.

Fig. 4.1.8(b2) shows a current-frequency curve of the presented model designed by the above method and Fig. 4.1.8(b1) shows a corresponding bifurcation diagram obtained by the theoretical bifurcation analysis. At the arrow (i) in Figs. 4.1.8(b1) and (b2), there exists a stable fixed point  $v_s$ , which corresponds to persistent non-firing. At the arrow (ii) in Figs. 4.1.8(b1) and (b2), an unstable fixed point  $v_u$  coincides with the value of the parameter  $B$ . In this case, the membrane potential  $V$  starting from the point  $v_u + 1$  increases, reaches  $N - 1$ , and is reset to  $B = v_u$ . After this reset, the membrane potential  $V$  increases or decreases depending on the value of the phase  $\phi$ . Such an orbit of the membrane potential  $V$  is called a homoclinic orbit [9]. At the arrow (iii) in Figs. 4.1.8(b1) and (b2), the homoclinic

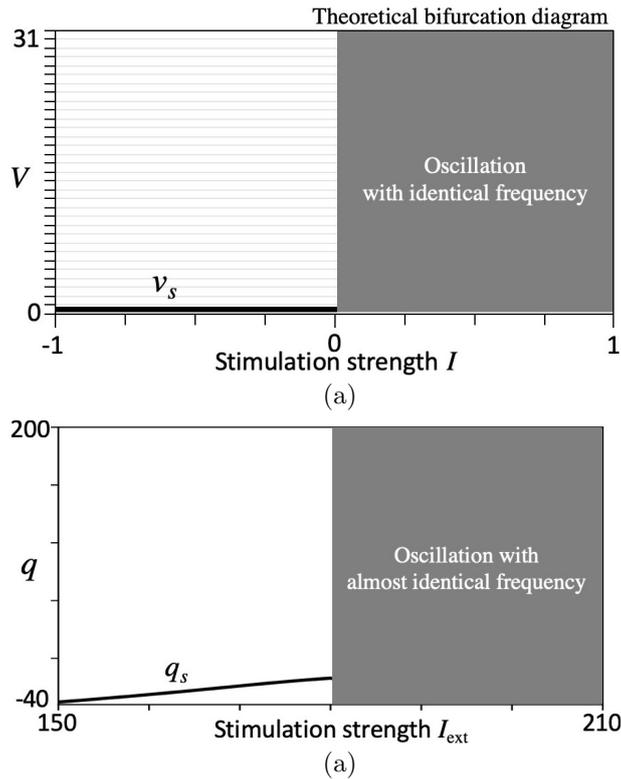


Figure 4.1.7: Border-collision bifurcation. (a)–(d) are figures of the presented model. (a) Theoretically obtained bifurcation diagram of the presented model. The parameter values are  $N = 5$ ,  $M = 10$ ,  $a = 0$ ,  $b = -1$ ,  $c = -1$ ,  $\alpha = 1.25^{-1}$ ,  $B = 30$ , and  $\gamma = 20^{-1}$ . The projected maps  $\mathbf{G}$  in (b)–(d) correspond to (b)–(d) in (a). The presented neuron model exhibits a border-collision bifurcation at (c). (a')–(d') are figures of the biologically plausible Hodgkin-Huxley-type neuron model [8]. (a') Bifurcation diagram of the biologically plausible neuron model. The parameter values are  $C_M = 20$ ,  $\bar{g}_{Ca} = 4$ ,  $\bar{g}_K = 8$ ,  $g_L = 2$ ,  $V_{Ca} = 120$ ,  $V_K = -80$ ,  $V_L = -60$ ,  $V_1 = -1.2$ ,  $V_2 = 18$ ,  $V_3 = 12$ ,  $V_4 = 17.4$ ,  $\phi = \frac{1}{15}$ , and  $V_E = 55$ . The Poincaré maps  $\mathcal{H}$  in (b')–(d') correspond to (b')–(d') in (a'), where their Poincaré sections are  $v$ -axes passing through the centers of rotations. The biologically plausible neuron model exhibits a Hopf bifurcation at (c'). Comparison between (b')–(d') and (b)–(d) suggests that the presented model can reproduce the occurrence mechanism of the Hopf bifurcation of the biologically plausible neuron model by the border-collision bifurcation qualitatively.

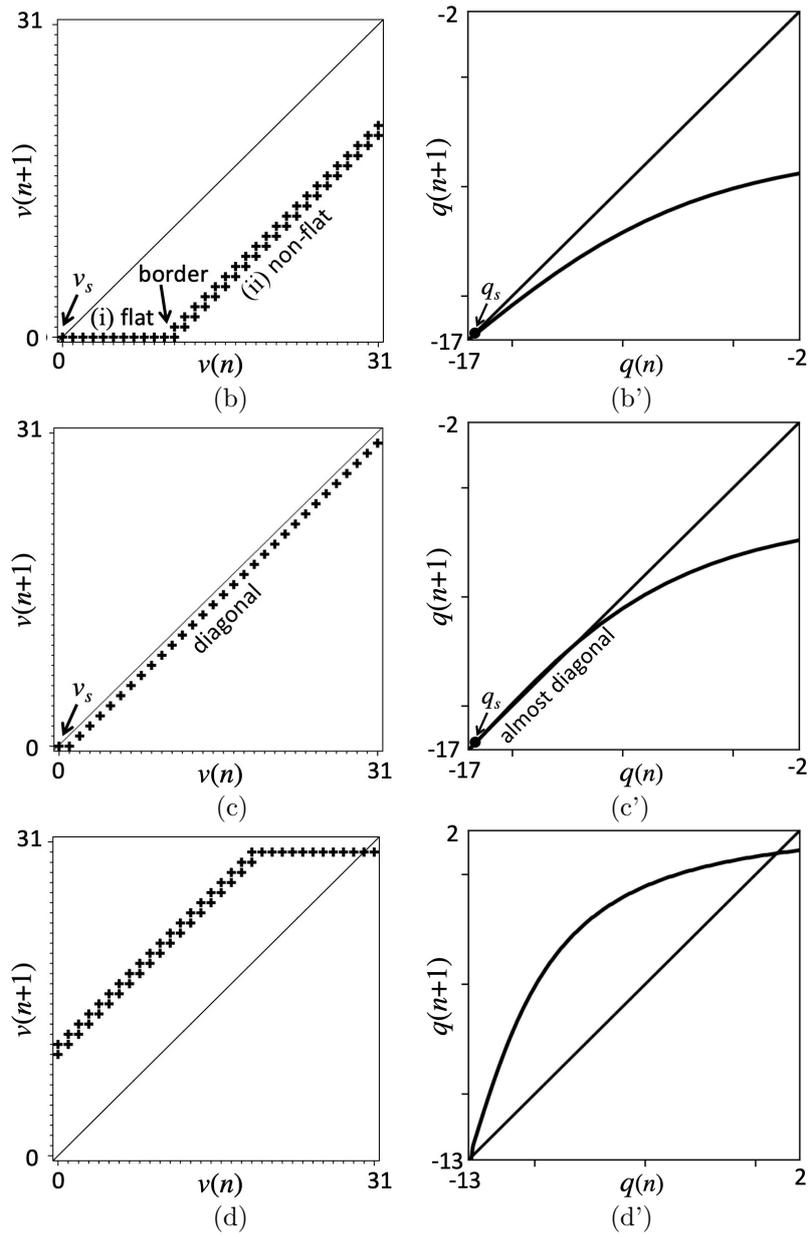


Figure 4.1.7: Continued.

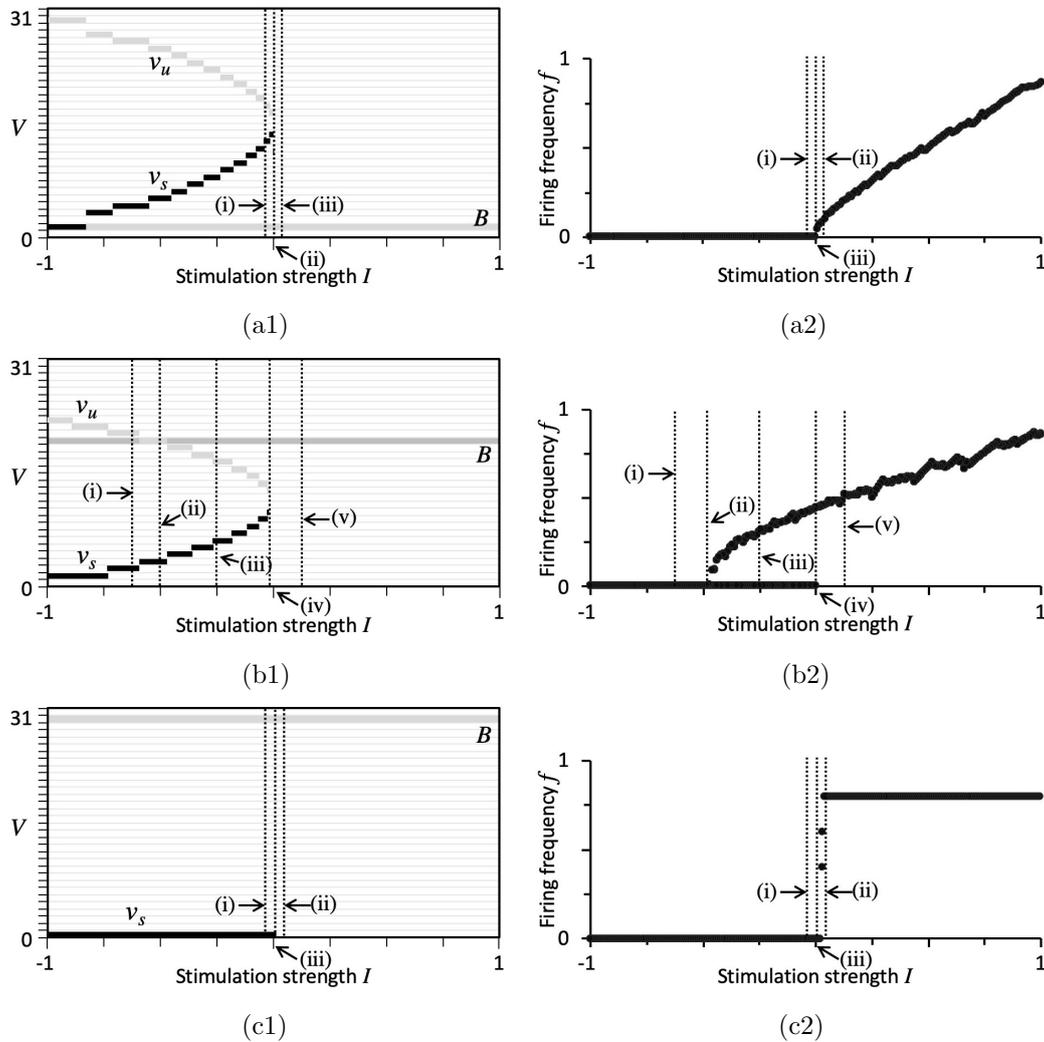


Figure 4.1.8: (a1) and (a2) show design of class 1 excitability and class 1 spiking without hysteresis. The parameter values are the same as those in Fig. 4.1.6(a). (a1) Bifurcation diagram obtained by the theoretical results in Section III. (a2) Current-frequency curve obtained by the step-by-step design method in Section IV-A. (b1) and (b2) show design of class 2 excitability and class 1 spiking with hysteresis. The parameter values are  $N = 5$ ,  $M = 10$ ,  $a = 0.0003$ ,  $b = 11$ ,  $c = 0$ ,  $\alpha = 0.010014015^{-1}$ ,  $B = 20$ , and  $\gamma = 4^{-1}$ . (b1) Bifurcation diagram obtained by the theoretical results in Section III. (b2) Current-frequency curve obtained by the step-by-step design method in Section IV-B. (c1) and (c2) show design of class 2 excitability and class 2 spiking without hysteresis. The parameter values are the same as those in Fig. 4.1.7(a). (c1) Bifurcation diagram obtained by the theoretical results in Section III. (c2) Current-frequency curve obtained by the step-by-step design method in Section IV-C.

orbit is changed into repeatedly firing, where the membrane potential  $V$  passes through a region in which  $V$  increases very slowly. This slow transition leads to the class 1 spiking as shown in Fig. 4.1.8(b2). At the arrows (vi) in Figs. 4.1.8(b1) and (b2), the presented model exhibits a saddle-node bifurcation, where the membrane potential  $V$  does not pass through a region in which  $V$  increases very slowly at the arrow (v). This sudden start of the fast repeatedly firing leads to the class 2 excitability as shown in Fig. 4.1.8(b2). The co-existence of the repeatedly firing and the persistent non-firing between the arrows (ii) and (vi) leads to a hysteresis as shown in Fig. 4.1.8(b2).

#### 4.1.4.3 CLASS 2 EXCITABILITY AND CLASS 2 SPIKING WITHOUT HYSTERESIS

A step-by-step systematic design method of the presented model so that it exhibits class 2 excitability and class 2 spiking without hysteresis like Fig. 4.1.1(c) is proposed as follows.

*Step 1:* Set  $a = 0$ . Also, set the values of the parameter  $c$  so that the presented model exhibits the border-collision bifurcation like Fig. 4.1.7(a), where effects of the parameters are in Table 4.1.1.

*Step 2:* Use the parameters  $\alpha$  and  $B$  to adjust the scale of the firing frequency  $f$  of the current-frequency curve.

*Step 3:* Use the parameters  $\beta$  and  $\gamma$  to adjust the scale of the stimulation strength  $I$  of the current-frequency curve.

Fig. 4.1.8(c2) shows a current-frequency curve of the presented model designed by the above method and Fig. 4.1.8(c1) shows a corresponding bifurcation diagram obtained by the theoretical bifurcation analysis. At the arrow (i) in Figs. 4.1.8(c1) and (c2), there exists a stable fixed point  $v_s$ , which corresponds to persistent non-firing. At the arrow (iii) in Figs. 4.1.8(c1) and (c2), the stable fixed point  $v_s$  vanishes via the border-collision bifurcation at the arrow (ii). In this case, the presented model exhibits repeatedly firing, where the membrane potential  $V$  does not pass through a region in which  $V$  increases very slowly. This sudden start of the fast repeatedly firing leads to the class 2 excitability as shown in Fig. 4.1.8(c2).

**Remarks on electromagnetic induction:** Analysis of effects of an electromagnetic induction to a neuron is one of the recent hot topic [8]. An impulsive electromagnetic induction will move a membrane potential of a neuron instantaneously. In Figs. 4.1.8(a2) and (c2), the current-frequency curve has no hysteresis and thus the presented model has only one attractor for all  $I$ , whereas in Fig. 4.1.8(b2), the current-frequency curve has the hysteresis and thus the presented model has two attractors for some  $I$ . Hence the presented models in Figs. 4.1.8(a2) and (c2) have stronger resistivity against some impulsive electromagnetic induction compared to the presented model in Fig. 4.1.8(b2). On the other hand, a persistent electromagnetic induction will work as a persistent external force (e.g., constant stimulation current) induced to a neuron. In Fig. 4.1.8(a2), the current-frequency curve is smooth, whereas in Figs. 4.1.8(b2) and (c2), the current-frequency curve has discontinuity. Hence the presented model in Fig. 4.1.8(a2) has stronger resistivity against some persistent electromagnetic induction compared to the presented neuron models in Figs. 4.1.8(b2) and (c2).

---

```

...
type arr is array(0 to N-1) of integer range -(M-1) to M-1;
signal C: std_logic; -- Eq. (1)
signal pre_C: std_logic;
signal V: integer range 0 to N-1; -- Eq. (2)
signal P: integer range 0 to M-1; -- Eq. (3)
signal F: arr := (4, 5, ..., 4, 3); -- Eq. (4)
signal S: std_logic_vector(1 downto 0); --Eq. (9)
signal pre_S: std_logic;
begin
clk_gen_C: clk_gen port map(clk100MHz, C, pre_C);
input_gen_S: input_gen port map(clk100MHz, S, pre_S);
process(clk100MHz, C, pre_C, S, pre_S) begin
if rising_edge(clk100Mhz) then
  if pre_C = '0' and C = '1' then -- Eq. (6)
    if P >= abs(F(V)) then
      if F(V) > 0 and V < N-1 then V <= V + 1;
      if F(V) < 0 and V > 0 then V <= V - 1;
      if V >= N-1 then V <= B;
    end if;
  end if;
  if pre_C = '0' and C = '1' then -- Eq. (7)
    if P < abs(F(V)) then P <= P + 1;
    else P <= 0; end if;
  end if;
  if pre_S = '0' and S(0) = '1' then -- Eq. (10)
    if S(1) = '0' and V < N-1 then V <= V + 1;
    if S(1) = '1' and V > 0 then V <= V - 1;
  end if;
end if;
end process;
...

```

---

Figure 4.1.9: Important parts of the register transfer level VHDL code and their relations to the dynamics equations of the presented model.

## 4.1.5 Implementation and Comparison

### 4.1.5.1 FPGA IMPLEMENTATION

Recall that the dynamic equations of the presented neuron model are Eqs. (4.1.2), (4.1.3), (4.1.4), (4.1.6), (4.1.7), (4.1.8), and (4.1.10). The membrane potential  $V$  and the auxiliary variable  $P$  are represented by unsigned integers to reflect Eqs. (4.1.2) and (4.1.3), and are implemented by  $n$ -bit and  $m$ -bit registers, where  $n = \lceil \log_2 N \rceil$  and  $m = \lceil \log_2 M \rceil$ . The return values of the vector field function  $F(V)$  are represented by two's complement signed integers to reflect Eq. (4.1.4) and thus  $F(V)$  is implemented by look-up-tables having  $n$ -bit unsigned inputs and  $(m + 1)$ -bit signed outputs. The values of  $n$  and  $m$  are shortened as short as possible under the condition that the presented model can realize the current-frequency curves and the bifurcation diagrams properly. The resulting values of  $(N, M)$  are  $(32, 1024)$  (i.e., 5-bit and 10-bit), which can be viewed as the values of  $(N, M)$  suitable for practical applications. The dynamics of the presented model is handwritten as a register transfer level VHDL code, where its important parts and their relations to the dynamic equations of the presented model are shown in Fig. 4.1.9. The VHDL code is compiled by Xilinx's design software environment Vivado 2018.3 and a resulting bitstream file is downloaded to Xilinx's FPGA Artix-7 XC7A100T-1CSG324C, where the FPGA has 15,850 slices and each slice contains four 6-input look-up-tables and eight flip-flops. Since the FPGA device and the HDL compiler used in this paper do not support asynchronous

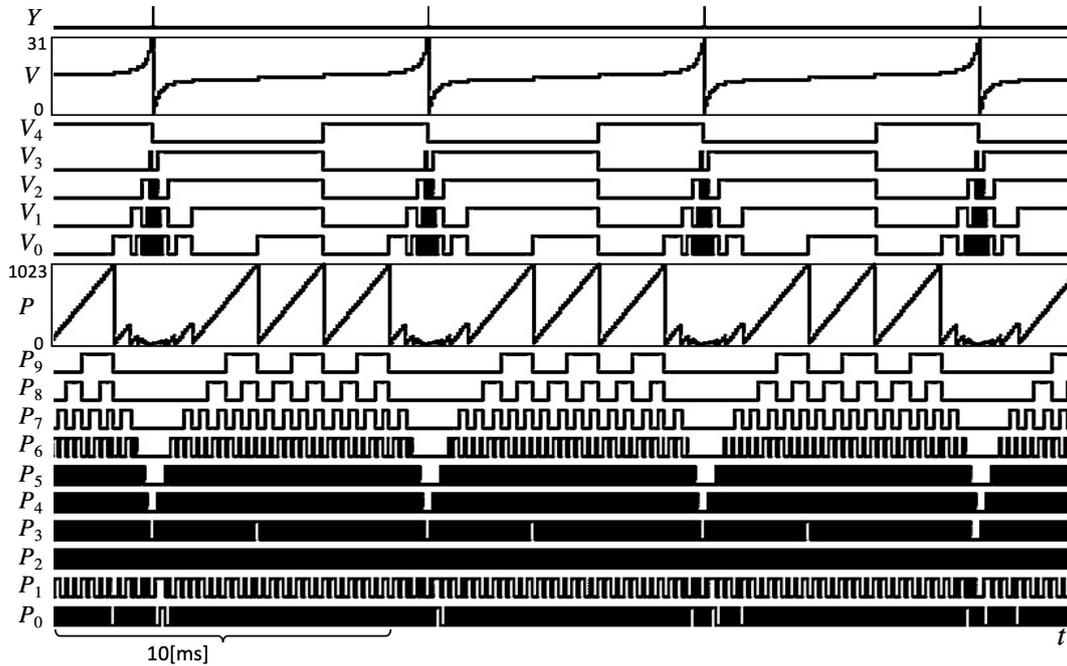


Figure 4.1.10: Experimental measurements of the presented model. Repeatedly firing corresponding to (e) in Figs. 4.1.8(a1) and (a2). The membrane potential  $V$  and the auxiliary variable  $P$  are extracted from the FPGA via 5-bit and 10-bit ports, which output their unsigned integer representations  $(V_4, \dots, V_0)$  and  $(P_9, \dots, P_0)$ , respectively.

triggering, the internal clock  $C(t)$  and the stimulation spike-train  $S(t)$  are generated from a common clock with a high frequency (100[MHz]), where periods of  $C(t)$  and  $S(t)$  are set so that their least common multiple is much longer than periods of typical repeatedly firing of the presented model. Note that so generated spike-trains  $C(t)$  and  $S(t)$  do not coincide during the least common multiple period and thus the resulting dynamics can be regarded to be almost identical with the asynchronous dynamics of the presented model. Fig. 4.1.10 shows measured waveforms of the FPGA-implemented presented model, which is designed by the bifurcation-analyses-based systematic design methods proposed in the previous section. In the figure, the membrane potential  $V$  and the auxiliary variable  $P$  are extracted from the FPGA via  $n$ -bit and  $m$ -bit ports, which output their unsigned integer representations  $(V_{n-1}, \dots, V_1, V_0)$  and  $(P_{m-1}, \dots, P_1, P_0)$ , respectively. By extensive experiments, we have confirmed that the FPGA-implemented presented model can exhibit the three typical neuron-like current-frequency curves in Fig. 4.1.1. As shown in Table II, the FPGA-implemented presented model consumes 38 look-up-tables, 23 flip-flops, and 16 FPGA slices. These results are compared with those of typical simple neuron models in the next subsection.

#### 4.1.5.2 COMPARISONS

Here the presented neuron model is compared with the following three typical simple neuron models.

**Leaky integrate-and-fire (LIF) model** [40]:

$$\begin{aligned} C \frac{dv}{dt} &= g_{\text{leak}}(E_{\text{leak}} - v) + I, \\ v(t^+) &= c \text{ if } v(t) \geq v_{\text{peak}}, \end{aligned} \quad (4.1.15)$$

where  $v$ ,  $C$ ,  $g_{\text{leak}}$ ,  $E_{\text{leak}}$ , and  $I$  are a membrane potential, a membrane capacitance, an ohmic conductance, a reverse potential, and an input DC current, respectively. When the membrane potential  $v$  crosses  $v_{\text{peak}}$  from below, it is reset to  $c$ . It is known that the leaky integrate-and-fire model (LIF model) can exhibit “class 1 excitability and class 1 spiking without hysteresis.”

**Quadratic integrate-and-fire (QIF) model** [41]:

$$\begin{aligned} C \frac{dv}{dt} &= k(v - v_{\text{rest}})(v - v_{\text{thresh}}) + I, \\ v(t^+) &= c \text{ if } v(t) \geq v_{\text{peak}}, \end{aligned} \quad (4.1.16)$$

where  $v$ ,  $C$ ,  $v_{\text{rest}}$ ,  $v_{\text{thresh}}$ ,  $k$ , and  $I$  are a membrane potential, a membrane capacitance, a resting potential, an instantaneous threshold potential, a scaling parameter, and an input DC current, respectively. When the membrane potential  $v$  crosses  $v_{\text{peak}}$  from below, it is reset to  $c$ . It is known that the quadratic integrate-and-fire model (QIF model) can exhibit “class 1 excitability and class 1 spiking without hysteresis” and “class 1 excitability and class 2 spiking with hysteresis.”

**Morris-Lecar (ML) model** [8]:

$$\begin{aligned} C_M \frac{d(v + V_E)}{dt} &= I_{\text{ext}} - \bar{g}_{\text{Ca}} M_{\infty}(v)(v + V_E - V_{\text{Ca}}) \\ &\quad - g_L(v + V_E - V_L) - \bar{g}_K N(v + V_E - V_K), \\ \frac{dN}{dt} &= \phi(N_{\infty}(v) - N)/\tau_N(v), \\ M_{\infty}(v) &= 0.5(1 + \tanh((v - V_1)/V_2)), \\ N_{\infty}(v) &= 0.5(1 + \tanh((v - V_3)/V_4)), \\ \tau_N(v) &= 1/\cosh((v - V_3)/(2V_4)), \end{aligned} \quad (4.1.17)$$

where  $v$ ,  $N$ ,  $C_M$ , and  $I_{\text{ext}}$  are a membrane potential, a potassium activation variable, a membrane capacitance, and an input DC current, respectively.  $\bar{g}_{\text{Ca}}$ ,  $\bar{g}_K$ , and  $g_L$  are conductances of ionic currents, respectively.  $V_{\text{Ca}}$ ,  $V_K$ ,  $V_L$  are reverse potentials of calcium, potassium, and leak current, respectively. Also,  $\phi$ ,  $V_1$ ,  $V_2$ ,  $V_3$ , and  $V_4$  are scaling parameters.  $V_E$  is an induced trans-membrane potential and consider  $C_M(dV_E/dt) = 0$  in this comparison [8]. It is known that the ML model can exhibit “class 1 excitability and class 1 spiking without hysteresis,” “class 2 excitability and class 1 spiking with hysteresis,” “class 2 excitability and class 2 spiking without hysteresis,” and “class 2 excitability and class 2 spiking with hysteresis.”

The dynamic equations of LIF, QIF, and ML models are handwritten as register transfer level VHDL codes using forward Euler formulae with two’s complement fixed-point number

representations, where the bit-lengths of the state variables are shortened as short as possible under the condition that the models can exhibit persistent non-firing and repeatedly firing properly. The resulting representations of the state variables are summarized in Table II. These codes are compiled by the design software environment used to compile the presented model. Then the resulting bitstream files are downloaded to the FPGA device used to implement the presented model. Table II and Fig. 4.1.11(a) summarize comparisons of implementation results. The table and the figure suggest the presented model consumes far fewer circuit elements compared to the LIF, QIF, and ML models as follows.

**Remarks on implementation:** The number of circuit elements needed to construct a flip-flop is far smaller than the number of circuit elements needed to construct a look-up-table and thus it is more efficient to decrease the number of look-up-tables to decrease the total number of circuit elements. Therefore, the total number of circuit elements used to construct each neuron model can be estimated by the numbers of look-up-tables and FPGA slices. As shown in Table II, the presented model uses far fewer look-up-tables and FPGA slices compared to the comparison models.

In addition, Fig. 4.1.11(b) shows power consumptions of the FPGA-implemented neuron models. The figure suggests the presented model consumes lower power compared to the LIF, QIF, and ML models.

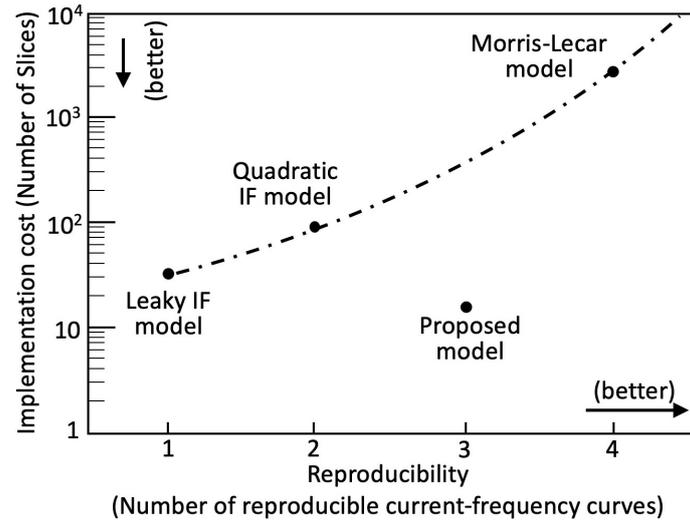
## 4.1.6 Conclusions

It is shown that the presented neuron model can realize the rich bifurcation phenomena of the biologically plausible neuron model (e.g., saddle-node bifurcation, saddle-node on invariant circle bifurcation, homoclinic bifurcation, and border-collision bifurcation) and related three typical neuron-like current-frequency curves. The theoretical analysis tools (e.g., the continuous-discrete hybrid Poincaré map and the two propositions) are utilized not only for the bifurcation analyses but also for the systematic designs of the presented model. It is shown that the presented model consumes fewer FPGA slices and lower power compared to the LIF, QIF, and ML models. These results suggest the presented model is a *theoretically designable, biologically plausible, and hardware-efficient neuron model*. So, the presented model will be a strong candidate to be used as a building block in a neural prosthetic device. In order to further develop the presented model, the following future problems should be investigated. (i) This paper presents the novel neuron model and the novel theoretical results. Since the development of the presented model is in such an early beginning phase, this paper focuses on the comparisons based on standard implementation methods, i.e., implementations by the standard HDL compiler with standard handwritten register transfer level HDL codes. Detailed comparisons using specific implementation techniques (e.g., pipelined circuit structure and single constant multiplication block) will be investigated in the next development phase. (ii) This paper reveals the projected map  $\mathbf{G}$  can be a strong mathematical tool to design the presented model. However, the projected map  $\mathbf{G}$  was used to design relatively low-dimensional bifurcations (bifurcations occurring on at most two-dimensional center manifolds). Hence the presented model and the theoretical results in this paper will be extended to higher dimensional ones in the next development phase.

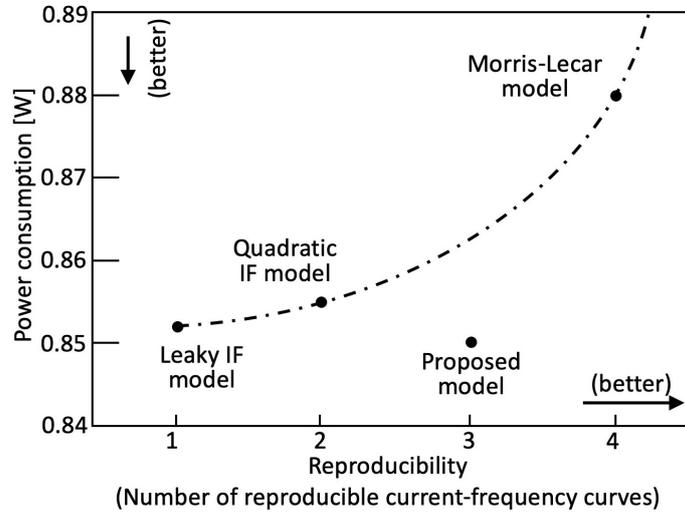
Table 4.1.2: Summary of implementation.

Models	Representation of state variable	Bit-length of state variable	# Slices	# LUTs	# FFs
<b>Presented</b>	Unsigned integer	$V$ : 5-bit, $P$ : 10-bit	16	38	23
<b>Leaky integrate-and-fire</b>	Signed two's compliment fixed-point	Integer part: 3-bit, Fractional part: 7-bit	32	93	10
<b>Quadratic integrate-and-fire</b>	Signed two's compliment fixed-point	Integer part: 4-bit, Fractional part: 7-bit	95	289	11
<b>Morris-Lecar</b>	Signed two's compliment fixed-point	Integer part: 11-bit, Fractional part: 15-bit	2830	9336	78

# Slices, # LUTs and # FFs means the number of FPGA slices, the number of look-up-tables and the number of flip-flops, respectively. DSP slices and block RAMs are not used. The parameter values of the proposed model is  $\beta = 0.05$  and the other values are the same those as Fig. 4.1.6(a). The parameter values of the LIF model are  $C = 1$ ,  $I = 1$ ,  $g_{\text{leak}} = 1$ ,  $E_{\text{leak}} = 1$ ,  $v_{\text{leak}} = 1.92$ , and  $c = 0$ . The parameter values of the QIF model are  $C = 1$ ,  $I = 1$ ,  $v_{\text{rest}} = 1$ ,  $v_{\text{thresh}} = 1$ ,  $k = 1$ ,  $v_{\text{peak}} = 4$ , and  $c = 0$ . The parameter values of the ML model are  $C_M = 2$ ,  $V_E = 0$ ,  $I_{\text{ext}} = 100$ ,  $g_L = 2$ ,  $g_{Ca} = 20$ ,  $V_L = -70$ ,  $V_{Ca} = 50$ ,  $V_K = -100$ ,  $\phi = 0.15$ ,  $V_1 = 0$ ,  $V_2 = 18$ ,  $V_3 = -10$ , and  $V_4 = 13$ .



(a)



(b)

Figure 4.1.11: Comparisons. The horizontal axis is the number of reproducible neuron-like nonlinear current-frequency curves (reproducibility). (a) Number of FPGA slices used to implement each neuron model. (b) Power consumption of each neuron model.

## 4.2 Spiking Neural Network Model<sup>2</sup>

### 4.2.1 Introduction

Persistent neural activity refers to neural activity that persists in the absence of the triggering stimulus. Various brain functions such as motor control, working memory, and decision-making involve a persistent neural activity [42–44]. Fig. 4.2.1(a) shows a well-known persistent neural activity, which is observed in position neurons of area I of the goldfish oculomotor system during spontaneous saccade [42]. Saccades are initiated by burst cells, where the on (off) burst cell activity triggers transient increases (decreases) in the firing rate  $F_{\text{extra}}$  of position neurons. The position neurons persist with this firing rate after the input burst cell activity disappears. The tonic drive to maintain eye positions  $E_{\text{contra}}$  and  $E_{\text{ipsi}}$  is supplied to motoneurons from the position neurons. Such a phenomenon that calculates the temporal integration of the triggering stimulus as the firing rate is said to be a neural integration, which has been observed in head direction cells [45] and entorhinal cortex cells [46] as well as the oculomotor system [42]. Several neural integrator (NI) models have been presented, e.g., [36, 47–51]. Koulakov *et al.* introduced a conceptual framework, a NI with robustness against parameter mistuning, which can be modeled by a continuous attractor network consisting of intrinsically bistable neurons [36]. They presented a recurrent spiking neural network based on NMDA-dependent bistability in [36]. Other studies attempted to model in a multicompartment neuron and a single multistable neuron, which are based on different biological perspectives but essentially the same mechanism, that is, an ensemble of bistable units [48–51].

Several circuit studies and engineering applications of a NI, such as an oculomotor system, head direction system, and decision-making system have been presented [52–54]. Bio-inspired systems including the ones mentioned above must utilize few hardware resources for circuit implementation and consume low power. In this study, we focused on the conceptual framework of Koulakov *et al.*. We first derived a recurrent network for a NI described by an ordinary differential equation (ODE) that can be solved with less computational resources than that of the previously presented NI models [36, 48–51]. Considering the implementation of a bio-inspired system on hardware, two important forms exist: analog implementation and digital implementation. For example, spiking neuron models for efficiently implementing on both the analog hardware [55, 56] and digital hardware [25, 57] have been developed. This study focuses on digital implementation using a field-programmable gate array (FPGA) because it is suitable for a highly scalable prototype implementation. The FPGA is also suitable for comparing different implementation methods (e.g., a custom-designed digital signal processor (DSP) and custom-designed CPU) of bio-inspired systems as long as the same platform and design environment are employed. Then, based on the above reduced ODE model, we present a novel hardware-oriented recurrent network model consisting of asynchronous cellular automaton (CA) neurons, which is designed to be suitable for digital implementation. A comparison of the proposed model with the reduced ODE and previously proposed NI models [36] reveals that the proposed model consumes fewer hardware resources and lower power than those of the other two models.

This paper includes the following contributions. (i) Recently, we have been developing

---

<sup>2</sup>© 2021 IEEE. Reprinted, with permission, from Kentaro Takeda and Hiroyuki Torikai, A novel hardware-oriented recurrent network of asynchronous CA neurons for a neural integrator, IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 8, pp. 2972–2976, Aug. 2021.

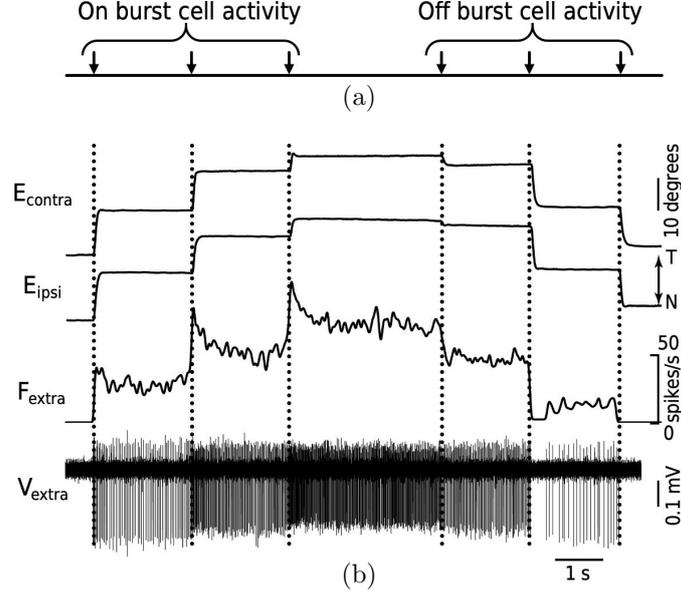


Figure 4.2.1: Experimental data showing persistent neural activity of area I position neurons during spontaneous saccades in goldfish oculomotor system adapted from [42]. (a) On (off) burst cell to trigger transient increases (decreases) in firing rate  $F_{\text{extra}}$  of position neurons. (b) Horizontal eye positions  $E_{\text{contra}}$  and  $E_{\text{ipsi}}$ , firing rate  $F_{\text{extra}}$ , and extracellular voltage  $V_{\text{extra}}$ .

asynchronous CA models of bio-inspired systems with discrete states and continuous time, which that can be implemented in an asynchronous sequential logic circuit, for example, [22, 25, 57, 58]. This paper presents a novel NI model implemented in such an asynchronous sequential logic circuit for the first time. (ii) The presented model can be implemented using fewer hardware resources for circuit implementation and consumes lower power than a numerical integration model with discrete states and a discrete time. This is implemented in a sequential circuit or a digital processor. (iii) The above advantage suggests that the presented model will be useful for developing future applications such as neural prosthetic devices and bio-inspired robots based on the novel asynchronous sequential logic circuit, which is implemented as a small-scale circuit and has low power consumption.

## 4.2.2 Reduced ODE model for NI

In this section, based on the conceptual framework of Koulakov *et al.*, we derive an ODE model of a recurrent network for a NI that can be solved with fewer computational resources. The network consists of one-dimensional integrate-and-fire-type neurons:  $C\dot{v}_k = g(v_k) + I_k^{\text{syn}} + I_k^{\text{ext}}$ , where  $v_k(t_+) \leftarrow v_k^{\text{reset}}$  if  $v_k(t) > v_k^{\text{th}}$ . A spike in the  $k$ -th neuron is assumed to be generated instantaneously if  $v_k$  reaches  $v_k^{\text{th}}$ . The following function, described by a polynomial of the lowest degree that gives the neurons hysteresis characteristics, is employed for the term  $g(v_k)$ :  $g(v_k) = v_k^2 + b_k$ . This term causes a pair of saddle-node bifurcation and a saddle-homoclinic bifurcation, which leads to the following nonlinear frequency-current characteristics with hysteresis, as shown in Fig. 4.2.2. If the input current  $I^{\text{ext}}$  increases gradually, then the firing rate increases discontinuously at the white triangle. In such a case, it is said that the neuron exhibits class 2 excitability [9]. If the input current  $I^{\text{ext}}$

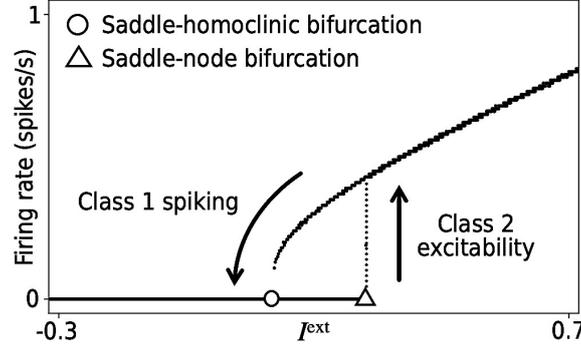


Figure 4.2.2: Class 2 excitability and class 1 spiking with hysteresis caused by saddle-node bifurcation and saddle-homoclinic bifurcation.

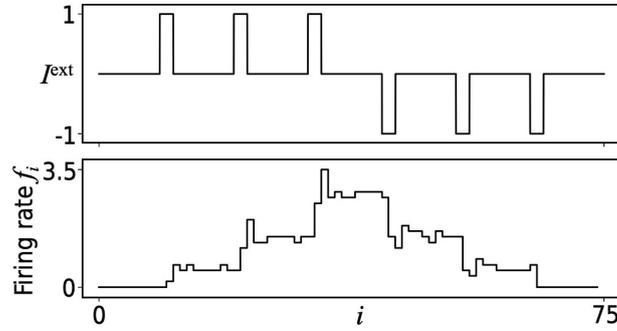


Figure 4.2.3: Neural integration of reduced ODE model.  $C = 1$ ,  $\tau = 6$ ,  $u_{\max} = 1$ ,  $b_1 = -0.2$ ,  $b_2 = -1.2$ ,  $b_3 = -2.2$ .  $v_k^{\text{reset}} = 0.3$  and  $v_k^{\text{th}} = 1$  for all  $k$ .  $w_{k,l} = 1$  for all  $k$  and  $l$ .  $\Delta = 2$ .

decreases gradually, then the firing rate decreases continuously and the neuron stops spiking at the white circle. In such a case, it is said that the neuron exhibits class 1 spiking [9]. Hence, the model is considered to be a neuron model exhibiting hysteresis frequency-current characteristics that can be solved with minimal computational resources. Further,  $I_k^{\text{syn}}$  has the following synaptic inputs:  $I_k^{\text{syn}} = \sum_l w_{k,l} u_{k,l}$ , where  $w_{k,l}$  and  $u_{k,l}$  represent synaptic weights and synaptic potentials, respectively. The dynamics of the synaptic potential  $u_{k,l}$  are described by a single exponential model as follows:  $\dot{u}_{k,l} = -\tau^{-1} u_{k,l} + \tau^{-1} \sum_{t_k^{(i)} < t} \delta(t - t_k^{(i)})$ , where  $\tau$  is a time constant, and  $\delta : \mathbb{R} \rightarrow \{0, 1\}$  is the unit impulse function  $\delta(t) = 1$  if  $t = 0$  and  $\delta(t) = 0$  if  $t \neq 0$ . The synaptic potential  $u_k$  is assumed to be saturated at  $u_{\max}$ . We refer to the model as the reduced ODE model throughout the paper. Fig. 4.2.3 shows the neural integration of the reduced ODE model consisting of three neurons. The firing rate in this figure is obtained from  $f_i = \text{Number of spikes of all neurons during } t \in [h_i, h_{i+1}) / \Delta$ , where  $\Delta > 0$  is the bin width, and  $h_i = i\Delta$ . In this figure, the firing rate  $f_i$  implies an integration value in which the input current  $I_{\text{ext}}$  is temporally integrated as with  $F_{\text{extra}}$  in Fig. 4.2.1(b).

### 4.2.3 Hardware-oriented model for NI

In this section, based on the reduced ODE model, a hardware-oriented recurrent network model for a NI is presented. The presented model consists of  $n$  neuron units, where the  $k$  ( $\in$

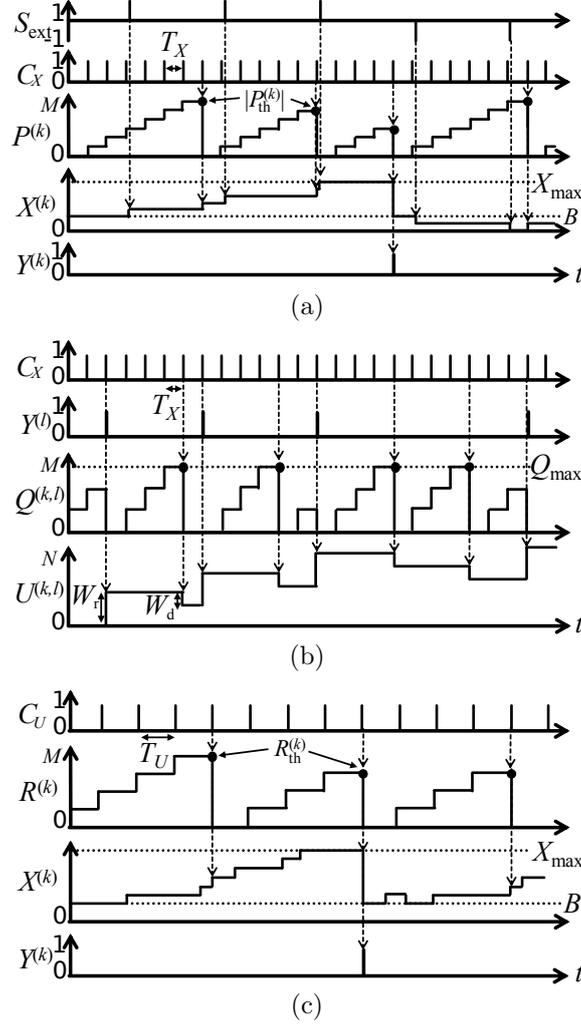


Figure 4.2.4: Typical timing chart of discrete states.

$\{1, \dots, n\}$ ) th neuron unit has two registers that store a discrete membrane potential  $X^{(k)} \in \mathbf{Z}_X \equiv \{0, 1, \dots, X_{\max}\}$  and a discrete frequency divider  $P^{(k)} \in \mathbf{Z}_P \equiv \{0, 1, \dots, P_{\max}\}$ . In addition, the  $k$  th neuron unit has lookup tables and/or logic gates, which are used as the following discrete vector field function  $F_X^{(k)} : \mathbf{Z}_X \rightarrow \mathbf{Z}_P^\pm \equiv \{-(P_{\max} + 1), \dots, 0, \dots, P_{\max}\}$ ,  $F_X^{(k)}(X) = \lfloor (a^{(k)}(X - b^{(k)})^2 + c^{(k)})^{-1} \rfloor$ , where  $\lfloor \cdot \rfloor$  represents an integer part,  $F_X^{(k)}(X)$  is clamped at  $\pm P_{\max}$ ;  $F_X^{(k)}(X)$  is  $-P_{\max}$  if  $a^{(k)}(X - b^{(k)})^2 + c^{(k)} = 0$ , and  $a \in \mathbb{R}$ ,  $b \in \mathbb{R}$ , and  $c \in \mathbb{R}$  are parameters. The membrane potential  $X^{(k)}$  and the frequency divider  $P^{(k)}$  accepts the internal clock  $C_X \in \{0, 1\}$  generated as follows:  $C_X(t) = \sum_{i=0}^{\infty} \delta(t - iT_X)$ , where  $T_X \in (0, \infty)$  is a clock period. The state update of the membrane potential  $X^{(k)}$  and the

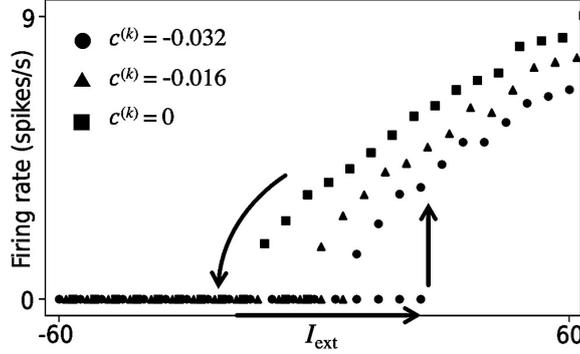


Figure 4.2.5: Hysteresis characteristics of presented individual neurons.  $N = 31$ ,  $M = 1023$ ,  $T_V = 0.00100314$ ,  $c^{(1)} = -0.032$ ,  $c^{(2)} = 0.016$ , and  $c^{(3)} = 0$ .  $a^{(k)} = 0.0003$ ,  $b^{(k)} = 11$ , and  $B^{(k)} = 19$  for all  $k$ .

frequency divider  $P^{(k)}$  is driven by the internal clock  $C_X$  as follows:

$$\begin{aligned} &\text{If } C_X(t) = 1, \text{ then} \\ &P^{(k)}(t_+) := \begin{cases} P^{(k)}(t) + 1 & \text{if } P^{(k)}(t) < |P_{\text{th}}^{(k)}|, \\ 0 & \text{if } P^{(k)}(t) = |P_{\text{th}}^{(k)}|, \end{cases} \quad (1) \\ &X^{(k)}(t_+) := \begin{cases} X^{(k)}(t) + \text{sgn}(P_{\text{th}}^{(k)}) & \text{if } X^{(k)}(t) \neq X_{\text{max}} \text{ and } P^{(k)}(t) = |P_{\text{th}}^{(k)}|, \\ B^{(k)} & \text{if } X^{(k)}(t) = X_{\text{max}} \text{ and } P^{(k)}(t) = |P_{\text{th}}^{(k)}|, \end{cases} \end{aligned}$$

where  $P_{\text{th}}^{(k)} \equiv F_X^{(k)}(X^{(k)}(t))$ ; the symbol “ $t_+$ ” is defined as “ $\lim_{\varepsilon \rightarrow +0} t + \varepsilon$ ,” and the symbol “ $:=$ ” is defined as an “instantaneous state update”. Moreover,  $B^{(k)} \in \mathbf{Z}_X$  is the parameter representing a reset potential. Fig. 4.2.4(a) shows a typical timing chart of the membrane potential  $X^{(k)}$  and the frequency divider  $P^{(k)}$  driven by the internal clock  $C_X(t)$ . The  $k$  th neuron unit accepts an input stimulation  $S_{\text{ext}} \in \{-1, 0, 1\}$ , which is described as follows:  $S_{\text{ext}}(t_+) = \sum_{i=0}^{\infty} \delta(t - t_+^{(i)}) - \delta(t - t_-^{(i)})$ , where  $t_+^{(i)} \in \mathbb{R}$  and  $t_-^{(i)} \in \mathbb{R}$  are spike positions and  $t_+^{(i)} \neq t_-^{(j)}$  for all  $i$  and  $j$ . The state update of the membrane potential  $X^{(k)}$  is driven by the input stimulation  $S_{\text{ext}}(t)$  as follows:

$$\begin{aligned} &\text{If } |S_{\text{ext}}(t)| = 1, \text{ then} \\ &X^{(k)}(t_+) := \begin{cases} X^{(k)}(t) + S_{\text{ext}}(t) & \text{if } X^{(k)}(t) \neq X_{\text{max}}, \\ B^{(k)} & \text{if } X^{(k)}(t) = X_{\text{max}} \text{ and } S_{\text{ext}}(t) = 1. \end{cases} \quad (2) \end{aligned}$$

Fig. 4.2.4(a) shows a typical timing chart of the membrane potential  $X^{(k)}$  driven by the input stimulation  $S_{\text{ext}}(t)$ . Then, the  $k$  th neuron unit produces an action potential  $Y^{(k)}(t) \in \{0, 1\}$ :

$$Y^{(k)}(t) = \begin{cases} 1 & \text{if } C_X(t) = 1 \text{ and } X^{(k)}(t) = X_{\text{max}} \text{ and } P^{(k)}(t) = |P_{\text{th}}^{(k)}|, \\ 1 & \text{if } S_{\text{ext}}(t) = 1 \text{ and } X^{(k)}(t) = X_{\text{max}}, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Fig. 4.2.4(a) shows a typical timing chart of the action potential  $Y^{(k)}(t)$ . Let the instantaneous pulse density of the three-state input stimulation  $S_{\text{ext}}$  be proportional to the

stimulation strength  $I_{\text{ext}} \in \mathbb{R}$ . Fig. 4.2.5 shows the frequency-current characteristics with hysteresis. Based on the results of a recently reported paper [57], it can be ensured that the asynchronous CA neuron exhibits a saddle-homoclinic bifurcation and a saddle-node bifurcation. The combination of these bifurcations leads the neuron to class 1 spiking and class 2 excitability with hysteresis, as shown in Fig. 4.2.2. Hence, the asynchronous CA neuron is considered to be a bistable unit.

The neuron units are connected all-to-all by  $n^2$  synapse units. The  $l$  ( $\in \{1, \dots, n\}$ )<sup>th</sup> synapse unit for the  $k$  th neuron unit has two registers that store the discrete synaptic potential  $U^{(k,l)} \in \mathbf{Z}_X = \{0, 1, \dots, X_{\text{max}}\}$  and the discrete frequency divider  $Q^{(k,l)} \in \mathbf{Z}_Q = \{0, 1, \dots, Q_{\text{max}}\}$ . The  $l$  th synapse unit accepts the internal clock  $C_X$ . The state update of the synaptic potential  $U^{(k,l)}$  and the frequency divider  $Q^{(k,l)}$  are driven by the internal clock  $C_X$  as follows:

$$\begin{aligned} &\text{If } C_X(t) = 1, \text{ then} \\ Q^{(k,l)}(t_+) &:= \begin{cases} Q^{(k,l)}(t) + 1 & \text{if } Q^{(k,l)}(t) < Q_{\text{max}}, \\ 0 & \text{if } Q^{(k,l)}(t) = Q_{\text{max}} \text{ or } Y^{(l)}(t) = 1. \end{cases} \\ U^{(k,l)}(t_+) &:= \begin{cases} U^{(k,l)}(t) + W_r^{(k,l)} & \text{if } Y^{(l)}(t) = 1, \\ U^{(k,l)}(t) - W_d^{(k,l)} & \text{if } Y^{(l)}(t) = 0 \text{ and } Q^{(k,l)}(t) = Q_{\text{max}}, \end{cases} \end{aligned} \quad (4)$$

where  $W_r^{(k,l)} \in \mathbf{Z}_X$  and  $W_d^{(k,l)} \in \mathbf{Z}_X$  are the rise and decay parameters of the synaptic potential, and  $U^{(k,l)}$  is saturated at  $X_{\text{max}}$ . Fig. 4.2.4(b) shows a typical timing chart of the synaptic potential  $U^{(k,l)}$  and the frequency divider  $Q^{(k,l)}$  driven by the internal clock  $C_X(t)$ . To receive synaptic inputs, the  $k$  th neuron unit has the register storing the discrete frequency divider  $R^{(k)} \in \mathbf{Z}_P$  and the discrete coupling function  $F_U^{(k)} : \{0, 1, \dots, nX_{\text{max}}\} \rightarrow \mathbf{Z}_P$ ,  $F_U^{(k)}(U) = \lfloor (f_U T_U (U/nX_{\text{max}}))^{-1} \rfloor$ , where  $F_U^{(k)}(U)$  is saturated at  $P_{\text{max}}$ ;  $F_U^{(k)}(U)$  is  $P_{\text{max}}$  if  $f_U T_U (U/nX_{\text{max}}) = 0$ ; and  $f_U \in [0, \infty)$  is a parameter. The membrane potential  $X^{(k)}$  and the frequency divider  $R^{(k)}$  accept the internal clock  $C_U \in \{0, 1\}$  generated as follows:  $C_U(t) = \sum_{i=0}^{\infty} \delta(t - iT_U)$ , where  $T_U \in (0, \infty)$  is a clock period. The state update of the membrane potential  $X^{(k)}$  and the frequency divider  $R^{(k)}$  is driven by the internal clock  $C_U$  as follows:

$$\begin{aligned} &\text{If } C_U(t) = 1, \text{ then} \\ R^{(k)}(t_+) &:= \begin{cases} R^{(k)}(t) + 1 & \text{if } R^{(k)}(t) < R_{\text{th}}^{(k)}, \\ 0 & \text{if } R^{(k)}(t) = R_{\text{th}}^{(k)}, \end{cases} \\ X^{(k)}(t_+) &:= \begin{cases} X^{(k)}(t) + 1 & \text{if } X^{(k)}(t) \neq X_{\text{max}} \text{ and } R^{(k)}(t) = R_{\text{th}}^{(k)}, \\ B^{(k)} & \text{if } X^{(k)}(t) = X_{\text{max}} \text{ and } R^{(k)}(t) = R_{\text{th}}^{(k)}, \end{cases} \end{aligned} \quad (5)$$

where  $R_{\text{th}}^{(k)} \equiv F_U^{(k)}(\sum_{l=1}^n U^{(k,l)}(t))$ . Fig. 4.2.4(c) shows a typical timing chart of the membrane potential  $X^{(k)}$  and the frequency divider  $R^{(k)}$  driven by the internal clock  $C_U(t)$ . To reflect the synaptic connection in the state update of the action potential  $Y^{(k)}$ , the following state update equation is added to Eq. (3):  $Y^{(k)}(t) = 1$  if  $C_U(t) = 1$  and  $X^{(k)}(t) = X_{\text{max}}$  and  $R^{(k)}(t) = R_{\text{th}}^{(k)}$ . Fig. 4.2.6 shows the firing rate  $f_i$  of the presented model in response to the stimulations. It can be confirmed that the presented model reproduces the neural integration, which is similar to that of the reduced ODE model (see also Fig. 4.2.3).

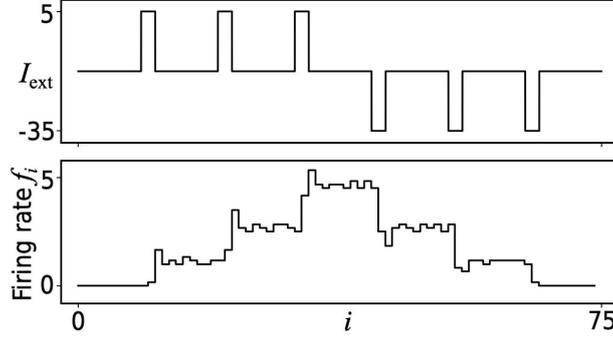


Figure 4.2.6: Neural integration of proposed model.  $n = 3$ ,  $N = 31$ ,  $M = 1023$ ,  $L = 255$ ,  $T_V = 0.00100314$ ,  $T_U = 0.0050271$ ,  $f_U = 50$ ,  $c^{(1)} = -0.032$ ,  $c^{(2)} = -0.016$ , and  $c^{(3)} = 0$ .  $a^{(k)} = 0.0003$ ,  $b^{(k)} = 11$  and  $B^{(k)} = 19$  for all  $k$ .  $\Delta = 2$ .

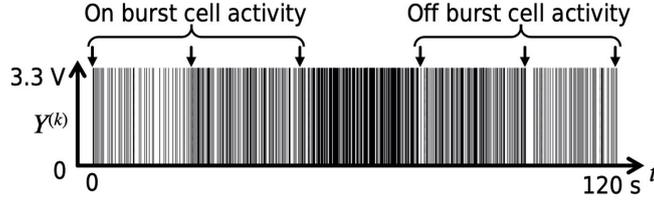


Figure 4.2.7: Signals  $Y^{(k)}$  in presented model running on FPGA (see also Fig. 4.2.1).

## 4.2.4 Implementation and comparison

### 4.2.4.1 FPGA IMPLEMENTATION

The dynamics of the presented model are written as a register transfer level (RTL) code using VHDL as follows: The discrete state variables  $(X^{(k)}, U^{(k)})$ ,  $(P^{(k)}, R^{(k)})$ , and  $Q^{(k)}$  are implemented by registers as  $X_{\max}$  bit,  $P_{\max}$  bit, and  $Q_{\max}$  bit unsigned integers, where  $X_{\max} = \lceil \log_2 X_{\max} \rceil$ ,  $P = \lceil \log_2 P_{\max} \rceil$ , and  $Q_{\max} = \lceil \log_2 Q_{\max} \rceil$ , respectively. The function  $F_X^{(k)}$  is implemented in lookup tables (LUTs) having a  $X_{\max}$  bit unsigned integer input and a  $(P_{\max} + 1)$  bit signed integer output in the two's complement format. The function  $F_U^{(k)}$  is implemented in LUTs having an  $nX_{\max}$  bit unsigned integer input and a  $P_{\max}$  bit unsigned integer output. The state updates in Eqs. (1), (2), (4), and (5) are written by sequential statements driven by the clocks  $C_X$  and  $C_U$ . The RTL code is synthesized by Xilinx Vivado Design Suite v2019.2, and a generated bitstream file is downloaded into Xilinx's FPGA, Kintex-7 XC7K325T-2FFG900C. Fig. 4.2.7 shows the waveforms measuring the signals  $Y^{(k)}$  of the presented model running on the FPGA. In this figure, the downward arrows represent stimulations to the network for approximately 4 s, where  $I_{\text{ext}} = 5$  if “On burst cell activity,”  $I_{\text{ext}} = -35$  if “Off burst cell activity,” and  $I_{\text{ext}} = -15$  otherwise, as in Fig. 4.2.6.

### 4.2.4.2 COMPARISON WITH REDUCED ODE MODEL

For implementation on an FPGA, the reduced ODE model is discretized using the forward Euler method with a step size of  $\Delta t = 2^{-5}$ , which is among the simplest numerical integration methods, as follows:  $v_k(t + \Delta t) = v_k(t) + \Delta t(g(v_k) + I_k^{\text{syn}} + I_{\text{ext}})$ ,  $u_k(t + \Delta t) = u_k(t) - \Delta t\tau^{-1}u_k(t) + \tau^{-1}\delta_{t,t_k^{(i)}}$ , where  $v_k(t + \Delta t) \leftarrow v_k^{\text{reset}}$  if  $v_k(t) > v_k^{\text{th}}$ . The discretized

Table 4.2.1: Comparisons of Hardware Resources and Power Consumption

	Proposed model implemented in asynchronous sequential logic	Reduced ODE model implemented in custom-designed hardware DSP	Reduced ODE model implemented as software running on custom-designed CPU	Koulakov model [36] implemented in custom-designed hardware DSP
Bit-length	$X_{\max} = 5\text{-bit}$ , $P_{\max} = 10\text{-bit}$ , $Q_{\max} = 8\text{-bit}$	17-bit fixed point number	32-bit floating point number	31-bit fixed-point number
# LUTs	438	3,482	1,124	1,248,229 (estimated)
# FFs	231	223	1,076	4,145 (estimated)
# Slices	146	1,100	438	Unavailable due to oversize
On-chip power	0.166 W	0.229 W	0.286 W	Unavailable due to oversize

On-chip powers are estimated using the Vivado Design Suite v2019.2 at the post-routing stage.

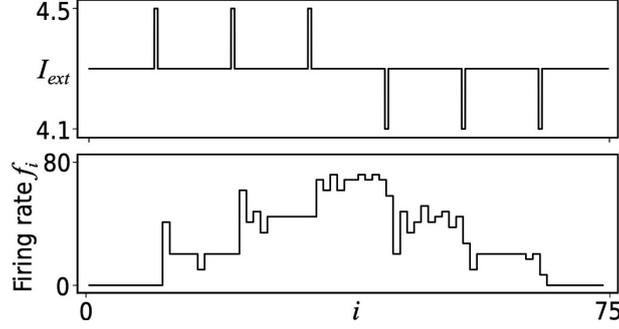


Figure 4.2.8: Neural integration of circuit-based bistability model [36] consisting of three neuronal units. Parameter value of NMDA synaptic conductance  $g_{NMDA,i}$  is  $g_2 = 5 \mu\text{S}/\text{cm}^2$  for interunit connections. Bias current values are  $0 \mu\text{A}/\text{cm}^2$ ,  $3.5 \times 10^{-2} \mu\text{A}/\text{cm}^2$ , and  $7.5 \times 10^{-2} \mu\text{A}/\text{cm}^2$ , respectively, for each unit. Other parameters are the same as those described in the Supplementary Methods of Koulalov *et al.* [36].  $\Delta = 0.097$ .

equation is implemented on the same FPGA in the following two ways.

*Custom-designed DSP* The discretized equation is described as an RTL code using VHDL, and the variables are represented by fixed-point numbers in two’s complement format, where the bit lengths of the state variables are reduced to be as short as possible under the condition that the model can exhibit the neural integration properly. The RTL code is synthesized by the same development environment, and the generated bitstream file is downloaded into the same FPGA.

*Custom-designed CPU* To calculate the discretized equation, a soft-core CPU called Microblaze is custom-designed using the same design environment. In this design, “Minimum area” in the predefined configuration is selected, where an integer multiplier and a floating-point unit are not included. The generated custom-designed CPU is implemented on the same FPGA. The discretized reduced ODE model is written in the C language, and the variables are represented by 32-bit floating numbers. The code is compiled by Xilinx Vitis IDE v2019.2, and the executable file is downloaded into the custom-designed CPU on the FPGA.

#### 4.2.4.3 COMPARISON WITH CIRCUIT-BASED BISTABILITY MODEL [36]

For further comparison, we implemented a circuit-based bistability model as presented by Koulakov *et al.* [36] consisting of the following two-compartment conductance-based neurons:  $C_m \dot{V}_i^s = -I_{Leak,i}^s - I_{Na,i} - I_{KDr,i} - \frac{g_c}{p}(V_i^s - V_i^d)$ ,  $C_m \dot{V}_i^d = -I_{Leak,i}^d - \frac{g_c}{1-p}(V_i^d - V_i^s) - I_{NMDA,i} - I_{ext}$ . The neurons, divided into a number of units (originally hundred) containing three neurons each, are connected all-to-all by NMDA synapses as follows:  $I_{NMDA,i} = g_{NMDA,i} \sum_j s_j (V_i^d - E_{syn}) / (1 + 0.3[\text{Mg}^{2+}] \exp(-0.08V_i^d))$ , where strong and weak NMDA synaptic conductances  $g_{NMDA,i}$  are applied for synapses connecting in the same unit and interunit connections, respectively. Further,  $s_j$  represents an NMDA gating variable whose dynamics are described by the following second-order kinetics:  $\dot{x}_j = F(V_j^s)(1 - x_j) - 0.5x_j$ ,  $\dot{s}_j = 0.5x_j(1 - s_j) - 0.01s_j$ . More details of the model can be found in the Supplementary Methods [36]. For a fair comparison, the number of units in the network is reduced from

100 to 3. Bias current values and NMDA synaptic conductances for interunit connections are arbitrarily chosen, while the other parameters are the same as those in [36]. Fig. 4.2.8 shows the neural integration of the circuit-based bistability model consisting of three units. As with B-1) in this section, the model is discretized using the forward Euler method with a step size  $\Delta t = 2^{-6}$  and is described as an RTL code using VHDL, where the variables are represented by fixed-point numbers in the two's complement format. A natural exponential function is approximated as a fifth-order Taylor polynomial. The bit lengths of the state variables are reduced to be as short as possible under the condition that the model can exhibit the neural integration properly in a simulation. The RTL code is synthesized by the same development environment. As a result, the estimated number of LUTs exceeds the available hardware resources on the FPGA. Thus, the number of slices and the power consumption cannot be measured, while the number of FFs and LUTs can be estimated. Table I summarizes the comparison results of hardware resources and power consumption.

#### 4.2.5 Discussion

Because the integration value is represented by the number of firing neurons (or groups for the circuit-based bistability model [36]), it can be close to a continuous representation if the number of neurons is large. In the case of the implementations shown in Table I, all models represent the integrated values at a resolution of three gradations in maximum performance (see also Figs. 4.2.3, 4.2.6, and 4.2.8). Hence, it can be said that the presented model achieves the same integration performance as that of the other two models while reducing the required hardware resources and power consumption.

Further, the required hardware resources for implementing the presented model and the reduced ODE model on the FPGA are proportional to the network scale. Fig. 4.2.9 shows the relationships between the number of neurons in the network and the FPGA resource utilization for the presented model and the reduced ODE model. The utilization of the LUTs, FFs, and slices for the presented model tends to increase at the same rate as that for the reduced ODE model. The utilization of the LUTs for the presented model is less than that for the reduced ODE model, while the utilizations of the FFs are approximately the same in both models. As a result, the number of slices (each slice contains four six-input LUTs and eight FFs) for the presented model is lower than that for the reduced ODE model independently of the network scales. The power consumption for the presented model is also lower than that for the ODE model, which is expected to be owing to the significant reduction in the number of LUTs.

Owing to the few required hardware resources, the presented model is suitable for implementing two types of major future applications: a neural prosthetic device [7] and a bio-inspired robot [59]. Specifically, an oculomotor system, head direction system, and decision-making system in which NIs are involved are eligible for a range of potential applications in the presented model, and engineering studies have been conducted [52–54]. In the presented model and the previous NI models [36, 47–50], it is necessary to choose the appropriate network size (i.e., the number of bistable units) depending on the application. However, the NI consisting of conductance-based neurons [36, 47–50], while having biological plausibility, is difficult to implement in terms of required hardware resources, as can be seen in Section IV.

In the early stages of a modeling study of a NI, several network models of rate-code

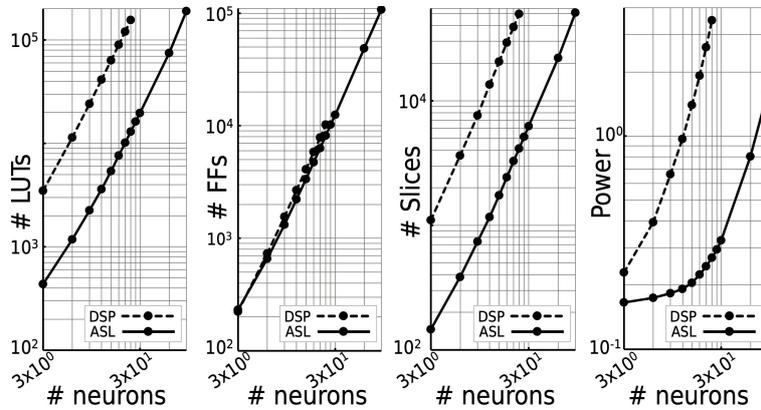


Figure 4.2.9: Utilization of LUTs, FFs, slices, and power consumption as number of neurons increases. Dashed (solid) lines indicate reduced ODE model implemented in custom-designed hardware DSP (presented model is implemented as asynchronous sequential logic circuit).

neurons recurrently connected with positive feedback, of which the dynamics of firing rate are described by an ODE, have been presented, for example, [60,61]. Such models work as a first-order filter with a long leakage time constant and can be easily implemented on a circuit in terms of the required circuit elements. However, the spike-timing mechanism cannot be applied in these types of systems. A persistent activity in the absence of external inputs exhibited by a NI is involved in working memory, as stated in Section I. In fact, several studies have attempted to observe the working memory function that emerges spontaneously by spike-timing-dependent plasticity, for example, [62,63].

Hence, these results suggest that the presented model is reasonably implementable and reasonably biologically plausible, that is, suitable for bio-inspired engineering applications.

## 4.2.6 Conclusions

In this study, a reduced ODE model was derived and a hardware-oriented recurrent network of asynchronous CA neurons for a NI was proposed. Both the models and the circuit-based bistability model were implemented on the same FPGA. It was shown that the proposed model consumes fewer hardware resources and lower power than those of the reduced ODE and circuit-based bistability models.

## Bibliography

- [1] B. S. Wilson, D. T. Lawson, R. D. Wolford, D. K. Eddington, and W. M. Rabinowitz, "Better speech recognition with cochlear implants," *Nature*, vol. 352, no. 18, pp. 236–238, 1991.
- [2] F. G. Zeng, S. Rebscher, W. Harrison, X. Sun, and H. Feng, "Cochlear Implants: System Design, Integration, and Evaluation," *IEEE Reviews in Biomedical Engineering*, vol. 1, pp. 115–142, 2008.

- [3] E. Zrenner, “Will Retinal Implants Restore Vision?” *Science*, vol. 295, no. 5557, pp. 1022–1025, 2002.
- [4] E. Noorsal, K. Sooksood, H. Xu, R. Hornig, J. Becker, and M. Ortmanns, “A Neural Stimulator Frontend With High-Voltage Compliance and Programmable Pulse Shape for Epiretinal Implants,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 1, pp. 244–256, 2012.
- [5] T. W. Berger, R. E. Hampson, D. Song, A. Goonawardena, V. Z. Marmarelis, and S. A. Deadwyler, “A cortical neural prosthesis for restoring and enhancing memory,” *Journal of Neural Engineering*, vol. 8, no. 4, 046017, 2011.
- [6] R. E. Hampson, D. Song, I. Opris, L. M. Santos, Dae C Shin, G. A. Gerhardt, V. Z. Marmarelis, T. W. Berger, and S. A. Deadwyler, “Facilitation of memory encoding in primate hippocampus by a neuroprosthesis that promotes task-specific neural firing,” *Journal of Neural Engineering*, vol. 10, no. 6, 066013, 2013.
- [7] R. E. Hampson, D. Song, B. S. Robinson, D. Fetterhoff, A. S. Dakos, B. M. Roeder, X. She, R. T. Wicks, M. R. Witcher, D. E. Couture, A. W. Laxton, H. Munger-Clary, G. Popli, M. J. Sollman, C. T. Whitlow, V. Z. Marmarelis, T. W. Berger, and S. A. Deadwyler, “Developing a hippocampal neural prosthetic to facilitate human memory encoding and recall,” *Journal of Neural Engineering*, vol. 15, no. 3, 036014, 2018.
- [8] Y. Che, H. Li, C. Han, X. Wei, B. Deng, and J. Wang, “Effects of DC electric fields on neuronal excitability: A bifurcation analysis,” *International Journal of Modern Physics B*, vol. 28, no. 18, 1450114, 2014.
- [9] E. M. Izhikevich, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. Cambridge, MA, USA: MIT Press, 2006.
- [10] E. Farquhar and P. Hasler, “A bio-physically inspired silicon neuron,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 3, pp. 477–488, 2005.
- [11] H. Soleimani, A. Ahmadi, M. Bavandpour, and O. Sharifipour, “A generalized analog implementation of piecewise linear neuron models using CCII building blocks,” *Neural Networks*, vol. 51, pp. 26–38, 2014.
- [12] C. Matsuda and H. Torikai, “A Novel Generalized PWC Neuron Model: Theoretical Analyses and Efficient Design of Bifurcation Mechanisms of Bursting,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 11, pp. 1738–1742, 2018.
- [13] Y. Babacan, F. Kaçar, and K. Gürkan, “A spiking and bursting neuron circuit based on memristor,” *Neurocomputing*, vol. 203, pp. 86–91, 2016.
- [14] A. Wagemakers and M. A.F. Sanjuán, “Electronic circuit implementation of the chaotic Rulkov neuron model,” *Journal of the Franklin Institute*, vol. 350, no. 10, pp. 2901–2910, 2013.

- [15] F. Folowosele, T. J. Hamilton, and R. Etienne-Cummings, "Silicon Modeling of the Mihalaş–Niebur Neuron," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 1915–1927, 2011.
- [16] C. Mayr, J. Partzsch, M. Noack, S. Hanzsche, S. Scholze, S. Hoppner, G. Ellguth, and R. Schuffny, "A Biological-Realtime Neuromorphic System in 28 nm CMOS Using Low-Leakage Switched Capacitor Circuits," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 1, pp. 243–254, 2016.
- [17] N. Korkmaz, İ. Öztürk, and R. Kılıç, "The investigation of chemical coupling in a HR neuron model with reconfigurable implementations," *Nonlinear Dynamics*, vol. 86, no. 3, pp. 1841–1854, 2016.
- [18] A. Zahedi, S. Haghiri, and M. Hayati, "Multiplierless Digital Implementation of Time-Varying FitzHugh–Nagumo Model," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 7, pp. 2662–2670, 2019.
- [19] H. Soleimani and E. M. Drakakise, "An Efficient and Reconfigurable Synchronous Neuron Model," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 1, pp. 91–95, 2018.
- [20] M. Heidarpour, A. Ahmadi, and R. Rashidzadeh, "A CORDIC Based Digital Hardware For Adaptive Exponential Integrate and Fire Neuron," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 1986–1996, 2016.
- [21] R. K. Weinstein, M. S. Reid, and R. H. Lee, "Methodology and Design Flow for Assisted Neural-Model Implementations in FPGAs," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 1, pp. 83–93, 2007.
- [22] K. Takeda and H. Torikai, "A Novel Hardware-Efficient Cochlea Model Based on Asynchronous Cellular Automaton Dynamics: Theoretical Analysis and FPGA Implementation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 9, pp. 1107–1111, 2017.
- [23] —, "A novel hardware-efficient CPG model based on asynchronous cellular automaton," *IEICE Electronics Express*, vol. 15, no. 11, pp. 1–11, 2018.
- [24] T. Matsubara and H. Torikai, "An Asynchronous Recurrent Network of Cellular Automaton-Based Neurons and Its Reproduction of Spiking Neural Network Activities," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 836–852, 2016.
- [25] —, "Asynchronous Cellular Automaton-Based Neuron: Theoretical Analysis and On-FPGA Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 736–748, 2013.
- [26] K. Takeda and H. Torikai, "A novel hardware-efficient spiking neuron model based on asynchronous cellular automaton dynamics exhibiting various nonlinear response curves," in *Proc. of the 2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil, 2018, pp. 1–8.

- [27] T. Hishiki and H. Torikai, “A Novel Rotate-and-Fire Digital Spiking Neuron and its Neuron-Like Bifurcations and Responses,” *IEEE Transactions on Neural Networks*, vol. 22, no. 5, pp. 752–767, 2011.
- [28] T. Noguchi and H. Torikai, “Ghost Stochastic Resonance From an Asynchronous Cellular Automaton Neuron Model,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 2, pp. 111–115, 2013.
- [29] W. Maass, “Networks of spiking neurons: The third generation of neural network models,” *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.
- [30] L. González-Estrada, G. González-Sanmiguel, L. M. Torres-Treviño, and A. Rodríguez, “Implementation of a Single Chaotic Neuron Using an Embedded System,” in *Advances in Computational Intelligence*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, Friedemann Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, I. Batyrshin, and M. G. Mendoza, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 7630, pp. 283–291.
- [31] E. Rodríguez-Orozco, E. García-Guerrero, E. Inzunza-Gonzalez, O. López-Bonilla, A. Flores-Vergara, J. Cárdenas-Valdez, and E. Tlelo-Cuautle, “FPGA-based Chaotic Cryptosystem by Using Voice Recognition as Access Key,” *Electronics*, vol. 7, no. 12, p. 414, 2018.
- [32] I. Ranaee, M. M. Nia, R. Jahantigh, and A. Gharib, “Introducing a new algorithm for medical image encryption based on chaotic feature of cellular automata,” in *Proc. of the 8th International Conference for Internet Technology and Secured Transactions (ICITST)*, London, United Kingdom, 2013, pp. 582–587.
- [33] I. Dogaru, R. Dogaru, and C. Damian, “FPGA implementation of chaotic cellular automaton with binary synchronization property,” in *Proc. of the 2010 8th International Conference on Communications (COMM)*, Bucharest, Romania, 2010, pp. 45–48.
- [34] L. O. Slepova and A. A. Zhilenkov, “Synthesis of model of hardware realization of LIF-model of biological neuron on the basis of FPGA,” in *Proc. of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, Moscow, Russia, 2018, pp. 992–996.
- [35] A. Farokhniaee and E. W. Large, “Mode-locking behavior of Izhikevich neurons under periodic external forcing,” *Physical Review E*, vol. 95, no. 6, p. 062414, 2017.
- [36] A. A. Koulakov, S. Raghavachari, A. Kepecs, and J. E. Lisman, “Model for a robust neural integrator,” *Nature Neuroscience*, vol. 5, no. 8, pp. 775–782, 2002.
- [37] A. Lasota and M. Mackey, *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*. NY, USA: Springer, 1994.
- [38] Y. A. Kuznetsov, *Elements of Applied Bifurcation Theory*, 3rd ed., ser. Applied Mathematical Sciences, S. S. Antman, J. E. Marsden, and L. Sirovich, Eds. New York: Springer, 2004, vol. 112.

- [39] M. D. Bernard, A. R. Champneys, C. J. Budd, and P. Kowalczyk, *Piecewise-Smooth Dynamical Systems: Theory and Applications*. London, U.K.: Springer-Verlag, 2008.
- [40] A. N. Burkitt, “A Review of the Integrate-and-fire Neuron Model: I. Homogeneous Synaptic Input,” *Biological Cybernetics*, vol. 95, no. 1, pp. 1–19, 2006.
- [41] B. Ermentrout, “Type I Membranes, Phase Resetting Curves, and Synchrony,” *Neural Computation*, vol. 8, no. 5, pp. 979–1001, 1996.
- [42] E. Aksay, G. Gamkrelidze, H. S. Seung, R. Baker, and D. W. Tank, “In vivo intracellular recording and perturbation of persistent activity in a neural integrator,” *Nature Neuroscience*, vol. 4, no. 2, pp. 184–193, 2001.
- [43] R. Romo, C. D. Brody, A. Hernández, and L. Lemus, “Neuronal correlates of parametric working memory in the prefrontal cortex,” *Nature*, vol. 399, no. 6735, pp. 470–473, 1999.
- [44] M. N. Shadlen and W. T. Newsome, “Neural Basis of a Perceptual Decision in the Parietal Cortex (Area LIP) of the Rhesus Monkey,” *Journal of Neurophysiology*, vol. 86, no. 4, pp. 1916–1936, 2001.
- [45] J. S. Taube, “Persistent Neural Activity in Head Direction Cells,” *Cerebral Cortex*, vol. 13, no. 11, pp. 1162–1172, 2003.
- [46] A. V. Egorov, B. N. Hamam, E. Fransén, M. E. Hasselmo, and A. A. Alonso, “Graded persistent activity in entorhinal cortex neurons,” *Nature*, vol. 420, no. 6912, pp. 173–178, 2002.
- [47] H. S. Seung, D. D. Lee, B. Y. Reis, and D. W. Tank, “Stability of the Memory of Eye Position in a Recurrent Network of Conductance-Based Model Neurons,” *Neuron*, vol. 26, no. 1, pp. 259–271, 2000.
- [48] Y. Loewenstein and H. Sompolinsky, “Temporal integration by calcium dynamics in a model neuron,” *Nature Neuroscience*, vol. 6, no. 9, pp. 961–967, 2003.
- [49] M. S. Goldman, “Robust Persistent Neural Activity in a Model Integrator with Multiple Hysteretic Dendrites per Neuron,” *Cerebral Cortex*, vol. 13, no. 11, pp. 1185–1195, 2003.
- [50] J. N. Teramae and T. Fukai, “A Cellular Mechanism for Graded Persistent Activity in a Model Neuron and Its Implications in Working Memory,” *Journal of Computational Neuroscience*, vol. 18, no. 1, pp. 105–121, 2005.
- [51] M. Nikitchenko and A. Koulakov, “Neural Integrator: A Sandpile Model,” *Neural Computation*, vol. 20, no. 10, pp. 2379–2417, 2008.
- [52] T. K. Horiuchi and C. Koch, “Analog VLSI-Based Modeling of the Primate Oculomotor System,” *Neural Computation*, vol. 11, no. 1, pp. 243–265, 1999.

- [53] T. M. Massoud and T. K. Horiuchi, “A Neuromorphic VLSI Head Direction Cell System,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 1, pp. 150–163, 2011.
- [54] W. Wojtak, F. Ferreira, P. Vicente, L. Louro, E. Bicho, and W. Erhagen, “A neural integrator model for planning and value-based decision making of a robotics assistant,” *Neural Computing and Applications*, vol. 33, no. 8, pp. 3737–3756, 2021.
- [55] H. Bao, C. Chen, Y. Hu, M. Chen, and B. Bao, “2-D Piecewise-Linear Neuron Model,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 4, pp. 1453–1457, 2021.
- [56] B. Bao, Y. Zhu, C. Li, H. Bao, and Q. Xu, “Global multistability and analog circuit implementation of an adapting synapse-based neuron model,” *Nonlinear Dynamics*, vol. 101, no. 2, pp. 1105–1118, 2020.
- [57] K. Takeda and H. Torikai, “A Novel Asynchronous CA Neuron Model: Design of Neuron-Like Nonlinear Responses Based on Novel Bifurcation Theory of Asynchronous Sequential Logic Circuit,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 6, pp. 1989–2001, 2020.
- [58] K. Takeda and H. Torikai, “A Novel Hardware-Efficient Central Pattern Generator Model Based on Asynchronous Cellular Automaton Dynamics for Controlling Hexapod Robot,” *IEEE Access*, vol. 8, pp. 139 609–139 624, 2020.
- [59] T. M. Morse, S. R. Lockery, and T. C. Ferrée, “Robust Spatial Navigation in a Robot Inspired by Chemotaxis in *Caenorhabditis elegans*,” *Adaptive Behavior*, vol. 6, no. 3-4, pp. 393–410, 1998.
- [60] B. Y. Kamath and E. L. Keller, “A neurological integrator for the oculomotor control system,” *Mathematical Biosciences*, vol. 30, no. 3-4, pp. 341–352, 1976.
- [61] S. C. Cannon, D. A. Robinson, and S. Shamma, “A proposed neural network for the integrator of the oculomotor system,” *Biological Cybernetics*, vol. 49, no. 2, pp. 127–136, 1983.
- [62] K. Kitano, H. Câteau, and T. Fukai, “Self-organization of memory activity through spike-timing-dependent plasticity,” *Neuroreport*, vol. 13, no. 6, pp. 795–798, 2002.
- [63] B. Szatmáry and E. M. Izhikevich, “Spike-Timing Theory of Working Memory,” *PLoS Computational Biology*, vol. 6, no. 8, p. e1000879, 2010.

# Chapter 5

## Neural Spike-Train Generator Model based on Quantum-dot Cellular Automaton

### 5.1 Neural Spike-train Generator Suitable for UWB-IR Applications <sup>1</sup>

#### 5.1.1 Introduction

Most computer devices are basically operated on the basis of the metal-oxide semiconductor field-effect transistor (MOSFET). The quantum-dot cellular automaton (QCA) is nanoscale technology, which has been expected to be alternative to the conventional MOSFET based architecture [1–4]. Fig. 5.1.1(a) shows a basic element of the QCA called a QCA cell. As shown in the figure, the QCA cell has four quantum dots arranged in a square pattern, where two dots are assumed to have electrons and the other dots are assumed to have no electrons. Due to Coulomb forces, there are two stable patterns of the dots as shown in Fig. 5.1.1(a), where these patterns are said to have polarizations  $Pol = +1$  and  $Pol = -1$ . Hence, the QCA cell can work as a binary memory, where the polarizations  $Pol = +1$  and  $Pol = -1$  may represent binary states  $Logic = 1$  and  $Logic = 0$ , respectively. In Fig. 5.1.1(b), QCA cells are located adjacent to each other. As shown in the figure, there are two stable patterns of the quantum dots in the located QCA cells due to the Coulomb forces. These patterns can be switched by appropriately low potential barriers among the dots and applying external Coulomb forces. Typically, the potential barriers are controlled by four-phase synchronized clocks, where the QCA cells driven by the same clock are said to belong to the same clock zone as shown in Fig. 5.1.1(c). By applying the four-phase synchronized clocks to the four clock zones appropriately, binary information can be transmitted through the QCA cells. It should be emphasized that the QCA cells can transmit the binary information without using current flows, whereas the conventional MOSFET based architecture can not avoid energy consumption due to current flows. This ultra low energy consumption property is one of the most significant advantages of the QCA based architecture. Then, using the QCA cells, many memory units and memoryless units have been designed so far [5–11]. In order to realize proper functions of such QCA units, designers are recommended to obey the following design guidelines [12–15].

---

<sup>1</sup>This section is based on “A novel spike-train generator suitable for QCA implementation towards UWB-IR applications,” by the same author, which appeared in *Nonlinear Theory and Its Applications*, IEICE, vol. 9, no. 4, pp.436–452, 2018, Copyright(C)2018 IEICE.

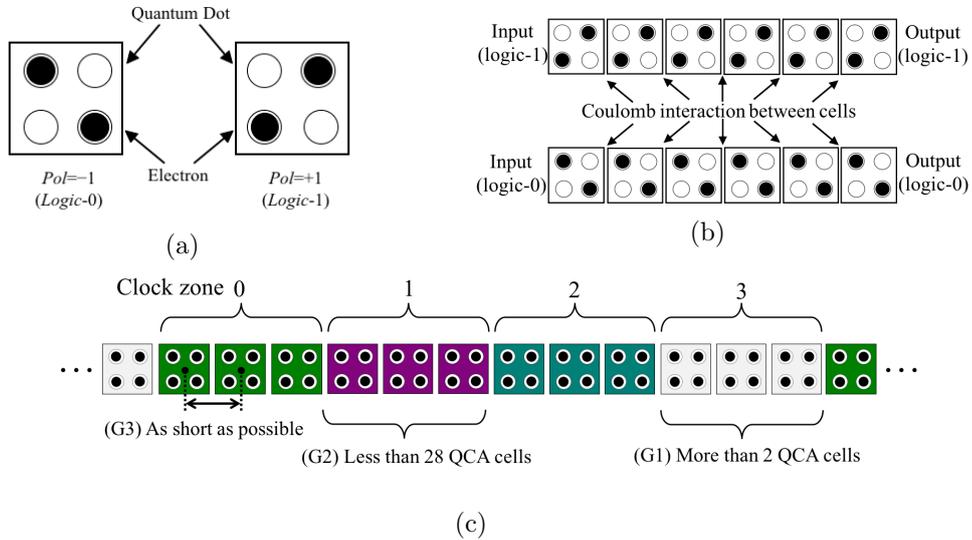


Figure 5.1.1: Quantum-dot cellular automaton (QCA) [1–4]. (a) QCA cell. The QCA cell has four quantum dots and two of them have electrons. There are two stable states characterized by polarizations  $Pol = +1$  and  $Pol = -1$ , which correspond to binary states  $logic = +1$  and  $logic = -1$ , respectively. (b) QCA cells located adjacent to each other. There are two stable states of the quantum dots. By changing potential barriers and applying external Coulomb forces, binary information can be transmitted without using current flows. (c) QCA circuit design guidelines (G1)–(G3) [12–15].

**(G1)** The number of adjacently located QCA cells in the same clock zone should be more than or equal to 2 as shown in Fig. 5.1.1(c).

**(G2)** The number of adjacently located QCA cells in the same clock zone should be less than or equal to 28 at a clock frequency of 1 THz as shown in Fig. 5.1.1(c).

**(G3)** The center-to-center distance between two adjacently located QCA cells should be as short as possible as shown in Fig. 5.1.1(c).

The cellular automaton (CA) has been investigated intensively from both fundamental scientific viewpoint and engineering application viewpoint. For example, since the CA is different from traditional dynamical systems such as ordinary and partial differential equations but can exhibit a huge variety of spatio-temporal phenomena, it is even called a “*new kind of science*” [16–18]. Also, the CA has been applied to many engineering systems such as traffic flow model, image classification, and music generation [19–21]. Among such applications, this paper focuses on application of the CA to spike-train generators and ultra wide band impulse radio (UWB-IR) communication, ranging, and positioning systems. The UWB-IR systems have been applied to various engineering systems such as vehicular radar, wireless sensor networks, and position estimation, where their advantages include high data rate, low power consumption, and high resistivity to noise [22–24]. Recall that the QCA is a nanoscale device and has the ultra low energy consumption property. So, if we design a UWB-IR system based on the QCA, such a system is expected to have the ultra small circuit area property and the ultra low power consumption property. An important building block

of such a QCA based UWB-IR system is a QCA based spike-train generator that generates spike-trains suitable for the UWB-IR. Our group has been developing design methods of the CA to generate spike-trains suitable for the UWB-IR systems [25–27]. However, these design methods can not be applied to the QCA based architecture since these methods can not handle transmission delays among QCA cells and thus they can not satisfy the design guidelines G1 and G2. In order to overcome this difficulty, this paper aims at

- proposing a novel CA spike-train generator that can be implemented by the QCA and can generate spike-trains with various spike patterns;
- proposing a rigorous analysis method of the proposed spike-train generator; and
- proposing a stochastic algorithm for parameter tuning for the proposed spike-train generator so that it can generate spike-trains suitable for UWB-IR systems.

This paper is organized as follows. In Section II, a novel spike-train generator suitable for the QCA implementation is proposed. Also, a rigorous analysis tool of the proposed generator is proposed. In Section III, a QCA layout of the proposed spike-train generator is designed and its operation is verified by a QCA simulator called QCADesigner [28]. Using the analysis tool, it is shown that the proposed generator can generate spike-trains with various spike patterns in terms of periods of spike-trains, second peaks of auto-correlation functions, and numbers of spikes. In Section IV, a parameter tuning algorithm for the proposed generator is proposed, where the analysis tool is used as its subroutine to accelerate parameter tuning speed. It is shown that the parameter tuning algorithm enables the proposed generator to generate spike-trains suitable for applications to the UWB-IR systems. Also, the resulting generator is designed as a QCA layout and its operation is verified by the QCA simulator.

## 5.1.2 Model description

### 5.1.2.1 PROPOSED SPIKE-TRAIN GENERATOR

In this subsection, a novel spike-train generator suitable for quantum-dot cellular automaton (QCA) implementation is proposed. The proposed generator has the following discrete time.

$$t = 0, 1, 2, \dots$$

Fig. 5.1.2(a) shows a circuit diagram of the proposed generator. As shown in the figure, the generator has  $M$  cells ( $M > 0$ ) called  $p$ -cells, which have the following binary states.

$$p_i(t) \in \mathbf{B} = \{0, 1\},$$

where  $i \in \{0, 1, \dots, M-1\}$  is an index of the  $p$ -cell. For simplicity, the following vector form of the binary states is introduced.

$$\mathbf{P}(t) = (p_0(t), \dots, p_{M-1}(t))^T \in \mathbf{B}^M.$$

As shown in Fig. 5.1.2(a), the  $p$ -cells are ring-coupled and thus the dynamics of the  $p$ -cells is described by

$$\mathbf{P}(t+1) = (p_{M-1}(t), p_0(t), \dots, p_{M-2}(t))^T. \quad (5.1.1)$$

In this paper, the initial conditions of the  $p$ -cells are fixed to  $\mathbf{P}(0) = (1, 0, \dots, 0)$  and thus the  $p$ -cells oscillate periodically with period  $M$  as shown in Fig. 5.1.2(b). As shown in Fig. 5.1.2(a), the proposed generator has reconfigurable wires from  $M$  left terminals (connected from the  $p$ -cells) to  $N$  ( $N \geq M$ ) right terminals (connected to the reset unit). It is assumed that each left terminal has one wire and each right terminal can accept any number of wires. Then the wires transform the binary state vector  $\mathbf{P}(t) \in \mathbf{B}^M$  of the  $p$ -cells into the following binary signal vector.

$$\mathbf{b}(t) \in \mathbf{B}^N = (b_0(t), \dots, b_{N-1}(t))^T.$$

In order to characterize the transformation by the wires, the following function  $a(j, i) : \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\} \rightarrow \mathbf{B}$  and its matrix form  $\mathbf{A}$  are introduced.

$$a(j, i) = \begin{cases} 1 & \text{if the } j\text{-th left terminal is wired to the } i\text{-th right terminal,} \\ 0 & \text{otherwise,} \end{cases}$$

$$\mathbf{A} = \begin{pmatrix} a(0, 0) & a(1, 0) & \cdots & a(N-1, 0) \\ a(0, 1) & a(1, 1) & \cdots & a(N-1, 1) \\ \vdots & \vdots & \ddots & \vdots \\ a(0, M-1) & a(1, M-1) & \cdots & a(N-1, M-1) \end{pmatrix}.$$

For example, the proposed generator in Fig. 5.1.2(a) is characterized by the following matrix  $\mathbf{A}$ .

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (5.1.2)$$

Then the binary state vector  $\mathbf{P}(t)$  is transformed into the binary signal vector  $\mathbf{b}(t)$  as follows.

$$\mathbf{b}(t) = \mathbf{A}\mathbf{P}(t). \quad (5.1.3)$$

In Fig. 5.1.2(b), black circles represent a time waveform of the binary signal vector  $\mathbf{b}(t)$ , which corresponds to the pattern of the wires in Fig. 5.1.2(a). As noted in Fig. 5.1.2(a), the signal  $\mathbf{b}(t) = (b_0(t), \dots, b_{N-1}(t))^T$  is transmitted to the reset unit with delay  $\rho$ . Then the reset unit accepts the following signal

$$\mathbf{b}(t - \rho) = (b_0(t - \rho), \dots, b_{N-1}(t - \rho))^T.$$

As shown in Fig. 5.1.2(a), the proposed generator has  $N$  cells called  $x$ -cells, which have the following tri-state states.

$$x_j(t) \in \mathbf{T} = \{0, 1, \phi\},$$

where  $j \in \{0, 1, \dots, N-1\}$  is an index of the  $x$ -cell. Also, “0” and “1” correspond to those in the binary set  $\mathbf{B}$ , while “ $\phi$ ” is used to represent a delay in the reset unit. For simplicity,

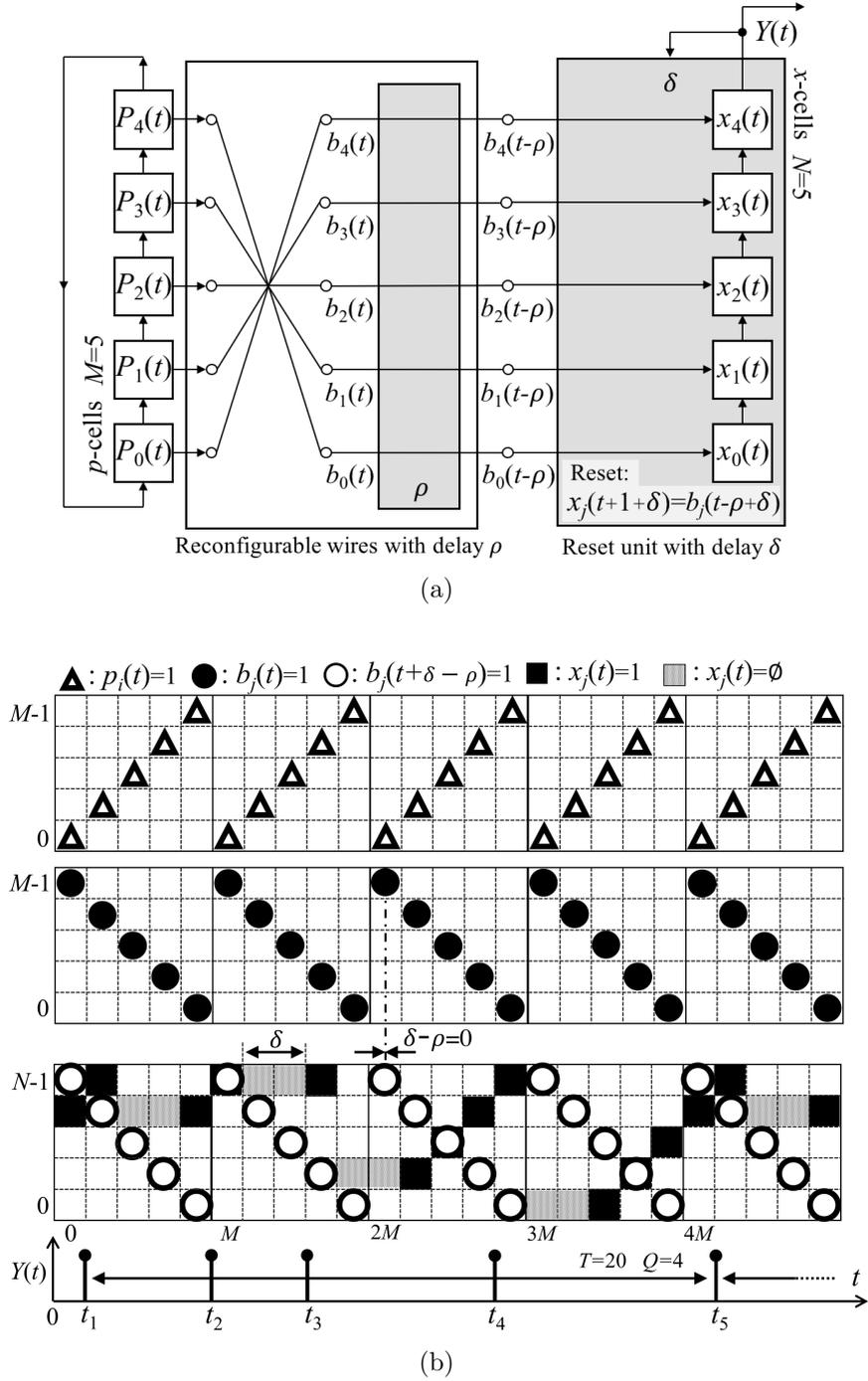


Figure 5.1.2: A novel spike-train generator suitable for quantum-dot cellular automaton (QCA) implementation. (a) Circuit diagram. The generator consists of  $M$   $p$ -cells,  $N$   $x$ -cells,  $M$  reconfigurable wires with delay  $\rho$ , and a reset unit with delay  $\delta$ , where  $M = N = 5$  in this figure. The wires are characterized by the matrix  $\mathbf{A}$  in Eq. (5.1.2). (b) Typical time waveforms.  $\rho = 2$  and  $\delta = 2$ . The white triangle represents  $p_i(t) = 1$ . The black circle represents  $b_j(t) = 1$ . The white circle represents  $b_j(t + \delta - \rho) = 1$ . The black box represents  $x_j(t) = 1$  and the gray box represents  $x_j(t) = \phi$ . The generator outputs a spike-train  $Y(t)$  with the period  $T = 20$  and the number  $Q = 4$  of spikes.

the following vector form of the tri-state states is introduced.

$$\mathbf{X}(t) = (x_0(t), \dots, x_{N-1}(t))^T \in \mathbf{T}^N.$$

As shown in Fig. 5.1.2(a), the  $x$ -cells are basically ring-coupled and then obey the following dynamics.

$$\text{If } \mathbf{X}(t) \neq (0, \dots, 0, 1), \text{ then } \mathbf{X}(t+1) = S(\mathbf{X}(t)), \quad (5.1.4)$$

where

$$S((x_0, \dots, x_{N-1})^T) = (0, x_0, \dots, x_{N-2})^T$$

is a shift operator. As shown in Fig. 5.1.2(a), the  $x$ -cells accept the binary signal vector  $\mathbf{b}(t-\rho)$  via the reset unit and then additionally obey the following dynamics.

If  $\mathbf{X}(t) = (0, \dots, 0, 1)$ , then

$$\mathbf{X}(t+1+d) = \begin{cases} (\phi, \dots, \phi) & \text{for } d \in \{0, 1, \dots, \delta-1\}, \\ \mathbf{b}(t+\delta-\rho) & \text{for } d = \delta, \end{cases} \quad (5.1.5)$$

where  $\delta \in \{1, 2, \dots, \rho\}$  is a parameter characterizing the delay of the reset unit, and is called a reset delay. In Fig. 5.1.2(b), black and gray boxes represent a time waveform of the tri-state state vector  $\mathbf{X}(t)$ , which corresponds to the pattern of the wires in Fig. 5.1.2(a). When the black box is below the highest position  $N-1$ , the black box is shifted upward due to Eq. (5.1.4). When the black box reaches the highest position  $N-1$ , the black box is reset to the white circle with the reset delay  $\delta$  due to Eq. (5.1.5). Repeating such dynamics, the state vector  $\mathbf{X}(t)$  oscillates as shown in Fig. 5.1.2(b). In addition, depending on the discrete state  $\mathbf{X}(t)$ , the proposed generator outputs the following spike-train  $Y(t)$  as shown in Fig. 5.1.2(b).

$$Y(t) = \begin{cases} 1 & \text{if } \mathbf{X}(t) = (0, \dots, 0, 1), \\ 0 & \text{otherwise.} \end{cases} \quad (5.1.6)$$

As a result, the proposed generator can be summarized as follows.

Time:	Discrete time $t$
States:	Binary states $\mathbf{P}(t)$ and Tri-state states $\mathbf{X}(t)$
Dynamics:	Eqs. (5.1.1), (5.1.3), (5.1.4), and (5.1.5)
Output:	Spike-train $Y(t)$ in Eq. (5.1.6)
Parameters:	Numbers $M$ and $N$ of the cells, Matrix $\mathbf{A}$ characterizing the wires, Transmission delay $\rho$ , and Reset delay $\delta$

**Remark 1 (Novelty):** It should be emphasized that our previous spike-train generators [25–27] can not be implemented by the QCA since they can not handle transmission delays among QCA cells and thus they can not satisfy the design guidelines G1 and G2. In order to overcome this difficulty, in this paper, the delays  $\rho$  and  $\delta$  are introduced in the reconfigurable wires and the reset unit, respectively. Due to the delays  $\rho$  and  $\delta$ , the proposed spike-train generators can not be analyzed by analysis methods in [25–27]. In order to overcome this difficulty, in the next subsection, a novel rigorous analysis tool of the proposed generator is proposed.

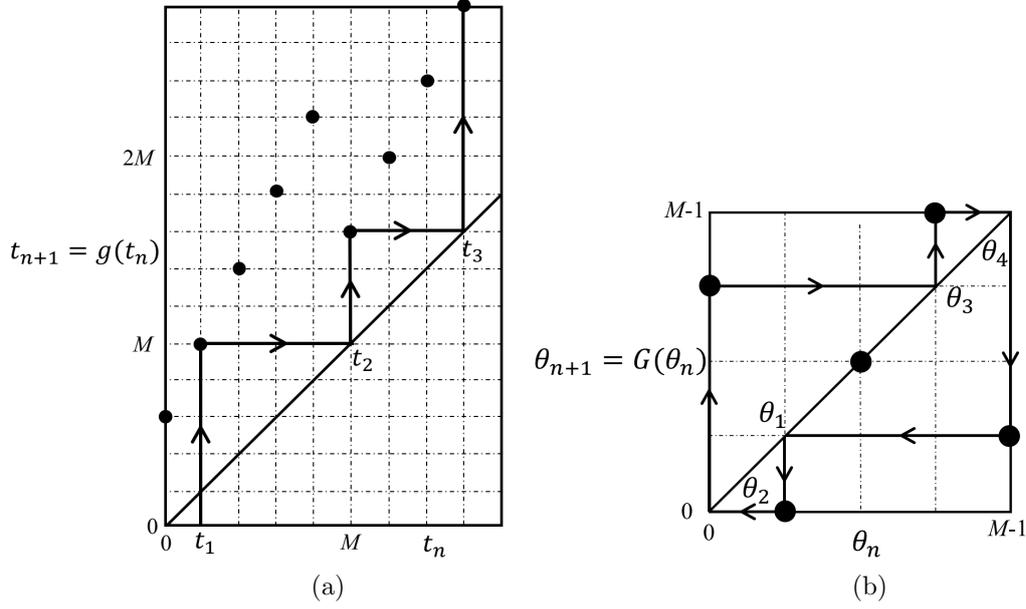


Figure 5.1.3: Spike maps corresponding to the proposed spike-train generator in Fig. 5.1.2(a), where the wires are characterized by the matrix  $\mathbf{A}$  in Eq. (5.1.2). (a) Spike position map  $g$ . (b) Spike phase map  $G$ .

### 5.1.2.2 NOVEL SPIKE MAPS FOR THE PROPOSED GENERATOR

As shown in Fig. 5.1.2(b), let  $t_n$  denote the  $n$ -th spike position of the output  $Y(t)$  of the generator. Then the dynamics of the spike position  $t_n$  can be described by the following spike position map  $g$ .

$$\begin{aligned} t_{n+1} = g(t_n) &= t_n + M - \beta(t_n + \delta - \rho(\text{mod } M)) + \delta, \\ g : \{0, 1, 2, \dots\} &\rightarrow \{0, 1, 2, \dots\}, \end{aligned} \quad (5.1.7)$$

where

$$\begin{aligned} \beta(j) &= i \quad \text{if the } j\text{-th left terminal is wired to the } i\text{-th right terminal,} \\ \beta : \{0, 1, \dots, M-1\} &\rightarrow \{0, 1, \dots, N-1\}. \end{aligned}$$

Fig. 5.1.3(a) shows an example of the spike position map  $g$ . Note that, by iterating the spike position map  $g$ , the spike position  $t_n$  ever increases and thus  $g$  is not suitable for analysis of the proposed generator. So, the following spike phase  $\theta_n$  is introduced.

$$\theta_n = t_n(\text{mod } M).$$

The dynamics of the spike phase  $\theta_n$  can be described by the following spike phase map  $G$  from a finite set into itself.

$$\begin{aligned} \theta_{n+1} = G(\theta_n) &= g(\theta_n) \pmod{M}, \\ G : \{0, 1, \dots, M-1\} &\rightarrow \{0, 1, \dots, M-1\}. \end{aligned} \quad (5.1.8)$$

Fig. 5.1.3(b) shows an example of the spike phase map  $G$ . Using the spike phase map  $G$ , fundamental characteristics (e.g., cycle  $Q$  of a periodic orbit and number of co-existing periodic orbits) of the spike phase  $\theta_n$  can be analyzed rigorously. For example, it can be shown that the spike phase map  $G$  in Fig. 5.1.3(b) has the following periodic orbit with cycle  $Q = 4$ .

$$(\theta_1, \theta_2, \theta_3, \theta_4) = (1, 0, 3, 4).$$

Then, using the periodic orbit  $(\theta_1, \dots, \theta_Q)$ , the corresponding sequence of the spike position  $t_n$  can be derived as follows.

$$t_n = \theta_1 + \sum_{k=1}^{n-1} (M - \beta(\theta_k - \rho(\bmod M)) + \delta). \quad (5.1.9)$$

For example, using the periodic orbit  $(\theta_1, \theta_2, \theta_3, \theta_4) = (1, 0, 3, 4)$  of the spike phase map  $G$  in Fig. 5.1.3(b), the following sequence of the spike position  $t_n$  can be derived.

$$(t_1, t_2, t_3, t_4) = (1, 5, 8, 14).$$

It can be confirmed that these spike positions are identical with those in Fig. 5.1.2(b). In addition, the period  $T$  of the spike train  $Y(t)$  can be derived as follows.

$$T = t_{Q+1} - t_1 = \sum_{k=1}^Q (M - \beta(\theta_k - \rho(\bmod M)) + \delta) - \theta_1. \quad (5.1.10)$$

For example, using the above mentioned periodic orbit  $(\theta_1, \theta_2, \theta_3, \theta_4) = (1, 0, 3, 4)$ , the period  $T = 20$  can be obtained and it can be confirmed that so obtained period  $T = 20$  is identical with that in Fig. 5.1.2(b).

**Remark 2 (Significance of the spike phase map):** As explained in this subsection, the spike phase map  $G$  can be used as a rigorous analysis tool for the proposed spike-train generator. It should be emphasized that the spike phase map  $G$  will be also utilized as a kind of subroutine to accelerate execution speed of a parameter tuning algorithm in section 4. As a result, it can be said that the spike phase map  $G$  is useful not only for analysis but also for design of the proposed spike-train generator.

## 5.1.3 QCA implementation of the proposed spike-train generator

### 5.1.3.1 QCA BASICS

In this section, a brief of QCA basics is introduced. Fig. 5.1.4(a) shows a QCA layout of a majority gate consisting of five QCA cells, where  $A$ ,  $B$ , and  $C$  are inputs and  $M = AB + BC + CA$  is an output. If  $C$  is fixed to 1 (i.e. polarization of the cell corresponding to  $C$  is fixed to 1), the majority gate works as an OR gate as shown in Fig. 5.1.4(b). If  $C$  is fixed to 0 (i.e. polarization of the cell corresponding to  $C$  is fixed to  $-1$ ), the majority gate works as an AND gate as shown in Fig. 5.1.4(c). Furthermore, by arranging QCA cells appropriately, other logic gates and complicated logic functions can be designed [5–8]. In order to design a D-type flip-flop by the QCA cells, we introduce the following definition.

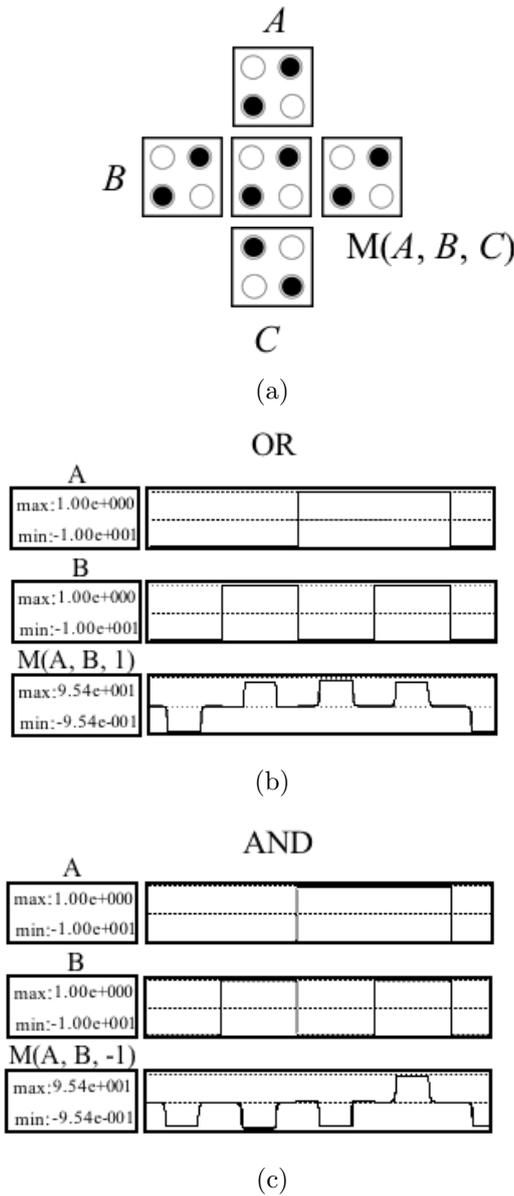


Figure 5.1.4: (a) QCA layout of a majority gate. The majority gate has three inputs  $A$ ,  $B$ , and  $C$ , and an output  $M = AB + BC + CA$  [1–4]. (b) and (c) show simulation results of the majority gate obtained by the QCADesigner. (b) The majority gate works as an OR gate if the input  $C$  is fixed to 0. (c) The majority gate works as an AND gate if the input  $C$  is fixed to 1.

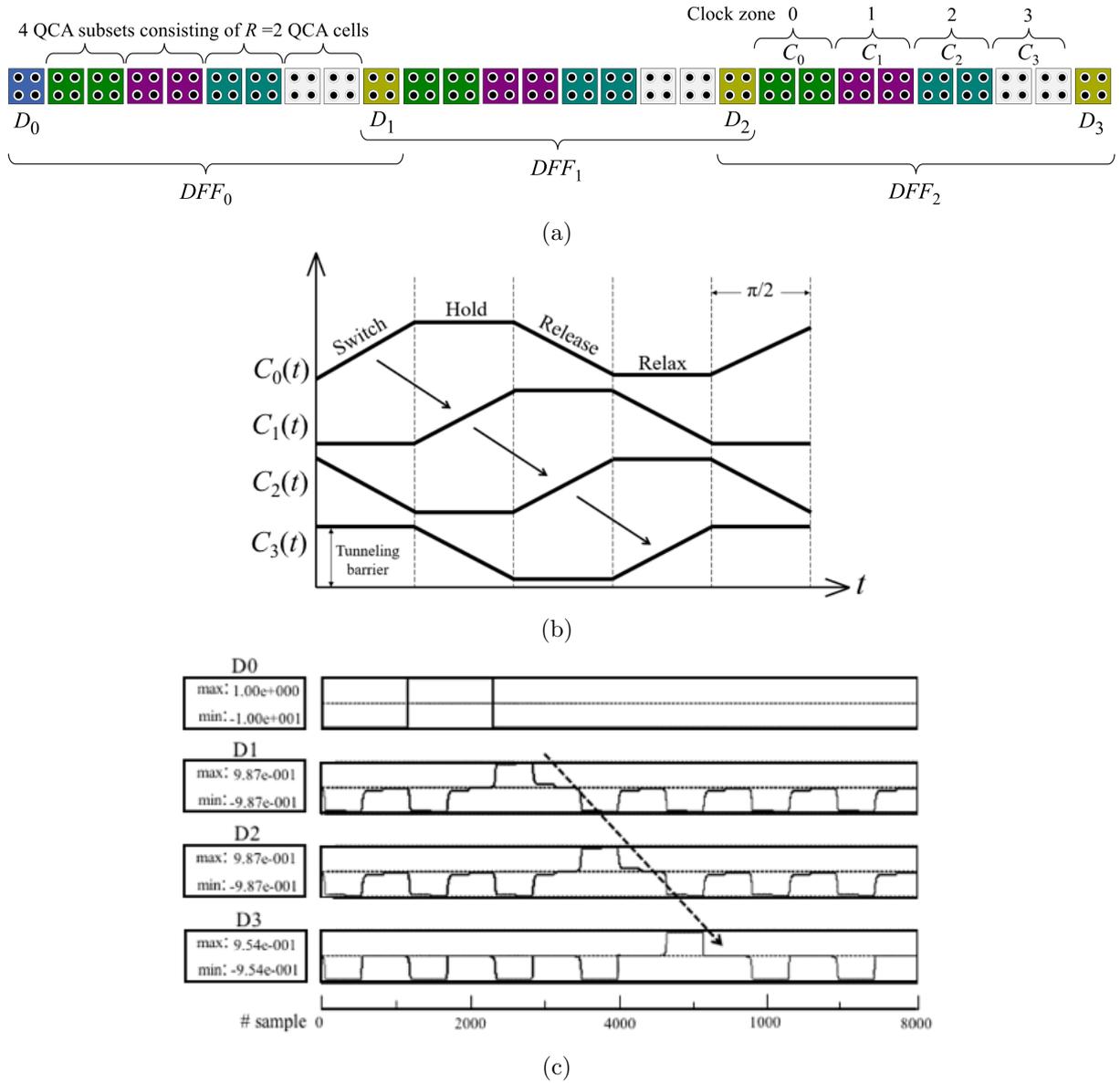


Figure 5.1.5: (a) QCA layout of a shift register consisting of D-type flip-flops  $DFF_0$ ,  $DFF_1$ , and  $DFF_2$ . (b) Clocks  $C_0(t)$ ,  $C_1(t)$ ,  $C_2(t)$ , and  $C_3(t)$ . (c) Typical time waveforms obtained by the QCADesigner. It can be seen that the QCA layout works as a shift register.

**Definition:** Suppose  $R$  QCA cells are arranged in serial and is said to form a QCA subset as shown in Fig. 5.1.5(a), where  $2 \leq R \leq 28$ . Suppose four QCA subsets are arranged in serial as shown in Fig. 5.1.5(a). Suppose a QCA cell  $D_i$  is arranged next to an end point of the four QCA subsets and a QCA cell  $D_{i+1}$  is arranged next to the other end point of the QCA subsets as shown in Fig. 5.1.5(a). Suppose the clock  $C_0(t)$  is applied to the QCA subset next to the QCA cell  $D_i$  as shown in Fig. 5.1.5(a). Suppose the clocks  $C_1(t)$ ,  $C_2(t)$  and  $C_3(t)$  are applied to the other QCA subsets in serial order as shown in Fig. 5.1.5(a). The set of  $4R + 2$  QCA cells arranged in the above manner is said to form a set  $DFF_i$ .

The set  $DFF_i$  works as the D-type flop-flop. Fig. 5.1.5(a) shows a QCA layout of a shift register consisting of  $DFF_i$ , where  $D_0$ ,  $D_1$ ,  $D_2$  are inputs of  $DFF_0$ ,  $DFF_1$ ,  $DFF_2$ , respectively, and  $D_1$ ,  $D_2$  and  $D_3$  are outputs of  $DFF_0$ ,  $DFF_1$ ,  $DFF_2$ , respectively. Fig. 5.1.5(b) explains how to provide clocks to the flip-flop. When a tunneling barrier between quantum-dots is high, the electrons in the dots are tightly confined and thus the polarization is locked. On the other hand, when the tunneling barrier is low, the electrons can be rearranged in the dots and thus the polarization can be changed. Hence, in order to work the set  $DFF_i$  of ten QCA cells as the flip-flops, it is necessary to provide four tunneling barriers with appropriate temporal patterns, where such tunneling barriers are called clocks. Fig. 5.1.5(b) shows examples of such clocks  $C_0(t)$ ,  $C_1(t)$ ,  $C_2(t)$ , and  $C_3(t)$ . Each clock  $C_k(t)$  has four phases called *Switch*, *Hold*, *Release*, and *Relax*. If a clock  $C_k(t)$  is in the *Hold* and the *Relax* phases, the tunneling barrier is high and low, respectively. As shown in Fig. 5.1.5(b), the four clocks  $C_0(t)$ ,  $C_1(t)$ ,  $C_2(t)$ , and  $C_3(t)$  are assumed to be synchronized with different phases. Also, as shown in Fig. 5.1.5(a), these four clocks are applied to the ten QCA cells in each set  $DFF_i$  of QCA cells. This clocking method realizes that each set  $DFF_i$  works as a D-type flip-flop. As shown in Fig. 5.1.5(a), the D-type flip-flops are connected in serial and then they work as a shift register. Fig. 5.1.5(c) shows a simulation result of the shift register, which is obtained by the QCA simulator QCADesigner [28]. It can be seen that the QCA layout in Fig. 5.1.5(a) works as a shift register, where  $D_0$  is its input. Other types of sequential logics can be also designed in a similar fashion [9–11].

### 5.1.3.2 QCA IMPLEMENTATION OF THE PROPOSED SPIKE-TRAIN GENERATOR

Fig. 5.1.6(a) shows a QCA layout of the proposed generator in Fig. 5.1.2(a). In this layout, the upper loop and the lower loop correspond to the  $p$ -cells and the  $x$ -cells in Fig. 5.1.2(a), respectively. Also, crossover cells are introduced in order to realize crosses of unconnected wires [4, 28], and delay cells (i) and (ii) are introduced in order to adjust the phases of the clocks  $C_0(t)$ ,  $C_2(t)$ ,  $C_3(t)$ , and  $C_4(t)$  appropriately. The delays caused by the delay cells (i) correspond to the transmission delay  $\rho$  of the reconfigurable wires in Fig. 5.1.2(a). The delays caused by the delay cells (ii) and the crossover cells correspond to the reset delay  $\delta$  of the reset unit in Fig. 5.1.2(a). Fig. 5.1.6(b) shows typical time waveforms of the QCA layout in Fig. 5.1.6(a) obtained by the QCADesigner. Comparing Fig. 5.1.6(b) with Fig. 5.1.2(b), it can be confirmed that the spike-train  $Y(t)$  generated by the QCA layout in Fig. 5.1.6(a) is equivalent to the spike-train  $Y(t)$  generated by the proposed generator in Fig. 5.1.2(a). It should be emphasized that the proposed generator with other values of the parameters  $(M, N, \mathbf{A}, \rho, \delta)$  can be also implemented by the QCA layout in the same fashion.

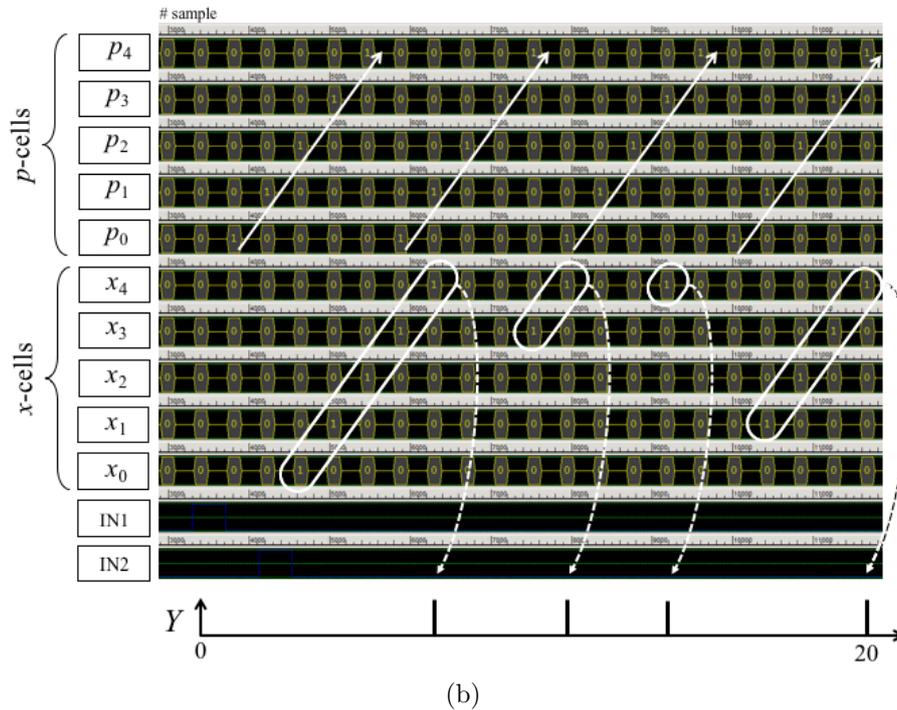
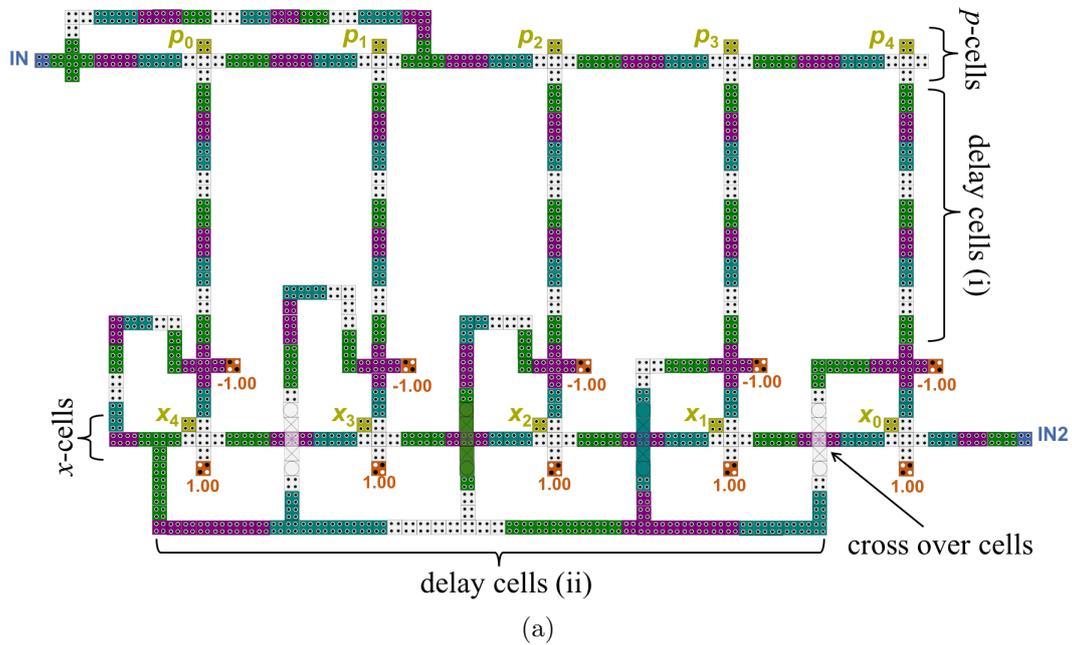


Figure 5.1.6: (a) QCA layout of the proposed generator in Fig. 5.1.2(a). The upper loop and lower loop correspond to the  $p$ -cells and the  $x$ -cells, respectively. The crossover cells realize crosses of unconnected wires [4, 28]. The delay cells are introduced in order to adjust the phases of the clocks. The delays caused by the delay cells (i) correspond to the transmission delay  $\rho$ . The delays caused by the delay cells (ii) and the crossover cells correspond to the reset delay  $\delta$ . (b) Typical time waveforms obtained by the QCADesigner. By comparing with Fig. 5.1.2(b), it can be confirmed that the QCA layout in (a) can realize the dynamics of the proposed generator in Fig. 5.1.2(a).

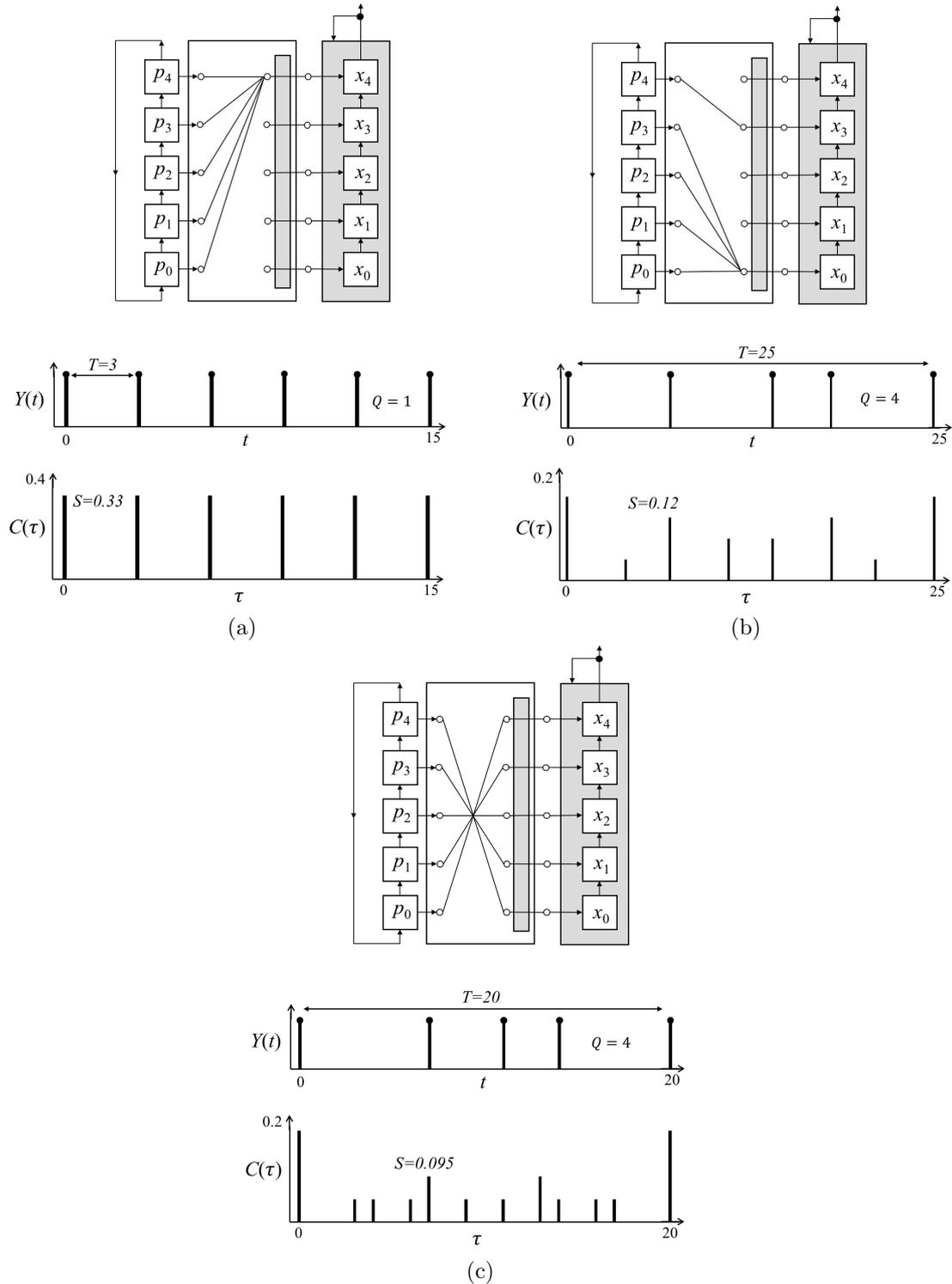


Figure 5.1.7: Generation of various spike-trains by the proposed generator. The numbers of the  $p$ -cells and the  $x$ -cells are  $M = N = 5$ . The transmission delay and the reset delay are  $\rho = \delta = 2$ . (a) The reconfigurable wires are characterized by  $a(0,4) = a(1,4) = a(2,4) = a(3,4) = a(4,4) = 1$ . The spike-train  $Y(t)$  is characterized by the period  $T = 3$ , the number  $Q = 1$  of spikes during the period, and the second peak  $S = 0.33$  of the auto-correlation function  $C(\tau)$ . (b)  $a(0,0) = a(1,0) = a(2,0) = a(3,0) = a(4,3) = 1$ .  $Y(t)$  is characterized by  $(T, Q, S) = (25, 4, 0.12)$ . (c)  $a(0,4) = a(1,3) = a(2,2) = a(3,1) = a(4,0) = 1$ .  $Y(t)$  is characterized by  $(T, Q, S) = (20, 4, 0.095)$ .

### 5.1.4 Generation of various spike-trains and parameter tuning for UWB application

Fig. 5.1.7 shows the proposed generator with different patterns of reconfigurable wires (i.e., different values of the parameter  $\mathbf{A}$ ) and corresponding spike-trains  $Y(t)$ . It can be seen that the proposed generator can generate spike-trains with various spike patterns by adjusting the pattern of the reconfigurable wires. In this section, a stochastic algorithm for parameter tuning for the proposed generator so that it can generate spike-trains  $Y(t)$  suitable for ultra wide band impulse radio (UWB-IR) communication, ranging, and positioning systems is proposed. In order to characterize the spike-train  $Y(t)$  of the proposed generator, the following quantities and function are introduced.

- The period  $T$  of the spike-train  $Y(t)$ , which can be obtained by using the spike phase map  $G$  as explained in the section 2.2.
- The number  $Q$  of spikes of the spike-train  $Y(t)$  during the period  $0 \leq t < T$ , which is identical with the cycle of the corresponding periodic orbit  $(\theta_1, \dots, \theta_Q)$  of the spike phase and thus can be obtained by using the spike phase map  $G$ .
- The autocorrelation function  $C(\tau)$  of the spike-train  $Y(t)$  and its second peak  $S$ , which are defined by

$$C(\tau) = \frac{1}{T} \sum_{t=0}^{T-1} Y(t)Y(t+\tau),$$

$$S = \begin{cases} \max_{0 < \tau < T} C(\tau) & \text{for } Q > 1, \\ C(0) & \text{otherwise.} \end{cases}$$

For example, the spike-trains  $Y(t)$  in Figs. 5.1.7(a), (b), and (c) are characterized by  $(T, Q, S) = (3, 1, 0.33)$ ,  $(25, 4, 0.12)$ , and  $(20, 4, 0.095)$ , respectively. Hence it can be confirmed that, by adjusting the pattern of the reconfigurable wires (i.e., the values of the parameter  $\mathbf{A}$ ), the proposed generator can generate spike-trains with various spike patterns in terms of the period  $T$ , the number  $Q$  of spikes, and the auto-correlation function  $C(\tau)$  and its second peak  $S$ .

Now, a parameter tuning algorithm for the proposed generator to generate spike-trains  $Y(t)$  suitable for the UWB-IR systems is proposed. In the UWB-IR systems, the following characteristics of the spike-train  $Y(t)$  are desired [29–31].

- (C1) The second peak  $S$  of the autocorrelation function  $C(\tau)$  of the spike-train  $Y(t)$  should be lower in order to realize higher resistivity against noise.
- (C2) The period  $T$  of the spike-train  $Y(t)$  should be longer in order to realize lower second peak  $S$  of the autocorrelation function  $C(\tau)$ .
- (C3) The number  $Q$  of spikes should be larger in order to realize higher resistivity against noise.

Then, the following objective function  $F$  of the spike-train  $Y(t)$  is introduced.

$$F = \alpha T + \beta Q - \gamma S, \quad (5.1.11)$$

where  $\alpha > 0$ ,  $\beta > 0$ , and  $\gamma > 0$  are parameters of the objective function  $F$ . Using the objective function  $F$ , the following parameter tuning algorithm is proposed.

**List 1. Stochastic Algorithm for Parameter Tuning.**

**Step 1: Initialization.** The numbers  $M$  and  $N$  of the cells and the delays  $\rho$  and  $\delta$  are given. Prepare  $L$  matrixes ( $\mathbf{A}_1, \dots, \mathbf{A}_L$ ) as follows:  $a_l(j, j) = 1$  for  $j = 0, \dots, M - 1$  and  $a_l(j, i) = 0$  for  $j \neq i$ , where  $a_l(j, i)$  is an element of the matrix  $\mathbf{A}_l$ . Also, initialize an iteration counter  $k$  to 0.

**Step 2: Evaluation.** A spike-train generated by the matrix  $\mathbf{A}_l$  is denoted by  $Y_l(t)$ . The values ( $F_1, \dots, F_L$ ) of the objective function  $F$  of the spike-trains ( $Y_1(t), \dots, Y_L(t)$ ) are calculated.

**Step 3: Selection and Mutation.** The matrix  $\mathbf{A}_{sel}$  corresponding to the maximum value  $F_{max} = \max_l\{F_l\}$  of the objective function  $F$  is selected. A column of the selected matrix  $\mathbf{A}_{sel}$  is randomly selected and the position of “1” in the selected column is randomly changed within the column.

**Step 4: Termination.** Let  $K$  be a given maximum iteration number. If  $k < K$ , then increment  $k$  by 1 and go to Step 2. If  $k = K$ , then terminate this algorithm.

Note that calculation speed of the objective functions ( $F_1, \dots, F_L$ ) can be accelerate by the spike phase map  $G$  (the calculation speed of  $T$  and  $Q$  becomes  $M$  times higher). Fig. 5.1.8 shows an example of parameter tuning result. Fig. 5.1.8(a) shows the proposed generator just after the initialization in Step 1. In this case, the period of the spike-train  $Y_l(t)$  is  $T_l = 5$ , the second peak of the auto-correlation function  $C_l(\tau)$  is  $S_l = 0.167$ , the number of spikes during the period is  $Q_l = 1$ , and the value of the objective function is  $F_l = 4.08$ , where  $\alpha = 0.8$ ,  $\beta = 0.1$  and  $\gamma = 0.1$ . Fig. 5.1.8(b) shows the proposed generator after the parameter tuning. In this case, the period of the spike-train  $Y_l(t)$  is  $T_l = 30$ , the second peak of the auto-correlation function  $C_l(\tau)$  is  $S_l = 0.066$ , the number of spikes during the period is  $Q_l = 5$ , and the value of the objective function is  $F_l = 24.49$ . It can be seen that, after the parameter tuning, the proposed generator generates the spike-train  $Y_l(t)$  with better characteristics, i.e., longer period  $T_l$ , larger number  $Q_l$  of spikes, and lower second peak  $S_l$  of the auto-correlation function. Fig. 5.1.9 shows the characteristics of the parameter tuning algorithm, where the dots indicate the characteristics of the proposed generator after the parameter tuning shown in Fig. 5.1.8(b). It can be seen that, as the iterations proceed, the parameter tuning algorithm can find patterns of the reconfigurable wires (i.e., values of the parameter  $\mathbf{A}_l$ ), which lead to better characteristics of the spike-trains  $Y_l(t)$ . It can be also seen that the average characteristics of the spike-trains almost converge after the parameter tuning. Fig. 5.1.10(a) shows a QCA layout of the proposed generator in Fig. 5.1.8(b) whose pattern of the reconfigurable wires is obtained by the parameter tuning algorithm. Also, Fig. 5.1.10(b) shows time waveforms of the QCA layout obtained by the QCA simulator QCADesigner. It can be confirmed that the QCA layout generates a spike-train  $Y(t)$ , which is equivalent to the spike-train  $Y_l(t)$  in Fig. 5.1.8(b).

**Remark 3 (Application):** The above result suggests that the QCA layout in Fig. 5.1.10(a) can generate a spike-train suitable for the UWB-IR systems in terms of the period  $T$ , the number  $Q$  of spikes, and the second peak  $S$  of the auto-correlation function. Potential

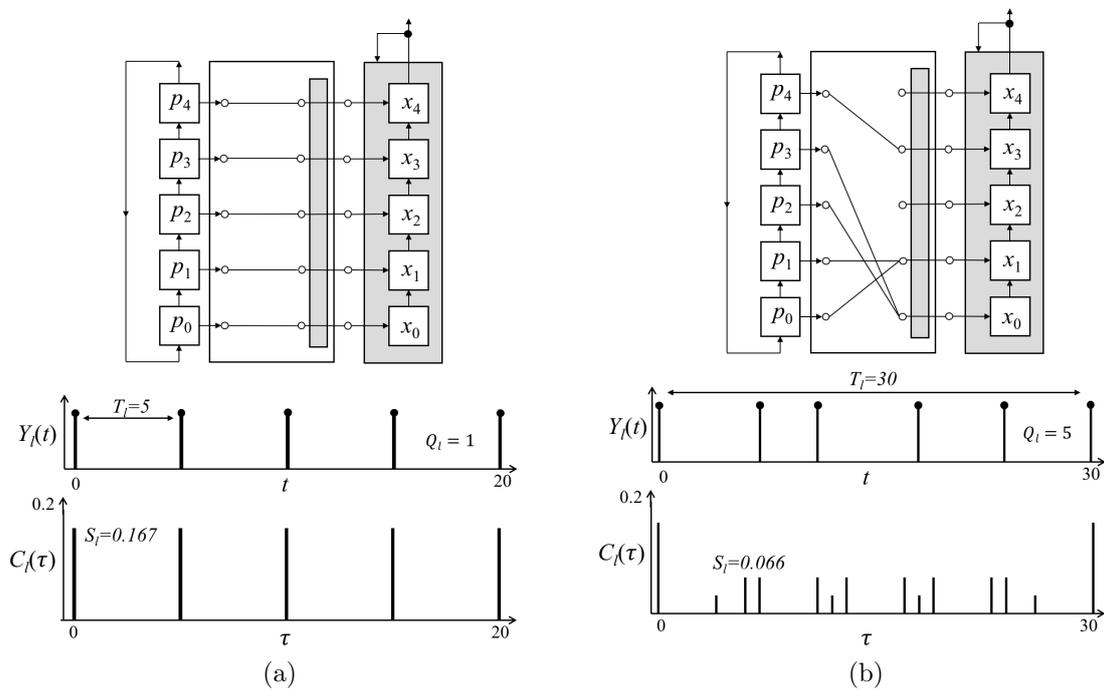


Figure 5.1.8: An example of parameter tuning result. The numbers of the cells are  $M = N = 5$  and the delays are  $\rho = \delta = 2$ . The number of the prepared matrixes is  $L = 5$  and the maximum iteration number is  $K = 5000$ . The parameters of the objective function  $F$  are  $\alpha = 0.8$ ,  $\beta = 0.1$  and  $\gamma = 0.1$ . (a) After initialization in Step 1. The spike-train  $Y_i(t)$  is characterized by  $(T_i, Q_i, S_i, F_i) = (5, 1, 0.167, 4.08)$ . (b) After tuning. The spike-train  $Y_i(t)$  is characterized by  $(T_i, Q_i, S_i, F_i) = (30, 5, 0.066, 24.49)$ .

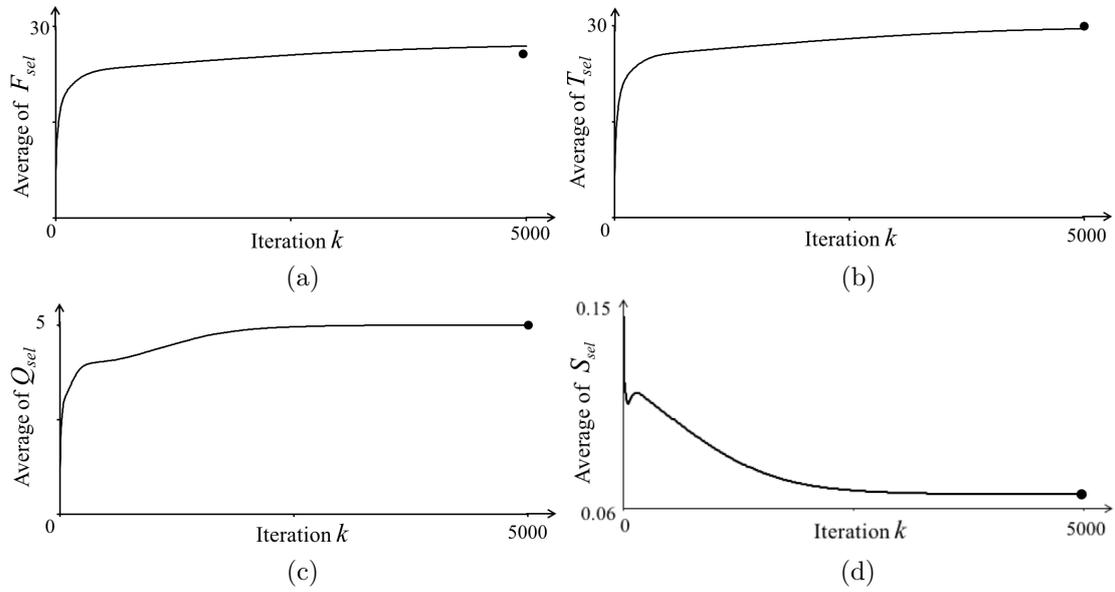


Figure 5.1.9: Parameter tuning characteristics. The numbers of the cells are  $M = N = 5$  and the delays are  $\rho = \delta = 2$ . The number of the prepared matrixes is  $L = 5$  and the maximum iteration number is  $K = 5000$ . The parameters of the objective function  $F$  are  $\alpha = 0.8$ ,  $\beta = 0.1$  and  $\gamma = 0.1$ . An execution of the parameter tuning for  $K$  iterations is called a trial. The graphs in (a)–(d) are averages for 1000 trials, where the dots indicate the characteristics of the proposed generator after the parameter tuning shown in Fig. 5.1.8(b). (a) Average of the maximum value  $F_{sel}$  of the objective function  $F$ . (b) Average of the period  $T_{sel}$  leading to the maximum value  $F_{sel}$  of the objective function  $F$ . (c) Average of the number  $Q_{sel}$  of spikes leading to the maximum value  $F_{sel}$  of the objective function  $F$ . (d) Average of the second peak  $S_{sel}$  of the auto-correlation function leading to the maximum value  $F_{sel}$  of the objective function  $F$ . The influence of the parameters  $\alpha$ ,  $\beta$  and  $\gamma$  is explained in Appendix.

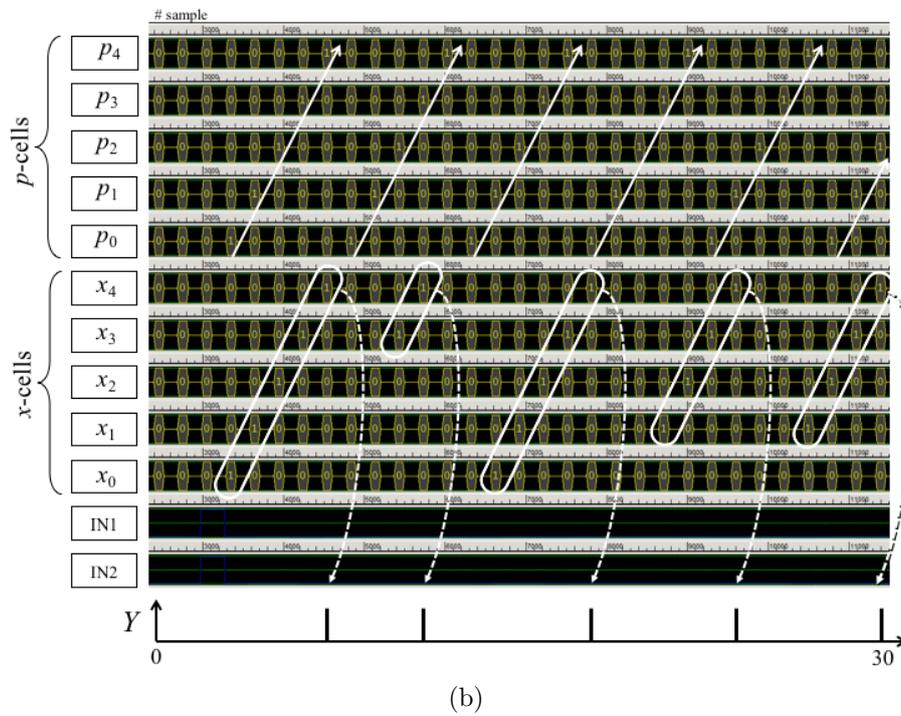
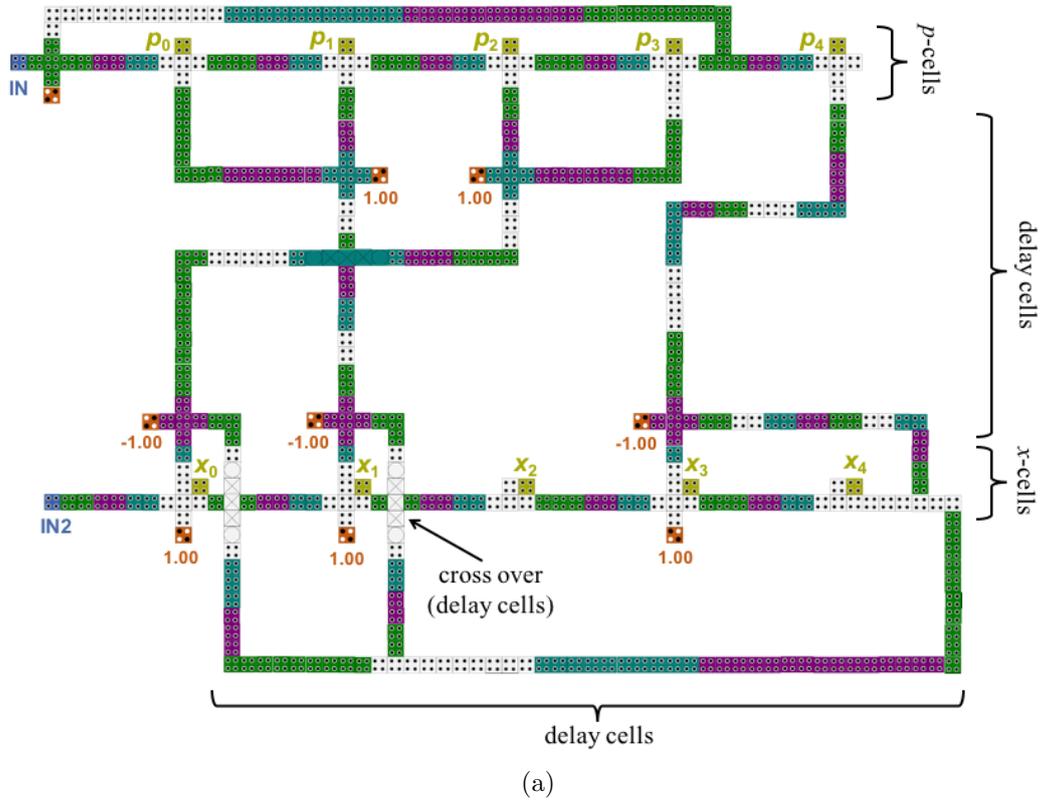


Figure 5.1.10: (a) QCA layout of the proposed generator in Fig. 5.1.8(b), where the pattern of the reconfigurable wires is obtained by the parameter tuning algorithm. (b) Time waveforms of the QCA layout obtained by the QCADesigner.

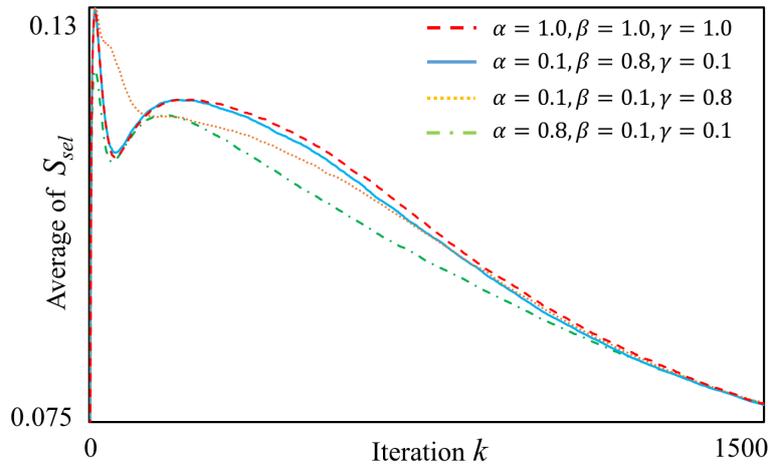


Figure 5.1.11: Comparison of parameter tuning characteristics for several parameters values.

applications of such QCA based UWB-IR systems include: intra and inter QCA chip wired communications, ultra low power wireless QCA based communications, and ultra low power QCA based ranging and positioning.

### 5.1.5 Conclusions

Recall that the purposes of this paper were to propose a QCA based spike-train generator that can generate spike-trains with various spike patterns, to propose an analysis tool of the generator, and to propose a parameter tuning algorithm for the generator so that it can generate spike-trains suitable for the UWB-IR systems. These purposes were achieved as the followings. The novel spike-train generator the dynamics of which is described by the cellular automaton with the delays  $\rho$  and  $\delta$  was proposed. Also, as its analysis tool, the spike phase map  $G$  was derived. Using the spike phase map  $G$ , it was shown that the proposed generator can generate various spike-trains in terms of the period, the number of spikes, and the auto-correlation function. Also, the parameter tuning algorithm for the proposed generator was proposed, where the spike phase map  $G$  is used as its subroutine, where the spike phase map  $G$  accelerates execution speed of the parameter tuning algorithm. It was shown that the parameter tuning algorithm enables the proposed generator to generate spike-trains suitable for the UWB-IR systems, i.e., spike-trains with longer period  $T$ , larger number  $Q$  of spikes, and lower second peak  $S$  of the auto-correlation function. Furthermore, the proposed generator after the parameter tuning was implemented as the QCA layout and its operation was validated by the QCA simulator. Future problems are including (a) calculation of bit error rates under specific pulse modulations, (b) more detailed analysis of the spike-train of the proposed generator, (c) development of more efficient parameter tuning algorithm, and (d) design of QCA based synchronizer and inverse spreader for UWB-IR communication applications.

## Appendix: Influence of the parameters in the tuning algorithm

Let us clarify influence of the parameters  $(\alpha, \beta, \gamma)$  of the objective function  $F$  in Eq. (11) used in the parameter tuning algorithm in List 1. It has been shown that lower auto-correlation (lower second peak of the auto-correlation function) of a spike-train leads to lower bit error probability in a UWB-IR communication system [32]. Therefore, the proposed parameter tuning algorithm searches values of the parameters  $\mathbf{A}$ , which lead to a lower second peak  $S$  of the auto-correlation function of the spike-train  $Y(t)$ . Fig. 5.1.11 shows the influence of the parameters  $(\alpha, \beta, \gamma)$  in the characteristics of the second peak  $S$ . It can be seen in the figure that the parameter values  $(\alpha, \beta, \gamma) = (0.8, 0.1, 0.1)$  lead to a better characteristics of the second peak  $S$  (see the green dashed curve). So, the parameter values  $(\alpha, \beta, \gamma) = (0.8, 0.1, 0.1)$  are used in the paper, e.g., Figs. 5.1.9–5.1.11 are obtained by the parameter tuning with these parameter values.

## Bibliography

- [1] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernstein, “Quantum cellular automata,” *Nanotechnology*, vol. 4, no. 1, pp. 49–57, 1993.
- [2] P. D. Tougaw and C. S. Lent, “Logical devices implemented using quantum cellular automata,” *Journal of Applied Physics*, vol. 75, no. 3, pp. 1818–1825, 1994.
- [3] C. S. Lent and P. D. Tougaw, “A device architecture for computing with quantum dots,” *Proceedings of the IEEE*, vol. 85, no. 4, pp. 541–557, 1997.
- [4] K. Walus and G. A. Jullien, “Design Tools for an Emerging SoC Technology: Quantum-Dot Cellular Automata,” *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1225–1244, 2006.
- [5] V. Pudi and K. Sridharan, “Low Complexity Design of Ripple Carry and Brent–Kung Adders in QCA,” *IEEE Transactions on Nanotechnology*, vol. 11, no. 1, pp. 105–119, 2012.
- [6] S. Perri and P. Corsonello, “New Methodology for the Design of Efficient Binary Addition Circuits in QCA,” *IEEE Transactions on Nanotechnology*, vol. 11, no. 6, pp. 1192–1200, 2012.
- [7] S. Perri, P. Corsonello, and G. Cocorullo, “Design of Efficient Binary Comparators in Quantum-Dot Cellular Automata,” *IEEE Transactions on Nanotechnology*, vol. 13, no. 2, pp. 192–202, 2014.
- [8] D. Abedi, G. Jaberipur, and M. Sangsefidi, “Coplanar Full Adder in Quantum-Dot Cellular Automata via Clock-Zone-Based Crossover,” *IEEE Transactions on Nanotechnology*, vol. 14, no. 3, pp. 497–504, 2015.
- [9] M. Mustafa and M. R. Beigh, “Novel linear feedback shift register design in quantum-dot cellular automata,” *Indian Journal of Pure & Applied Physics*, vol. 52, pp. 203–209, 2014.

- [10] S. Angizi, M. H. Moaiyeri, S. Farrokhi, K. Navi, and N. Bagherzadeh, "Designing quantum-dot cellular automata counters with energy consumption analysis," *Microprocessors and Microsystems*, vol. 39, no. 7, pp. 512–520, 2015.
- [11] V. Vankamamidi, M. Ottavi, and F. Lombardi, "A Line-Based Parallel Memory for QCA Implementation," *IEEE Transactions On Nanotechnology*, vol. 4, no. 6, pp. 690–698, 2005.
- [12] K. Kim, K. Wu, and R. Karri, "Quantum-Dot Cellular Automata Design Guideline," *IEICE Transactions on Fundamentals of Electronics*, vol. E89-A, no. 6, pp. 1607–1614, 2006.
- [13] W. Liu, L. Lu, M. O'Neill, and E. E. Swartzlander Jr, "Design Rules for Quantum-Dot Cellular Automata," in *Proc. of the 2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, Rio de Janeiro, Brazil, 2011, pp. 2361–2364.
- [14] V. Vankamamidi, M. Ottavi, and F. Lombardi, "Two-Dimensional Schemes for Clocking/Timing of QCA Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 34–44, 2008.
- [15] A. Roohi, R. Zand, S. Angizi, and R. F. DeMara, "A Parity-Preserving Reversible QCA Gate with Self-Checking Cascadable Resiliency," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 4, pp. 450–459, 2018.
- [16] L. O. Chua, *A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science Volume I*, ser. World Scientific Series on Nonlinear Science Series A, 2006, vol. 57.
- [17] —, *A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science Volume II*, ser. World Scientific Series on Nonlinear Science Series A, 2006, vol. 57.
- [18] —, *A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science Volume III*, ser. World Scientific Series on Nonlinear Science Series A, 2009, vol. 68.
- [19] H. T. Zhao, S. Yang, and X. X. Chen, "Cellular automata model for urban road traffic flow considering pedestrian crossing street," *Physica A: Statistical Mechanics and its Applications*, vol. 462, pp. 1301–1313, 2016.
- [20] N. R. da Silva, P. Van der Weeën, B. De Baets, and O. M. Bruno, "Improved texture image classification through the use of a corrosion-inspired cellular automaton," *Neurocomputing*, vol. 149, pp. 1560–1572, 2015.
- [21] H. Dina Bezerra, N. Nedjah, and L. de Macedo Mourelle, "Reconfigurable hardware architecture for music generation using cellular automata," in *Proc. of the 2014 IEEE 5th Latin American Symposium on Circuits and Systems (LASCAS)*, Santiago, Chile, 2014, pp. 1–4.
- [22] I. Gresham, A. Jenkins, R. Egri, C. Eswarappa, N. Kinayman, N. Jain, R. Anderson, F. Kolak, R. Wohlert, S.P. Bawell, J. Bennett, and J. P. Lanteri, "Ultra-Wideband Radar Sensors for Short-Range Vehicular Applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 9, pp. 2105–2122, 2004.

- [23] J. Barral, P. V. Orlik, Z. Sahinoglu, A. F. Molisch, and P. Kinney, "UWB Systems for Wireless Sensor Networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 313–331, 2009.
- [24] S. Gezici and H. V. Poor, "Position Estimation via Ultra-Wide-Band Signals," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 386–403, 2009.
- [25] H. Torikai, H. Hamanaka, and T. Saito, "Reconfigurable Digital Spiking Neuron and Its Pulse-Coupled Network: Basic Characteristics and Potential Applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 8, pp. 734–738, 2006.
- [26] H. Torikai, "Basic Characteristics and Learning Potential of a Digital Spiking Neuron," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 10, pp. 2093–2100, 2007.
- [27] H. Torikai, A. Funew, and T. Saito, "Digital spiking neuron and its learning for approximation of various spike-trains," *Neural Networks*, vol. 21, no. 2-3, pp. 140–149, 2008.
- [28] K. Walus, T. J. Dysart, G. A. Jullien, and R. A. Budiman, "QCADesigner: A Rapid Design and Simulation Tool for Quantum-Dot Cellular Automata," *IEEE Transactions on Nanotechnology*, vol. 3, no. 1, pp. 26–31, 2004.
- [29] T. Iguchi, A. Hirata, and H. Torikai, "Theoretical and Heuristic Synthesis of Digital Spiking Neurons for Spike-Pattern-Division Multiplexing," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E93-A, no. 8, pp. 1486–1496, 2010.
- [30] M. G. Benedetto and G. Giancola, *Understanding Ultra Wide Band Radio Fundamentals*. Prentice Hall, 2004.
- [31] D. Porcino and W. Hirt, "Ultra-wideband radio technology: Potential and challenges ahead," *IEEE Communications Magazine*, vol. 41, no. 7, pp. 66–74, 2003.
- [32] X. Huang and Y. Li, "Performances of impulse train modulated ultra-wideband systems," in *Proc. of the 2002 IEEE International Conference on Communications*, vol. 2, New York, NY, USA, 2002, pp. 758–762.

# Chapter 6

## Overall Conclusions and Future Work

### 6.1 Overall conclusions

This thesis has used asynchronous cellular automata (ACA) to design hardware efficient CTDS-class biological system models which produce smoother nonlinear vector fields than those of DTDS-class models.

Chapter 2 presented hardware-efficient ACA-based cochlea models. It was shown that the proposed cochlea models can reproduce the characteristics of the nonlinear bandpass filters and the two-tone distortion products of the mammalian cochlea, and that these models can be implemented using fewer hardware resources than required for implementing the numerical integration formula of the conventional Hopf oscillator cochlea model.

Chapter 3 presented hardware-efficient ACA-based CPG models and showed that these CPG models can reproduce serpentine, tripod and wave gaits for snake-like and hexapod robots. It was also shown that the proposed CPG models can be implemented using fewer hardware resources than needed to implement the numerical integration formula of the conventional CPG model.

Chapter 4 presented a hardware-efficient ACA-based neuron model and demonstrated that this model is able to reproduce the typical nonlinear response curves of a biological neuron. Additionally, the proposed neuron model was successfully implemented using fewer hardware resources than required for conventional neuron models (i.e. the leaky integrate-and-fire model, quadratic integrate-and-fire model, and Morris-Lecar model). It was shown that the proposed coupled ACA-based NN model can reproduce neural integration as observed in biological neural networks. Furthermore, the proposed NN model was demonstrated to require fewer hardware resources in implementation than required for the conventional NN models based on coupled quadratic integrate-and-fire neurons and Hodgkin-Huxley neurons.

To further explore hardware-efficient biological system models, this thesis also investigated a biological system modeling approach based on the quantum-dot cellular automata (QCA). Chapter 5 demonstrated that a neural spike-train generator based on QCA is able to generate spike trains able to produce neural spike patterns varying in period, density, and correlation. Additionally, it was shown that the parameter-tuning algorithm can enable the proposed generator to generate spike trains suitable for UWB-IR communication, ranging, and positioning systems. After implementation of the parameter-tuning algorithm, a QCA layout for the proposed generator was designed, and its operation verified using a QCA simulator.

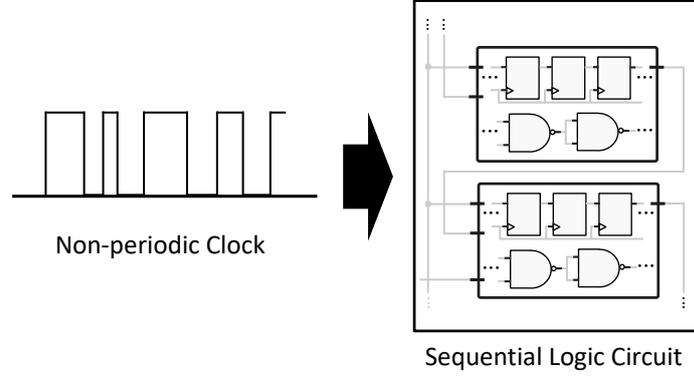


Figure 6.1.1: Non-periodic clock-driven CA.

It can be concluded from the obtained results that CTDS-class biological system models are more suitable for integrated circuit implementation than DTDS-class biological system models. Furthermore, this thesis has analyzed theoretically the following nonlinear phenomena in CTDS-class biological system models: the supercritical Andronov-Hopf bifurcation (Chapter 2) and the saddle-node on invariant circle (SNIC) bifurcation (Chapter 4). Using a dynamical system theory lens indicates that these bifurcation phenomena are important for biological systems, and those observed in CTDS-class models have not yet received sufficient analysis. The presented biological oscillators can be conceptualized as Hopf and SNIC oscillators, both typical types of nonlinear biological oscillators, while nonlinear phenomena were analyzed for coupled Hopf oscillators (Chapter 3), coupled SNIC oscillators (Chapter 4), and coupled phase oscillators (Chapter 3). Chapter 3 investigated coupled CTDS-class phase oscillators in addition to the Hopf oscillators and the SNIC oscillators described above. It is hoped that these results will contribute to the development of a theoretical framework for CTDS-class biological system models and prove useful in future neuromorphic hardware designs.

## 6.2 Future work

As introduced in Section 1.1.2, the biological system model belonging to class CTDS treated in this thesis can be described as follows.

$$\begin{cases} X_1(t + h_1) = X_1(t) + F_1(X_1(t), X_2(t + h_1)), & (6.2.1a) \\ X_2(t + h_1) = X_2(t) + F_2(X_1(t), X_2(t))s(\phi_2(t)), & (6.2.1b) \\ \phi_2(t + h_1) = \phi_2(t) + h_1 \pmod{h_2}. & (6.2.1c) \end{cases}$$

Eq. (6.2.1c) is regarded as an irrational rotation, which generates a quasiperiodic sequence if  $h_1/h_2 \in \mathbb{Q}$ . Roughly speaking, the map in (6.2.1c) works as a weak noise for the transitions of the discrete states  $X_1$  and  $X_2$ . The system in Eq. (6.2.1) can be realized by a sequential logic circuit to which uncoupled periodic clocks are supplied. In the future work, the generator of the quasiperiodic sequence  $\phi_2$  in Eq. (6.2.1c) will be replaced by other types of non-periodic clock generators. Such a system is considered to be realized by a sequential

logic circuit to which the following generators are supplied (see Fig. 6.1.1):

- Chaotic signal generator,
- Stochastic random number generator,
- Pseudorandom number generator,

Future work includes the generalization of the ACA treated in this thesis to the non-periodic clock-driven cellular automaton (NCA). The author intends to investigate whether the NCA can be applied to biological system models such as a cochlea model, a CPG model, and a SNN model. One of my goals in the future is to analyze the NCA-based biological system models and find their advantages. Furthermore, it would be desirable if the findings, including those made in this thesis, could be applied to a cryptographic generator and QCA.

## Publication List

### Journal Paper

1. Sho Komaki, Kentaro Takeda, and Hiroyuki Torikai, “A Novel Ergodic Discrete Difference Equation Model of Central Pattern Generator: Theoretical Analysis and Efficient Implementation,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2021. (early access)
2. Kentaro Takeda and Hiroyuki Torikai, “Two-Tone Distortion Products in Hardware-Efficient Cochlea Model based on Asynchronous Cellular Automaton Oscillator,” *IEICE Electronics Express*, vol. 18, no. 18, 20210310, 2021.
3. Kentaro Takeda and Hiroyuki Torikai, “A novel hardware-oriented recurrent network of asynchronous CA neurons for a neural integrator,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 8, pp. 2972–2976, 2021.
4. Kentaro Takeda and Hiroyuki Torikai, “Smooth Gait Transition in Hardware-Efficient CPG Model based on Asynchronous Coupling of Cellular Automaton Phase Oscillators,” *Nonlinear Theory and Its Applications, IEICE*, vol. 12, no. 3, pp. 336–356, 2021.
5. Kentaro Takeda and Hiroyuki Torikai, “A Novel Hardware-Efficient Central Pattern Generator Model based on Asynchronous Cellular Automaton Dynamics for Controlling Hexapod Robot,” *IEEE Access*, vol. 8, pp. 139609–139624, 2020.
6. Kentaro Takeda and Hiroyuki Torikai, “A Novel Asynchronous CA Neuron Model: Design of Neuron-like Nonlinear Responses based on Novel Bifurcation Theory of Asynchronous Sequential Logic Circuit,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 6, pp. 1989–2001, 2020.
7. Kentaro Takeda and Hiroyuki Torikai, “A novel spike-train generator suitable for QCA implementation towards UWB-IR applications,” *Nonlinear Theory and Its Applications, IEICE*, vol. 9, no. 4, pp.436–452, 2018.
8. Kentaro Takeda and Hiroyuki Torikai, “A Novel Hardware-Efficient CPG Model based on Asynchronous Cellular Automaton,” *IEICE Electronics Express*, vol. 15, no. 11, 20180387, 2018.
9. Kentaro Takeda and Hiroyuki Torikai, “A Novel Hardware-Efficient Cochlea Model based on Asynchronous Cellular Automaton Dynamics: Theoretical Analysis and FPGA Implementation,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 9, pp. 1107–1111, 2017.

## International Conference

1. Kentaro Takeda and Hiroyuki Torikai, “Synchronization phenomena of asynchronously coupled cellular automaton phase oscillators and its theoretical analysis,” in *Proc. of 2021 Nonlinear Science Workshop (NLSW)*, p. 42, Dec. 2021.
2. Sho Komaki, Kentaro Takeda, and Hiroyuki Torikai, “A novel asynchronous sequential logic model of central pattern generator for quadruped robot: systematic design and efficient implementation,” in *Proc. of 2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Jul. 2021.
3. Kentaro Takeda and Hiroyuki Torikai, “Asynchronous CA Model of Central Pattern Generator,” in *Proc. of 2020 International Symposium on Nonlinear Theory and its Applications (NOLTA)*, pp. 303–306, Nov. 2020.
4. Kentaro Takeda and Hiroyuki Torikai, “A novel hardware-efficient CPG model based on asynchronous coupling of cellular automaton phase oscillators for a hexapod robot,” in *Proc. of 2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Jul. 2020.
5. Kentaro Takeda and Hiroyuki Torikai, “Asynchronous cellular automaton neuromorphic circuit: design of nonlinear responses based on design of bifurcation phenomena,” in *Proc. of 2019 International Symposium on Nonlinear Theory and its Applications (NOLTA)*, pp. 324–325, Dec. 2019.
6. Kentaro Takeda and Hiroyuki Torikai, “A novel hardware-efficient CPG model for a hexapod robot based on nonlinear dynamics of coupled asynchronous cellular automaton oscillators,” in *Proc. of 2019 International Joint Conference on Neural Networks (IJCNN)*, pp.1–8, Jul. 2019.
7. Kentaro Takeda and Hiroyuki Torikai, “Asynchronous cellular automaton models of biological system,” in *Proc. of 2018 International Symposium on Nonlinear Theory and its Applications (NOLTA)*, p. 427, Sept. 2018.
8. Kentaro Takeda and Hiroyuki Torikai, “A novel hardware-efficient spiking neuron model based on asynchronous cellular automaton dynamics exhibiting various nonlinear response curves, ” in *Proc. of 2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 5036–5043, Jul. 2018.
9. Kentaro Takeda and Hiroyuki Torikai, “A Neuromorphic Quantum-dot Cellular Automaton,” in *Proc. of 6th Korea-Japan Joint Workshop on Complex Communication Sciences (KJCCS)*, Paper ID P29, Jan. 2018.
10. Kentaro Takeda and Hiroyuki Torikai, “Basic Analysis of a Spike-train Generator based on Quantum-dot Cellular Automaton,” in *Proc. of 2017 IEEE Workshop on Nonlinear Circuit Networks (NCN)*, pp. 104–105, Dec. 2017.
11. Kentaro Takeda, Chiaki Matsuda and Hiroyuki Torikai, “Design of Electronic Circuit Model of Neural System based on Hybrid Dynamical System,” in *Proc. of 2017*

- International Symposium on Nonlinear Theory and its Applications (NOLTA)*, p. 628, Dec. 2017.
12. Kentaro Takeda and Hiroyuki Torikai, “A Novel Hardware-Efficient CPG Model based on Nonlinear Dynamics of Asynchronous Cellular Automaton,” in *Proc. of 24th International Conference on Neural Information Processing (ICONIP)*, Part III, Springer LNCS 10639, pp. 812–820, Nov. 2017.
  13. Kentaro Takeda and Hiroyuki Torikai, “Experiments on a Hardware-Efficient Bio-Inspired Snake-like Robot,” in *Proc. of 2017 Taiwan and Japan Conference on Circuits and Systems (TJCAS)*, p. 53, Aug. 2017.
  14. Kentaro Takeda and Hiroyuki Torikai, “Reproduction of Nonlinear Cochlea Response by Asynchronous Bifurcation Processor,” in *Proc. of 2016 International Symposium on Nonlinear Theory and its Applications (NOLTA)*, pp. 279–282, Nov. 2016.
  15. Kentaro Takeda and Hiroyuki Torikai, “Design of Biomimetic Digital Hardware based on Asynchronous Bifurcation Processor,” in *Proc. of 5th Japan-Korea Joint Workshop on Complex Communication Sciences (JKCCS)*, p. 83, Oct. 2016.
  16. Hiroyuki Torikai, Kentaro Takeda, and Taiki Naka, “Asynchronous Bifurcation Processor: Fundamental Concepts and Application Examples,” in *Proc. of 4th International Conference on Applications in Nonlinear Dynamics (ICAND)*, pp. 217–229, Aug. 2016.

## Domestic Conference

1. 武田健太郎, 鳥飼弘幸, 非同期順序回路の非線形現象とその応用, CCS/IN 合同ワークショップ, 2019年8月.
2. 武田健太郎, 鳥飼弘幸, 非同期セルオートマトンに基づいたニューロンモデルの安定性解析について, 2019年電子情報通信学会 NOLTA ソサイエティ大会, B-13, 長岡, 2019年6月.
3. 武田健太郎, 鳥飼弘幸, 非同期セルオートマトン発振器の結合系に基づいた CPG モデルについて, 電子情報通信学会総合大会講演論文集, N-2-4, 2019年3月.
4. 武田健太郎, 鳥飼弘幸, 非同期セルオートマトンに基づいたニューロンモデルの分岐現象について, 電子情報通信学会技術研究報告 複雑コミュニケーションサイエンス研究会, Vol. 118, No. 316, CCS-2018-33, pp. 1–5, 2018年11月.
5. 武田健太郎, 鳥飼弘幸, 遅延を有する QCA スパイク列発生器のリターンマップについて, 2018年電子情報通信学会 NOLTA ソサイエティ大会, B-1, 2018年6月.
6. 武田健太郎, 鳥飼弘幸, 非同期セルオートマトン発振器の結合系の解析と応用について, 電子情報通信学会技術研究報告 複雑コミュニケーションサイエンス研究会, Vol. 117, No. 288, CCS-2017-31, pp. 55–59, 2017年11月.
7. 武田健太郎, 鳥飼弘幸, 非同期セルオートマトンに基づく CPG モデルを用いたヘビ型ロボット, 2017年電子情報通信学会 NOLTA ソサイエティ大会, B-7, 2017年6月.

8. 武田健太郎, 鳥飼弘幸, 非同期分岐プロセッサを用いた Hopf 蝸牛モデル, 電子情報通信学会技術研究報告 複雑コミュニケーションサイエンス研究会, Vol. 116, No. 285, CCS-2016-33, pp. 17–21, 2016 年 11 月.