PHYSICAL COMPUTING SYSTEMS: THEORY, IMPLEMENTATION AND FUNCTIONALITY

Shaohua Kan

PHYSICAL COMPUTING SYSTEMS: THEORY, IMPLEMENTATION AND FUNCTIONALITY

DISSERTATION

to obtain
the degree of doctor at Graduate School of Information Science and Technology, Hokkaido University,
on the authority of the Dean,
prof. dr. ir. M. Haseyama,
on account of the decision of the Doctorate Board,
to be publicly defended
on Tuesday 22 of August 2021 at 13.05 hours

by

Shaohua Kan

This dissertation has been approved by:

Supervisor	:	Prof. Dr. Megumi Akai-Kasaya	Osaka University
Co-supervisors	:	Prof. Dr. Tetsuya Asai	Hokkaido University
		Prof. Dr. Kohei Nakajima	The University of Tokyo

Graduation Committee

Chairman	Prof. Dr. Tetsuya Asai
Associate chair	Prof. Dr. Seiya Kasai
	Prof. Dr. Masayuki Ikebe
	Prof. Dr. Hiromichi Ohta
Additional committee members	Prof. Dr. Keiji Sasaki
	Prof. Dr. Kazuhisa Sueoka
	Prof. Dr. Akihisa Tomita
	Prof. Dr. Akihiro Murayama
	Prof. Dr. Fumitaro Ishikawa
	Prof. Dr. Junichi Motohisa
	Prof. Dr. Hiroki Arimura

People with great and firm faith although in adversity,
Will eventually usher in their own moment and seize it.
他日卧龙终得雨，今朝放鹤且冲天。

Yuxi Liu, 772 AD

CONTENTS

PREFACE

Moving to an unfamiliar city during the covid-19 pandemic and staring a doctoral program with completely new and uncertain existence was not easy. However, as I near the completion of my PhD, I have come to realize that this experience has instilled in me the courage and confidence to confront future setbacks and adversities. I would like to take this opportunity to express my heartfelt gratitude to all those who have been instrumental to me throughout this challenging yet rewarding journey.

First and foremost, I could not have undertaken this journey without my supervisor Prof. Dr. Megumi Akai-Kasaya. Words cannot express my gratitude for your mentorship, support, care and necessary resources you have provided me over the past five years. Your patience, respect, and empowerment have inspired me to strive for excellence and persevere through every obstacle.

I would like to express my deepest appreciation to my official supervisor and chair of my committee Prof. Dr. Tetsuya Asai, whose sophistication and empirical guidance has been pivotal in shaping the direction of this journey. Special thanks are due to LALSIE laboratory for providing a conducive environment for conducting this research. The administrative personnel and lab mates have been unfailingly helpful in facilitating various aspects of my academic journey.

My deepest gratitude also extends to Prof. Dr. Kohei Nakajima, whose expertise, academic supports and constructive suggestions have enriched my understanding of the subject matter and improved the quality of my papers.

In addition, I am extremely grateful to my parents for their financial and mental support, and to my friends for their constant companionship. Their unwavering love and continuous encouragement have sustained me during the highs and lows of this academic pursuit.

My heartfelt gratitude goes to Otsuka Toshimi Scholarship Foundation and Japan Society for the Promotion of Science that financed my research, and to Core-to-Core program and the project director Prof. Dr. Takuya Matsumoto for the opportunity and financial assistance for my academic visit to the Netherlands in the last year of my PhD.

Additionally, I am greatly indebted to Prof. Dr. Wilfred van der Wiel and his team, Nano-Electronics group, for their camaraderie, enthusiasm and intellectual discussions that have made my moments in the Netherlands incredibly warm and pleasant and inspiring.

Last but not least, I wish to acknowledge the anonymous reviewers whose constructive suggestions have laid the foundation for this thesis. Their rigorous attitude and dedication

to advancing knowledge in the field has inspired me to make my modest contributions to the academic community.

In conclusion, this thesis would not have been possible without the collective efforts and support of these remarkable individuals and institutions. Thank you all for being an integral part of this transformative journey.

Shaohua Kan
Sapporo, July, 2023

SUMMARY

Unconventional computing (UC) is a novel formalism as a post-Moore solution that stems from a perspective of smaller, faster, and more energy-efficient methodology. Exploring unbounded physical phenomena serving as computing resources is the main goal of UC research, but it is difficult to take these findings as guidelines due to the diversity of physical phenomena and the lack of a unified paradigm. Before UC formalism can grow and refine into a unified formal theory, it is necessary to explore the computational paradigms available for existing physical devices, but the most important aim should be providing guidance for future system design. One UC paradigm that utilizes dynamic systems, reservoir computing, is widely discussed due to its arbitrary and diverse physical implementations. An important objective of this paper is to comprehensively analyze different schemes of RC structure design and nonlinear function selection, and to observe their respective effects on information processing. The ultimate purpose is to validate the usability of proposing schemes and design rule, then develop a small, simple, fast and low power-consumption physical device based on them. Additionally, attempts to find a suitable UC paradigm for existing devices according to their own characteristics is also necessary for future system design.

In Chapter 2, we defined a simple structure of reservoir from its mathematical definition matrix and give it the simplest form of realization, that is, one pair of diodes. Instead of treating RC as a "black box" like most schemes of physical RC design, we realized a simple controllable physical system by external parameters so as to provide $I-V$ curve of diodes with dynamics and to observe their effects. This scheme originates from the design idea of independent processing nodes, and on this basis, the influence of nonlinear function of processing node and node arrangement structure are explored respectively. Nodes structure has greater effects on NARMA2 task, short-term memory capacity task and classification task than its nonlinear function. A large number of random and strong connections between nodes ensure the echo state duration of original information in the network, which is conducive to short-term memory capacity and tasks requiring high memory capacity but interferes with the classification of processed information. Reservoir with sparser and weaker connections on the other hand solves both NARMA2 task and classification task well. Results indicate that a sparse interconnectivity of all nodes that are commonly accepted in reservoir can be replaced by regular connections of segmental nodes. Besides, even parallel-node structure showed acceptable accuracy on classification task, although it

did not solve the NARMA task well. Nevertheless, this opens up a new path for physical RC design in the future because the parameters of the parallel structure are directly reflected in the definition matrix of reservoir and thus can be better correlated with inherent parameters of the physical system, facilitating the adjustment and control of the system.

In Chapter 3, a new physical device, sets of planar parallel electrodes, is designed based on the parallel structure to further verify its practicability. The size and distribution of the electrodes cannot be changed, which means that the structure of the reservoir is fixed. Solution that dripped to the surface of the electrodes provides different nonlinear functions owing to their specific $I-V$ characteristic, and the slightly different electrochemical reactions occurred on different electrodes act as the feedback gain of processing nodes. It is shown in short-term memory capacity results that no matter what solution is added on the surface, there is a certain degree of interaction between the adjacent electrodes, which implies the feedforward gain in the parallel-group structure described in Chapter 2. Therefore this device exhibited a prediction error matching that of parallel-group structure in the same NARMA2 task. Compared to distilled water, solution with complex REDOX reactions display more dynamical $I-V$ characteristics, which aid in reproduction of periodic signals but pose a challenge in solving higher-order nonlinear problems. In the approximation of those higher-order nonlinear problems, protons contribute greatly to the computational power of this system. This is concluded because system lost computing power when the solvent was replaced by nonprotonic solvent.

Next we demonstrated how to apply appropriate UC paradigm to existing physical systems based on their own characteristics. Chapter 4 explored the feasibility of performing stochastic computing paradigm on a SWNT/POM network with stochastic transfer process of charges inside. At the level of real device, the magnitude of output current was controlled by the voltage on the grid under the condition of a stable source-drain voltage. The output current showed sinusoidal under the control voltage of sinusoidal excitation. In addition, the spike density of the output current increased with the number of times applying the gate voltage of the same magnitude. Above observations are realized in experiment and the results are repeatable. On the simulation level, we achieved the gradual growth or decay of the current spike density along epoch with big V_G or small V_G. Each epoch includes 3 clock cycles for input stimulus and 256 clock cycles for weights updating. Such results show a great potential of SWNT/POM network device to replace encoders, decoders and memory in fully-implemented hardware circuits for stochastic computing.

Overall, the outcomes of this research can serve as a basis for enhancing the design and evaluation of various physical UC schemes. Starting from simple design principle and small device, the theory of UC is explored and supplemented, and a different way is pointed out for the future system design and development. The findings, insights, and methodologies obtained from this study can be applied to refine and optimize future physical UC designs.

1

Introduction

1

THE advancement of artificial intelligence (AI) heavily relies on the capabilities of computing scale and speed. As AI continues to flourish, the surging computational complexity poses challenges in achieving efficient AI performance on mobile devices. Traditional digital computers, meanwhile, has reached the physical limits of chip size and density due to its own atomic structure. Therefor it is necessary and urgent to develop practically applicable post-Moore's Law technologies before it reach the end. The quest for new computing systems that offer faster processing, reduced costs, and lower power consumption has driven researchers and innovators to explore alternative approaches. The development landscape of these computing systems is illustrated in Figure 1.1. Neuromorphic computing was developed first, focusing on mimicking the brain function. Indeed, such exploration is limited because nature exhibits a wide variety of computing mechanisms, and besides our understanding of the information processing methods employed by the brain is still quite limited. Unconventional computing (UC), on the other hand, is more diverse because it explores unbounded physical phenomena and nonlinear effects rather than just moving toward a single rule: implementing discrete-value logic gates (digital computer) or brain function (neuromorphic computer) (Jaeger, 2021). In this case, unconventional computing is expected to bring advantages over classical computing way by breaking the paradigms, architectures, and even the underlying substrates.

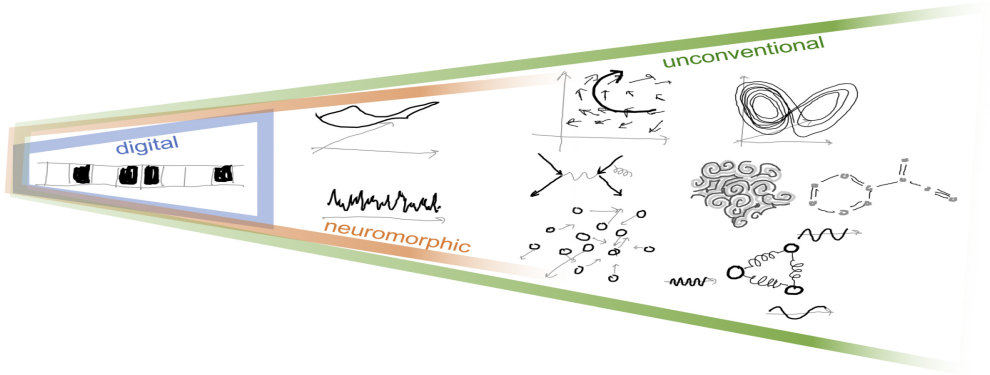

Figure 1.1: The scopes of digital, neuromorphic and UC in a napkin drawing (from Jaeger, 2021).

Information processing is widespread in nature, but lack of general unifying theory frameworks and concrete guidance for system design and utilization, therefore, how to exploit and utilize the computational resources of nature systems, as well as how to design and train such systems remains the most attractive and challenging subjects (K. Nakajima and Fischer, 2021).

1.1. UNCONVENTIONAL COMPUTING

THE term unconventional computing (UC) refers to those new formalisms different from conventional electronic digital computers to perform information processing. New formalisms generally include: analogue (Tomayko, 1985), chemical ((Margulies et al., 2006), biological (Shah et al., 2020), biochemical (Sangwan and Hersam, 2020), physical phenomena (Miller, 2010; Bhatti et al., 2017; Nielsen and Chuang, 2010) and mathematical exploration (van Daalen et al., 1993). As we have discussed, UC aims to enable information processing explicitly in various physical systems, utilizing them as computation resources. This approach offers several advantages. Firstly, UC expands our exploration of smaller, faster, and more energy-efficient natural computing resources, presenting a wider range of options for the next generation of computers. Secondly, these natural computing resources provide valuable inspiration, particularly from a neurological perspective, as they exhibit robustness and efficiency in their operations. Additionally, imperfections such as noise and defects are embraced as inherent phenomena in UC. Research has demonstrated that noise can be utilized as a computing resource in its own right (H. Tanaka et al., 2018; Kan, Sasaki, et al., 2021). Instead of trying to eliminate them, our design principles focus on leveraging these natural physical phenomena that incorporate these imperfections. This paradigm shift in embracing and leveraging imperfections opens up new possibilities for designing innovative and resilient computing systems.

Despite the fertility on UC field, those contributions are barely systematically interrelated due to their different theoretical frameworks and underlying principles. Research on the definition, design and theoretical guidance of UC system just gets started (Zhang et al., 2020) and needs more time and experiments to grow and refine into a unified formal theory. While most of the research at this stage is focused on the respective physical systems and their respective frameworks, there are still plenty of details and overviews of the systematical commonality to discuss.

1.1.1. COMPUTATION AND PHYSICAL PROCESS

SINCE UC makes the computational processes more like physical processes (Blum et al., 1998), we must made clear the distinction and connection between computational processes and physical processes. While computation is fundamentally a physical process, it does not necessarily require a specific material realization (Adamatzky, 2017). Modern computers, for example, achieve computation through programming languages, highlighting the separation between the abstract computations and their physical implementation. In essence, when we discuss computation in the context of a physical process, we are referring to the ability to derive alternative functions through abstract transformations of that physical process to accomplish our processing goals. Declaring a physical system as com-

putational implies that it can be replaced by other surrogate realizations while preserving some of the algebraic structure or incomplete description of the original physical system. This concept is often referred to as realization homomorphism (Hartmanis, 1993). It is worth noting that BJ MacLennan has provided insightful guidance on designing embodied computing systems, a concept within UC, in his book (MacLennan, n.d.). The approach involves understanding how information processing occurs in physical reality, abstracting the process from specific physical realizations, and finally realizing these abstract computational principles in an appropriate system. MacLennan's perspective offers valuable direction in the design and implementation of computational processes in physical systems.

Summarizing, physical processes are means to accomplish the end goal, which is the desired computation. Tradeoffs are different between computation serving physical system purpose and physical system serving the computation purpose (Adamatzky, 2017). For physical realization purpose, the approach to computing is based on the idea that physical systems can perform computational tasks using their inherent properties and interactions. In this paper we only focus on physical realization being the goal and will be discussed further in the next subsection.

1.1.2. COMPUTATION ON PHYSICAL ENTITIES

ACCORDING to computationalism, all physical systems perform computations. Although it may seem counterintuitive, this view is quite popular among some philosophers and physicists (Piccinini and Maley, 2021). If computing is a configuration of physical entities, we can develop new computing paradigms that can address some of the most pressing challenges facing our society today. Computation on physical entities involves the use of physical systems, such as atoms, molecules, or biological cells, to not only process but also store information. If computing needs can be implemented directly at the physical level, rather than using the centralized processing of information based on von Neumann architecture – moving data from memory to the processor and back again – the computational speed can be greatly improved, and the power consumption can be greatly saved. This is a vision of the rise of intelligent matter after AI.

At present, while new hardware paradigms from AI magnates mainly developed their neuromorphic hardware chips in a top-down way (like TrueNorth from IBM (Merolla et al., 2014) and TPU from Google), academic researches on the other hand, are concentrating on bottom-up approaches and expecting effcient UC by exploiting nanomaterials. Inspiration from nature is very valuable for the design of intelligent matter. The macroscopic functions of natural substances arise from complex internal structures and the interaction of molecular, nano-scale, and macro-scale building blocks. In a review (Kaspar et al., 2021) authors concluded that intelligent matter includes four functional elements: (1) sensors to interact

with environment; (2) actuators to respond to input; (3) network to process feedback; (4) long-term memory for information storage. Besides, such matter should also exhibit a high level of complexity and functionality. The essence of intelligent matter is still the development of neuromorphic computing devices. However, as explained at the beginning of this section, this limits the exploration of the intrinsic computational value of the broader physical systems. Besides, the lack of complete system-level demonstrations greatly limits the development and practical application of intelligent matter.

Returning to the discussion of physical realization of UC, a framework that exploits the dynamics of a physical system called physical reservoir computing has evolved to the exploration of a variety of physical computing systems (K. Nakajima, 2020). This introduces two concepts: reservoir computing (1.2) and dynamical systems (1.3), each of them will be covered in the next two sections.

1.2. RESERVOIR COMPUTING

RESERVOIR computing (RC) has become one of the most promising approaches in UC, owing to its many advantages, such as the relatively small computational burden of the training process, lack of a fading-memory problem, and relatively simple physical implementation (Kan, Nakajima, et al., 2021). RC is a framework derived from recurrent neural-network theory, providing rich dynamics and solving arbitrary problems relying on its inherent sensitivity to time, delays, and temporal structure (Dominey, 1995; Pascanu et al., 2014). Subsequent experiments show that the temporal recurrent network structure also requires working memory and a recognizing mechanism to the current element with the previous elements to learn sequence identity and its underlying rules (Dominey et al., 2003). In this context, Echo State Network (ESN) (Jaeger, 2001b) and Liquid State Machine (LSM) (Maass et al., 2002) were presented under different motives but were collectively unified as RC (Verstraeten et al., 2007). Computing on entities can be achieved by most physical systems by modeling RC as dynamical systems, whereupon implementing RC with various physical systems are actively discussed by researchers from physics, computer science, and neuroscience. In this section, we would like to start with the basic concepts and terms to introduce the origin and development of RC, and show their application values.

1.2.1. MATHEMATICAL FORMULATION

ECHO state network (ESN) originates from a discrete time, sigmoid-activation-function-unit recurrent neural network, where the network state can be metaphorically understood as "echo" of the input history (Jaeger, 2001b). There are four parts in the ESN framework: an input layer to feed the input signal to the reservoir, an recurrent network called

reservoir to process input data, an readout layer to collect and train the reservoir states for specific targets, and a feedback layer to project the output back to reservoir. From the point of view of mathematical expression, the four parts are defined by four matrices respectively. Input weights W^{in} usually involves the pre-processing of input data, reservoir weights representing the internal connections are an $N \times N$ matrices $W = w_{ij}$ (N is the number of internal processing nodes), the output weights W^{out} are used for the connections to output units, and the feedback weights W^{back}. When we have a teacher input $u(k)$ for $k = 0, ..., n_{max}$, the network states x at the time unit k is formulized as follows:

$$x(k) = F(W^{in} \times u(k) + W \times x(k-1) + W^{back} \times y(k)) \tag{1.1}$$

where $F = (f_1, ..., f_N)$ are the internal units' activation functions, typically are sigmoid functions. The output y is given by this formula:

$$y(k) = F^{out}(W^{out} \times (x(k), u(k), y(k-1)) \tag{1.2}$$

where the $(x(k), u(k), y(k-1)$ is a concatenation result of current input, current reservoir state, and the previous output. The training process, like all neural network schemes, is the most important process for testing results. Although an important concept in RC is the fast and constructive linear regression algorithm for its readout process, other new learning algorithms such as incorporating gradient descent algorithm are constantly being proposed with the structure design of RC becomes more complex and multi-layered (M. Nakajima et al., 2022).

In the definition of this reservoir (i.e., $N \times N$ matrix W), internal nodes are sparsely and randomly connected. Connectivity between nodes ensures the memory of past information in the reservoir, also called echo, while sparsity provided reservoir with individual dynamics by decoupling some of the sub-networks. There is a key factor that determines the performance of information processing, the spectral radius. Under the condition that the largest singular value σ_{max} of the matrix is less than 1, implying all states $x, x' \in [-1, 1]^N$. Spectral radius $\rho(W)$ is mathematically defined as the largest absolute eigenvalues of the weight matrix W. If the spectral radius is less than 1, the influence of inputs and previous states will disappear after a long enough time in the reservoir, implying the echo state theory holds. Otherwise, the impact accumulates after many iterations, resulting in confusion or even a failure ESN due to an asymptotically unstable null state. While the matrix weights is easy to set while holding this necessary condition, it is difficult to implement in the actual physical system because the adjustment to physical system parameter is difficult to exactly correspond to the weights value. Mathematically thinking, things would be easier if the weights matrix is a diagonal matrix, because the weights themselves are the eigenvalues

and feedback gain is the only inner parameter of each node. This scheme will be proved its practicability in the Chapter 2.

1.2.2. ECHO STATE PROPERTY

ABOVE discussion assumes that the $tanh$ activation function is used in the netwrok, so this criterion is no longer valid for other schemes such as reservoir construction using other activation functions or dynamical systems. Fortunately, a concept named Echo State Property (ESP) has been proposed to help us determine if the conditions for the echo state are met. ESP requires that the reservoir states, i.e., the responses of all the internal nodes, becomes dependent solely on the input sequence as the input sequence being injected. Figure 1.2 briefly and graphically shows how ESP happens and works in RC, where T is the target signal, input echo function ϕ is an intrinsic characteristic of the system, and outputs can be adjusted using the readout function ψ. As can be seen from this diagram,

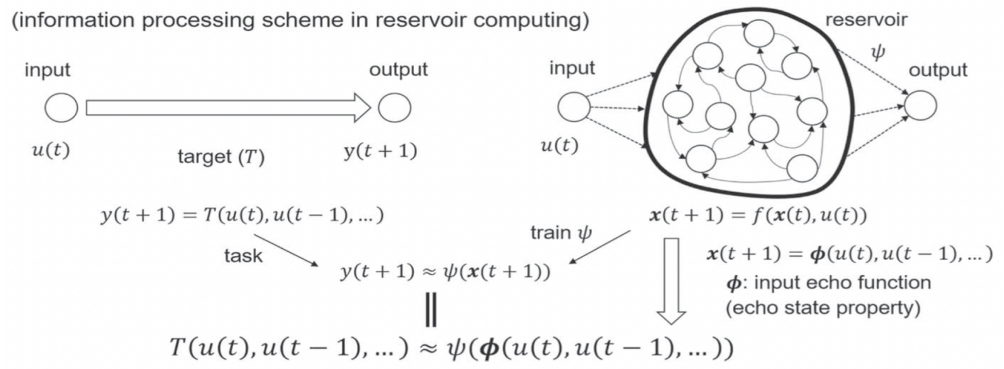

Figure 1.2: How ESP works in RC (from K. Nakajima, 2020).

when the reservoir fulfills ESP, $x(t) = \phi(u(t-1), u(t-2), ...)$, means that the current state of the reservoir is its own intrinsic dynamics to a series of past inputs. In this case, even if the reservoir is driven by different initial states (e.g., $x'(t)$ and $x(t)$), the reservoir state will gradually become synchronized under the drive of the same input sequence. It is expected to see $\|f(x'(t), u(t)) - f(x(t), u(t))\| \approx 0$ after sufficiently long time t. Finally, the aim is to train the output to approximate the target, i.e., $\psi(\phi(u(t), u(t-1), ...)) \approx T(u(t), u(t-1), ...)$. The mathematical expression $\rho(W) < 1$ that explained in the above section is the necessary condition for achieving ESP in RC, but it is not the sufficient condition for optimal realization of best performance. A definition of "ESP index" is proposed to chase ESP (Gallicchio,

2018) and defined as below:

$$Index = \frac{1}{P}\sum_{j=1}^{P}(\frac{1}{L-T}\sum_{t=1}^{L-T}\|\hat{F}(x_0,s)-\hat{F}(z_0,s)\|) \tag{1.3}$$

where P is the number of random initial states of each set, L is time-steps of the input, T is transient time-steps, x_0 and z_0 are two different initial states, s is the input sequences, $\hat{F}$ is the network state after iteration from the initial state under the influence of input sequences. In simple terms, this index represents the difference of the system states after T transition time between two sets of P initial states. Simulation results show that the value of ESP index is consistent with system computing performance in practical tasks, and the domain of empirical ESP validity well beyond commonly adopted ESP condition, i.e., $\rho(W) < 1$. With ESP index, we can easily estimate its asymptotic stability and assess whether the internal parameters of this system satisfy ESP in a specific physical system.

1.2.3. SHORT-TERM MEMORY

NOW, if a reservoir fulfills ESP, there is a relatively long-lasting echoing of the input signal inside the system, that is, the system itself has a certain memory function, which we call short-term memory, or fading memory. This property is important for an RC system to learn the underlying time-dependent characteristics. The concept of short-term memory is based on a statistical measure Pearson product-moment correlation coefficient squared, known as Coefficient of Determination. It examines how differences in one variable can be explained by the difference in a second variable when predicting the outcome of a given event, ranging from 0 to 1. Therefore, short-term memory conceptually reflects the proportion of the variance for the current reservoir output (at time unit t) that's explained by past inputs (at k time units before t, i.e.$(t-k)$). Mathematically expressed as follows:

$$MC_k = \frac{cov^2(y_k,\hat{y_k})}{\sigma^2(y_k)\sigma^2(\hat{y_k})} \tag{1.4}$$

where y_k is the target testing signal, $\hat{y_k}$ is the fitted signal and k is the unit of delayed input, which theoretically could be from 0 to infinity, but in experiments we usually set it from 0 to 10. It may be a little confusing to call k the delay unit, but it makes sense from an experimental point of view. When we calculate MC_k by the above equation, target signal y_k is constructed from the k^{th} delayed original input signal, i.e., $u(t-k)$. New concept introduced in RC, short-term memory capacity (STMC) of one RC system, is defined as the sum of all MC_k (Fette and Eggert, 2005):

$$MC = \sum_{k=1}^{\infty} MC_k \tag{1.5}$$

In standard ESN, STMC will be positively correlated with the size of the spectral radius λ_{max}. This is easy to understand because the abstract understanding of spectral radius is the sparsity degree of random internal connection between nodes in the definition matrix W, and the smaller the spectral radius, the more sparse the internal connection and the longer "echoing" time in the network, that is, the smaller the STMC. However, larger spectral radius is not always better because different tasks require different levels of memory capacity. I will discuss it in combination with specific tasks in future sections.

1.2.4. NARMA TASK

THE ultimate purpose of RC should be information processing ability. A widely used benchmark task in RC is nonlinear autoregressive moving average (NARMA) task (Atiya and Parlos, 2000). The key idea of NARMA is to build a dynamical model whose outputs are related not only to past inputs, but also to past outputs. Thus, a feedback output from this model would be another input signal to this model. This task evaluates the reservoir's ability to duplicate a higher-order dynamical model constructed from current and previous inputs. The number following NARMA represents the order of this nonlinear model, for example, NARMA2 is a second-order nonlinear model stated as:

$$y(t+1) = 0.4\,y(t) + 0.4\,y(t)\,y(t-1) + 0.6\,u(t)^3 + 0.1 \tag{1.6}$$

where $u(t)$ is the input signal along time sequences which in general a sequence of random digits in the interval of [0, 0.5], and $y(t)$ uses as the target signal. By training the output weights to render the output signal of reservoir as close as possible to the target signal. The output signal, i.e. the predicted values based on trained weights, is calculated following the equation $\hat{y}(t) = X_{test} \times W^{out}$. By quantifying the error between the target signal and the trained fitting signal, the performance of the reservoir for the NARMA2 task can be evaluated. Here introduce an example of how to the deviation between the target and predicted signals, namely, normalized-mean-square error (NMSE). NMSE calculates the mean square error of the predicted value and the target value at all discrete time points, and the formula is as follows with a testing length of M,:

$$\Delta E = \frac{1}{M} \frac{\sum_t [\hat{y}(t) - y(t)]^2}{\sigma^2(y)}. \tag{1.7}$$

1.2.5. PHYSICAL IMPLEMENTATIONS

DIFFERENT from recurrent neural networks, the input and reservoir inner connection weights do not need to be trained. Hence, the training process can be simplified to the training of output weights without affecting the inner states. This makes the training

process faster and training results more stable, since the training is only for the readout part. The faster training process is due to the greatly reduced training parameters and simplified training method (e.g., linear regression), and the more stable results are compared to the well-known shortcomings of back propagation algorithm, that are, the optimal results do not always happen after training and training is easy to fail for various reasons. Moreover, multitasking can be safely implemented under this framework because no interference occurs among the tasks (K. Nakajima, 2020). Furthermore, associating reservoirs with dynamical systems increases the arbitrariness and diversity of the physical realization, enables the development of physical reservoir implementation, providing promising candidates for next-generation AI paradigms (G. Tanaka et al., 2019).

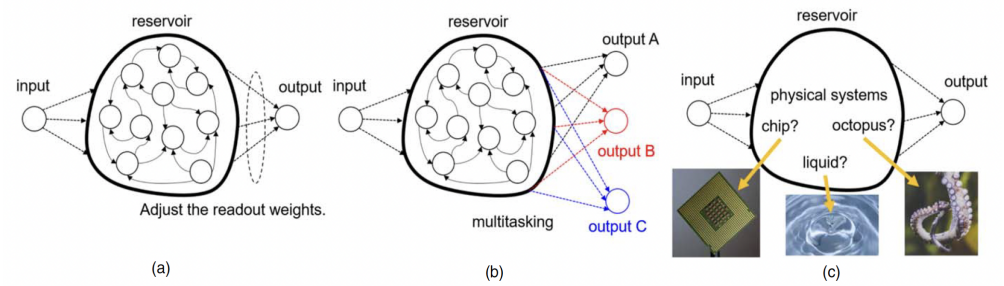

Figure 1.3: Advantages of RC (from K. Nakajima, 2020).

For physical implementations, RC concept is much simpler and clearer than the original ESN programming, and it generally includes three parts: input layer, reservoir, and output layer, where the feedback layer is ignored for easier realization. Input layer is usually linear functions or regularization of input data. Through a reservoir with massive amount of nonlinear processing nodes coupled with each other, input data are mapped into a high-dimensional space to facilitate the separation of states (Cover, 1965). The reservoir state at time unit t is expressed as follows:

$$x(t) = F(u(t), x(t-1)) \tag{1.8}$$

where F describes the nonlinear transformation ($R^N \times R \rightarrow R^N$) that determined by the inner physical law of the physical system. The reservoir state at time k is the general name for the state of all nodes (with the number of N) at time t, i.e., $x(t) = \{x_n(t)\}_{n=1}^{N}$. The output of reservoir $\hat{y}(t)$ at time t, which is the approximation to the target output $y(t)$, can be extracted from a linear combination with adjustable output weights W_{out}, coupled to reservoir states $x(t)$, expressed as:

$$\hat{y}(t) = W^{out} \times x(t). \tag{1.9}$$

A key advantage of using the RC paradigm is treating the system as a entirety and does not require control over individual elements, making physical implementation of RC suitable for many complex systems. Increasing physical reservoirs have been reported worldwide, for example, dynamical memristors (Du et al., 2017), neuromorphic materials (Sillin et al., 2013), soft body (K. Nakajima et al., 2015), spintronics (Tsunegi et al., 2019), quantum dynamics (K. Nakajima et al., 2019), and analog circuits and digital circuits (Appeltant et al., 2011; Alomar et al., 2016). Although the selection of physical systems is wide and the internal working mechanisms vary, there is a common pattern black box, which transforms and maps the input data into high-dimensional space with observable system states. In this case, even if we could not explicitly tell how to do or what to do on this physical level in advance, experience-oriented instructions on how to induce and exploit the intrinsic information processing ability are still achievable. One thing clear is that it is the variety of available intrinsic properties from these physical systems that make computing process diverse,powerful and useful. However, one of the challenges this presents is how to design the physical RC and how to evaluate and compare across different schemes. The conception of dynamical systems is a useful analytical tool to analyze a variety of physical RC from the perspective of physical concept and mathematical expression. In the next section, I will go into more details about dynamical systems theory, recur to RC, and how to analyze and evaluate an RC system with the help of its dynamical behavior.

1.3. DYNAMICAL SYSTEMS

At first, let a historical tour help us understand dynamical systems from its origin (Devaney, 2021): In 1980, Henri Poincaré had discovered that the stable and unstable manifolds might not match up and could actually cross at an angle, what we now call “chaos”. This is the beginning of chaos theory . In 1960, meteorologist E. N. Lorenz discovered that very simple differential equations could exhibit chaos and observed sensitive dependence on initial conditions, raising that very famous phrase: the flap of a butterfly’s wings in Brazil could possibly set off a tornado in Texas. In fact, chaos is a behavior frequently observed in dynamical systems. Chaos theory deals with the study of these highly complex and unpredictable systems and involves the analysis of the long-term behavior of a system, including its stability, periodicity, and sensitivity to initial conditions. This has a lot of similarities and relevance to the concept of many UC models, especially RC.

A lot of UC related concepts, as a matter of fact, are based on dynamical systems principles. For instance, genetic algorithms, use evolutionary processes to optimize solutions to complex problems and their behavior can be analyzed using techniques such as phase transitions and criticality. Another example, neural networks (e.g. RC), inspired by the structure and function of the human brain, are composed of interconnected nodes that can learn

1

and adapt, like a dynamical system. Their behaviors, therefore, can also be analyzed using techniques from dynamical systems theory, such as bifurcation analysis and chaos theory. Moreover, dynamical systems can be used as a substrate for UC, physical RC that we discussed earlier is one representative. Furthermore, as we mentioned in previous sections, physical implementation of RC thrives on the idea of harnessing the intrinsic nonlinear responses and dynamics of physical systems. The research basis of physical RC is that the reservoir part can be regarded as a huge hyper parameter of a dynamical system. It is necessary to gain the knowledge of nonlinear dynamical systems before we could analyze or harness this system. Some related concepts, definitions and properties suitable for physical RC are introduced in the next subsections.

1.3.1. CONDITIONAL LYAPUNOV EXPONENT

IN physical RC system, one necessary property is the similar output trajectory generated by the same input sequences, independent to the initial state of the dynamic system, which was the Echo State Property (ESP) of RC discussed in Section1.2. In the field of dynamical system theory there is a technical term relates to this property, called Conditional Lyapunov exponent.

To explain this property mathematically, we give two different initial states of reservoir $r(0)$, $\tilde{r}(0)$. If these two states always converge to the same state under the common input signal $s(k)$, where k is the time sequences, the signal-driven dynamical system shows common-signal-induced synchronization, i.e., $\|r(k) - \tilde{r}(k)\| \to 0$ $(k \to \infty)$. Such property demonstrates the reproducibility of the output sequence to the input sequence (except the initial transient period), also known as the asymptotically stability. Derived from dynamical system, Lyapunov exponent is a mathematical tool that can help us characterize asymptotically stability. Conditional Lyapunov exponent, conditioned by the stochastic property of input sequences $s(k)$, measures the exponential growth (or decay) rate of an infinitesimal disturbance $\delta r(k)$ to the state $r(k)$ and the mathematical definition is as follows:

$$\lambda := \lim_{t\to\infty} \frac{1}{T} ln\|\delta r(T)\| \tag{1.10}$$

This equation can be interpreted as $\|\delta r(k)\| \propto exp(\lambda k)$, where $\lambda < 0$ implies asymptotically stability.

Revisit the concept physical RC system, any input-driven physical systems with a negative conditional Lyapunov exponent is appropriate.

1.3.2. EDGE OF CHAOS

ANY dynamical system can appear synchronized state ($\lambda < 0$) and asynchronized state ($\lambda > 0$) by adjusting its internal parameters. In the asynchronized state, the response of this dynamical system is no longer related to the input sequence, but rather appears disordered, i.e. chaotic. There exist a transition stage where the system transites from synchronization to asynchronization, which we call edge of chaos. The edge of chaos is an empirically defined state that is about to reach the transition point, i.e. ($\lambda < 0$) and ($\lambda \simeq 0$). This empirical law has been considered a state in which dynamic systems exhibit highest information processing performance. M. Inubushi et al. demonstrated this law by an experiment on an ESN model, as shown in Fig.1.2.

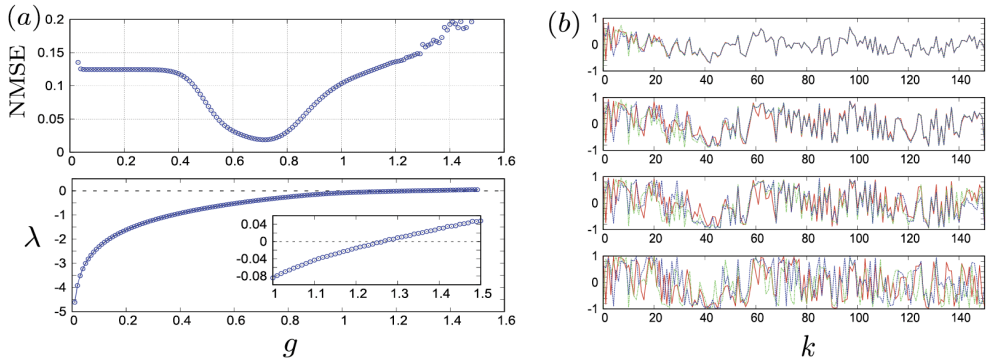

Figure 1.4: Demonstration of Edge of Chaos law (from K. Nakajima and Fischer, 2021).

The information conveyed in Fig.1.4(a) is that ESN model shows the best performance in information processing at $g \simeq 0.7$ (upper figure), corresponding to a negative Lyapunov index $\lambda \simeq -0.36$ (lower figure) and approaching the transition point of $g = 1.25$. Given three different initial states on this ESN model, Fig.1.4(b) intuitively exhibits the time evolutions of three ESNs with these initial states with the parameter g increasing as 1.0, 1.2, 1.3, 1.5 (from top to down). When $g < 1.25$, convergence occurs and the rate of convergence slows down as g increases (in consistent with the conditional Lyapunov exponent λ goes from negative to 0). When $g > 1.25$, conditional Lyapunov exponent λ becomes positive, convergence no longer exists and the system becomes chaos.

Conditional Lyapunov exponent is also associated with short-term memory, the concept we introduced in the previous section (1.2). In the above example, the time required for convergence affects short-term memory to some extent and is roughly speaking positively correlated with the reciprocal of the conditional Lyapunov exponent. This makes sense, because when the system no longer converges, the influence of past input signals on the system also disappears. Studies have confirmed that the memory capacity of the system

reaches its maximum at the edge of chaos (Toyoizumi and Abbott, 2011).

1.3.3. MEMORY OR NONLINEARITY

REVISIT the previous discussions, nonlinearity is essential both from the perspective of dynamical systems and from the perspective of RC construction. Memory capacity, on the other hand, is also necessary for computing on physical entities. However, the nonlinearity of system dynamics inevitably destroys the original information and makes it difficult to maintain long-term memory. The Memory-Nonlinearity trade-off has been clearly shown by Inubushi and Yoshimura in their numerical experiment (Inubushi and Yoshimura, 2017), and the behind mechanism has also been described with the use of variational equation relating to information theory. Besides, they also verified that a linear-nonlinear-combined system performs better in Santa Fe Laser data set and the NARMA task (with respective optimal mixture rates p) by replacing a small number of nonlinear nodes to linear nodes. This improvement less dependent on the details of the reservoir (i.e., topology: random sparse coupling or the ring coupling) and tasks (performance is always improving) (K. Nakajima and Fischer, 2021), but the best performance still need an optimized mixture rates p.

In summary, the trade-off between memory capacity and nonlinearity should depend on the specific task, because different tasks require various amount of memory capacity and nonlinear transformation. Therefore, when we discuss the information processing capability of a known physical system, we need to carefully explore its internal dynamic characteristics and memory characteristics in respective tasks.

1.4. RESEARCH GAP

UNCONVENTIONAL computing (UC) has received increasing attention due to the variety of new formalisms as well as their rich but undeveloped potential of breaking through the bottleneck of traditional computing paradigm at bottom. UC systems, in contrast to digital systems, tend to rely more on continuous values and take the advantages of intrinsic physical properties of the system itself rather than of Boolean logic to do the calculation. This requires that the paradigms for analog computing must be robust to thermal and electrical noise and jitter. Moreover, due to the lack of a standardized and unified paradigm like the digital computer, the development of UC is diversified, focus not only on conceptual innovations, but also on breakthroughs at the device level. However, the overall framework is still missing.

Among those requirements, physical dynamical systems with the implementation of artificial neural networks for information processing is a good integration in this aspect. A

kind of artificial neural network called reservoir computing can not only utilize the concept of dynamic system, but also conform to the framework of artificial neural network with the training process simplified to only in the readout process. Hence, the main impact of the network performance comes from the "reservoir", i.e., the physical dynamical system, and it's the only part requires configuring. These systems can vary greatly in terms of physical property, architecture, and complexity, while they all share one common capability: information processing, or computing ability. Despite the diversity of physical system choices, electronics stands out as the most attractive candidate due to the mutually reinforcing development of electronics and computers. However, we are entering a new era of electronics: nanoelectronics, where their extremely small size allows the creation of a highly dense and complex system for future unconventional processors. Nonlinear response of nanomaterials shows potential for information processing (Bose et al., 2015; H. Tanaka et al., 2018), but lack of unified implementation framework and has not been applied to difficult practical tasks. In this case, designing nanoelectronic systems, determining their realistic computational potential, and evaluating the performance across different systems and architectures have also become crucial topics of discussion.

Fortunately, with the help of dynamic systems theory, we have a more concrete and clear direction for the analysis of physical systems. The physical system is considered a reservoir substrate with a huge hyper parameter. By selecting the internal parameters of the system, it can exhibit computationally desirable characteristics, commonly referred to as the "Edge of Chaotic" or "Echo State Property" in technical terms. Most unconventional systems take the approach of configuring the parameters simply randomly, with the increase of complexity, however, the probability of the reaching the optimal parameters also decreases. Therefore, the main goal of this doctoral project is to start with the simplest rules and nanoelectronic system design, gradually explore the principle of information processing under the framework of reservoir computing, uncover the mystery of "black box" and find a general, understandable, and simple paradigm for its utilization. Furthermore, it is also necessary to analyze the characteristics of the system from many aspects, evaluate the computing power of the system from a comprehensive point of view, and find the most suitable computing tasks for each physical system. The ultimate purpose is to achieve complex and powerful computing functions through the simplest devices and principles.

1.5. RESEARCH MOTIVE AND TASKS

Research motive:

To explore the information processing potentials of simple physical systems with straightforward principles.

Research tasks:

T1 Design information processing systems with devices that are easy to interpret and operate.

T2 State the intrinsic nonlinearities and desirable working principles for different architectures and schemes.

T3 Quantify the information processing ability in different kinds of tasks.

T4 Understand the strengths and limitations of different physical schemes for effective utilization in the aspect of computing.

1.6. METHODOLOGY

To exploit the nanoparticle device with the simplest rules and paradigms for information processing, intrinsic dynamical behaviors of specific nanoparticle should be explicit at first. Observe and regulate its behavior, and design an appropriate scheme that achieves the desired computing function in accordance with the law of its change (T1). Take the implementation scheme of reservoir computing framework as an example, the dynamic response expression of the inner nodes and control factors of system dynamics should be investigated first. Besides, a surrogate electronic circuit or simulation model can be also used to replace a real nanoparticle device for easy control of the nonlinear response and detailed investigation of the system structure (T2). After determining the structure of the reservoir and the regulation of the dynamic characteristics of the nodes, the computing system is constructed on the surrogate electronic circuit, simulation mode, or even the real nanoparticle device. Next is to evaluate the utilization value and potential of unconventional systems on information processing tasks, which can be studied by some basic benchmark tasks (T3). The last and most important step is to select the most appropriate information processing task (for instance, time-sequential signal prediction, classification task, image recognition, and speech recognition) for the system and test its performance based on the above investigations (T4). The methodology of each research task is further stated below.

T1 Polyoxometalate (POM) exhibits multiple oxidation and reduction characteristics owing to the unique structure, which could be potentially useful for computing system. Based on the characteristics shown on the neuromorphic device that our group designed earlier (H. Tanaka et al., 2018), we try to explore unconventional computing solutions with the use of such molecular or the network device. Under the concept of dynamical system, the nonlinear current-voltage ($I-V$) responses of this device can be utilized to build a "reservoir"

composed of a large number of nonlinear processing nodes. Generally, these processing nodes are randomly connected with each other. However, in order to design the simplest dynamical system, the internal nodes are first designed as a group of parallel nodes that are not connected to each other, and the I-V characteristic working as activation function can be easily reproduced by a surrogate electronic circuit composed of two diodes in parallel back to back. By artificially introducing several parameters (input gain, feedback gain, and feedforward gain), the I-V curves representing the dynamics of the system, as well as the system structure can be adjusted. After confirming its feasibility and practicability by research tasks T2 and T3, the computing scheme can also be implemented on real molecular devices. Under the concept of stochastic computing, random spikes and noise generated spontaneously by the charge transfer can be used to replace the stochastic number generator and calculation on hardware level. Considering the execution period and difficulty modifying the real molecular device, a surrogate model simulating the charge transfer process between electrodes is established as the unconventional computing system.

T2 As introduced in Chapter 1.2, echo state property (ESP) is an important and useful analytical tool to estimate the computing power of a dynamical system. Hence, for a given designed dynamical system, the first thing to verify is whether it satisfies ESP. The mathematically straightforward way is to calculate the Lyapunov exponent (Equation 1.10) and the ESP index of this system (Equation 1.3). In addition, according to the definition of ESP, there is another simple and intuitive observation method, that is, to observe whether the system shows convergence after a initial period of chaos. As Fig.1.4(b) shows, convergence and non-convergence can be clearly observed through the output states of the system. Another thing to figure out is where the internal nonlinearity comes from and the controllable parameters. For one pair of diodes circuit, the nonlinearity comes from the current response characteristics of the diode as well as the external control parameters, so it is necessary to show the variation of the response under different parameters. However, for real molecular devices or nanoparticle networks, the factors affecting nonlinear response is unknown, thus it is necessary to analyze and explore more comprehensively, from hardware device (electrode material, structure, distance, setup device, and so on) to nanoparticle properties (molecular structure, electrochemical properties, and so on).

T3 ESP is a property related with temporal structure, hence the utilization value of RC is focusing on solving time-dependent problems. Short-term memory capacity and NARMA task introduced in Chapter 1.2 are common benchmark tasks with time-sequential signals. However, there are many kinds of information processing methods, and it is obviously not scientific or comprehensive to evaluate computing ability by the performance of only one benchmark test. The most commonly used index to quantify prediction performance is to calculate the normalized error between the test signal and the teacher signal. There are also

certain considerations for the selection of test signals and training signals. Training signals should wait until the system being stable, that is, after the system displaying convergence state, so as to avoid the initial chaos interfering with the training weights. In addition, the ratio of training data to test data is also critical. Insufficient training data will lead to the weight can not be trained to the optimal solution, too much training data will lead to overfitting problem and thus affect the test accuracy. Some other tasks like the reproduction of periodic signals, prediction for sensor signal with hysteresis, are also time-sequential signals. For other tests such as classification tasks, it can be directly calculated by the accuracy of statistical classification results.

T4 As discussed in Chapter 1.3.3, there is a trade-off between memory capacity and nonlinearity in all dynamical systems. Thus the performance in T3 can vary from task to task and the selection of the most appropriate task for the system needs to be considered comprehensively by the investigation results of the system memory capacity and nonlinear characteristics. For example, sensing signals that are strongly related to past input history requires strong memory capacity and relatively weak nonlinearity, while some signals that have no strong historical relationship with the past inputs, such as preprocessed data for classification, do not require high memory capacity. This explains why relatively weak performance in NARMA task or short-term memory capacity does not mean the failure of information processing ability of system. Therefore, the ultimate task is to interpret the computational paradigm through the analysis of the above investigations and the task that the system performs best.

1.7. THESIS OUTLINE

THE overall structure of this paper can be summarized as follows: Chapter 2 introduces a physical computing system constructed using basic circuit elements, specifically diodes, starting from mathematical concepts. It provides a comprehensive investigation of the system intrinsic characteristics and examines its performance across different tasks. Following that, in Chapter 3, one simple, practical, and easily achievable computational processing architecture discovered in Chapter 2 is utilized to implement the same approach on a micro-scale device. In this micron-scale device, different solutions exhibit performance differences due to their distinct current-voltage response characteristics. Unlike the previous two chapters, Chapter 4 introduces an example of applying the known dynamic characteristics of a nano-device within existing computational paradigms. It explores the utility of existing devices and potential as a computing system by resorting to their dynamic characteristics. Chapter 5 provides with an overall discussion on these entire works, evaluating all physical systems from physical device level to computing performance. Last, Chapter 6

responses the research motive and tasks proposed in this chapter and also points out the limitations of this study as well as recommendations that extend to future research directions that may inspire other researchers.

2

RESERVOIR COMPUTING WITH SIMPLE STRUCTURE AND NONLINEAR RESPONSE

This chapter is published as:
S. Kan, K. Nakajima, Y. Takeshima, T. Asai, Y. Kuwahara, M. Akai-Kasaya, "Simple Reservoir Computing Capitalizing on the Nonlinear Response of Materials: Theory and Physical Implementations", *Physical Review Applied*, **2021**, 15, 2.

ABSTRACT The potential of nonlinear dynamical systems serving as reservoirs has attracted much attention for the physical realization of reservoir computing (RC). Here, we propose a hardware system working as a reservoir with one simple form of nonlinearity that reflects the intrinsic characteristics of the materials. We show that insufficient dynamics in such physical systems can perform like complex dynamical systems with the assistance of external controls. Based on the idea of spatial multiplexing, this dynamical system is studied under two frameworks. The correlation between structural adjustments of the reservoir and system performance in processing various types of task is proposed. Our results are expected to enable the development of material-based devices for RC.

2.1. INTRODUCTION

ARTIFICIAL neural networks (ANNs) have been actively developed as the foundation of artificial intelligence (AI) systems. ANNs can be grouped according to their architecture into feedforward (Schmidhuber, 2015) and recurrent networks (Pascanu et al., 2014), making them suitable for various tasks. Reservoir computing (RC), a framework derived from recurrent neural-network theory, has generated significant interest, owing to its many advantages, such as the relatively small computational burden of the training process, lack of a fading-memory problem, and relatively simple physical implementation. Generally, there are three parts in a RC implementation: an input layer to feed the input signal to the reservoir, a reservoir to process input data, and an output layer to read out the reservoir states and use them for training (Verstraeten et al., 2007). Through RC, input data are mapped into a high-dimensional space to facilitate the separation of states (Cover, 1965). Generally, the output signal, y, at time t can be extracted from a linear combination with adjustable output weights, W_{out}, coupled to reservoir states x at time t, expressed as

$$\begin{aligned} y(t) &= f_{out}[x(t)], \\ f_{out} &= W^{out} x(t) \end{aligned} \tag{2.1}$$

Different from recurrent neural networks, the RC input and reservoir weights do not need to be adjusted. Hence, the training process can be simplified to the training of output weights, such that it does not affect the node states. Various reservoir-model architectures are proposed, such as multilayer (Gallicchio et al., 2017; Malik et al., 2017)), parallel time delay (Grigoryeva et al., 2016)), single-node structures with time multiplexing (Appeltant et al., 2011), and growing echo state networks (ESNs) (Qiao et al., 2017). A physical RC scheme (Figure 2.1) was recently proposed (K. Nakajima, 2020), in which the reservoir part could be any physical dynamical system serving as a computational resource instead of a conventional recurrent network. This concept enables the development of physical reservoir implementation, providing promising candidates for next-generation AI paradigms (G. Tanaka et al., 2019). Typical examples of dynamical systems serving as the reservoir (Figure 2.1) include dynamical memristors (Du et al., 2017)), neuromorphic materials (Sillin et al., 2013), and a soft body (K. Nakajima et al., 2015). Criteria for the normal operation of physical dynamical systems as reservoirs are studied (K. Nakajima, 2020). One key criterion is that the system should generate different responses with trends similar to a given input signal, as shown by the different system states in Figure 2.1. The nonlinear response of a signal with its complex network or individual but relational dynamical response generates the required dynamics for a RC. Such a large number of product sums of diverse response signals improves the RC performance. Current research on physical RCs relies heavily on software

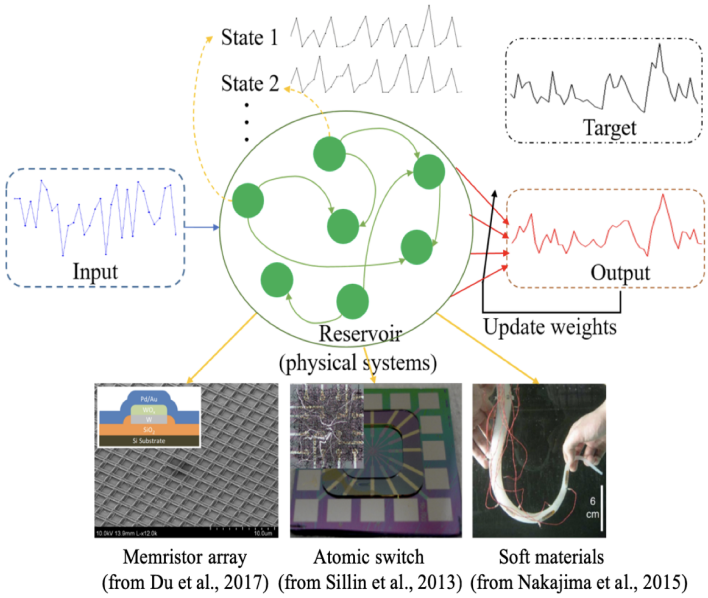

Figure 2.1: Concept of a physical RC. Reservoir can be replaced with any physical system that responds differently to a given input signal. Output signal is the linear combination of all system states coupled to adjustable output weights. There can be multiple output signals, depending on the specific task.

to replicate nodes (Appeltant et al., 2011;Torrejon et al., 2017), imposing a burden on performance. Reductions in signal conversion load and memory consumption are required for hardware RCs.

The nonlinearity of the signal response of nanomaterials is proposed for information processing, where their extremely small size allows the creation of a highly dense and complex system for future neuromorphic information processors (Bose et al., 2015; H. Tanaka et al., 2018). Because the conductance of a nanomaterial is sensitive to an electrical potential difference, the current-voltage ($I-V$) characteristics can be used to distinguish nonlinearity. Various nonlinear responses, such as temperature-independent tunneling current through a molecularmonolayer that functionalizes nanoparticles (Nishijima et al., 2018) and stochastic switching via ion movement dynamics (Du et al., 2017), are reported for such systems.

However, nanomaterials with a significant nonlinear conversion capacity do not always show sufficient dynamics, and they are not easy to control or observe. Therefore, to examine the availability of a simple nonlinear $I-V$ response of materials (H. Tanaka et al., 2018; Nishijima et al., 2018; Crljen et al., 2005; H. Fujii et al., 2017), as well as its usage in RC systems, we construct an electronic circuit working as the nonlinear node and introduce an external control to generate a dynamical response. In this study, we use a single phys-

ical node constructed from an electronic circuit, and a large number of virtual nodes are generated to form a reservoir by time and spatial multiplexing.

Compared with other physical reservoir schemes, the dynamics in our physical systems are more controllable and simpler. The much lower external memory consumption and signal processing complexity of our control allow the construction of a parallel process for multiple devices, so that a future information processing device can be developed utilizing highly integrated nanomaterials. We investigate the feasibility and effectiveness of the proposed scheme by testing its performance in various tasks and thereby gain insights into the physical realization of a RC. The broader aim of this work is to an alternative RC system, so that appropriate nanomaterials can be used for building them.

2.2. DESIGN SCHEME AND TESTING METHOD

In this section, we provide an overview of our design scheme to clarify the concept of our physical reservoir. The basic principle of our scheme is to build a dynamical system based on the $I-V$ curves of materials with a simple external control. Two parameters (i.e., input gain, e, and feedback gain, a) are introduced to increase the dynamics and represent individual processing nodes in the reservoir. The $I-V$ curve-based model is defined by the following functions:

$$V_{in}(t) = aV_{out}(t-1) + eu_t \tag{2.2}$$

$$V_{out}(t) = \begin{cases} V_{in}(t) + V_{th}, & \text{if} \quad V_{in}(t) \leq -V_{th} \\ 0, & \text{if} \quad -V_{th} < V_{in}(t) < V_{th} \\ V_{in}(t) - V_{th}, & \text{if} \quad V_{in}(t) \geq V_{th} \end{cases} \tag{2.3}$$

where u_t (random sequences within [-5,5]) are the input signals, and V_{th} is the threshold voltage. t represents the sequential number. Figure 2.2 shows the simulated results of this model, where three $I-V$ curves are generated from three values of e and a, and the value of V_{th} is fixed. In this model, the input gain, e, determines the slope of the curve, and the feedback gain, a, affects the distribution around the curve.

Next, we propose a RC system that offers tunable internal nonlinearity for each node. The nonlinearity needed is shown for each node in Figure 2.3. Figure 2.3(a) displays a diagram of our first scheme: parallel node. Nodes are assigned with different values of input gain, e, and feedback gain, a. Hence, the n^{th} node is defined by e^n and a^n. This separately processes the input stream, $u(t)$. The t^{th} input fed into the nth node can be read as

$$V_{in}^n(t) = a^n V_{out}^n(t-1) + e^n u(t) \tag{2.4}$$

The transient-node state, $V_o^n ut$, which characterizes the transient response to a certain

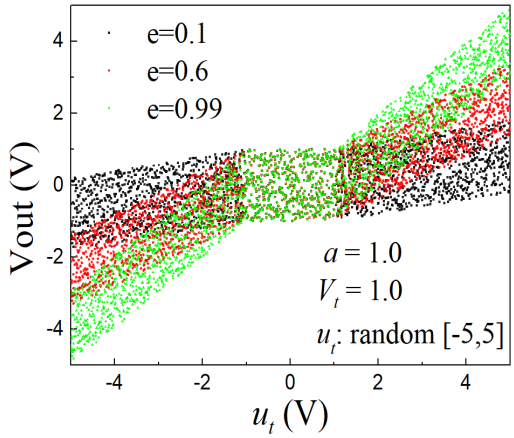

Figure 2.2: Response function settings. V_{th} is the threshold voltage and u_t is the random input signal. $I-V$ curves in black, red, and green represent different values of e, which are reflected in the different slopes. Larger e corresponds to a steeper slope. All three curves have the same shape because they have the same value of a.

input, can be expressed as

$$V_{out}^{n}(t) = f_{out}^{n}[V_{in}(t)] \tag{2.5}$$

where f^n denotes the nonlinear transition function of the nth node in the reservoir. During the nth-node process, each input is fed into the node state by coupling it to the same input gain, e^n, and the past-node state is fed by coupling it to the same feedback gain, a^n. The output layer reads out all of these transient states, in which the final output can be obtained as the linear summation of the states of all nodes weighted by an output weight. The output weights are optimized using a training procedure to best fit the output signal to the target signal.

In our second scheme (i.e., parallel group), another external parameter, feedforward gain, is introduced to provide connections between nodes in the reservoir. The node connection structure inside the reservoir is shown in Figure 2.3(b), where the nodes are divided into groups of six. These groups are parallel, and we call it the *parallel-group structure*. In each group, e^n and a^n serve as input and feedback gains for the first node, respectively. For the remaining nodes (2–6), a^n serves as the feedforward gain coupled to the output from the previous node at the same time sequence, and λ serves as the feedback gain. The input voltage in each group can be formulated as

$$\begin{aligned} V_{in}^{6n+1}(t) &= a^{6n+1}V_{out}^{6n+1}(t-1) + e^{6n+1}u(t), \quad \text{if first node} \\ V_{in}^{6n+x}(t) &= a^{6n+x}V_{out}^{6n+x-1}(t-1) + e^{6n+x}u(t) + \lambda V_{out}^{6n+1}(t-1), \quad \text{if rest nodes} \end{aligned} \tag{2.6}$$

Although the input stream is processed sequentially, the central idea of our design includes

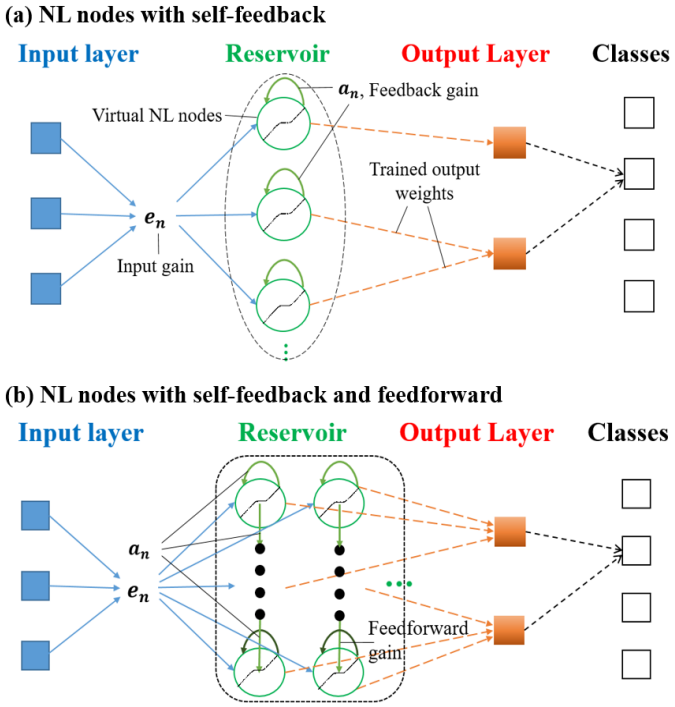

Figure 2.3: Diagram of RC scheme. (a) Parallel-node structure. Nodes in this scheme are parallel to each other and independent. Different nodes are defined by different value pairs of a and e. Each node also has self-feedback. Nodes are independent. (b) Parallel-group structure. Nodes are divided into some groups, which are parallel. Six nodes in each group have feedforward from the previous node.

a solution based on a parallel implementation. Such an implementation derives from a spatial multiplexing research model (K. Nakajima et al., 2019). Both implementations are related to the ESN framework (Jaeger, 2001a) and satisfy the echo-state property. This system is rather simple and lacks complex nonlinear dynamics, allowing observation of the effects of parameter tuning and architectural changes. Our aim is to verify the effectiveness of such simple dynamics (i.e., single nonlinear response and independent feedback) during information processing. Furthermore, such a parallel framework requires minimum external memory because it needs to store only the previous state. The required external memory will not swell under such parallel processing.

Next, we realize the physical RC system with tunable nonlinearity of a hardware system. Considering the similarity of the $I-V$ characteristic of diodes to the nonlinear response of materials and its easier implementation, we construct our physical system using an electronic circuit in which the nonlinear response is realized with two antiparallel diodes (1S1585). This part is called the physical nonlinear (NL) node. The output-to-input relation we need is plotted in the top panel of Figure 2.4, with an input voltage range [-2.1, 2.1], and

thus, an output voltage range [-1.5, 1.5]. The input range is defined between -2.1 and 2.1 V due to the limitation of our hardware system (i.e., DAC PCF8591), but some inputs will go beyond the range with the effects of external control. To enable the node to return to the normal range, we use a regression setting in our hardware system, that is, if the input voltage exceeds the defined range, the input voltage will return to zero; if not, the input is its original value. The entire procedure of information processing is shown in Figure 2.4 under the control and monitoring of the Ardunio Mega 2560 development board. To save space, we draw on the concept of time multiplexing (Appeltant et al., 2011), so that only one single NL node is used to create our RC system. We define N individual virtual nodes by generating N value pairs of a and e, which means that the input signal is processed N times through the physical NL node, whereas the e and a values are updated each time. Thus, a single physical NL node serves as different virtual nodes at different times. Owing to implementation under one single physical node, compared with the parallel-node scheme, the parallel-group scheme requires external memory for all output states from the ante nodes, $V_{out}^{6n+x}(t-1)$, to create the connection mentioned in Equation 2.6.

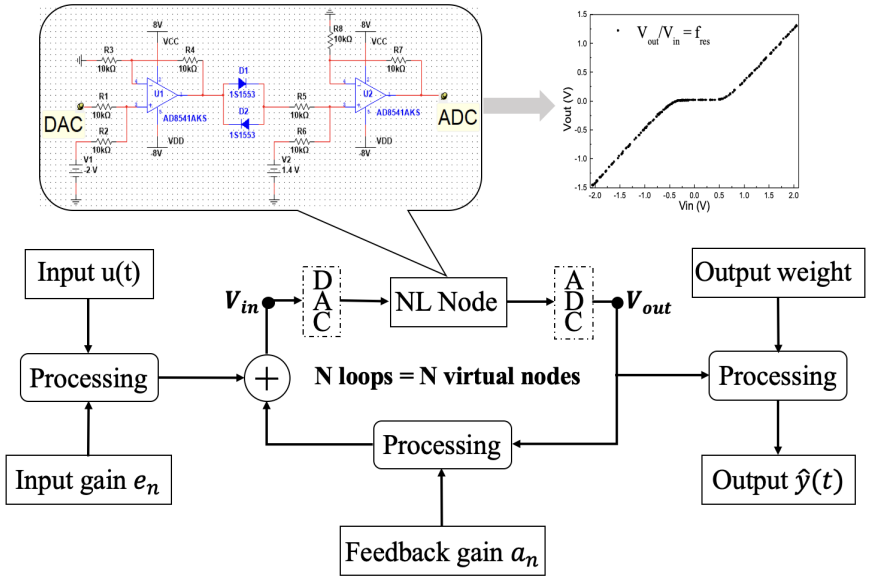

Figure 2.4: Schematic of physical realization. NL node with output-to-input characteristics generates N virtual nodes via time multiplexing, so that the entire input signal is processed N times. Each input data item passing a node is coupled with the same input gain. Next, it is fed into the node with the previous node state coupled with the same feedback gain. Inputs are transformed using a digital-to-analog converter (DAC) when fed into the NL node and by an analog-to-digital converter (ADC) to obtain output voltage from nodes. Training process is then conducted using all collected digital data.

Because the input and feedback gains have a direct impact on the processing results, an

optimization of their values is necessary. Even if a and e are randomly selected, an optimal range should be determined. We leverage a mapping method that simultaneously displays the changes in dynamics along both e and a. Parameter a ranges from 0 to 2.6, and parameter e ranges from 1.1 to 3.1, both having a step of 0.2. For each pair of e and a, we feed an input signal of 300 sequences randomly generated from -0.5 to 0.5 to our reservoir and calculate the variances of the node states (300 output voltage values). A variance mapping graph (Figure 2.5) is then plotted for various parameter values. A greater variance indicates richer dynamic characteristics of the nodes because the output voltage distribution is more scattered. Based on the distribution of variance, different ranges (i.e., shapes, colors, and their combinations) are selected for testing in the benchmark nonlinear autoregressive moving average (NARMA2) task, introduced in Section 1.2.4, to help us adjust the range selection. After 30 tests under different ranges, we determine the parameter ranges that give the best performance, as marked with a black parallelogram in Figure 2.5.

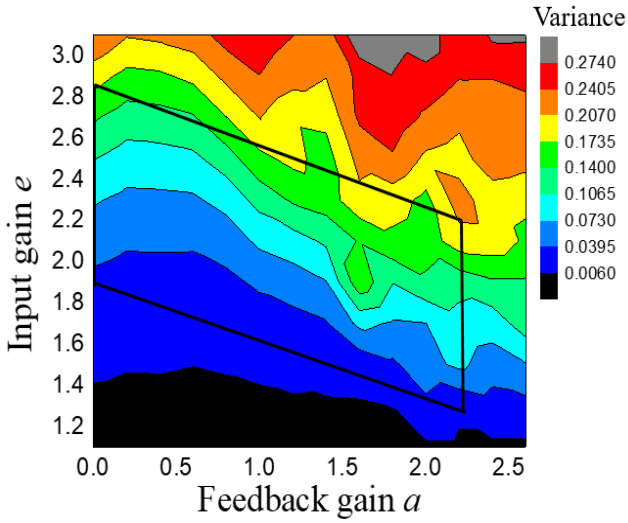

Figure 2.5: Variance mapping. We calculate 14 × 11 groups of variance of node states for this mapping graph; 14 because parameter a ranges from 0 to 2.6 in increments of 0.2 and 11 because parameter e ranges from 1.1 to 3.1. After around 30 tests under different ranges in the NARMA2 task, we determine the parameter ranges that give the best performance (marked with black parallelogram).

This method determines the values of input gain e and feedback gain a under the parallel-node scheme. In the parallel-group structure, parameters a and e have the same ranges as those determined for the parallel-node structure. Parameter λ is 0.05, 0.1, 0.12, 0.12, and 0.12 for the remaining five nodes in each group. Gradually increasing the feedback parameter λ is helpful for remembering previous input information. These values are selected according to several tests under the benchmark NARMA2 task. Owing to the limitation of

physical realization for finding the precise range, we conduct limited groups of tests to determine all parameters. However, we believe that the range choice can be further improved.

2.3. PERFORMANCE ANALYSIS

AFTER parameter ranges are determined, we test the performance of our diode-based physical system. The results of the following tasks become experimental data from our hardware reservoir. The values of the parameters in all tests, if not specified, are taken from the parameter ranges discussed in Section 2.2.

2.3.1. FEATURES OF OUTPUT OF NODE STATE

WW first test the output features of our physical RC to the input signal, $u(t)$, comprising sequential random data in the interval [0, 0.5]. This range is used for the input signal because it is a common range used in NARMA tasks, although it goes against our design principles, which state that both positive and negative values should be used, as in Figure 2.2. To use both positive and negative sides of this nonlinear response, we randomly set some of the e values to be negative.

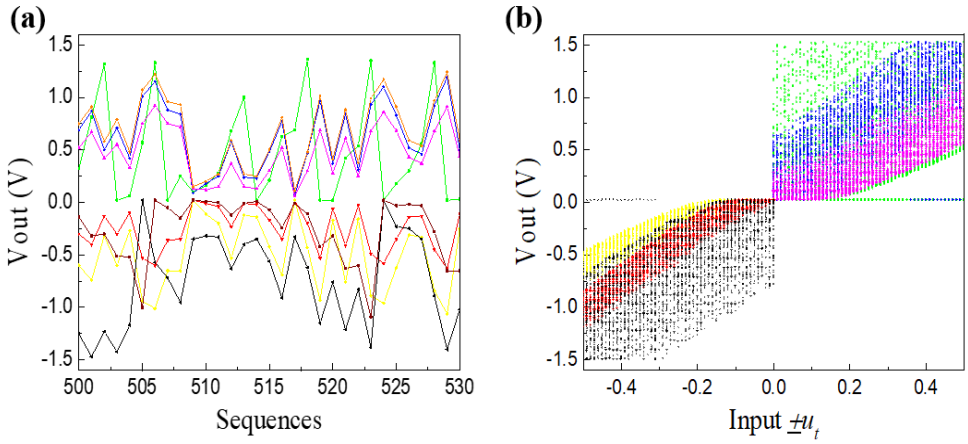

Figure 2.6: Output features. (a) Sequential node states. This graph plots 10 node states out of 300 nodes along the time sequences from our physical reservoir based on the scheme in Figure 2.3(a). Input signal to this reservoir includes 4000 random data items, $u(t)$, in the interval [0, 0.5]. Output data are divided into either all positive or all negative based on the sign of the input gain, e. (b) Typical input-to-output characteristics. Outputs of six nodes selected from 300 nodes. Negative part, caused by a negative sign of input gain e, is supposed to be on the right side of the x axis, because $u(t)$ is in the interval [0, 0.5]. However, it is flipped along the vertical axis for a clear $I-V$ curve.

Figure 2.6(a) displays the sequential states of eight selected nodes from 300. All states having the same abscissa use the same input value, but each node responds differently.

Figure 2.6(b) shows the relation between output and input, which is similar to that of the $I-V$ curves (Figure 2.2), where the slopes in both the graphs reflect feedback gain a. However, unlike Figure 2.2, the input gain, e, is not fixed in this case.

2.3.2. RESULTS OF NARMA2 TASK

HERE, we report the performance of our physical reservoir on the NARMA task (Atiya and Parlos, 2000), which is a widely used and effective benchmark for RC when evaluating the reservoir's ability to duplicate a higher-order dynamical model constructed from current and previous inputs. Generally, the NARMA2 and NARMA10 tasks require different optimal parameter settings (Appeltant et al., 2011; Yamanaka et al., 2018). NARMA10, as a higherorder dynamical system, is a fairly difficult target to train for. Hence, we use the NARMA2 task as our benchmark. The parameters in this paper are set by observing the performance for the NARMA2 task. NARMA2 is a second-order nonlinear model defined by Equation 1.6.

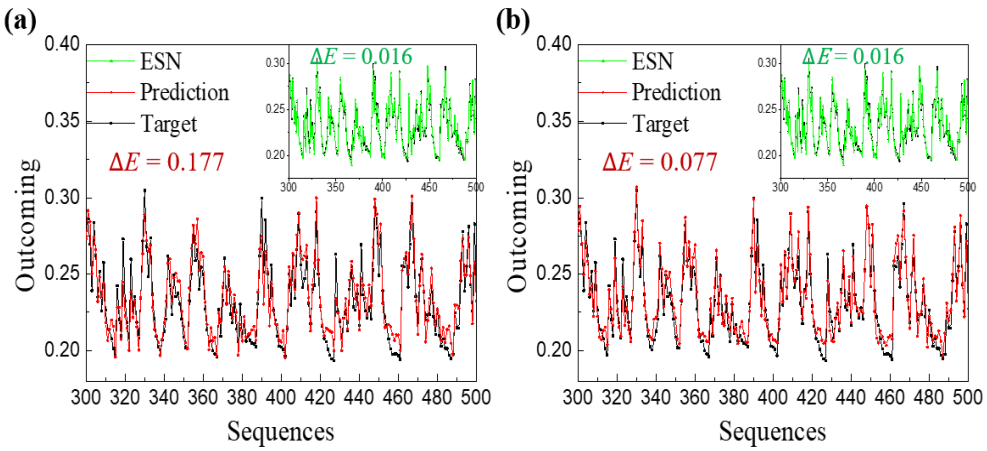

Figure 2.7: Performance in NARMA2 task. Results for (a) parallel-node and (b) parallel-group structures. Red line is the trained prediction signal, and black line is the supervisor signal (NARMA2 model). Green lines in the inset are testing results from ESN code. ΔE is used to reflect the extent of deviation between predicted and target values.

The input signal, $u(t)$, is a sequence of random digits in the interval [0, 0.5] generated by a random function, and the model output, $y(t)$, serves as the target. We train the system's output signal by selecting the output weights to render the signal as close as possible to the target one. Output weights, W_{out}, are determined from training data using ridge regression. The predicted signal is calculated as $\hat{y}(t) = X_{test}W_{out}$. Details of the training and testing process are given in the Appendix. We introduce the normalized-mean-square error (NMSE) to quantify the deviation between the target and predicted signals, expressed as

Equation 1.7.

A total of 300 nodes are used for all schemes to perform this task. Figure 2.7 shows the testing results for the parallel-node and -group structure reservoirs. Black lines reflect the NARMA2 model serving as the target signal in this prediction task. We train the output weights during the training step and use them to calculate the prediction signal from the testing data (shown in red). Green lines in the inset are the testing results for the standard ESN (spectral radius: 0.99; input weight interval: [-1,1]; activation function: tanh function); see the settings in the Appendix. Performance is reflected by the reproducibility of the prediction to the target. To assess the performance, the ΔE, defined by Equation 1.7, is introduced to evaluate the deviation between the target and calculated values. The parallel-group structure [Fig. 2.7(b)] has a smaller ΔE value, and thus, outperforms the parallel-node structure in this prediction task. Not surprisingly, the standard ESN obtains an obviously smaller ΔE. However, if we were to replace the activation function in this programmed ESN with the $I-V$ response used in our physical schemes, the ΔE would increase to 0.121.

2.3.3. SHORT-TERM MEMORY CAPACITY

A well-known feature of RC is its fading (short-term) memory, meaning that the current reservoir state contains information from recent past inputs, but it is unrelated to older ones. The short-term memory capacity (STMC) is introduced to reflect the correlation between the current reservoir states (at time t) and past inputs [at time $(t-k)$], ranging from zero to one. The STMC in RC networks is defined to be represented by the squared correlation coefficient of the target testing signal, y_k, and the fitted signal, $\hat{y}_k$, as proposed by Jeager Atiya and Parlos, 2000. Based on this definition, we obtain STMCs for higher-order target signals Grigoryeva et al., 2016, where the nonlinear target function is a Legendre polynomial of a time-delayed input. Specifically, target signal y_k^q is constructed from the kth delayed original input signal $[u(t-k)]$, according to the qth order of Legendre polynomials (see the Appendix). The fitted signal, $\hat{y}_k^q$, is trained using ridge regression by reservoir-state X, and the procedure is the same as that used in the NARMA2 task. Under the same nonlinear order, q, $r(y,\hat{y})_k$ is used to calculate the degree of correlation between the optimally trained output signal, $\hat{y}_k$, and the target signal, y_k. MC^q, summed over all MC_k with the same q, signifies how much of the delayed input signal can be recovered from $\hat{y}_k$. The input signal, $u(t)$, here, is the same random sequence as that used in the NARMA2 task, but mapped from its original range [0, 0.5] to [-1, 1], which is the standard input range for a memory capacity task. Under the same order of nonlinearity, a STMC ranging between zero and one represents the correlation coefficient of the trained output and target signals. This is the k-delay STMC and is given in Equation 1.4. STMC of the same order of the nonlinear target

is the sum of all delayed STMCs is expressed as:

$$r(y,\hat{y})^q = \sum_{k=1}^{\infty} r(y,\hat{y})_k^q. \tag{2.7}$$

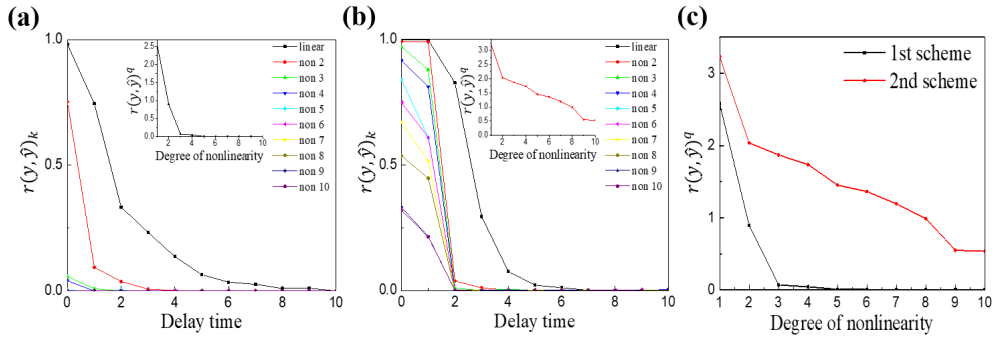

Figure 2.8: Memory-capacity performance of hardware schemes. Results for (a) parallel-node and (b) parallel-group structures. $r(y,\hat{y})_k$ represents the fraction of variance explainable in one signal by another. $r(y,\hat{y})_k^q$ sums all delayed $r(y,\hat{y})_k$ (k from 0 to 10). (c) $r(y,\hat{y})^q$ results for two schemes. Degree q (1 to 10) represents the target signal given by qth-order Legendre polynomials.

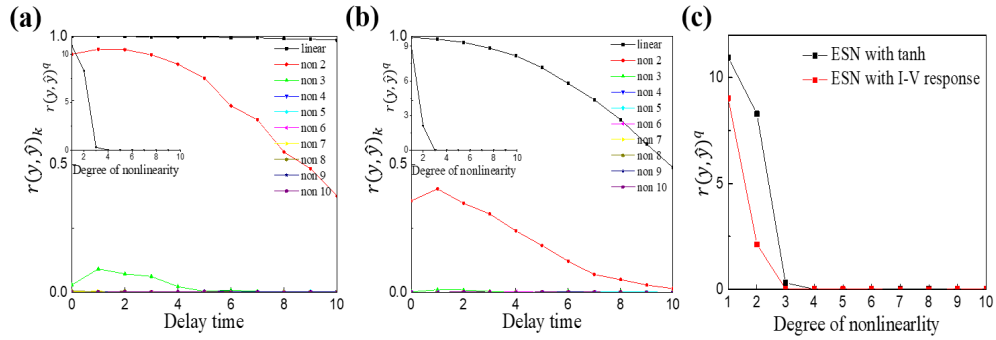

Figure 2.9: MC performance of programmed ESN (a) with hyperbolic tangent function and (b) with $I-V$ response as the activation function. (c) MC to different orders of target signal (i.e., $r(y,\hat{y})^q$) for these two ESNs.

Figures 2.8(a) and 2.8(b) show the STMCs for the parallel-node and -group structures, respectively, both with 300 nodes. Although their total memory capacities are not high, the parallel-group structure suppresses memory decline, especially against forgetting the two prior memory spans. The curves in Fig. 2.8(c) are the $r(y,\hat{y})^q$ results of the two structures. The MC $r(y,\hat{y})$ of the second scheme is improved, and the higher-order target is better remembered. For comparison, we measure the memory capacity of our programmed ESNs with 300 nodes. The activation of one is a hyperbolic tangent function [Fig. 2.9(a)], and the

other one uses the $I-V$ response we use in our physical schemes [Fig. 2.9(b)]. Both ESNs demonstrate significant abilities to remember inputs after a longer time, compared with our hardware systems. However, they perform poorly in the memory-capacity-to-higher-order target.

2.3.4. TESTING THE ACCURACY OF SPEECH RECOGNITION

AN isolated-word recognition task is performed to further analyze their performances. Five hundred audio files comprising ten English spoken words ("zero" to "nine") are downloaded from the internet (Cramer et al., 2022). These audio files are recorded from five people (three men and two women), and each word is repeated ten times by the same speaker. Preprocessing, feature extraction, and training procedures are performed using MATLAB 2019a. See the Appendix for a complete description.

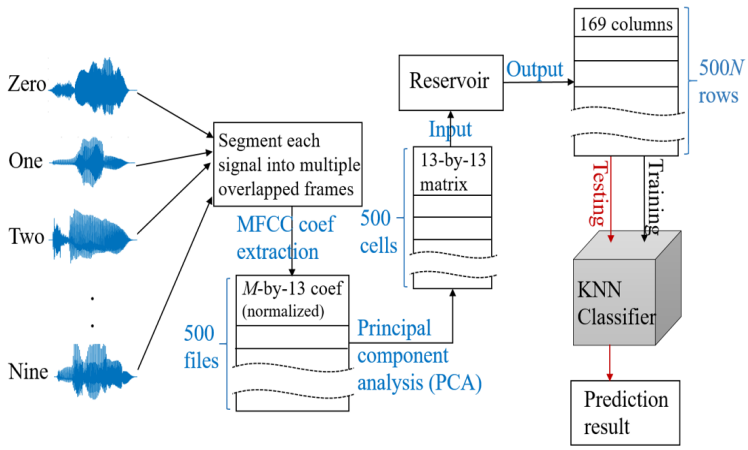

Figure 2.10: Schematic of the isolated-word recognition process. $M\times13$ features are extracted separately for each of the 500 audio signals, where M is the number of segmented frames. After principal component analysis, all coefficient matrixes with different sizes are organized into the same size of 13×13 (i.e., 169 coefficients) for one speech sample. Reservoir of N nodes would give a reservoir state of $500\times169\times N$ data items.We rearrange the reservoir state into 500 cells (each given the same label) and use them for training and testing sets. Detailed description of this classification process is given in Fig. 2.11.

Figure 2.10 shows a general flowchart of this task. Five hundred feature vectors are extracted from 500 audio files input into the reservoir. Five hundred output cells are collected from the reservoir for training and testing, where each cell consists of N vectors processed by N nodes (N =50 for this task). A k-nearest-neighbor (KNN) classifier combined with 20-fold stratified cross validation realizes the training and testing of samples. The KNN classifier is trained by specifying the distance metric as Euclidean, the distance weighting function as squared inverse, and the number of neighbors as k. The best performance is obtained when k = 58 for the parallel-node structure, 36 for the parallel-group structure,

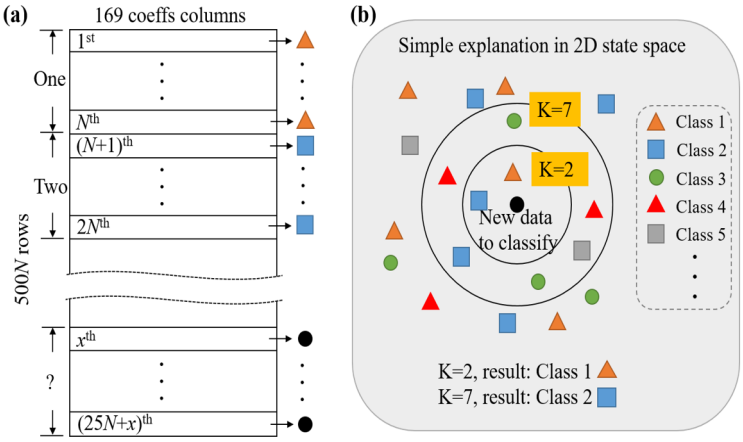

Figure 2.11: Classification process using KNN model. (a) 500 output cells with N rows and 169 columns. After identifying training subsets, each row with 169 variables is projected to a point in 169-dimensional space. Rows or observations with the same label are projected to points of the same color and shape. (b) Simplified schematic of the KNN model. This illustration represents two-dimensional (2D) space for easier understanding, but the actual space is 169-dimensional, and the coordinates come from 169 variables.

and nine for the ESN, as applied to the *fitcknn* function in MATLAB. All samples are divided into 20 disjointed subsamples (folds) chosen randomly, but with the same size. The training and testing process is repeated 20 times, each with a different assignment of subsamples as the testing set, and the remaining subsamples use the training set. Training data are mapped into 169-dimensional space based on their 169 predictor variables represented by the colored points in Fig. 2.11(b), and testing data, represented by black points, are used for classification. According to the calculated distances between each testing point and all other training points, the class with the largest weighted value is specified as the predicted class. In this case, each speech cell has 50 predicted labels; the most frequent label is the final class for this speech cell. The final accuracy rate is calculated as the average of these 20 cross validations.

The results for three RC systems are listed in Table I. The accuracies for the parallel-node and -group structures (both having 50 nodes) are 81% and 83.6%, respectively, and those for the standard ESN programming are 66.8% with 50 nodes and 72.6% with 200 nodes. When we change the spectral radius (SP, an eigenvalue of the network weight matrix with the largest absolute value) in the defined ESN, the accuracy under 50 nodes reaches 81.6%. To evaluate the contribution of the reservoir to performance, we test it in accordance with the procedure in Fig. 2.9, but without the reservoir. The testing accuracy is 74% (68%–80%), which represents the contribution from the preprocessing and postprocessing of data.

Table 2.1: Cross-validation results of three different reservoir structures.

System	ESN($\rho(W) = 0.99$)	ESN($\rho(W) = 0.19$)	Parallel-node RC	Parallel-group RC
Accuracy	66.8%(52%-80%)	81.6%(68%-92%)	81%(64%-92%)	83.6%(76%-92%)

2.4. DISCUSSION AND CONCLUSION

THE motivation for this work is to find a suitable method for using the inherent nonlinear responses of materials to construct suitable dynamical systems for physical RCs. Initial design idea for the RC structure is to utilize only input gain and feedback gain as the system parameter. Unlike the standard structure of randomly connected nodes, each node only has its own feedback gain, which is reflected as a diagonal matrix in the weight matrix defining the reservoir. Hence the largest absolute value of feedback gain is the spectral radius. We demonstrate the use of a simple nonlinear $I-V$ response and a simple external control to increase its internal dynamics. This opens the door to using molecular devices as processors.

Conventional RCs have several important properties, such as varied dynamical responses, fading memory, and nonlinear transformations, which are reflected by parameters such as input scaling, spectral radius, and sparsity. In our scheme, we reduce the RC to only two parameters: feedback gain a and input gain e. These two parameters can be used to adjust nonlinearity, control the dynamics of nodes, and generate virtual nodes. Their values have a significant effect on the final performance. Owing to the limitations of physical implementation, we find the optimal range for these values based on a limited number of trials. However, this does not affect our understanding of the relationship between dynamic properties and final performance. We hope to develop a practical method for finding the optimal range of parameters in future studies.

Effective reservoirs are generally considered to be necessary for achieving richly varied signals, which are ensured by a sparse interconnectivity of all nodes. However, in this work, we show that a scheme that uses independent nodes in the reservoir can work well for the isolated-word recognition task. An accuracy of 83.6% is achieved with only 50 nodes, indicating the potential of using basic nonlinearity. Moreover, we show that this kind of nonlinear response is more suitable for a structure with regular connections between nodes. The parallel-group RC outperforms the standard ESN with an activation function of the $I-V$ response in a NARMA task, and it has a larger memory capacity for higher-order target signals. The standard ESN has a high STMC and, therefore, has a low ΔE in the NARMA2 task. However, its improvement to STMC for a higher-order nonlinear target ($q \geq 3$) is quite limited. The performance of these schemes in the speech-recognition task is inconsistent with that

of previous tasks. Although the pre- and post-processing of speech information plays a positive role in this task, our physical schemes need only a small number of nodes to improve the accuracy rate. Our physical reservoir does not require an increased number of nodes to show higher accuracy in the recognition task, whereas the standard ESN does. Nevertheless, the highest accuracy for this ESN (obtained with 200 nodes) is still lower than that of our physical reservoir with 50 nodes. Notably, the standard ESN that performs well in the NARMA task performs poorly in our isolated-word recognition task. When the ESN reaches an accuracy of 81.6% (under a spectral radius of 0.19), the performance in the NARMA task becomes worse (ΔE =0.0778). This demonstrates that, for tasks such as the isolated-word recognition task, the extraction of current information is important, and information from the previous state may undermine the characteristics of the current information. Therefore, although increasing the number of nodes in the ESN facilitates the extraction of information features, it can improve only the recognition accuracy to a level similar to the contribution of pre- and post-processing. These results can be used to develop heuristics for building specific physical reservoirs and selecting the training and testing method for a specific task.

We use a single electronic circuit to generate numerous nodes as the reservoir, where the signals are serially fed into it. Processing under the sequential feeding of input is slow. However, in our schemes, nodes or node groups have an independent parallel architecture, which facilitates the fabrication of parallel hardware of physical nodes to compensate for the scaling of the total computation time with the number of virtual nodes. For the parallel-node structure, because the response is instantaneous and determined only by the most recent output and the current input, there is no need to access memory to extract the collected output states. In the parallel-group scheme, the memory is accessed only once for each current input calculation to read the state of the previous node.

Although real molecular devices will have nonidealities, such as noise and variability of characteristics, we think that the inherent properties of molecules can correspond to the randomness in our parameter selection, which would make the reservoir states more separable. A similar point of view is also proposed for a physical RC comprising a memristor array (Appeltant et al., 2011). Noise that occurs during signal processing may degrade its performance by violating the rules of processing, but it may still be possible to utilize. For example, we have previously investigated a nanomaterial device with a noise-assisted nonlinear signal response, which is a promising approach for decreasing energy consumption through informational signal processing and transference (H. Fujii et al., 2017). However, based on current research, we believe that materials with stable and adjustable $I-V$ properties, such as a nanoparticle bridge with temperature-dependent properties (Nishijima et al., 2018), are a more appropriate choice. The results of this work can be used for the development of single-molecule devices for informational processing and the application of nanomaterials to physical RCs.

2.5. ACKNOWLEDGEMENTS

This work is partially based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). K.N. is supported by JSPS KAKENHI Grant No. JP18H05472.

2.A. APPENDIX TO CHAPTER 2

2.A.1. DISCUSSION OF SETUP DETAILS IN NARMA2

Initially, in order to directly use both positive and negative voltage ranges, the input signal $u(t)$ is set to a range of [-0.5,0.5]. However, the original definition and discussion of all NARMA models were based on the generation of inputs from a uniform density in [0,0.5]. Therefore, the input signal range of the physical system is no longer consistent with that of the NARMA model if we normalize the input signal into the range of [0,0.5], which interferes with the test error. If the input signal is normalized to [0,0.1], test error could be further decreased because of the reduced standard deviation of input signal. But this makes it difficult to compare the results with those of other similar articles at the same level. Therefore, we standardize the range of input signal range as [0,0.5] and exploit the negative voltage range by a negative input gain. Moreover, the definition of normalized mean square error (NMSE) also affects the comparison of results with similar articles. In addition to the expression introduced in Section 1.2, which is based on the standard deviation of target signal, there is another definition expressed as below:

$$\Delta E = \frac{M\sum_t [\hat{y}(t) - y(t)]^2}{\sum_t \hat{y}(t) \sum_t y(t)}. \tag{2.A.1}$$

The above expression is based on the mean values of both target signal and test signal. The disadvantage of the definition is that it relies heavily on the mean value, thus a small range of input signal would lead to an incredibly slight value of the test error, while the actual performance is mediocre. The definition based on standard deviation is almost stable between 0 and 1, which is easier to analyze and compare the results. Therefore, we also standardize the definition of NMSE to be Equation 1.7.

2.A.2. TRAINING PROCEDURE FOR NARMA2 TASK

This section describes the training process for the NARMA2 task. The input signal, u(t), is a sequence of T = 4000 digits randomly generated in the interval [0, 0.5]. By substituting the input sequence into the NARMA2 model [Equation 1.6], we can obtain the target signal, $y(t)$. If the input signal is fed into the reservoir with N = 300 nodes, they will respond to the

input signal, producing 300 vectors of node states. Each vector contains 4000 data items, which include the output voltages of each physical node recorded at 4000 time steps. This 4000 × 300 matrix is the reservoir state, X_{out}. A total of 100 data items in each vector are used to initialize the reservoir, 2000 data items are used for training, and the remaining 1900 are used for testing.

With a 300 × 1 output weight matrix, the output signal can be calculated using matrix multiplication. During the training process, the optimal output weights (i.e., those that render the output signal as close as possible to the target signal) are determined. We use ridge regression, which is essentially an improved least-squares estimation method, to train the output signal. The aim of least-squares estimation is to minimize the objective function, $\sum[y_{train} - (\beta_0 + \sum_{i=1}^{N} \beta_i X_{train\,i})]^2$, so the coefficients can be obtained by choosing $\beta = (X_{train}^T X_{train})^{-1} X_{train}^T y_{train}$. Ridge regression addresses the problem of multicollinearity by estimating regression coefficients as:

$$\beta = (X_{train}^T X_{train} + kI)^{-1} X_{train}^T y_{train} \tag{2.A.2}$$

where k is the ridge parameter and I is the identity matrix. k is 22, 5, 0.6, and 1, respectively, for the parallel-node structure, parallel-group structure, standard ESN with hyperbolic tangent function, and ESN with $I - V$ response. The output weight matrix, W_{out}, is $\beta_{1:N} \times 1$. To test the performance of this system, we apply the optimal output weights determined during the training stage to the testing data and obtain the predicted signal, $\hat{y}(t) = \beta_0 + X_{test} W_{out}$.

2.A.3. MEMORY CAPACITY TASK

The definition of short-term memory is based on coefficient of determination (i.e., squared correlation coefficient, also called r-squared), normally used to analyze how differences in one variable can be explained by a difference in the other variable. In short-term memory, it examines the coefficient of determination between the current observed outcome $\hat{t}(t)$ and input of k-units ahead (i.e. $u(t-k)$), thus the result assesses how strong the relationship is between these two variables. The process of memory capacity measurement is similar to the NARMA2 task, except that the target signal is obtained by Legendre polynomials instead of the NARMA2 model. The expressions of the Legendre polynomials of orders 1-10 are listed in Table 2.A.1.

For a given delay, k, the input variable, x, in the Legendre polynomials is replaced by $u(t-k)$. The reservoir still receives $u(t)$ as its input signal. This means that the reservoir state, X, is the response under input sequence $u(t)$. The aim is to fit the target signal given by the sequence $u(t-k)$. The training and testing process is exactly the same as that for the NARMA2 task, but the ridge parameter in ridge regression training is taken as one for all

Table 2.A.1: Legendre polynomials.

q	$P_q(x)$
1	x
2	$\frac{1}{2}(3x^2-1)$
3	$\frac{1}{2}(5x^3-3x)$
4	$\frac{1}{8}(35x^4-30x^2+3)$
5	$\frac{1}{8}(36x^5-70x^3+15x)$
6	$\frac{1}{16}(231x^6-315x^4+105x^2-5)$
7	$\frac{1}{16}(429x^7-693x^5+315x^3-35x)$
8	$\frac{1}{128}(6435x^8-12012x^6+6930x^4-1260x^2+35)$
9	$\frac{1}{128}(12155x^9-25740x^7+18018x^5-4620x^3+315x)$
10	$\frac{1}{256}(46189x^10-109395x^8+90090x^6-30030x^4+3465x^2-63)$

schemes in this task.

2.A.4. DETAILS OF SPEECH-RECOGNITION TASK

This task uses the extracted features of Mel-frequency cepstral coefficients (MFCCs) and linear classification using KNN, as described below.

We download our dataset from a website (Cramer et al., 2022) that hosts a large number of audio files with words (i.e., digits “zero” to “nine”) spoken by various speakers. We randomly pick 500 audio files (three male speakers and two female speakers) as our dataset. Each spoken digit is recorded 10 times in the dataset (i.e., 100 samples for each speaker and 500 samples in total).

We first use the *audioread* function in MATLAB to read data from each audio file and return the sampled data with a obtained sampling rate of 8 kHz. Then, we use the MFCC function to extract the log energy and MFCCs of each signal. In order to more intuitively show the role of each step in the feature extraction process of original speech signal, the following plot is drawn after each step using one speech "zero" as an example, given in Figure 2.A.1. The extraction process is performed frame by frame, each of 30 ms with 1440 sample times. Next frame share an overlap of 75% with the previous one, as shown in Figure 2.A.1 (a) and (e). Log energy and MFCCs are calculated in each frame, the process is exhibited in Figure 2.A.1 (d)-(h) and the mathematics involved are also listed below. The first coefficient in the coefficient vector is replaced with the Log energy value, so that each coefficient vector

includes 13 coefficients.

$$S_1(k) = \sum_{n=1}^{N} s_1(n)h(n)e^{\frac{-j2\pi kn}{N}}$$
$$P_1(k) = \frac{1}{N}\|S_1(k)\|^2 \tag{2.A.3}$$

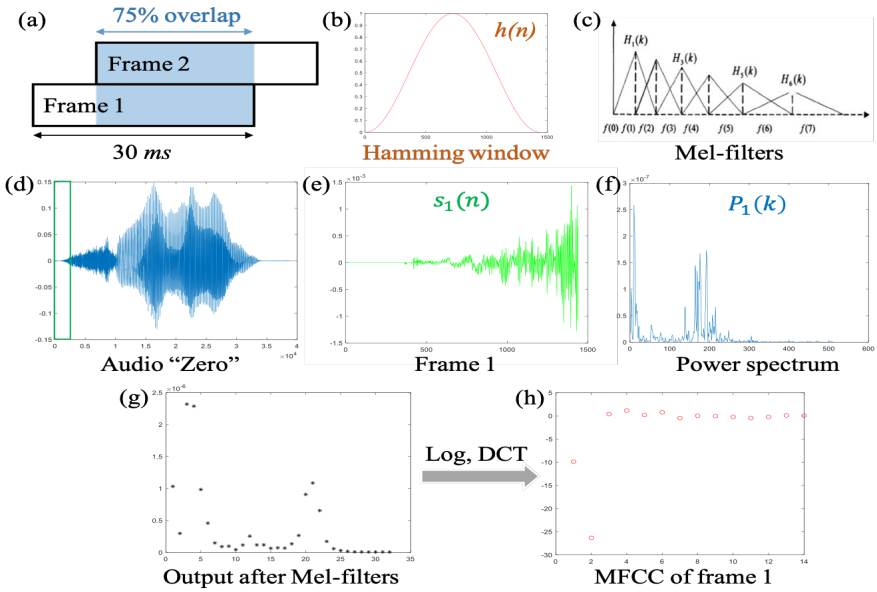

Figure 2.A.1: The extraction process of audio "Zero" from original speech signal to MFCCs. (a) Window length and overlap length of frames. (b) Hamming window used in Fast Fourier Transform (FFT) algorithm. (c) Mel filter bank. (d) Audio "Zero" in time-domain. (e) Audio signal extracted in frame 1. (f) Power spectrum estimation obtained by FFT. (g) Results of Mel filter bank acting on power spectrum. (h) Coefficients calculated by logarithm (Log) and discrete cosine transformation (DCT) on results in (g).

This method is widely used to extract parameters in speech-recognition tasks, including delta and delta-delta coefficients. We perform three trials using our reservoir system under the same settings. One is done using MFCCs and log energy; one is done using MFCCs, delta, and log energy; and one is done using MFCCs, delta, delta-delta, and Log energy. The results of these trials show that the accuracy decreases with an increasing number of parameters. We believe that isolated words are not complicated enough to extract many parameters. Hence, we extract only MFCCs and Log energy as the features in this task. We notice that Log energy is not on the same scale as MFCC, which may bias the classifier. Thus, we normalize the features by subtracting the mean and dividing the standard deviation of each column.

After MFCC extraction, we have M coefficients for each audio file. Because the length

of each sampled speech is different, M is not fixed. To reduce dimensionality and normalize the size of coefficients of each speech signal, principal component analysis is used to organize the coefficients extracted from each audio file into a matrix of the same size (i.e., 13 × 13). Before inputting these data into the reservoir, we rearrange each 13 × 13 matrix into a row vector of 169 elements, labeled with its spoken word (digits "zero" to "nine"). The input data are hence a 500 × 169 matrix, which is fed into the reservoir row by row after the row minimum and maximum values are mapped to [-0.5, 0.5]. This normalization is necessary, because, in this task, all parameters take positives values. Recall that, when we choose parameters based on the NARMA2 task, the range of the input signal for the NARMA task is [0, 0.5], and the input gain, e, is randomly assigned a plus or minus sign within the selected range. Next, the output from the reservoir with 50 nodes is rearranged into 500 50 × 169 submatrices, where the rows represent the observations, and the columns represent the predicted variables. After the features and their labels are collected, they are used to train the classifiers.

Ten classes (labels), representing ten speech targets (digits "zero" to "nine") are trained using the KNN algorithm (*fitcknn* function in MATLAB). The hyperparameters for the KNN classifier include the number of nearest neighbors, the distancemetric used to compute the distance to the neighbors, and the weight of the distance metric. In this task, the number of neighbors is set to k (k =58 for the parallel-node structure, 36 for the parallel-group structure, and 11 for the ESN, optimized for the best performance), and the metric for the selected distance is the squared-inverse-weighted Euclidean distance. An n-fold (n=20) stratified cross validation is used in the KNN classifier to reduce the error caused by the selection of training data and testing data.

To decrease the sensitivity of model performance to data partitioning, 500 output submatrices are divided randomly into 20 disjointed subsamples, each consisting of 25 submatrices. The training and testing process is repeated 20 times, with a different set of subsamples as the testing set each time. The remaining subsamples are used as the training set. This training of 20 classifiers under different combinations of training subsets continues until each subset is predicted. All 25 000 rows of observations are mapped onto 169-dimensional space based on their 169 predictor variables as the relative coordinate value. We calculate the squared-inverse-weighted Euclidean distances between each testing point and all other training points and rank them in ascending order by the KNN model. Next, for the KNN, the class with the largest weighted value is specified as the predicted label. The most frequent label in one submatrix with 50 labels is used as the final class for this speech cell. The final accuracy rate is the average of prediction results of these 20 trained classifiers. The accuracies of the parallel-node and -group structures are 81% (64%–92%) and 83.6% (76%-92%), respectively, and that of the ESN, which performs much better in the NARMA2 task, is only 66.8% (52%-80%) for this task.

In addition to the training method of KNN based on linear regression for RC system output, we also tried to use a training method via logistic (LR). Using function $fmincg$ in MATLAB to minimize cost function by gradient descent method then apply the trained parameters to test data. Table 2.A.2 lists the test accuracy of LR training and the corresponding accuracy by KNN, where ESNs possess with spectral radius of 0.99.

Table 2.A.2: Test accuracy of four reservoirs.

Type	Parallel node	Parallel group	ESN(tanh)	ESN(diodes $I-V$)
LR	74%	80%	55%	53%
KNN	81%(64%-92%)	83.6%(76%-92%)	66.8%(52%-80%)	51%(32%-80%)

2.A.5. PARAMETER SETTINGS IN ESN SYSTEMS

Most parameters for our ESN are fixed (activation function, *tanh* function; spectral radius, 0.99; input weight interval, [-1,1]; node connection weight interval, [-0.99, 0.99]). Input scaling is introduced in the speech recognition task to normalize the input signal within the interval [-0.5, 0.5], which is consistent with the input scale in our physical RC systems. Input scaling is not used in the NARMA2 task because the input signal, $u(t)$, is generated within the range of [0, 0.5], and the same signal, $u(t)$, is used for all physical systems and the ESN system.

3

Reservoir Computing through Electrochemical Reaction

This chapter is published as:
S. Kan, K. Nakajima, T. Asai, M. Akai-Kasaya,
"Physical Implementation of Reservoir Computing through Electrochemical Reaction",
Advanced Science, **2022**, 9, 2104076.

ABSTRACT Nonlinear dynamical systems serving reservoir computing enrich the physical implementation of computing systems. A method for building physical reservoirs from electrochemical reactions is provided, and the potential of chemical dynamics as computing resources is shown. The essence of signal processing in such systems includes various degrees of ionic currents which pass through the solution as well as the electrochemical current detected based on a multiway data acquisition system to achieve switchable and parallel testing. The results show that they have respective advantages in periodic signals and temporal dynamic signals. Polyoxometalate molecule in the solution increases the diversity of the response current and thus improves their abilities to predict periodic signals. Conversely, distilled water exhibits great computing power in solving a second-order nonlinear problem. It is expected that these results will lead to further exploration of ionic conductance as a nonlinear dynamical system and provide more support for novel devices as computing resources.

3.1. INTRODUCTION

IT is becoming increasingly difficult to satisfy the requirements of high efficiency for computing power and storage capacity while developing hardware systems in artificial neural networks. Reservoir computing (RC) originates from recurrent neural networks (RNNs) with input, hidden, and output layers, and has attracted considerable attention owing to its unique advantages (Bodén, 2001). First, the output layer of RC is the only part that needs to be tuned toward the target signal, making the learning process faster and simpler while the computing power is no less than that of conventional RNNs (Vlachas et al., 2020b). Another attractive aspect of RC is that the hidden layer is replaced by a massive number of nonlinear nodes coupled with one another (referred to as reservoir), as shown in Figure 3.1. Instead of a conventional recurrent structure, various physical systems that utilize their nonlinear dynamics to work as a computational resource have been discussed extensively (Jaeger and Haas, 2004; Uchida et al., 2004b; Dambre et al., 2012; K. Fujii and Nakajima, 2017; Goto et al., 2021; Furuta et al., 2018; Sillin et al., 2013; K. Nakajima and Fischer, 2021). In fact, any dynamical system has the potential to be a reservoir for information processing (Dambre et al., 2012). This basic concept introduces the concepts of arbitrariness and diversity in the choice of a reservoir and leads to the thriving development of physical reservoir computing (Kawai et al., 2019b; Tsunegi et al., 2019; Appeltant et al., 2011; Du et al., 2017; Tsunegi et al., 2018; K. Nakajima et al., 2015; K. Nakajima, 2020). Hardware implementation of RC utilizes a variety of physical systems, devices and materials, and introduces more possibilities for hardware breakthroughs. Different feats displayed by different systems or designed schemes have been discussed extensively during recent years, among which organic electrochemical devices began to attract an attention. There is a study using semiconductive dendritic organic electrochemical networks for bio-signal classification, proved that the nonlinearities created by electrolyte when electrically excited could be an advantage in RC (Cucchi et al., 2021). However, rich dynamic properties of electrochemical reactions have not been widely studied, and their greater application potential deserves to be further explored.

3.2. DESIGN AND REALIZATION SCHEME

NEUROMORPHIC devices based on molecular dynamics or chemical reactions have great advantages in reducing power consumption and area reduction. The possibility of constructing complex neural networks based on molecular devices has been discussed in our previous research (H. Tanaka et al., 2018). According to this conjecture, we discussed a reservoir construction method that can take advantage of the current-to-voltage property presented by molecules in the following research, further demonstrating the ability of

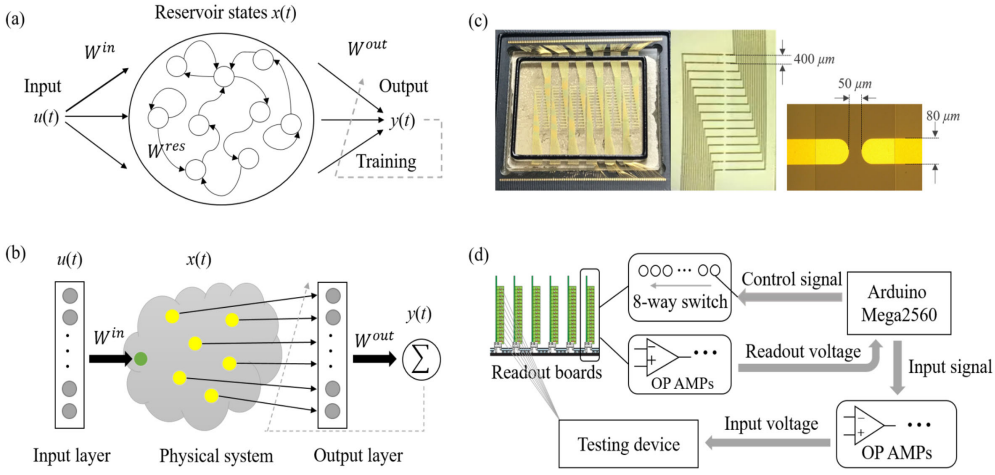

Figure 3.1: Illustration of physical reservoir computing (RC) and construction of molecular-based reservoir. a) Structure of traditional reservoir computing. b) Concept of our physical RC system. c) Testing device of planar electrodes. d) Measurement process chart of proposed solution reservoir.

chemical dynamics in the field of computing (H. Tanaka et al., 2018). Based on the prior studies, in this paper we build a physical RC system to utilize diverse electrochemical reactions of actual materials and experimentally analyze their computing ability. This section gives the concept of physical RC and the realization scheme of our system.

3.2.1. CONCEPT OF PHYSICAL RC

FROM Figure 3.1(a), we can observe that the structure of RC includes input layer, reservoir and output layer. A novel scheme exploring physical dynamics in the reservoir, as a computational resource, was called physical RC, in which the reservoir part is replaced by a physical dynamical system (K. Nakajima, 2020). Such a physical dynamical system could act as nonlinear nodes in the traditional reservoir with varying degrees of data processing. Therefore, the physical dynamical system should have sufficiently rich dynamics with high effective dimension to enable diverse responses of the system to the same input. However, to make the reservoir works successfully, there is one prerequisite, that is, the reproducibility of the input–output relation (Inubushi et al., 2021), which is also termed an echo state property (Jaeger, 2001a; Yildiz et al., 2012). When the system becomes chaotic and exhibits considerable dynamics, it loses the echo state in general and its relevance to the input data, making it an unreliable computing resource. When this condition is satisfied, the output signals behave like a series of signals synchronized with the input series.

The working principle of RC can be described as follows. The output signal $\hat{y}$ should be trained to be as close as possible to the target signal y by adjusting the output weights

W_{out}, and the optimal values can be obtained by a linear regression algorithm, such as the least-squares method (i.e.,minimum $\|y_{train} - \hat{y}^2_{train}\|$). Therefore, at the temporal unit t, the output signal $\hat{y}(t)$ and reservoir state $x(t)$ can be expressed as follows

$$\hat{y}(k) = W^{out} \times x(t) \tag{3.1}$$

$$x(t) = f_{res}(W^{in} \times u(t) + W^{res} \times x(t-1) \tag{3.2}$$

where f_{res} is the nonlinear function of reservoir, W_{in} and W_{res} are the input weights and weights between nodes, respectively. $u(t)$ is the input value at the current temporal unit, while $x(t-1)$ represents the reservoir state at the previous temporal unit. To quantize the deviation between the target signal and the testing signal as well as to evaluate the performance of RC, mean square error (MSE) is generally used. With the testing data a length of M, the expression of MSE is as follows

$$\Delta E = \frac{\sum_t [\hat{y}(t) - y(t)]^2}{M}. \tag{3.3}$$

Based on this expression, the normalized mean square error (NMSE) is defined to measure the mean relative scatter and reflect the random errors, expressed as Equation 1.7.

3.2.2. PROPOSED RC SCHEME AND STRUCTURE PRINCIPLE

FIGURE 3.1(b) shows the concept of our proposed RC system, which is based on the planar electrodes shown in Figure 3.1(c). The process of obtaining the response data can be simply described as follows: Inject the input voltage signal to one electrode and read the response current signals from the other electrodes as the reservoir state. The solution sample is dropped on the surface of the metal electrodes. 3.1(c) shows the testing device, which was composed of 90 pairs of planar electrodes (6 groups × 15 pairs). One side is selected as the input electrodes (e.g., the right) to the input voltage and the other side as the readout electrodes (e.g., the left) to read the response current. One pair of electrodes was recognized as one node in the reservoir. To achieve our design scheme, we introduced a multiway data acquisition system attached to a number of readout electrodes in our testing system to facilitate the control of readout nodes, which also made parallel testing possible. Figure 3.1(d) presents a general flowchart: after the input voltage is imposed on the input electrodes, the readout boards (i.e., multiway data acquisition system) consisting of a current-to-voltage converter and differential amplifier circuits would transfer the response current to voltage within a range that can be detected. In this case, the response currents from many readout electrodes can be captured using fewer pins. The entire process was controlled by a microcontroller (ArduinoMega2560), and a detailed process description is

provided in the Supporting Information. A complex task generally requires a large number of nodes in RC, and the definition of nodes, such as defined by time-multiplexing and external parameters, greatly affects system performance (Appeltant et al., 2011; (Kan, Nakajima, et al., 2021)). Therefore, selection of appropriate node definition is also one of the most important explorations in physical RC system, and the work in this paper also covers this.

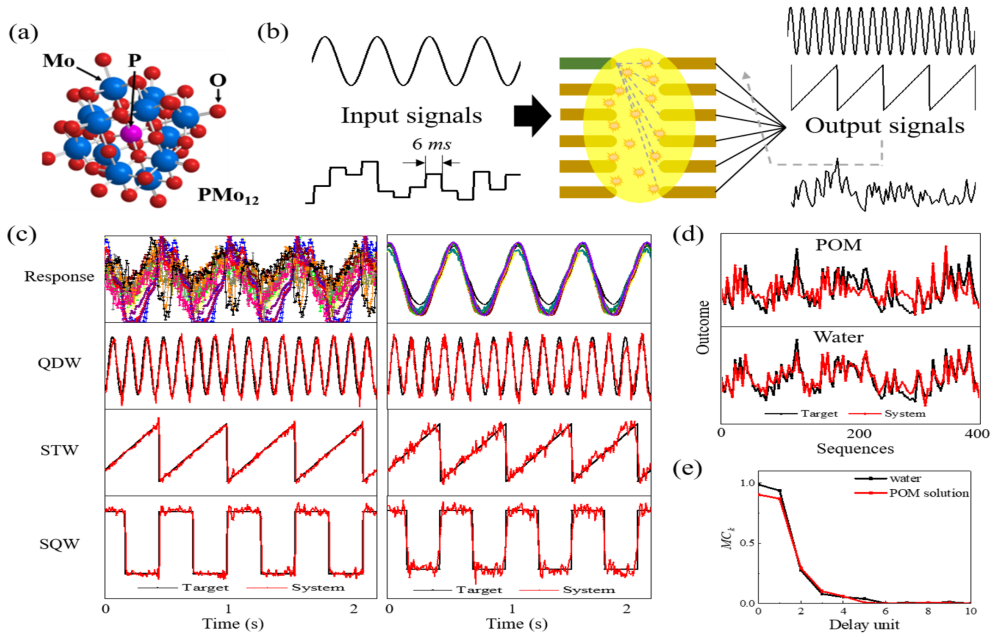

Figure 3.2: Schematic of electrochemical-reaction-based reservoir. a) A structure of the polyoxometalate (POM) molecule. b) Process diagram of two tasks performed by testing systems. c) Responses of POM solution (left) and deionized (DI) water (right) to sinusoidal signal and their predicted performance to target signals of quadruple sine (QDW), saw tooth (STW), and square waves (SQW). d) Predicted performance of POM solution and DI water to a nonlinear target. e) Short term memory for the linear target signal of DI water and POM solution.

3.3. WORKING PRINCIPLE AND VERIFICATION

POLYOXOMETALATE (POM) exhibits several advantages owing to the unique structure, as shown in Figure 3.2(a). Multiple oxidation and reduction characteristics enrich the dynamics of the RC system as well as promote the information processing capability (H. Tanaka et al., 2018; Wang and Yang, 2015). In this study, we utilize the electrochemical current in the solution, of which complex and large nonlinear response is expected even at low voltages owing to the high and multiple redox activities of POMs (Fernandes et al., 2015). A simple signal prediction test was performed first to verify the effectiveness of our system and POM molecule, and the general process is shown in Figure 3.2(b). The pro-

cessing ability of our system is embodied in the prediction results of periodic signals and of a second-order nonlinear auto-regressive moving average (NARMA2) model (Atiya and Parlos, 2000). The concentration of the molecule is not specified in this study because the specific concentration values do not result in discernable difference within a certain range. The solution concentrations between 1-3 mg/mL^{-1}, used for the performance tests, do not affect the results of the experiment; this conclusion is proved succinctly in the last section of Supporting Information 3.A.

For periodic signal prediction, the input $u(t)$ is a sinusoidal signal with a period of 1.8 *s*, while the target signals are sine wave signals with quadruple frequency of input signal, saw tooth wave signals and square wave signals of the frequency same to input signal, respectively. The response currents were normalized to the range of [0, 1] as the training and testing data, respectively. To evaluate the performance, we calculated the MSEs between the predicted and target values. The response of DI water (Figure 3.2(c), right) to a sinusoidal input signal corresponds to a signal with smooth sinusoidal shape and varying amplitudes, while the POM solution (Figure 3.2(c), left) exhibited more diverse and complicated responses, thus leading to smaller errors. Additionally, with 112 nodes, the MSEs of the POM solution for the three target signals are 0.023, 0.002, and 0.006, and those of DI water are 0.0512, 0.0391, and 0.0565, respectively. These results confirm what we expected, that is, POM can enhance the computing power of our physical system. However, in the NARMA2 task, where the input $u(t)$ is a random signal with a step width of 6 ms, the prediction performance of the POM solution is inferior to that of DI water. The NMSE was introduced to evaluate the NARMA2 performance, and the calculated results were 0.2106 and 0.3865 for DI water and POM solution, respectively (with 112 nodes). Figure 3.2(d) shows the NARMA2 performance of POM solution and DI water.

To analyze the reasons for the inconsistent performance, a short-term memory capacity (MC) test was performed Fette and Eggert, 2005. MC_k corresponds to the squared correlation coefficients between the current reservoir states (represented as temporal unit, t) and kth unit of past input (i.e., $(t-k)$, where $k \in [0, 10]$), which ranges from 0 to 1. It reflects the extent to which the current states can be explained from past inputs. From this calculation, we found two coefficients in showed significant differences in this task. The results of DI water and POM solution are plotted in Figure 3.2(e). The two coefficients MC_0 and MC_1, which play the most important roles in the NARMA2 task, are higher in DI water than in POM solution and yield a better performance in NARMA2 (see Equation 1.4 in Chapter 1.2) (Atiya and Parlos, 2000; Akashi et al., 2020). Low values of these two coefficients indicate a lack of partial information, which means that the dynamic characteristics caused by redox reaction of POM molecules are too strong and the consistency of information is kind of destroyed. Nevertheless, these results exceeded our expectations because the value of MC_1 was similar to MC_0. This was achieved by a "feedforward" parameter in our previous work

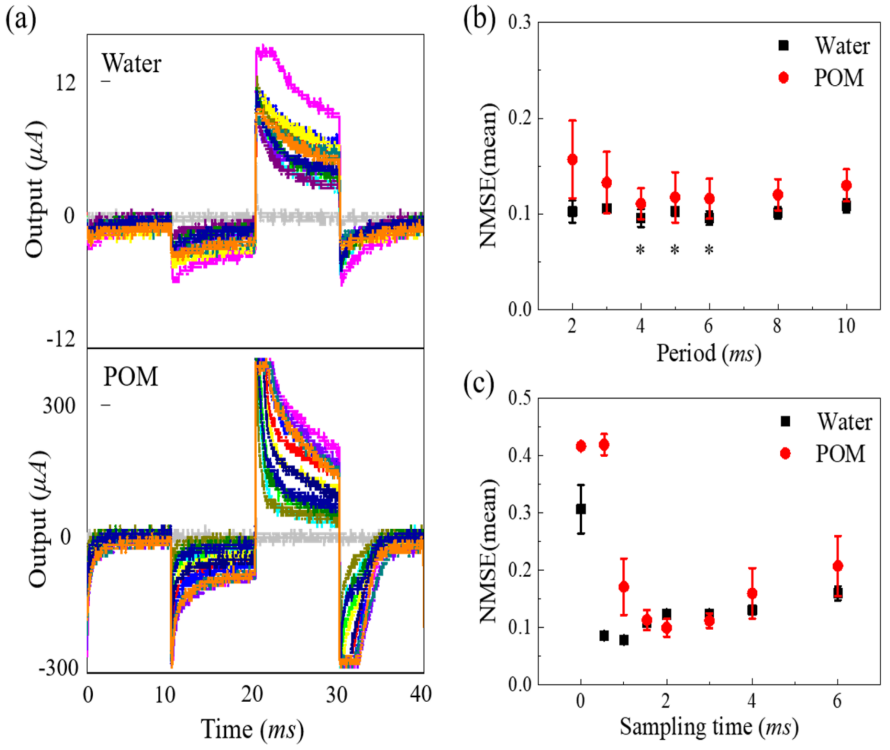

Figure 3.3: Investigation of computing ability. a) Response signals of DI water and POM solution to a series of input voltages. b) Prediction results of DI water and POM solution at different response periods but with the same sampling time. c) Prediction results of DI water and POM solution for different sampling times at a response period of 6 ms. Plots in b,c) show averaged values using three trials and the error bars show standard deviations.

based on the use of electronic devices which lacked response dynamics, and the values of the parameter was selected based on a series of analyses (Kan, Nakajima, et al., 2021). In this solution RC scheme, this feature existed naturally and was independent of the choice of the solution sample. This highlighted the potential of our system and demonstrated that our system deserves further development. To clarify the differences in these responses as well as further improve the system performance, we examined various measurement parameters.

Figure 3.3a shows the plotted response waveforms of 15 output electrodes. Data were derived directly from the oscilloscope, wherein we input a series of voltage values to test one specific electrode, and each input voltage was maintained for 10 ms. The current of DI water was significantly smaller than that of the POM solution because of the lack of electrolytes. The measured current through DI water does not exhibit a clear potential window due to the existence of non-Faradic currents; the electric current flowing in water should be zero within a potential window (i.e., less than $\approx 1\ V$) during an ideal electrochemical measure-

ment. An electric double layer induced at the solution interface with gold should contribute to a large capacitive current. Our measurement setup utilized the variation of condition in each pair of electrodes to yield signal response complexity that serves as a computational resource for RC, which was not technically absolute electrochemical information.

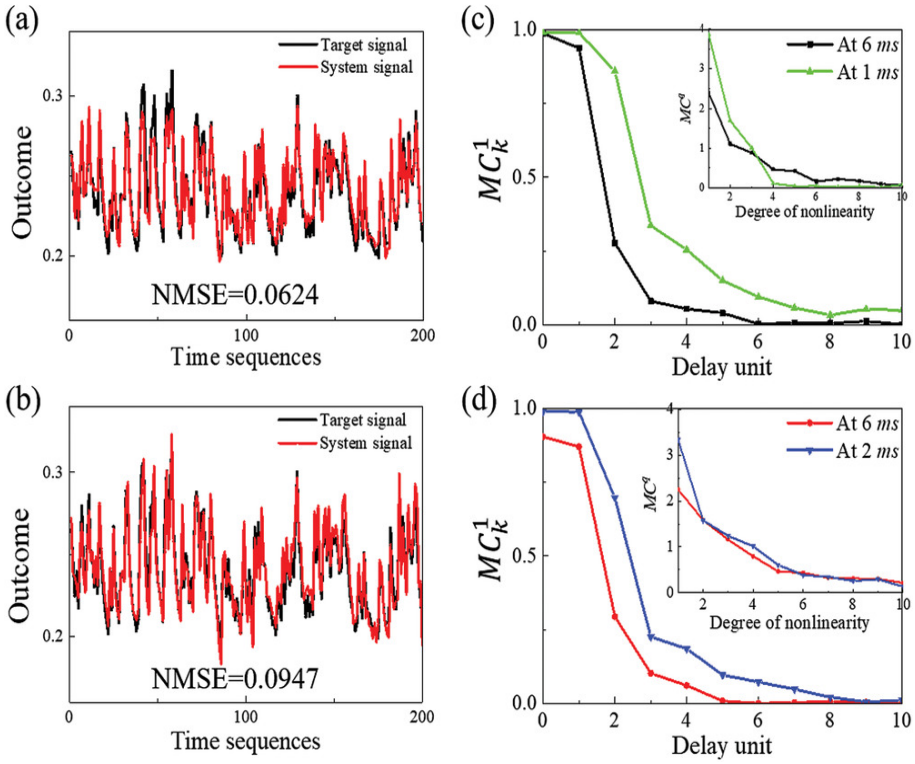

Figure 3.4: Comparison of DI water and POM solution outcomes. Prediction performance and prediction errors of a) DI water and b) POM solution. Short-term memory to first-order target and the memory capacity (inset) of c) DI water, d) POM solution.

The response waveform of the samples shows different transient characteristics. Specifically, the upper and lower voltage limits are due to the limitation of the $I-V$ amplifier. The input voltage range andmagnitude factor of the amplifier were adjusted to make the output from all nodes the most adequate value on average. Compared with the gradual trend during the entire response of DI water, the response of the POM solution was more intense in the initial response. The abrupt initial spike was caused by the double layer on the electrode surface and by an interaction of the capacitive resistance circuits within the parallel external circuit, as explained in the first section of Supporting Information. It is considered that the intense capacitive current led to a poor POM solution computing power at early sampling times.

From the observation of the response waveform, we speculated that the performance of this system may be affected by the length of the reaction time, so it was necessary to further control the parameters of the time. This is also the exploration of the more appropriate node definition that mentioned earlier. To clarify how the reaction time of the solution affects the computing power, we tested the performance of the NARMA2 task in terms of the test period and sampling time, and the statistical results are plotted in 3.3(b,c), respectively. In Figure 3.3(b), the sampling time was kept at 1ms for DI water and at 2ms for the POM solution. In 3.3(c), the response period was maintained at 6 ms in both samples. In addition, the results of the two samples in 3.3(b) were obtained from the same set of electrodes, and 3.3(c) was from another set of electrodes to ensure the consistency of the test conditions. Both sets of electrodes were brand new, and DI water was tested first before POM solution because the redox reaction of the POM solution would cause damage to the electrode surface thus affects the stability and reliability of the results. At the end of all tests, there was no serious damage to the electrode surface. However, after numerous measurement repetitions, we found that the response current of the solution started to increase significantly. To avoid the interference attributed to this factor, we changed the testing solution for each test from the same prepared solution sample. Moreover, in 3.3(b) the three tests for two samples (24 NMSEs in total) at sampling period of 3 to 6 ms are marked with an asterisk to indicate the data used for analyzing variance. The calculated P-value of 0.310199 indicates the response of the two samples to be no significant difference under their respective optimum settings in NARMA2. 3.3(b) indicates that the response period had no correlation with the computing power of DI water, and only affected that of the POM solution when the period was less than 4 ms. 3.3(c) reveals the dependence of computing power on sampling time for both samples: the best sampling time (the lowest error and most stable performance) for DI water was around 1 ms, and for POM solutions was $\approx 2\ ms$. Although the lower optimal sampling time for DI water than that for POM can be attributed to the rapid migration of protons and hydroxide ions during the conduction of electricity, it is difficult to investigate the details clearly at this stage because their migration process remains largely unknown (Agmon et al., 2016). There is no doubt, however, that proton migration plays a positive role in this process, as the system computing power deteriorates in the case of non-protonic solvent dimethylformamide (see the first section of Supporting Information 3.A).

From the above analysis, we fixed the response period to 6 ms and repeated the same tests many times. Finally, we got the conclusion that better computational performance can be achieved by selecting better reading time points. For example, in NARMA2 task, NMSE of DI water was decreased from 0.2106 to 0.0624, and that of POM solution greatly decreased from 0.3865 to 0.0947. Figure 3.4(a,b) shows the best NARMA2 results of DI water and POM solutions, respectively. Their corresponding MC calculation results are plotted in Figure 3.4(c,d), which incorporated the MC results of Figure 3.2(e) for comparison purposes.

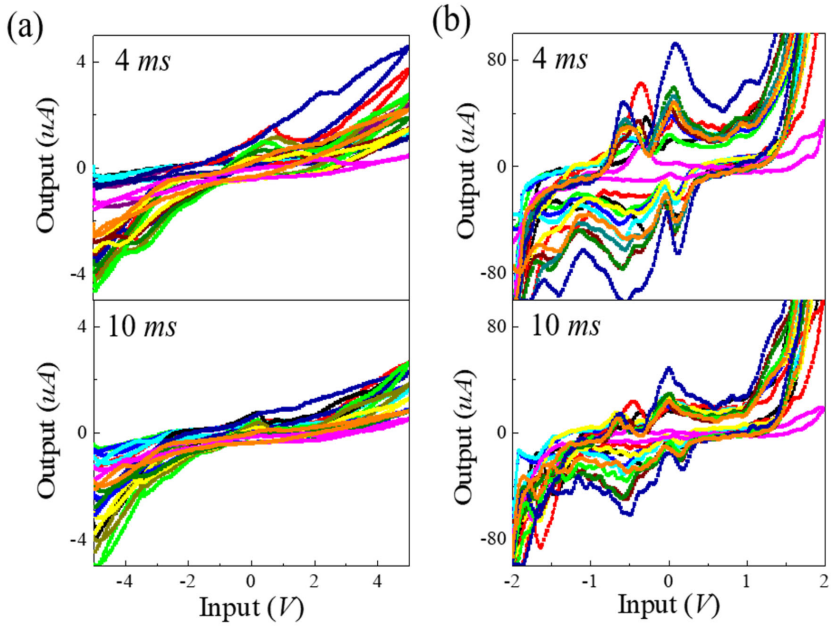

Figure 3.5: I–V characteristics at two testing periods. The sampling point was at 4 ms (upper graphs) and 10 ms (lower graphs) of a) DI water, b) POM solution.

In addition to the linear MC calculation, we use the Legendre polynomial to construct the target signal $P_q(t-k)$ in order to calculate the MC of higher order nonlinearities, where q is the degree of polynomial. The sum of MC_k values at the same q of past units $(t-k)$ is labeled as MC^q (Grigoryeva et al., 2016). This expression was used to extend the MC measurement for evaluating the nonlinear memory capacities, and the results are shown in the insets of Figure 3.4(c,d). The results confirm that MC^1 (i.e., the linear MC results), especially the MC_1^1 and MC_1^2 (i.e., the preceding MC_0 and MC_1), could be greatly increased by adjusting the sampling time. It did not affect the MC^q of the POM solution for higher-order signals (i.e., $q \geq 2$), but it had an adverse effect on the higher-order MC^q of DI water.

The information processing capability of our system was evaluated using a real-world task which was pneumatic artificial muscle (PAM) length prediction (Sakurai et al., 2020). PAM is well known to exhibit hysteresis, and to estimate its length, we need to consider the history of applied pressures. Therefore, this task is appropriate to evaluate the memory-required computational capability. In order to perform the investigation, 3000 data points were chosen from the open-source data file provided by Akashi et al. (Akashi et al., 2020). The training and testing processes were maintained the same to the NARMA2 task, and NMSE was used to represent the error. The results from Figure 3c were used to select the readout time; the magnitudes of current in DI water were measured at 1 ms intervals, and

that in POM solution were measured at 2 ms intervals. The duration of each input voltage was 4 ms, and 112 nodes were used, consistent with the previous tests. The NMSE values were obtained as 0.1102 from DI water and 0.2507 from POM solution, which are consistent with the NARMA2 results and MC calculations. A previous work reported the NMSE values, from the echo state network, between 0.28 to 0.003 using only air pressure to emulate the PAM length. Although the values obtained in this work are slightly lower, these results were obtained using a natural physical computing system rather than a software-based neural network. Furthermore, we only used 112 nodes for the computations compared to their 3000. Additionally, both results were significantly better than those predicted by linear regression system, which indicates that our system is capable of implementing the nonlinearity and memory capacity required for this task. The present work also demonstrates the strong computing power of distilled water. Therefore, with further development of our system, DI water can be competent for this task. The procedure of analysis are discussed in the first section of the Supporting Information and the results are shown in and Figure 3.A.1(c), Supporting Information.

To explore the effect of reaction time on the current-to-voltage response, we tested the $I-V$ characteristics of these samples at two periods equal to 4 and 10 ms, respectively. The current curves plotted in Figure 3.5 are similar in shape to the cyclic voltammetry curves (Fernandes et al., 2015). However, the shapes are not exactly the same with respect to the peak position and magnitude ratio. The two-terminal measurement lost definitive control of the potential at the working electrode surface; therefore, the chemical reaction in our system took place at input voltages which deviated slightly from the norm at each pair of electrodes with different distances. The complex but reproducible higher nonlinearity of the current from the POM solution was advantageous in the periodic signal reconstruction task. Conversely, the $I-V$ characteristics of DI water exhibit less nonlinearity but greater help to higher order tasks. This is directly reflected in NARMA2 task and PAM length task: the dynamics offered by chemical reactions of POM were less helpful to performance than proton migration in DI water. Our system provides a novel perspective on the reservoir application of Faradic currents in solution. Although an organic electrochemical reservoir has recently been reported (Cucchi et al., 2021), the detected signal is the electrical current flowing through an organic-ionic-electronic conductor, where the organic fibers within the electrolyte create strong nonlinear responses corresponding to multiple input signals. Our system detects the formation and transients of the electric double layer along with the electrochemical reaction. This results in good computing performance even at fast measurement speeds, and the power consumption is also expected to reduce owing to the dynamics of ions. Considering the high sensitivity of the computing power of electrochemical reaction to various factors, a more rigorous exploration ought to be conducted to maximize its computing ability, and to enhance the coupling of the functionality of complex chemical

reactions and the dynamics of light ions.

3.4. CONCLUSION

WITH the help of a multiway data acquisition system and electronic circuits, we realized a physical RC system based on ionic current or proton migration. This system displayed "feedforward connection" between nodes, which was proved to be independent of sample selection in this study. However, in our previous scheme, whichwas designed according to the nonlinear $I-V$ characteristics of molecules, this condition was achieved by introducing external parameters of "feedforward gain" (Kan, Nakajima, et al., 2021). We think this is a natural advantage of our system and deserves further development and exploitation. Moreover, we found that the complex and diverse chemical reactions of POM molecules contributed to the enhancement of information processing ability for periodic signals and memory in higher-order, nonlinear systems, but the computing power of higher-order tasks has not been fully utilized. Conversely, DI water showed strong computing power in the NARMA2 task. Other studies have shown that good performance in time-serials prediction task also enables handwriting font recognition, isolated word recognition, and other classification tasks (Appeltant et al., 2011; Du et al., 2017; Cucchi et al., 2021; Kan, Nakajima, et al., 2021). Therefore, our proposed device has the potential to serve as a computing system. Furthermore, in this device with the solution sample, electrons barely flow from the electrodes to the surface of the material owing to the fact that protons travel much faster than ions (Agmon et al., 2016). Based on this discovery, it may be possible to design amore powerful computing system using the proton or ion transfer with minimal electrochemical reactions within very fast time sequential range to provide a low-cost, low-power consumption, and highly integrated hardware devices for increasingly important edge computing. We hope that our research provides a new perspective combining chemical properties for the future development of hardware computing systems with low-power consumption.

3.5. ACKNOWLEDGEMENT

This work was supported by a JSPS KAKENHI grant (no. JP19K22877 and JP18H01135).

3.A. SUPPORTING INFORMATION TO CHAPTER 3

Herein, we supplement the main text listings, including the definition of performance evaluation indicators, training and testing methods, the setting of specific parameters, and others.

3.A.1. SUPPLEMENT TO THE PERFORMANCE AND TESTING RESULTS

We declared in the text that the prediction ability of the Polyoxometalate (POM) solution to periodic signals improved (Figure 3.2(b)) compared with that of distilled (DI) water. In this task, the input sinusoidal signal had 100 periods in total, and each included 300 time steps. Each time step lasted for 6 ms. The response currents were normalized in the range of [0, 1] for training and testing, as described in the next subsection.

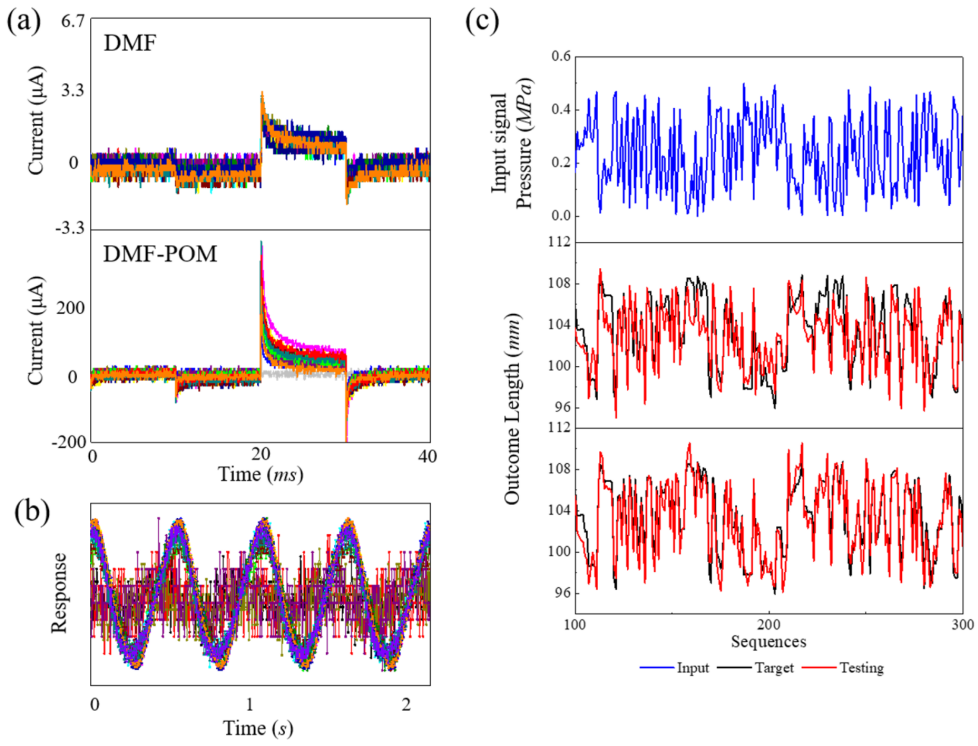

Figure 3.A.1: (a) Current responses of dimethylformamide (DMF) and DMF with Polyoxometalate (POM). (b) Responses of DI water to sinusoidal signal in which four readout pins are in the "OFF" state. (c) Presentation of pneumatic artificial muscle (PAM) length task: input pressure (top) as well as results of POM solution (middle) and DI water (bottom).

Figure 3.3(a) shows the current responses of DI water and POM solution with the original data captured by the oscilloscope. In the main text we declared that proton migration should play a significant role in the information processing capacity of the system. For comparison, we chose the nonprotonic solvent dimethylformamide (DMF) to test its current response waveform with and without POM, as shown in Figure 3.A.1(a). This was under the same process that led to Figure 3.3(a): imposed a string of input voltages (i.e., five random numbers) on one input electrode and read the current responses of 15 output electrodes.

Similar to the effect of POM in DI water, the addition of POM in DMF could greatly increase the current response and cause an initial spike in the response. However, the addition of POM did not make the responses at each electrode more distinct as it did in DI water, and it did not show any advantage in computing ability of DMF and DMF with POM. Thus, we believe that proton transfer is worthy of further study in the future.

Not all parts of the output current shown in these four figures are the response currents from the solutions. For example, the light gray output in the four graphs in Figure 3.3(a) and Figure 3.A.1(a) is an invalid node with no readout value. This was owing to an improper working state of the device, that is, the selector circuit used in the readout boards did not switch the specific port. Such an invalid node is acceptable in RC, so we kept these invalid nodes in the NARMA tasks and MC calculation. However, we eliminated this part of the response current from the sinusoidal periodic prediction task. This is because a weak current can still be read out even if the port is not "ON" because of the existence of leakage current on the circuit, and the weak current will lead in phase compared with the current that is being normally read out (i.e., when the switch is "ON"). We illustrate the responses of DI water in both cases in Figure 3.A.1(b), whereby the four weak, noisy, and phase-shifting waveforms are the read leakage currents. It is obvious that this feature is detrimental to the processing of periodic signals. Furthermore, we experimentally investigated the abrupt initial spikes shown in the insets of Figure 3.3(a) and Figure 3.A.1(a) on the breadboard. By connecting a series of side-by-side resistors on the breadboard with a series of readout ports on one output board, we compared the output current with and without the neighboring resistors. The results show that the spikes appeared on the readout port when a resistor was also inserted on the neighbor port, and became larger with an increasing number of neighbors with a resistor. Therefore, in the system with solution, POM caused capacitive resistance circuits on the parallel electrodes of our device, which interacted with the adjacent electrodes, and resulted in an initial current spike.

3.A.2. DETAILS OF PAM TASK

Pneumatic artificial muscle (PAM) is a retractable, extendable soft actuator operated by pressurized air inside a pneumatic bladder, as shown in Figure 3.A.2. Pressure is not the only factor that determines the generated force of PAM, but also the inflation state that determined by the existing pressurized air. In other words, the mathematical model of PAM functionality is a nonlinear dynamic system. Besides, the behavior of PAM can hardly control because of the difficulties in status sensing accurately caused by their mechanical nonlinearity and anisotropy of the generated force. However, reservoir computing that operates on the principle of approximating any arbitrary dynamical system, making it a suitable approach to capture and model the complex behavior of PAM by learning its intrinsic dynam-

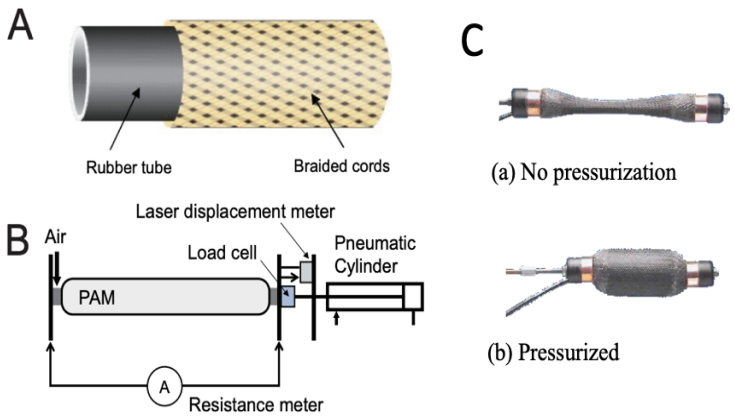

Figure 3.A.2: Description of pneumatic artificial muscle (PAM). (A) Schematic of first developed McKibben PAM. (B) PAM measuring equipment. (from Sakurai et al., 2020) (C) Air muscle contracting and extending. (from Rocketmagnet, 2007)

ics. Therefore, we introduced this task in our work to further evaluate the computing ability of our system in a real task.

The data we used was from an article entitled as "Input-driven bifurcations and information processing capacity in spintronics reservoirs". In this task, the input data, air pressure (the blue data in the upper Figure 3.A.1(c)), will be transferred into the voltage from -2 to 2 V input into our system. The period of each input voltage is 4 ms, and read the current of DI water at 1 ms, and read that of POM solution at 2 ms. The number of nodes, length of training data and testing data, and other parameters are consistent with those in NARMA2 task. POM solution and distilled water test results are also shown in Figure 3.A.1(c): the middle illustration shows the prediction result of POM solution and the bottom one shows that of DI water. Their NMSEs are 0.2507 and 0.1102 respectively.

3.A.3. TRAINING AND TESTING PROCESS

Two training methods were used in this study. One was the least-squares method used for the training of periodic target signals, and the other was ridge regression, which was used for the training of high-order target signals. An input signal with M data (M = 3000 in this study) was applied to a reservoir with N (N = 112 in this study) nodes produced $M \times N$ reservoir states X. In this case, an output signal $\hat{y}$ with M data can be obtained by multiplying these reservoir states by N output weights W_{out}. The training aim was the minimization of the objective function $\|y_{ytrain} - \hat{y}_{train}\|^2$, where subscript "train" refers to the training part of the target signal and output signal. For the least square method, the output weight W_{out}

was calculated by,

$$W_{out} = (X_{train}^T X_{train})^{-1} X_{train}^T y_{train} \tag{3.A.1}$$

The other method used for the NARMA task and MC calculation was ridge regression. Compared with the least-squares method, it artificially adds a non-negative factor k to the main diagonal element of the independent variable matrix, as follows,

$$W_{out} = (X_{train}^T X_{train} + kI)^{-1} X_{train}^T y_{train} \tag{3.A.2}$$

The predicted output signal is obtained by multiplying the testing signal by the output weights calculated from the training signal. The prediction performance is evaluated by comparing the deviation between the predicted and target signals. In the training process of periodic signal prediction, the first 20 data points were used as the initial states to initialize the reservoir, 2500 as the training part, and 400 as the testing part. The remaining 80 data points were ignored because several predicted values of them sometimes were far off the target. In the NARMA task, the first 100 data points were used for the initialization of the reservoir, 2500 for the training part, and the remaining 400 for the testing part.

Ridge regression used a non-negative factor k, and different values of k were used for different situations. k was set to the value of 0.1 for the NARMA2 task, and to the value of 1 for the MC task.

3.A.4. DETAILED DESCRIPTION OF TESTING PROCESSING

In this section, we provide a detailed explanation of the process flow shown in Figure 3.1(d). Arduino Mega 2560, as the main controller in our testing system, takes charge of sending the input, control, and receiving response signals. Arduino sent the "begin transmission" order and the input voltage (digit value) to a digital-to-analog converter (DAC) PCF8591, and the analog voltage can then be obtained from the DAC. Because the output range from the DAC is in the range of 0-4 V, we also used a differential amplifier circuit to convert the voltage to -2 to 2 V and then amplify it; the amplification factor can be adjusted freely between 1 and 3. In this case, voltages from negative to positive values are imposed on the input electrodes. Following the imposed voltage, the response current was detected on the readout electrodes. Each readout electrode was connected to a corresponding pin of the readout boards, with a total of 48 pins (six readout boards, each with eight pins). The integrated current to voltage converter and differential amplifier circuit on each readout board converted the response current read to a voltage of 0 to 4 V owing of the reading range limitation of Arduino analog ports. There are three options for the magnification factor: 10^4, 10^5, and 10^6. The converted voltage on eight ports of each board will be selected in turn through our 8-way selector circuit, which includes eight switches controlled by the

shift register SN74HC595. The Arduino sends the control signal to the shift register to alternately select one switch out of eight to turn on to transfer the conversion voltage of the corresponding port to the output port. In this case, the response currents from a total of 48 readout electrodes on the six boards can be captured through just six analog pins on the Arduino.

3.A.5. COMPOSITION OF THE TEST END OF THE ELECTRODE

Figure 3.A.3 shows an array of electrodes. One testing board was composed of 90 pairs of rounded-tip electrodes, each with a gap of 50 μm. The distance between adjacent electrodes was 400 μm. Each electrode was 80 μm wide, and their corresponding terminals were wire-bonded to external hard wirings. The electrodes, composed of Cr/Pd/Au, were deposited on Pyrex glass and then covered by a polyimide film that left the facing area bare.

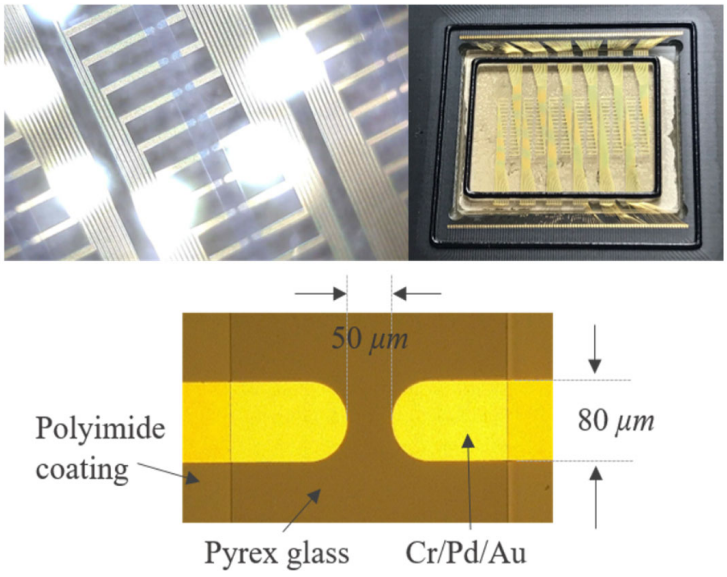

Figure 3.A.3: Actual electrode array graph and the magnified parts of one of its pairs. The test end of electrode array for polymer wire growth consists of 90 pairs of rounded tip Au electrodes with a gap of μm. The wiring area is covered by polyimide film.

3.A.6. INSTRUCTIONS ON SELECTION OF POM SOLUTION CONCENTRATION

In the main text, we stated that the concentration of POM solution was not specified strictly in this study. However, this does not mean that the selection of solution concentration is arbitrary or unreliable. During our testing, we found that there was no significant difference

in current responses for solution concentrations greater than 1 mg/mL (but not greater than 10 mg/mL). In this section, we did a simple proof. The cyclic current responses at different solution concentrations was tested on a new set of electrodes. The cycle voltage at each concentration was repeated 100 times, and the cycle data of the last time was selected. This allows the electrochemical reaction to go on for a while, so that conform to our performance testing process shown in the text. Eight groups of different solution concentrations were selected for testing. After each test, the surface of the electrodes was washed with distilled water and sucked it dry before the new test solution was added. The results of this test are shown in Figure 3.A.4.

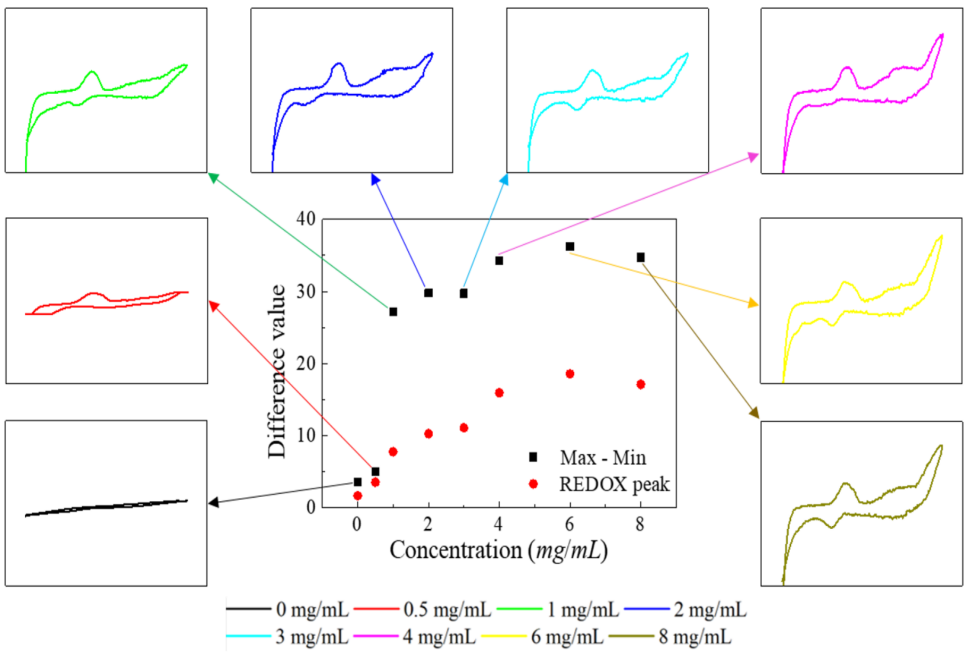

Figure 3.A.4: $I-V$ characteristics of POM solution at different concentrations from a fixed pair of electrodes. 8 surrounding illustrations have the same axis values: x-axis is the input voltages from -2.5 to 2.5 V, y-axis is the output currents from -20 to 25 μA.

The middle illustration in Figure 3.A.4 is the statistical results. The black dots represent the difference between the maximum and minimum value of each curve, while the red dots are the peak difference between the two most prominent REDOX peaks during voltage increase and decrease. The 8 illustrations surrounding the middle one are the circulating currents at 8 sets of solution concentrations, which are 0, 0.5, 1, 2, 3, 4, 6, 8 mg/mL, respectively. The voltage increases and decreases at 0.2 V intervals between -2 and 2 V, each voltage value lasting 2 ms then readout the current. Intuitively, there is no significant differ-

ence between the cyclic voltammetry curves between 1 and 8 mg/mL. This may be due to the fact that only about 0.1 mL of the solution is dripped on the electrode surface, making the difference in the amount of solute insignificant.

4

APPLYING STOCHASTIC COMPUTING FOR UNCONVENTIONAL AI SYSTEM

This chapter is published as:
S. Kan, Y.Sasaki, T. Asai, M. Akai-Kasaya, "Applying a Molecular Device to Stochastic Computing Operation for a Hardware AI System Design", *Journal of Signal Processing*, **2021**, 25, 6.

ABSTRACT Stochastic computing (SC) has attracted much attention of researchers for decades and plays an important role in the field of information science. The relative large cost on area and power of stochastic number generators (SNGs) greatly offset the advantage of SC for hardware implementation. In this paper, we proposed a scheme replacing the SNG's function during weight updating process by a molecular device, which can spontaneously generate random spikes and noise. A control electrode (gate electrode) is added to previous electron-cascading model by simulation, the frequency of charges collected on collector electrode that higher than threshold are related to the number of times voltage is applied on this gate electrode. Specifically, as the voltage applied again and again, the frequency may increase or decrease gradually. This is basically consistent with the basic principle of weight update. After some parameter adjustment and model optimization, we believe that this molecular device is expected to replace the part of network weight update in the hardware SC circuit.

4.1. INTRODUCTION

COMPARED to the traditional software implementation, the hardware implementation of artificial neural network (ANN) becomes a breakthrough to further accelerate the training speed and reduce the power consumption. However, an ANN system generally requires a complex hardware which involves thousands of neurons in processing tasks such as classification. Researches in stochastic computing (SC) has been rekindled owing to the increasing computational complexity in deep learning and the bottleneck of circuit miniaturization. SC is a low-cost form of computation in which numbers are represented in bit-streams and interpreted as probability. It can use very simple logic to perform complex operations (Li et al., 2016), such as multiplication can be used in SC with an AND gate. Complex functions, exponents, absolute values, square roots and hyperbolic tangents can all be calculated using very small number of logic gates (Alaghi and Hayes, 2013). Another attraction of SC is its fault tolerance (Najafi and Salehi, 2016). Bit flipping introduces least-significant bit (LSB) errors, whereas in traditional binary representations, a single bit error can make a big difference.

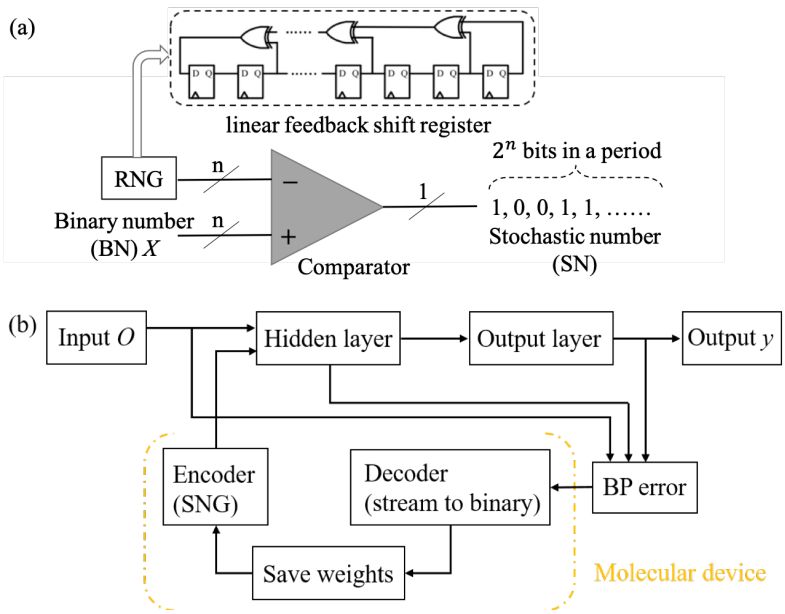

Figure 4.1: (a) Traditional digital SNG consists of random number generator (RNG) and comparator, (b) Weights computation and updating processing in traditional stochastic computing based digital circuits

The basic unit in SC that converts a binary number into stochastic bit-stream is stochastic number generators (SNGs) as shown in Figure 4.1(a), the random number generator (RNG) of which is composed of CMOS, such as the widely used linear feedback shift reg-

ister (LFSRs) (Zhakatayev et al., 2018). The n-bit binary number (BN) X and n-bit random number A are fed into the digital comparator, if $X > A$, the digital comparator outputs one bit '1', otherwise it outputs one bit '0', completing one cycle. A bit-stream of 2^n length can be obtained by repeating 2^n period and keeping X constant, thus the conversion of n-bit BN to a bit-stream SN is completed, and the probability of each bit being 1 in this bit-stream is $X/2n$. In SC, although relatively simple logic circuits may be sufficient to perform complex calculations on stochastic number (SN), larger circuits are needed to generate the input bit-streams. Therefore, generating SN is often more expensive than calculation on SN. Their relative large cost on area and power greatly offset the benefits of computing for hardware implementation, so much lower cost method is needed to develop bit-stream generators. With the development of nanotechnology, it will be a trend to extract true randomness from physical viewpoints to further reduce the overhead of bit-stream generators used for SC.

A common ANN training method is state updating based on backward propagation (BP). For example, weight $w_i(t)$ in a BP based ANN is adjusted as

$$w_i(t+1) = w_i(t) + \eta \Delta x_i \tag{4.1}$$

where η is the learning rate, Δ is the deviation between output and target. The adjusting method by SC digital circuit is exhibited in Figure 4.1(b). Output y is calculated by the hidden layer and output layer in ANN based on input O and BP error in the form of stochastic bit-streams. Specifically, when the clock signal arrives, the latest weight value obtained from the operation of the adder to the previous weight value is fed into the register, completing an iteration update. The initial value of weight w_i should be represented as SN through a SNG and then input to a register waiting for update, though this process is not shown in the diagram. As widely known that SC provides a coding way of operating with analog quantities (since probabilities are defined between 0 and 1) using digital circuitry. Unfortunately the range of representation is restricted to a limited interval [0, 1] or [-1, 1], make it unable to use the proper weights and bias values provided by traditional training algorithms in the application in SC-based hardware ANN, that is why encoder and decoder are necessary.

Inspired by the spontaneous randomness of charge transfer between two electrodes and the accumulative characteristic of charges, we bring up an idea of replacing the traditional CMOS circuit by a molecular device (dotted in Fig. 4.1(b)) to help update weights during the training process. This scheme can save the transistors consumed by the encoding and decoding circuits and even the memory. In this paper, we verify the feasibility of this idea through simulation and try to find a more effective implementation scheme.

4.2. PROPOSED METHOD

In the following, we introduce the principles of our proposed model in detail, then we show our analysis to the simulation results and practical planning of its application in SC hardware systems.

4.2.1. SIMULATION MODEL

A neuromorphic network, single-walled carbon nanotubes complexed with polyoxometalate (POM), was found being able to spontaneously generate random spikes and noise (H. Tanaka et al., 2018). To represent the qualitative electrical behaviors in such random network of POM-particle cells, a cellular automata (CA) model on a 2D regular grid was conducted. In this model, the 2D grid of cells are defined as a structure with random defects, charges in each cell will transfer to their neighbor cell(s) with the largest (or also the second largest) charge difference. Due to the voltage applied between the left and right electrodes, charge will spontaneously shift from right to left (i.e. from source electrode to collector electrode). The charges collected at the collector electrode is recorded at each moment. After about 1,000 cycles, the frequency of the higher changer numbers began to slightly increase over time. Although the transition is spontaneous and random, unfortunately, such process is not controllable and cannot be used directly. Therefore, based on this previous model, we introduce another electrode called gate electrode to enhance and control the transfer of charges. The revised model is represented by Figure 4.2(a). The effect of V_G acts on the transfer process: when voltage V_G is imposed on gate electrode, number of charges that transfer to neighbor cell(s) will be related to voltage V_G instead of mechanism in original model. In this case, the charge transfer process will become orderly and controllable on a random basis.

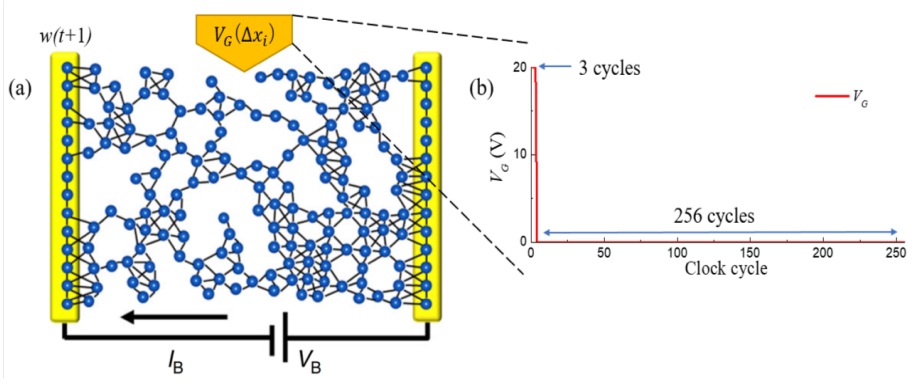

Figure 4.2: (a) 2D CA-model with an additional gate electrode V_G, (b) The method of action on electrode V_G of one epoch

We found that if we hold the voltage V_G on this electrode for 3 clock cycles, the charge accumulation effect will add up. Otherwise, a gradual increase or decrease trend in the number of higher charges cannot be observed. Considering 8 bits of data are generally used in the digital system, this means at least 256 clock cycles are required to generate stochastic numbers. Therefore, in our simulation model, we define each epoch of 259 clock cycles as shown in Figure 4.2(b), first three are the application time imposed to the gate electrode, the rest 256 cycles are the observe time. But of course the number of cycles is not fixed, it can be extended as the case may be, and the longer the cycles are extended, the more "high bits" will be obtained.

4.2.2. RESULTS ANALYSIS

In the original CA model without V_G, the number of charges N_C observed in the collector electrode of only a small number of occasional increases over time. In this new model introduced with V_G, the density of high bits becomes higher during the observation cycles as the epoch number increases (when V_G is a high value). Figure 4.3 gives four examples selected from epoch axis to show the spike density variation. These four points are circled in Figure 4.4(a). These four examples show clearly different density of high bits, especially the numbers that reach a higher N_C. It can be easily found that the location of the higher charge appears randomly so it can generate the SN with high quality for computation.

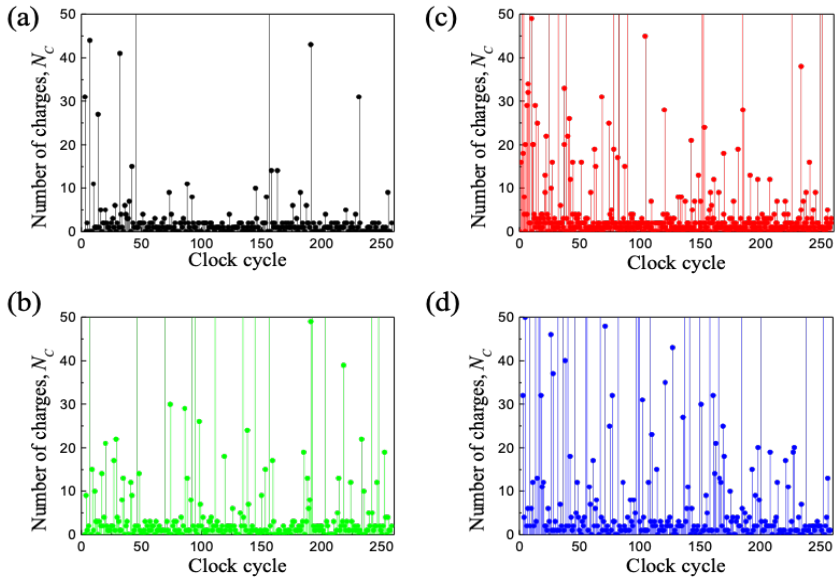

Figure 4.3: Voltage spike density increasing along epoch: (a), (b), (c) and (d) show four cases selected from epoch axis.

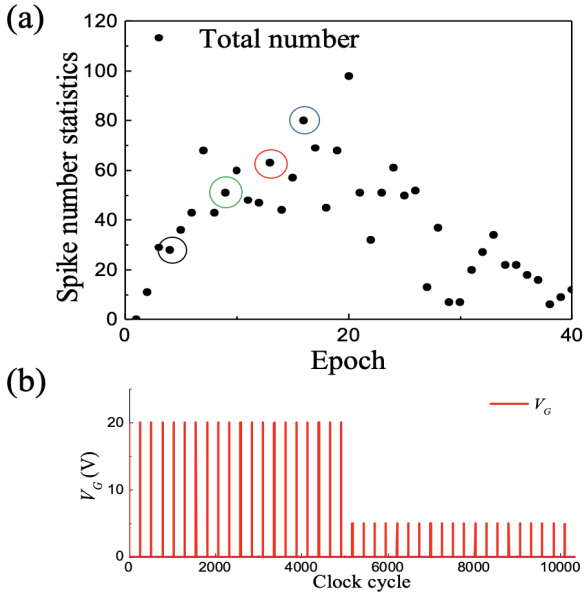

Figure 4.4: (a) Statistic results along epochs during the observation cycles, (b) Action process of voltage to gate electrode.

To clearly show how the density varies along epochs under the control of V_G, we counted the number of voltage spikes that over the threshold voltage (i.e. 5 V) during the observation cycles (i.e. 4-259 cycles in each epoch), and plot the statistic results in Figure 4.4(a). Figure 4.4(b) is the corresponding applied voltage of V_G. For simplicity, hereafter when we mention the frequency of spikes, it refers by default to the number of spikes that the collected charges greater than the threshold voltage. The voltage applied to V_G during the first 20 epochs is 20 V, and applied 5 V for the rest 20 epochs. Such result demonstrates the persistence of the effects of the previous state, so that the influence of the previous state is consequent to involve in each iteration. Although there are some disturbances along the way up and down, this trend (the density increases or decreases with each iteration) is a good representation of the process of updating weights as shown in Equation 4.1.

4.3. MODEL IMPROVEMENT AND DEVICE REQUIREMENT

ALTHOUGH the current model has achieved a well control under the gate electrode to charge transfer, it cannot be practically applied at present. There are still several requirements needed to achieve for the next step. In this part, we would like to state these requirements and clarify what we will do for the next step.

4.3.1. ACCURACY

As exhibited in Figure 4.4(a), the frequency of statistics in the spikes does not steadily increase nor decrease. And the results of many tests show that such deviations from the expected trajectory are irregular. Random behaviors and noise existed in the model significantly interfered with the stability of the trend, which is bound to have a negative impact on the training results of the network. Therefore, it is necessary to enhance the controllability of the charge transfer process in this model.

4.3.2. REPEATABILITY

THE network states usually needs to be iterated repeatedly until the maximum number of iterations is met or the prediction accuracy of the model in the training sample set reaches the preset requirements. In this case, the spike density changing process should be repeatable. Unfortunately, the current model cannot realize such process. When performing more rounds of the process in Fig. 4.4 (i.e. V_G =20 V for 20 epochs and V_G =5 V for 20 epochs) controls, the statistical number of spikes start to be unstable since the second round, and the increasing and decreasing trend was almost not visible since the third round. This phenomenon is also harmful to the updating of weights.

4.3.3. LONG-TERM MEMORY

CURRENT model shows short-term memory, but to utilize such system in practice, long-term memory is needed to play the role of registers in the traditional digital circuits. Therefore, one important progress we need to make is to keep the density of spikes for longer time after V_G applied. It is obvious to know that the decay of density is due to the lack of charges replenishment. So one effective solution is to keep the density of pikes by giving V_G a suitable value during the observation cycles. This idea has been confirmed by experiments of real molecular device, and long-term memory has also been found and verified in similar molecular devices Hagiwara et al., 2021. This suggests that long-term memory is possible to achieve, but how to properly apply this situation in simulation combined with the actual situation in digital circuits will be a challenge.

4.4. CONCLUSION

IN this paper we discussed about the charge transfer process in a molecular device with three electrodes and its potential utilization value by our simulation model. Firstly, we introduce the method of neural network weight updating using SC in traditional digital circuits, and then propose our scheme, that is, the inherent characteristics of charges transfer in molecular devices can be used for updating weights. Next, we present some simulation

results to prove the feasibility of such a scheme. By modifying the previous CA model, basic conditions have been met for supplication: the occurrence frequency of charges on the collector electrode that over the threshold can gradually increase or decrease as the number of iterations increases. In addition, higher spikes (i.e. over the threshold) occur randomly during the observation cycles because of the natural random process in charge transfer.

However, there are still some problems to be solved before such a scheme can be applied in practice. We itemized them in this paper and they will be our next direction of work. The first problem is to show a smoother trend with less deviations from the trend line so that they can be applied in real tasks with higher accuracy and reliability. Second is to realize the repeatability of updating process. This is because the training of a network involves thousands of iterations, so repeatability of trends is necessary to ensure that the best fit to be found. Last, we also need to find a way to achieve long-term memory so that the previous weight can be kept before needed for updating.

Compared to a fully-implemented SC digital system, using this proposed scheme could help save the number of transistors used and reduce the energy consumption. These above results display the great potential of molecular device for implementing compact and energy efficient in-memory neural networks, and the potential of stochastic approaches for hardware artificial intelligence.

4.A. SUPPORTING INFORMATION TO CHAPTER 4

As this is a conference paper with a limited word count, many works could not be covered in detail. Therefore, a supplementary section has been added to provide a detailed introduction on the background, specific experiment details, and the comprehensive thought process that led to the birth of this schematic concept.

4.A.1. INTRODUCTION OF REVISED CELLULAR AUTOMATA MODEL

A cellular automata (CA) model consisting of source and drain electrodes connected by SWNT/POM network was constructed by H. Tanaka et al. This CA model reproduces stochastic transfer of limited number of charges from a cell to its neighbouring cell with the largest gradient for state $\Delta a_{i,j}^{k}$ among all k neighbours, as illustrated in Figure 4.A.1(a). Based on this original CA model, a gate electrode (Figure 4.A.1(b)) is added. To observe the effect of gate voltage V_G on the collecting state of charges on the drain electrode, the number of charges on the drain electrode under different voltage values are statistically compared. Figure (4.A.1(c)) shows the number of charges N_C along time (left) and the probability of N_C exceeds the threshold of 25 (right) under two different gate voltages, $V_G = 11$ and $V_G = 14$. N_C of $V_G = 14$ obviously larger and denser than that of $V_G = 11$, and the probability also

increases right after V_G increase from 11 to 14 V in the middle stage. This implies that the probability can be controlled by V_G. Another thing worth noting is that the effect of gate voltage on the probability persists for a while after V_G are gone. Although the probability decreases immediately after V_G is stopped, the magnitude of the decrease is significantly smaller when V_G = 14 than when V_G = 11. If we further complete this device to achieve the combination function as encoder, decoder and memory, this molecular device could replace them in SC hardware circuits, as shown in Fig. 4.1(b).

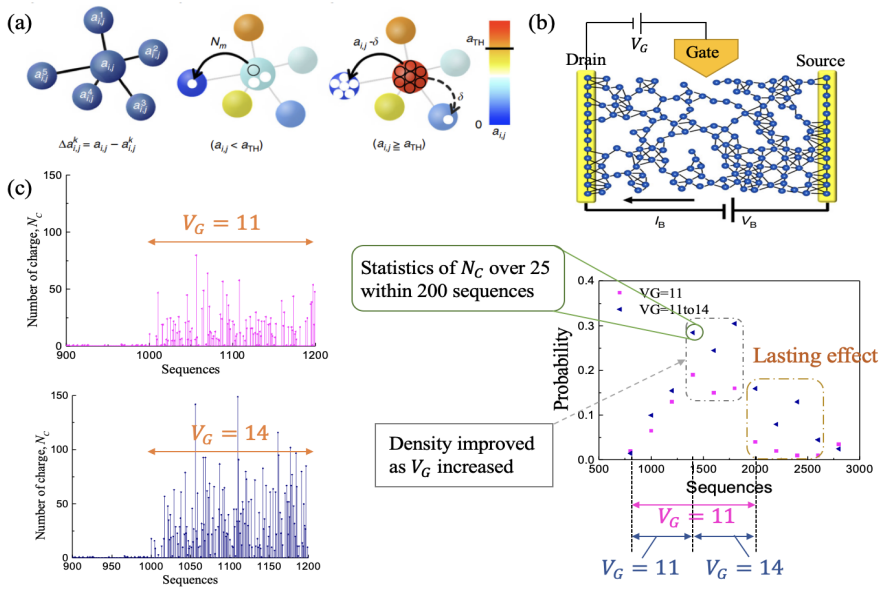

Figure 4.A.1: Revised CA model and simulation results.

4.A.2. BACKGROUND OF STOCHASTIC COMPUTING

With the wide application of deep learning and big data and its development in various fields, the hardware implementation of neural network (NN) is highly anticipated, but it becomes a difficult challenge due to the increasingly complex computing requirements, which require high computing performance and low power consumption. In recent years, research on SC, an arithmetic method devised in the 1960s, has been reignited. With SC, data can be encoded into stochastic number (SN) using bit strings generated probabilistically, making it possible to perform complex operations at a low cost using very simple logic (Alaghi and Hayes, 2013). For example, in conventional arithmetic, a huge digital multiplier was required to perform multiplication, but in SC, multiplication can be performed with just a single AND gate (Figure 4.A.1). As the name suggests, SC computations are based on

probability values rather than Boolean values taken in conventional digital circuits. Therefore, it is expected that by utilizing SC, the current bottleneck caused by high computational complexity can be fundamentally overcome.

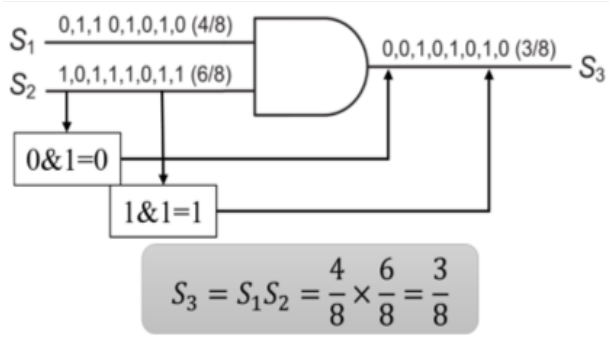

Figure 4.A.2: Example of multiplication of two SNs performed by AND gate.

4.A.3. BIRTH OF SCHEMATIC CONCEPT

The computation method of SC has theoretical advantages such as simplified calculations and low power consumption, but it is currently not considered to have absolute advantages in the implementation of hardware neural networks (van Daalen et al., 1993). This is because SC is a probabilistic calculation, and a module corresponding to the mutual conversion between binary numbers (BN) and SN is required. These two processes are called encoding and decoding, respectively. Taking the encoder as an example, as shown in Figure 4.1(a), the basic unit is a SNG which converts BN to SN by a RNG (Alaghi and Hayes, 2013). The comparator subtracts BN and the random number, and outputs an SN of 0 or 1 depending on the positive or negative result of the subtraction. When n is set to 8 in Figure 4.1(a), over 1000 transistors are required for a single SNG, whereas NN requires SNGs and memory cells for each input data, resulting in a significantly higher implementation area and power consumption. Therefore, the cost incurred by existing hardware significantly offsets the advantages of SC calculation in hardware implementation, necessitating lower-cost hardware to optimize SC implementation. For example, new devices that can directly store SN states can be used to avoid using encoders and decoders.

According to previous research by our study group, it has been revealed that the electrical characteristics of network components made from phosphomolybdic acid (PMo12) molecules (POM) can mimic brain function (H. Tanaka et al., 2018). Such networks can theoretically reproduce memory functions, as they are similar to capacitors. Additionally, these molecular components can generate random current spikes at varying intervals when voltage is applied, and their frequency of occurrence is determined by the applied voltage

and duration. This process is similar to the generation of SN in SC, and if this characteristic can be controlled effectively, a series of processes such as SN generation, calculation, and memory can be achieved using these molecular component. If such control can be achieved, a circuit can be constructed and implemented that targets weight updates in NN. Having this implementation would enable the processing of information in the probability domain, making the circuit no longer requiring conventional encoders, decoders, and memory. Such an integrated circuit constructed in this manner can reduce computational complexity using SC, meeting the demands for small area and low power consumption.

4.A.4. TENTATIVE STUDY AND METHOD

M1 *Control of spike generation frequency and improvement of long-term memory in molecular devices through simulation.*

A molecular device fills the amorphous crystal of POM molecules (which can store a lot of charge) in the gold electrode gap on the SiO_2/Si substrate. By applying a voltage between the source and drain electrodes, it generates random spike signals. Based on the output characteristics of the device, previous studies by H. Tanaka et al. have constructed a CA model (Figure 4.2(a)) that reflects the redox reactions of the molecules and reproduces charge transfer in the molecular device. While it is already clear that the spike density of the molecular device changes with the channel bias voltage V_B, simulation results in Figure 4.3 show that injecting charges from the gate changes the amount of charge accumulation in the molecular film and affects the spike density. The applicant has developed a new model to control the current spike density at the simulation level and showed that the amplitude of spike generation varies with the size of the applied pulse voltage from the gate voltage (V_G) and also depends on the number of pulse applications.

By adjusting the voltage applied to the source, we were able to achieve the function of long-term memory, maintaining a relatively stable state of spike frequency for an extended period. This suggests the possibility of using it as an alternative to memory. We successfully reproduced the output that changes depending on the oxidation-reduction state of the molecules by changing the rules in the CA model. By providing feedback between these results, we could determine the most reliable operating conditions for actual devices.

M2 *Realization prospect of memory effect.*

Figure 4.A.2(a) shows an overview of the molecular device and its output. Figure 4.A.2(b) shows the dependency of V_B on the observed current spikes. The spike current is believed to be an electron accumulation effect due to the multi-electron reduction of molecules within the device, with each molecule acting as a capacitor. The density of output spike signals from the device is varied by V_G, and efforts are made to manifest the memory effect shown

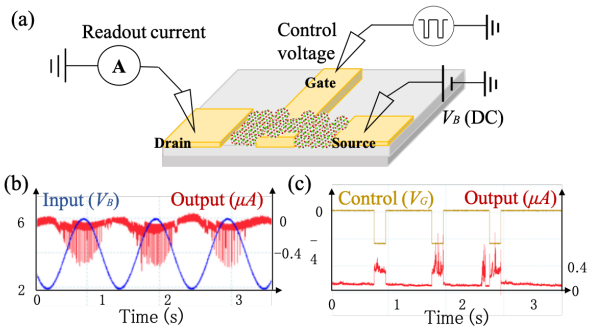

Figure 4.A.3: (a) schematic diagram of POM devices; (b) output current corresponding to input voltage of sinusoidal signal; (c) output current when the gate voltage is repeatedly applied.

in simulations. Figure 4.A.2(c) shows the charge injection effect by the gate, but there are many elements that need to be determined, such as the voltage dependency of spike frequency, gap length dependency, and the effects of molecular film thickness, operating atmosphere, and temperature. In particular, the influence of moisture in the atmosphere on the device output is significant, and controlling these factors still remains a challenge.

M3 *Construction of weight-updating SN circuits using molecular devices aimed at practical application of SC-NN.*

In this study, we try to verify the effect of gate voltage in actual molecular devices and aim

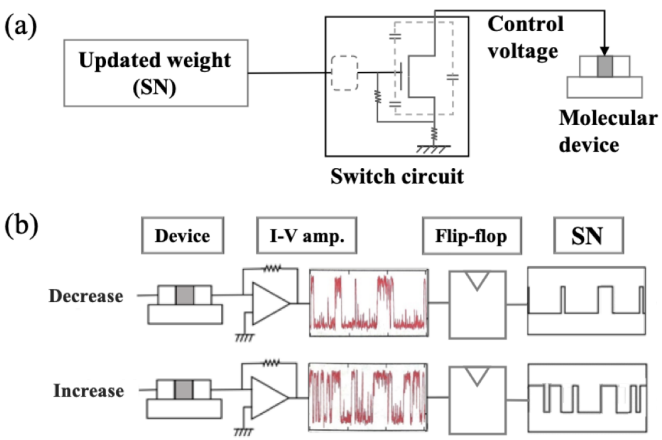

Figure 4.A.4: Realization method of weight update SN circuit using molecular element.

to demonstrate the memory effect shown in the calculation. If molecular devices have this function, it is possible to realize the weight update circuit in Figure 4.A.3 without a signif-

icant change in size. To apply molecular devices to the hardware design of SC-NN, it is necessary to connect them with two interface circuits, as shown in Figure 4.A.3. Interface 1 requires an MOSFET switch circuit that can achieve charge injection and release using the signal based on SN, as shown in Figure 4.A.3(a). The spike signal width converted by the spike voltage conversion from the output current of the device is wide, from $1\mu s$ to $1ms$, and is converted to SN by a flip-flop circuit. This part of the circuit is interface 2 (Figure 4.A.3(b)). As a result, the back and forth of SN and BN in NN is avoided, and it is expected that a hardware NN based on complete SC calculation will be constructed. By avoiding the use of encoder, memory, and decoder, this achieves a reduction in size and power consumption of about one-tenth of the current hardware structure design based on SC for NN.

5

Discussion

5.1. PHYSICAL SYSTEM DESIGN

DESIGN of a physical computing system requires a global perspective, meaning one should be aware of the computational forms and rules of the physical system and have a clear understanding of how to utilize them effectively. This is precisely the foundation for the wide application of digital circuits. However, as physical systems are analog in nature, the diversity of choices, complexity of inherent computational methods, and the unknown impact of factors such as noise and imperfection make it difficult to design a physical system from scratch. The physical implementation of reservoir computing, with its simplified computational rules and less attention on control over details, has led to extensive exploration as a candidate for the next generation of hardware computing paradigms. However, most existing works treat the existing physical system as a dynamic system that does not require adjustment or precise control, utilizing its dynamic characteristics directly to evaluate performance within the framework of reservoir computing. Although it expanded the breadth of choices for physical systems, it provided limited assistance in the design. This work starts exploring simple and effective design rules from a top-down perspective and develop and explore applicable physical systems based on these design rules.

5.1.1. MATHEMATICAL POINT OF VIEW

ECHO state network (also called reservoir) is initially defined by a $N \times N$ matrix W^{res}, representing N inner processing nodes sparsely connected with each other, where the values in this definition matrix reflect the degree of connection. As introduced in Section1.3, the behavior of dynamical systems in chaotic state no longer depends on the input sequence, but leads to different trajectories under the influence of different initial states. Dynamical systems for computing purposes need to have echo state property so that the network can still appear the convergence even with different initial states, and they have the best computational performance when they are about to reach the edge of chaos. In order to let the reservoir working as a dynamical system with computing ability, a commonly adopted necessary condition is that the spectral radius of the weight matrix less than 1, i.e., $\rho(W^{res}) < 1$. This condition can be easily hold true and the values can be easily adjusted because diagonal factorization of matrices is easy to implement at the code level, but it is hard to apply on physical systems. Although any physical dynamic system can work like a "black box" under the condition of echo-state-property (ESP), this is unhelpful for system regulation and let alone the design of unknown systems, because it is difficult to finely adjust the specific values in the matrix only through the limited parameters of the dynamical system. From this, we thought of a method to avoid the diagonal factorization and simplify the structure, which is to define the weight matrix itself as a diagonal matrix. In this case, the values on the diagonal are the eigenvalues, representing the feedback gain of each

processing node itself, while the elements off the diagonal are all zero, representing the independence of each node. Figure 5.1 schematically shows the above concept and this gives rise to the research work of structure shown in in Figure 2.3(a).

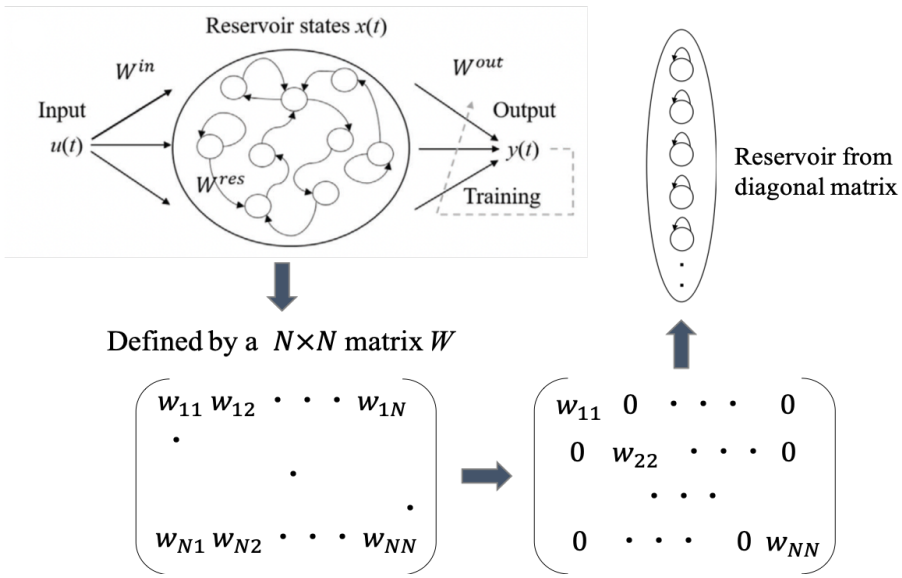

Figure 5.1: Reservoir structure corresponds to weight matrix

5.1.2. ROLES OF INTER-CONNECTIVITY

IN Chapter2, all results of information processing performance are compared under three different basic frameworks: parallel-node structure, parallel-group structure, and randomly-connected-nodes structure. For the randomly-connected-nodes structure, their sparsity of the connection is adjusted by the value of spectral radius. The result of commonly used benchmark task shows that a degree of connection between processing nodes plays an important role in solving this nonlinear problem, and even sparse connections with regularity significantly increase the performance. Proper connection not only helps to solve nonlinear problems, but also improves the performance of classification tasks. Dense connection retains a longer echo-state duration of the original data thus the best for processing nonlinear signals, but this makes the current data less informative so that loss the accuracy in classification task. Therefore, the most limiting factor for this parallel-nodes structure is that its insufficient short term memory capacity cannot approximate higher-order nonlinear systems well.

It is worth noting that emphasizing only the impact of structure is not sufficient because these three structures also employ different activation functions: standard echo state

network uses the activation function of hyperbolic tangent function while that of parallel-node and -group structure is based on the $I-V$ response of diodes. Results suggest that the degree of connectivity between nodes has a greater impact than the intrinsic nonlinear function. Therefore, a feedforward gain is applied regularly to make the scheme a parallel-group structure (Figure2.3(b)) on the basis of the independent nodes so that the effects of connectivity between nodes on computing performance can be further observed. Chapter2 and Chapter3 of this thesis are consequential exploration of this schematic idea and they both verifies its application value.

Moreover, since the activation function of these processing nodes is not fixed between -1 and 1 as the hyperbolic tangent function is, and the reservoir weight is not set in accordance with the spectral radius of less than 1, the system itself is prone to chaos. Problem can be solved by manually setting a regression to the system input. When the input value calculated following Equation 2.4 exceeds the detection range of ADC, it goes to 0. In this sense, ESP index (Equation 1.3) will always be 0 for this system no matter the value of feedback gain a and input gain e, in other words, chaos is avoided in this system.

5.1.3. EXTENSION TO EXISTED PHYSICAL DEVICE

ALTHOUGH the design scheme mentioned in Chapter 2 can be easily implemented through running code, we still conducted experimental testing on electronic circuits. This is because leveraging real physical responses, even as noise, as existing phenomena, is an important application value in physical reservoir computing. Therefore, parallel-node and -group schemes are achieved on electronic circuits with op-amps and a pair of anti-parallel diodes to provide activation function to nodes. Scheme in Chapter 3 improves the approximation capability to higher-order nonlinear systems under the parallel-node structure concept. Working process diagram is illustrated in Figure 3.2(b), after selecting an input electrode, each output electrode is defined as each individual processing node. The existence of a feedback gain can be considered due to the continuity of ion or proton transport and the dependence on the voltage applied at the previous moment. Benefit from the double-layer capacitance effect caused by drops of solution on the surface of a set of parallel electrodes and the interaction of ionic or protonic migration in the solution between the electrodes, mutual influence existed between nodes. Besides, our multichannel data acquisition system that sequentially collects responses of output electrodes to the same input voltage also generate sort of connection between nodes. That is to say, there is also a feedforward gain between parallel electrodes, although they are originally conceived in parallel and independent way. When we select the next input electrode, a new group of processing nodes with both feedback gain and feedforward gain are generated, which is conceptually consistent with the diagram Fig. 2.3(b). Moreover, analysis to the short-term memory capacity of this

planner electrodes device also shows that the inner working principle of the scheme is similar to that of parallel-group structure in Chapter2.

5.1.4. DESIGN FOR STOCHASTIC COMPUTING

WHILE unconventional computing aims to harness analog signals and create brand-new computing paradigms, digital-based computing is a technology that we find it hard to abandon completely at the moment. An interesting and less common compromise is called stochastic computing (SC), which deals with analog probabilities through digital circuits. SC is an example of a new computing paradigm that is explored from a statistical perspective: The Law of large numbers. The hardware costs of random number generator (RNG) offset the computational advantages of using simple logic gates, hence we think of a low-cost hardware scheme using a POM/SWNT network that can spontaneously generate stochastic number of charges with a controllable probability. In Chapter 4, the feasibility of this scheme using for updating weights of neural network is preliminarily proved by improving cellular automata (CA) model in simulation level. However, as discussed in Section 4.3, there are still steps to go for a practical guidance.

Figure 5.2: Improvements necessary for practical application

First is that the statistic stability should be improved. The black dots distributed in Figure 5.2(a) are the result of real statistics, indicating that there is still irregular deviation from the ideal trajectory in orange color. The second is to maintain this statistical stability so that the iteration in neural network training process is repeatable, like the green trends in Figure 5.2(b). These problems are expected to be solved by fine-tuning the CA model, such as reducing the interference of random behavior and noise in the model and enhancing the control of conductance between gate and drain. This is because the original model is a two-dimensional model that relies on random behavior, and how the real internal structure and behavior changes after imposing a gate voltage has not been fully demonstrated by real experiment. If it can cooperate with the guidance of more experimental results from real device, it will be helpful to the improvement of the model and the stability of the performance. Last, not only do we need the probability of N_C above the threshold to be stable and repeatable after adjustment to V_G, but more importantly, we need it to exist for as long

as possible. In other words, this device should be equipped with long-term memory so that it can play the role of registers in digital circuits. This is difficult to achieve through simulation model because it involves fundamental adjustment from material to structure, thus it is necessary to redesign and fabricate this device from the perspective of memory device.

5.2. EVALUATION OF INFORMATION PROCESSING PERFORMANCE

5.2.1. APPROXIMATION TO DYNAMICAL SYSTEMS

THERE are typically two kinds of neural network (NN) structures: feedforward (FNN) and recurrent (RNN). They are theoretically different in application purpose: RNNs are generally used to approximate dynamical systems because of their dynamic mapping of input data while FNNs are used to approximate nonlinear functions because of their static mapping implementation (Jaeger, 2001b). Reservoir computing (RC), as a derivation of RNN, is regarded as the most suitable to solve arbitrary problems relying on its inherent sensitivity to time, delays, and temporal structure. Hence the most commonly adopted benchmark task for RC is the reproduction ability to a dynamical model called NARMA whose outputs are related not only to past inputs, but also to past outputs. Since the past output states are also included in the construction model of NARMA, the system needs to have a certain short-term memory capacity, and the higher the order of NARMA signal, the greater the memory capacity is needed. Figure 2.8 2.9 and Figure 3.4(c) and (d) are the plots of short-term memory changing along time. It is explicit to find that the memory capacity is not only related to the nonlinear function of nodes, but also to the connection between nodes, that is, the reservoir structure. Compared with parallel-node structure, increasing connections between nodes, whether random or regular, enhances the system's ability to “remember” linear signals. Regular connections improved the system's ability to "remember" higher-order signals better than random connections, though not for longer time. The results of the short-term memory capacity test help to reveal why a system did not perform well in the NARMA test. For example, lower MC_0 and MC_1 of the parallel-node structure led to a bigger prediction error in NARMA2 task, and lower MC^q ($q \geq 2$) is the reason for worse performance in NARMA10.

Beside the mathematically defined model NARMA, there is a real-world dynamical system "pneumatic artificial muscle (PAM)". Their behavior that exhibits hysteresis is a complex dynamic system with past input pressure and present input pressure acting together thus difficult to express by mathematical formula nor to monitor by sensor. This task has a consistent testing result with NARMA2 task, so it is reasonable to assume that any system that solves NARMA2 problems perfectly will have the ability to predict PAM behavior.

5.2.2. PERFORMANCE IN CLASSIFICATION TASK

ANOTHER description of the working principle of RC is that the originally inseparable data becomes linearly separable by projecting the original data into a higher dimensional state space by nonlinear transformation. In order to utilize the nonlinear transformations of reservoir intuitively, we select a classifier that divides the data label according to its neighbors in high dimensional state space, called KNN (shown in Figure 2.11). K is the number of similar samples taken when dividing the data in the high dimensional sample space. When there are enough samples and the distribution of sample categories is good enough, the larger the K value is, the more correct the classification will be. The classification accuracy of several schemes in Chapter 2 is tested under the condition of their own appropriate K values. ESN defined by a spectral radius of 0.99, which performed best in short-term memory capacity and NARMA2 tests, unexpectedly failed to demonstrate its classification ability. Accuracy can be improved with the increase of the number of processing nodes, but it is still not as high as the accuracy of "diodes". When the spectral radius is reduced, the classification performance under the same number of processing nodes (i.e. 50) is improved to an accuracy that close to that of parallel-node structure. But there is no doubt that the parallel-group structure reaches the best classification result. It is undeniable that this parallel-group scheme can not only solve the high-order nonlinear problem ($\Delta E = 0.077$), but also solve the classification problem (accuracy = 83.6%), though the preprocessing and postprocessing of input data also play an important role on the classification accuracy (74%). This can provide different perspectives for future system design and scheme selection, such as reducing the dependence on complex dynamical systems and trying to make controllable adjustment to simple regular systems. Considering the similarity of short-term memory capacity analysis to the parallel-group structure and planar-electrodes-device in Chapter 3, we are convinced that our planar-electrodes-device has the great potential to solve the classification tasks, which deserves a further study.

5.2.3. TRADE-OFF BETWEEN MEMORY AND NONLINEARITY

As introduced in Section 1.3, there is a trade-off between short-term memory and nonlinear process. Based on our experimental results and observations, we hypothesize that achieving accurate approximations of higher-order dynamical systems (e.g. NARMA task) requires a larger memory capacity in the physical UC system. On the other hand, classification task, which involves nonlinear computations on raw data such as dimensionality reduction, high-dimensional projection, convolution, and pooling, necessitate more robust and specific nonlinear process. In addition, results in Chapter 3 shows that richer nonlinear responses also plays a more important role on the approximation to periodic signals. DI water with higher MC_0 and MC_1 performs better in NARMA2 but worse in periodic signal

prediction, because their responses to the sinusoidal input is almost identical. Responses of POM solution on the other hand show rich dynamics so that the linear fitting to periodic signals works better. In short, such known trade-off can be used not only as a tool for system analysis, but also for system application development.

6

Conclusion

6.1. RESPONSE TO RESEARCH TASKS

THIS thesis explores the potential of utilizing specific characteristics of various physical devices for unconventional computing (UC) system, starting from different computing paradigms and theories. Our research works provide new ideas and directions for the physical design of UC systems and fill the unifying theoretical gap in the field of UC. The research aim described in this thesis is to utilize the inherent information processing of physical systems in order to design an unconventional computing paradigm and ultimately develop a simple and powerful physical computing system. The selection and design of the system always focus on the nanoelectronic systems, following the principle of simple operation and low power consumption. To achieve our research tasks, working principles and the inherent characteristics of different physical systems are investigated respectively and suitable computing paradigms are designed based on those investigations. Eventually, the feasibility, computing power and practical application value of each system under its computing paradigm are evaluated based on benchmark tests and realistic task. In this section we conclude the research tasks formulated in section 1.5.

T1 Design information processing systems with devices that are easy to interpret and operate.

In order to design the physical system reasonably, we first trace back to the mathematical expression and necessary condition of echo state network: A matrix representing randomly sparsely interconnected nodes in a network, with a spectral radius of less than 1. A sketchy idea is proposed: mathematically reduce the definition matrix to a diagonal matrix, so that spectral radius is directly from the diagonal values, implying a structure of independently parallel nodes which can be easily implemented. Besides, a parallel-group structure is also expanded based on this idea and implemented them on an electric circuit. Feasibility and practicability of such structures are verified theoretically and experimentally in Chapter 2. On this basis, an electronic device is proposed in Chapter 3: pairs of planar parallel micrometer electrodes facing each other. Intrinsic working principles of such device are similar with that of parallel-group structure because of our multichannel data acquisition system and the double-layer effects on the surface. In Chapter 4, we enhance a 2D network, single-walled carbon nanotubes complexed with polyoxometalate (POM/SWNT), by incorporating an additional gate electrode. Amorphous crystals of POM molecules filled between two gold electrodes on SiO2/Si substrate make internal charge transfer to the drain stochastically but directionally under the action of the source voltage, so that current pulses at stochastical time can be observed at the drain. One additional gate electrode adds an extra level of control and functionality to the previous system, which initially only consisted of a source and drain. Specific operation details on both the experimental level as well as the

simulation cellular automata model level need to be further improved, but the computing potential of the device has been demonstrated in the field of stochastic computing (SC).

T2 State the intrinsic nonlinearities and desirable working principles for different architectures and schemes.

Both schemes in Chapter 2 are implemented on an electric circuit based on one pair of antiparallel diodes, and the conductivity of this circuit works as the activation function in the network. In order to increase the diversity between nodes and to facilitate the control to satisfy the echo state attribute, we artificially introduce external control parameters: input gain can change the slope of $I-V$ response curve of diodes, feedback gain changes the response distribution area along the curve, and feedforward gain in the parallel-group structure make nodes in each group connected in series in one direction. With such a simple, controllable physical operating system, we demonstrate the effectiveness of parallel-node and -group structures, especially the improvement of short term memory capacity and high dimensional projection with a small number of one-way series connections. Electrodes device in Chapter 3 follows the working principle of group parallel structure, but its feedback gain and feedforward gain are unadjustable parameters and determined by the established system, such as the size of electrodes and the distance between them. Besides, the $I-V$ responses are generated essentially by the electrochemical properties of the solution itself, and also affected by electrode spacing, sampling time and other factors. Therefore, compared with the controllable diodes circuit, the dynamical system formed by the planar electrode is more dynamic and complex. Original POM/SWNT device behaves like a capacitor with a variable capacity due to the unique function of POM molecules as "electron sponges". Since the charges stored between the source and drain elecyrodes are constantly changing, so is the current detected on the drain, representing a series of varying peaks on the timeline. In Chapter 4, one gate electrode is added on the top of the SiO2/Si substrate as the control voltage. Due to the influence of the control voltage, the stochastic spontaneous transfer of electrons is enhanced in different degrees, thus realizing the control of density, peak value and duration of current spikes. If more precise control of these three parameters can be achieved through period design and voltage selection, then this device can be applied in the hardware circuit of neural network operation based on stochastic computation.

T3 Quantify the information processing ability in different kinds of tasks.

NARMA2 task refers to training reservoir computing (RC) systems to reproduce a second order nonlinear model defined by Equation 1.6 and the results are represented by the normalized mean square error (NMSE) between the trained fitting curve and the NARMA2 model. Chapter 2 illustrates that performance of NARMA2 is directly related to the degree of connectivity between noes in the reservoir, specifically, NMSE of the reservoir with indepen-

dent nodes is 0.177, and the error decreases by 56.5% when the nodes are connected regularly and sparsely. The sparsity of connections between nodes is represented by spectral radius in standard echo state networks (ESN), which uses hyperbolic tangent function as the activation function. The connection sparsity between nodes is reflected by the spectral radius in ESN, and a larger spectral radius means a stronger connection, which also makes the error of NARMA2 smaller. NMSE of parallel-node structure in Chapter 2 shows an insufficient information processing ability to solve higher order dynamical problems, but it exhibits acceptable accuracy of classifying isolated spoken words. This is mainly due to the pre-processing (Mel-Frequency Cepstral Coefficients extraction and Principal Component Analysis) of the original speech signals, which reduces dependence on temporal features and utilizes the high-dimensional projection ability of the reservoir to process information more effectively. Under such computing paradigm, testing accuracy of relatively weak connections in reservoir are much higher than strong connections. Parallel-group structure in this task achieved the highest classification accuracy of 83.6%. Chapter 3 shows the difference of NMSE in NARMA2 caused by the variation of activation function. Even for the same solution, due to the effect of double-layer capacitance, the extent of $I-V$ response varies at different sampling times. There is no difference in the performance of NARMA2 at a certain range of sampling frequency for the same solution, but sampling frequencies that are too fast or too slow are detrimental to computing power. The most important difference in information processing ability comes from the nonlinear $I-V$ responses of the solution itself. Compared to the complex redox reaction of POM solution, the $I-V$ response of distilled water is much simpler and its performance on NARMA2 task is also better. However, the complex redox characteristics lead to more dynamic processing of input signals, which makes the prediction of periodic signals more accurate. Since the target signal is no longer random numbers, there is no need to normalize the error, mean square error (MSE) can represent the performance of this task well. The lowest MSE of POM solution in periodic signals prediction is 0.002.

T4 Understand the strengths and limitations of different physical schemes for effective utilization in the aspect of computing.

The experimental results show that a few regular connections between nodes can play a very high computational performance both in classification tasks and in the fitting of nonlinear dynamic systems. The advantages of strong random interconnection and disconnectedness between nodes in computing performance are quite different. Strong interconnection between nodes makes the echo states of information in the network last longer, and correspondingly has a higher short-term memory capacity, so it is beneficial to the prediction of time-dependent nonlinear systems. Although the disconnection of nodes shortens the echo states of the information, they still have the ability to project the information into the

high-dimensional state space and can effectively classify the information after proper training. Strong and dense interconnections, on the other hand, make the network less able to extract useful information at the present moment, which is not conducive to classification in high-dimensional space.

6.2. LIMITATIONS OF THIS THESIS

6.2.1. RESEARCH SCOPE

THE the RC systems presented in this thesis are all within the framework of traditional RC, where they work as recurrent neural networks. For example, only a single-layer network structure and linear regression for output training are used, without incorporating multi-layer recursive network designs or other novel training methods (e.g. M. Nakajima et al., 2022). Additionally, the concept of reservoir computing involves treating a large number of randomly sparsely connected nodes as a collective entity - a reservoir, therefore tolerating a few ineffective nodes or redundant nodes. In software implementations of reservoir computing, this approach does not result in any additional computational costs or resource wastage, as there is no distinction in the runtime of the code. However, in hardware implementations, due to the constraints of conversion between digital signal and analog signal, as well as the storage and retrieval of data between the computer and the hardware, significant computational costs and wastage can occur. Indeed, by strategically choosing nodes with specific characteristics and capabilities, it is possible to enhance the overall performance of the system. This approach allows for more efficient allocation of computational resources and can lead to improved processing speed, reduced time consumption, and better utilization of hardware capabilities. Unfortunately, this thesis does not explore the processing performance of individual nodes and their optimal combination. For instance, the selection of parameters that used to define processing nodes in Chapter 2 is determined by the range of parameters rather than specific values, short-term memory capacities in Chapter 2 and Chapter 3 are also calculated from the fit results of all processing nodes without checking one by one. The difficulties in fine control over the hardware and time consumption might have restricted the ability to conduct such analysis, but thoughtful selection of processing nodes can lead to optimization of computational resources, cost savings in hardware, and improved information processing efficiency.

Moreover, since the physical system is treated as a whole in the thesis, quantitative analysis of the various factors and interactions within the system is not conducted. In particular, for micron-scale electronic components like those in Chapter 3, the internal influencing factors are more complex thus cannot be accurately described by several parameters and structural diagrams like those schematically described in Chapter 2. The processing of in-

formation on this physical device, planar electrode group, is not only caused by the $I-V$ characteristic of the solution, but also affected by the double-layer capacitance. The presence of double-layer capacitance may be the reason why such device exhibits "feedforward" gain compared to the diodes circuit in Chapter 2. However, we can only read the output signal under the simultaneous action of all factors, so the influence of each factor is difficult to analyze in detail.

Last, the three-electrode device proposed in Chapter 4 is a supplementary and extended idea based on the existing two-electrode structure. The stochastic transfer process of charges from source to drain is described by a cellular automata (CA) model in which each cell discharges limited number of charges and transfers to its neighbors with the largest gradients. However the probability of charge transfer between cells is also based on the hypothesis of experimental observation results rather than the detailed reproduction of theoretical process. Therefore, this model cannot be used to explain the complex chemical phenomena on real POM/SWNT network. In addition, the establishment of this model is based on on a two-dimensional regular grid with random defects set by random numbers, which does not represent the real network structure. Especially when a base electrode is added, still using the original 2D structure to define the network is somewhat unsatisfactory. More accurate models require the joint efforts from many disciplines such as interface chemistry and computational chemistry, which are unfortunately outside the author's current field of research and knowledge. One final point is that the initial intention behind this proposed design was to construct an AI system entirely implemented on hardware. Unfortunately, with the current state of research, this objective cannot be fully achieved. As discussed in Section 4.3, the current design and functionality of molecular devices are not yet refined enough to meet the requirements for real AI training accuracy. This presents new challenges for the design and fabrication of such devices.

6.2.2. RESOURCES ACCESS

THE research of this thesis focuses on the realization of AI function on fully hardware realization, thus the running time, computing efficiency and accuracy are restricted by the hardware level, explained specifically below.

Running time: Due to the limited hardware resources, the system described in Chapters 2 relies on the concept of 'spatial multiplexing'. This approach involves using the same component repeatedly to fulfill various functions, which inevitably results in high time costs. Another significant running time consumption occurs in the serial peripheral interface (SPI), and each additional external device requires several extra operating cycles to wait for the handshake and transmit the control signal. Chapters 2 and 3 both use the Arduino Mega2560 micro-controller to control, send and read data. An external digital-to-analog

converter (DAC) is required because the Mega2560 board lacks an onboard DAC. In those two chapters, PCF8591, an 8-bit data acquisition DAC, is utilized. Additionally, due to the limited memory capacity of the Mega2560 and the significant data volume of speech signals, an external memory becomes essential for the speech recognition task. To simplify the operation, the author directly utilizes MATLAB, where the memory capacity expands to that on computer, for preprocessing the original audio signals, controlling Arduino, and facilitating data transmission and reception via the I2C bus. However, this approach significantly increases the communication time, exacerbating the existing issue of time overhead resulting from 'hardware savings at the cost of time'.

Computing efficiency: The fact that there is only one controller executing all the computations in a sequential manner limits the efficiency of the calculations, although the design is parallel executable. Therefore, the computational efficiency cannot be fully realized in this case. Furthermore, the excessive communication duration observed speech recognition task highlights how the concatenation of hardware devices intensifies the low computing efficiency, especially in more complex tasks. Therefore, we did not perform the speech recognition task in Chapter 3, as the introduction of additional PCBs including switching circuit and amplifying circuit further complicates the concatenation and control of hardware system. Due to the rapid changes in the current response on the electrodes, in order to ensure consistent reading times on each electrode, the same input signal needs to be re-entered every time we move to the next readout electrode. This also reduces computing efficiency.

Computing accuracy: Neither DAC nor ADC in this thesis is equipped with sufficient resolution. Both the 8-bit DAC and the onboard 10-bit ADC map voltages within the range of 0 to 5 V. The 8-bit DAC has a resolution of less than 20 mV, while the 10-bit ADC has a resolution of less than 5 mV. Obviously, such a low resolution affects computing accuracy, particularly in Chapter 3, where the processing of information are reflected by the response current of molecule or proton on the micron-scale electrodes. The limited resolution results in the neglect of small current changes, thereby diminishing the precision of information processing. Furthermore, the limited magnification adjustment on the PCB, combined with the influence of the electrode distance on the size of the current response, poses challenges in ensuring that the current response on each electrode is fully amplified within a detectable voltage range. Moreover, for such a system with extremely weak and sensitive current variations, accompanying inevitable noise, reading the data only once during the output instead of multiple times within a specific time interval to obtain an average value introduces a certain random component to the output current.

One additional limitation of the work presented in this thesis is the sole reliance on an oscilloscope for observing current and voltage signals. This approach lacks more comprehensive and thorough observation and investigation methods that could potentially pro-

vide a deeper understanding of the phenomena under study at the micro level. A specific example is given to illustrate this. The planar electrode array in Chapter 3 consists of very fine gold wires, making it fragile and susceptible to damage. POM molecules undergoing oxidation-reduction reactions for an extended period can cause surface damage to those electrodes. However, in our study, the investigation into the integrity of the electrodes after each experiment is not conducted. Consequently, the impact of electrode damage on the reaction current and computational performance is not quantified either. Furthermore, although a comparison was made between POM solution and distilled water in terms of computing performance, the lack of suitable equipment and conditions prevented the differentiation of how much of the current was attributed to POM molecules and how much was contributed by protons during the current detection in the POM solution. As a result, this work only discusses the computational functionality brought about by protons as solvents in a macroscopic sense, making it difficult to conduct a quantitative analysis.

6.3. RECOMMENDATIONS

THIS chapter concludes with importance and substantive significance of our work, suggestions to further research on our physical systems and recommendations for future UC system design and development.

6.3.1. SIGNIFICANCE OF THIS THESIS

IN this thesis, we discusses the design and utilization of physical UC systems from two perspectives. Firstly, we start by examining existing computational paradigm (i.e. RC) and simplifying its mathematical definition to a level that is conducive to physical implementation. Based on the simplified definition, physical systems are designed, and the concomitant hardware is constructed. Secondly, we explore the behavior characteristics of existing device (i.e. POM/SWNT) and uses them as a basis to design computational paradigms, thus achieving a comprehensive UC physical system.

In the first aspect, which pertains to the design of a physical system based on the RC paradigm, our approach differs from most single-layer physical reservoir designs. Instead of constraining the physical system to operate within specific configurations, we allow the system to exhibit near-arbitrary behavior, enabling the construction of a large number of sparsely connected nodes or the formation of a mutually coupled cyclic node structure through feedback delays. In our work in Chapter 2, we trace back to the original definition of RC and meticulously analyze the internal workings of reservoirs. In this context, a novel structure of independent and parallel nodes that are mathematically interpretable and physically controllable is proposed and developed firstly. The significance of this re-

search lies in its potential to serve as theoretical guidance and reference for future physical RC system designs. For example, the RC scheme of planar parallel electrode array device in Chapter 3 was inspired and constructed based on this research. By comparing its performance with the various structural schemes presented in Chapter 2, it becomes easy to infer the internal structural characteristics of the new designed physical RC system. This approach avoids treating the reservoir as a "black box" and enhances our understanding of its internal workings. It is worth emphasizing that this work also provides insights from the perspective of device design for the development of UC systems, specifically on how to utilize simple and easily implementable devices. By exploring the potential of such devices, we can develop efficient and practical computing systems that leverage the strengths of simple yet powerful components. Furthermore, its contribution extends beyond the field of physical RC and provides a design approach for other UC paradigms in the realm of physical implementations. Specifically, it offers insights into how to leverage foundational concepts or definitions to provide a starting point for designing physical systems. This approach serves as a breakthrough for the design of physical systems, enabling the exploration of novel UC paradigms and their potential applications.

In the second aspect, a UC system is developed specifically based on the unique properties and behaviors exhibited by the molecular device that are utilized as the foundation for designing the computing paradigm. The integration of molecular devices, as proposed in Chapter 4, with conventional digital circuits to form an integrated circuit system represents a very new and unexplored approach. This opens up pathways for future applications in AI chips based on SC, providing new options for information processing and long-term memory using the characteristics of molecular materials. Indeed, one of the most significant contributions of this work is providing a comprehensive thought process and conceptual methodology for the design and development of UC systems based on specific devices. We believe that this will inspire more researchers to explore the application of their unique devices in the field of UC.

6.3.2. RECOMMENDATIONS FOR FURTHER RESEARCH

THIS work provides several research directions that are worth further exploration. Below, we outline these potential future research paths one by one.

The first recommendation is that the network structure of isolated parallel nodes presented in Chapters 2 can be further explored and improved. It has demonstrated comparable speech classification capabilities to randomly sparse interconnected nodes of Echo State Networks (ESNs) and outperformed deeper interconnected nodes, indicating that sparse interconnectivity among nodes is not necessary for classification tasks. The functioning of these isolated processing nodes is similar to filters in convolutional neural networks

(CNNs), as they both aim to explore and extract hidden features. Furthermore, this discrete structure is easily implementable in hardware, allowing for exploration of various physical systems or devices. Additionally, since these isolated nodes are randomly defined within a certain range without individual selection, there is potential for further improvement in performance or reduction in the number of nodes. We believe that in the future, efficient filter banks can be developed for image, audio or video recognition by analyzing and selecting nodes individually.

In addition, the micron-scale planar electrode set in Chapter 3 allows even deionized water to perform powerful information processing capabilities, which undoubtedly opens up a new direction of exploration, namely, proton-based computing systems. Such computing systems have two advantages: Firstly, protons, as the smallest ions, possess extremely fast transmission speeds, enabling them to transfer data at high speeds and potentially serving edge computing applications. Secondly, in the extremely short response time, only a small amount of electrons pass through the electrodes, avoiding the heat generation and power consumption issues typically associated with traditional CMOS devices due to higher current flow. This presents new requirements and challenges for material selection and structural design of such functional device.

Last but noe least, the potential utilization of polyoxometalate (POM) molecules, also known as the "electronic sponge," for information processing is worth attention and further development. Within the RC computational paradigm, their complex and nonlinear current response to input voltage can greatly enhance the accuracy of predicting periodic signals. We suggest exploring alternative computational paradigms to harness the capabilities of POM molecules in a broader range of computational tasks. As each POM molecule can store a large amount of charge, its 2D network composed of single-walled carbon nanotubes (SWNT/POM) exhibits negative differential resistances (NDRs), leading to periodic or non-periodic current oscillations. The occurrence of current pulses is random, making it suitable for use as a random number generator, with the density of current pulses depending on the magnitude or frequency of applied voltage. The ultimate goal of this research is to significantly reduce the number of transistors and computational complexity in hardware implementation of SC, leading to a chip architecture that can achieve substantial reductions in both. Hence we recommend two avenues for future improvements of this device: Firstly, redesigning the device structure to achieve reproducible control over the pulse density. Secondly, selecting composite materials for the device to enable memory functionality, allowing the density of current pulses to be sustained over a longer period of time.

List of References

Adamatzky, A. (Ed.). (2017). *Advances in unconventional computing*. Springer International Publishing. https://doi.org/10.1007/978-3-319-33924-5

Agmon, N., Bakker, H. J., Campen, R. K., Henchman, R. H., Pohl, P., Roke, S., Thämer, M., & Hassanali, A. (2016). Protons and hydroxide ions in aqueous systems. *Chemical Reviews*, *116*(13), 7642–7672. https://doi.org/10.1021/acs.chemrev.5b00736

Akashi, N., Yamaguchi, T., Tsunegi, S., Taniguchi, T., Nishida, M., Sakurai, R., Wakao, Y., & Nakajima, K. (2020). Input-driven bifurcations and information processing capacity in spintronics reservoirs. *Phys. Rev. Res.*, *2*, 043303. https://doi.org/10.1103/PhysRevResearch.2.043303

Alaghi, A., & Hayes, J. P. (2013). Survey of stochastic computing. *ACM Trans. Embed. Comput. Syst.*, *12*(2s). https://doi.org/10.1145/2465787.2465794

Alomar, M. L., Canals, V., Morro, A., Oliver, A., & Rossello, J. L. (2016). Stochastic hardware implementation of liquid state machines. *2016 International Joint Conference on Neural Networks (IJCNN)*, 1128–1133. https://doi.org/10.1109/IJCNN.2016.7727324

Appeltant, L., Soriano, M. C., Van der Sande, G., Danckaert, J., Massar, S., Dambre, J., Schrauwen, B., Mirasso, C. R., & Fischer, I. (2011). Information processing using a single dynamical node as complex system. *Nature Communications*, *2*(1), 468.

Atiya, A., & Parlos, A. (2000). New results on recurrent network training: Unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks*, *11*(3), 697–709. https://doi.org/10.1109/72.846741

Bhatti, S., Sbiaa, R., Hirohata, A., Ohno, H., Fukami, S., & Piramanayagam, S. (2017). Spintronics based random access memory: A review. *Materials Today*, *20*(9), 530–548. https://doi.org/https://doi.org/10.1016/j.mattod.2017.07.007

Blum, L., Cucker, F., Shub, M., & Smale, S. (1998). *Complexity and real computation*. Springer New York. https://doi.org/10.1007/978-1-4612-0701-6

Bodén, M. (2001). A guide to recurrent neural networks and backpropagation.

Bose, S. K., Lawrence, C. P., Liu, Z., Makarenko, K. S., van Damme, R. M. J., Broersma, H. J., & van der Wiel, W. G. (2015). Evolution of a designless nanoparticle network into reconfigurable boolean logic. *Nature Nanotechnology*, *10*(12), 1048–1052.

Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers, EC-14*(3), 326–334. https://doi.org/10.1109/PGEC.1965.264137

Cramer, B., Stradmann, Y., Schemmel, J., & Zenke, F. (2022). The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems, 33*(7), 2744–2757. https://doi.org/10.1109/TNNLS.2020.3044364

Crljen, Z. Ž., Grigoriev, A., Wendin, G., & Stokbro, K. (2005). Nonlinear conductance in molecular devices: Molecular length dependence. *Phys. Rev. B, 71*, 165316. https://doi.org/10.1103/PhysRevB.71.165316

Cucchi, M., Gruener, C., Petrauskas, L., Steiner, P., Tseng, H., Fischer, A., Penkovsky, B., Matthus, C., Birkholz, P., Kleemann, H., & Leo, K. (2021). Reservoir computing with biocompatible organic electrochemical networks for brain-inspired biosignal classification. *Science Advances, 7*(34), eabh0693.

Dambre, J., Verstraeten, D., Schrauwen, B., & Massar, S. (2012). Information processing capacity of dynamical systems. *Scientific Reports, 2*(1), 514. https://doi.org/10.1038/srep00514

Devaney, R. L. (2021). *An introduction to chaotic dynamical systems*. Chapman; Hall/CRC. https://doi.org/10.1201/9780429280801

Dominey, P. F. (1995). Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biological Cybernetics, 73*(3), 265–274. https://doi.org/10.1007/BF00201428

Dominey, P. F., Hoen, M., Blanc, J.-M., & Lelekov-Boissard, T. (2003). Neurological basis of language and sequential cognition: Evidence from simulation, aphasia, and erp studies [Understanding Language]. *Brain and Language, 86*(2), 207–225. https://doi.org/https://doi.org/10.1016/S0093-934X(02)00529-1

Du, C., Cai, F., Zidan, M. A., Ma, W., Lee, S. H., & Lu, W. D. (2017). Reservoir computing using dynamic memristors for temporal information processing. *Nature Communications, 8*(1), 2204.

Fernandes, D. M., Nunes, M., Carvalho, R. J., Bacsa, R., Mbomekalle, I.-M., Serp, P., De Oliveira, P., & Freire, C. (2015). Biomolecules electrochemical sensing properties of a pmo11v@n-doped few layer graphene nanocomposite. *Inorganics, 3*(2), 178–193.

Fette, G., & Eggert, J. (2005). Short term memory and pattern matching with simple echo state networks. In W. Duch, J. Kacprzyk, E. Oja, & S. Zadrożny (Eds.), *Artificial neural networks: Biological inspirations – icann 2005* (pp. 13–18). Springer Berlin Heidelberg.

Fujii, H., Setiadi, A., Kuwahara, Y., & Akai-Kasaya, M. (2017). Single walled carbon nanotube-based stochastic resonance device with molecular self-noise source [133501]. *Applied Physics Letters*, *111*(13). https://doi.org/10.1063/1.4986812

Fujii, K., & Nakajima, K. (2017). Harnessing disordered-ensemble quantum dynamics for machine learning. *Phys. Rev. Appl.*, *8*, 024030. https://doi.org/10.1103/PhysRevApplied.8.024030

Furuta, T., Fujii, K., Nakajima, K., Tsunegi, S., Kubota, H., Suzuki, Y., & Miwa, S. (2018). Macromagnetic simulation for reservoir computing utilizing spin dynamics in magnetic tunnel junctions. *Phys. Rev. Appl.*, *10*, 034063. https://doi.org/10.1103/PhysRevApplied.10.034063

Gallicchio, C. (2018). Chasing the echo state property. *CoRR*, *abs/1811.10892*. http://arxiv.org/abs/1811.10892

Gallicchio, C., Micheli, A., & Pedrelli, L. (2017). Deep reservoir computing: A critical experimental analysis. *Neurocomputing*, *268*, 87–99.

Gauthier, D. J., Bollt, E., Griffith, A., & Barbosa, W. A. S. (2021). Next generation reservoir computing. *Nature Communications*, *12*(1), 5564.

Goto, K., Nakajima, K., & Notsu, H. (2021). Twin vortex computer in fluid flow. *New Journal of Physics*, *23*(6), 063051. https://doi.org/10.1088/1367-2630/ac024d

Grigoryeva, L., Henriques, J., Larger, L., & Ortega, J.-P. (2016). Nonlinear Memory Capacity of Parallel Time-Delay Reservoir Computers in the Processing of Multidimensional Signals. *Neural Computation*, *28*(7), 1411–1451. https://doi.org/10.1162/NECO_a_00845

Hagiwara, N., Sekizaki, S., Kuwahara, Y., Asai, T., & Akai-Kasaya, M. (2021). Long- and short-term conductance control of artificial polymer wire synapses. *Polymers*, *13*(2).

Hartmanis, J. (1993). Some observations about the nature of computer science. *Proceedings of the 13th Conference on Foundations of Software Technology and Theoretical Computer Science*, 1–12.

Inubushi, M., & Yoshimura, K. (2017). Reservoir computing beyond memory-nonlinearity trade-off. *Scientific Reports*, *7*(1), 10199. https://doi.org/10.1038/s41598-017-10257-6

Inubushi, M., Yoshimura, K., Ikeda, Y., & Nagasawa, Y. (2021). On the characteristics and structures of dynamical systems suitable for reservoir computing. In K. Nakajima & I. Fischer (Eds.). Springer Singapore.

Jaeger, H. (2001a). *The "echo state" approach to analysing and training recurrent neural networks* (GMD Report No. 148). GMD - German National Research Institute for Computer Science. http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf

Jaeger, H. (2001b). *The "echo state" approach to analysing and training recurrent neural networks* (GMD Report No. 148). GMD - German National Research Institute for Computer Science. http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf

Jaeger, H. (2021). Towards a generalized theory comprising digital, neuromorphic and unconventional computing. *Neuromorphic Computing and Engineering*, *1*(1), 012002. https://doi.org/10.1088/2634-4386/abf151

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, *304*(5667), 78–80. https://doi.org/10.1126/science.1091277

Kan, S., Nakajima, K., Asai, T., & Akai-Kasaya, M. (2022). Physical implementation of reservoir computing through electrochemical reaction (adv. sci. 6/2022). *Advanced Science*, *9*(6), 2270036. https://doi.org/10.1002/advs.202270036

Kan, S., Nakajima, K., Takeshima, Y., Asai, T., Kuwahara, Y., & Akai-Kasaya, M. (2021). Simple reservoir computing capitalizing on the nonlinear response of materials: Theory and physical implementations. *Phys. Rev. Appl.*, *15*, 024030. https://doi.org/10.1103/PhysRevApplied.15.024030

Kan, S., Sasaki, Y., Asai, T., & Akai-Kasaya, M. (2021). Applying a molecular device to stochastic computing operation for a hardware ai system design. *Journal of Signal Processing*, *25*(6), 221–225. https://doi.org/10.2299/jsp.25.221

Kaspar, C., Ravoo, B. J., van der Wiel, W. G., Wegner, S. V., & Pernice, W. H. P. (2021). The rise of intelligent matter. *Nature*, *594*(7863), 345–355. https://doi.org/10.1038/s41586-021-03453-y

Kawai, Y., Park, J., & Asada, M. (2019a). A small-world topology enhances the echo state property and signal propagation in reservoir computing. *Neural Networks*, *112*, 15–23. https://doi.org/https://doi.org/10.1016/j.neunet.2019.01.002

Kawai, Y., Park, J., & Asada, M. (2019b). A small-world topology enhances the echo state property and signal propagation in reservoir computing. *Neural Networks*, *112*, 15–23.

Li, B., Najafi, M. H., & Lilja, D. J. (2016). Using stochastic computing to reduce the hardware requirements for a restricted boltzmann machine classifier. *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 36–41. https://doi.org/10.1145/2847263.2847340

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, *14*(11), 2531–2560. https://doi.org/10.1162/089976602760407955

MacLennan, B. J. (n.d.). Bodies — both informed and transformed embodied computation and information processing. In *Information and computation* (pp. 225–253). https://doi.org/10.1142/9789814295482_0009

Malik, Z. K., Hussain, A., & Wu, Q. J. (2017). Multilayered echo state machine: A novel architecture and algorithm. *IEEE Transactions on Cybernetics*, *47*(4), 946–959. https://doi.org/10.1109/TCYB.2016.2533545

Margulies, D., Melman, G., & Shanzer, A. (2006). A molecular full-adder and full-subtractor, an additional step toward a moleculator. *Journal of the American Chemical Society*, *128*(14), 4865–4871. https://doi.org/10.1021/ja058564w
doi: 10.1021/ja058564w

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., Jackson, B. L., Imam, N., Guo, C., Nakamura, Y., Brezzo, B., Vo, I., Esser, S. K., Appuswamy, R., Taba, B., Amir, A., Flickner, M. D., Risk, W. P., Manohar, R., & Modha, D. S. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, *345*(6197), 668–673. https://doi.org/10.1126/science.1254642

Miller, D. A. B. (2010). Are optical transistors the logical next step? *Nature Photonics*, *4*(1), 3–5. https://doi.org/10.1038/nphoton.2009.240

Najafi, M. H., & Salehi, M. E. (2016). A fast fault-tolerant architecture for sauvola local image thresholding algorithm using stochastic computing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *24*(2), 808–812. https://doi.org/10.1109/TVLSI.2015.2415932

Nakajima, K., & Fischer, I. (2021). *Reservoir computing: Theory, physical implementations, and applications*. Springer Nature Singapore. https://books.google.nl/books?id=AQc8EAAAQBAJ

Nakajima, K. (2020). Physical reservoir computing—an introductory perspective. *Japanese Journal of Applied Physics*, *59*(6), 060501. https://doi.org/10.35848/1347-4065/ab8d4f

Nakajima, K., Fujii, K., Negoro, M., Mitarai, K., & Kitagawa, M. (2019). Boosting computational power through spatial multiplexing in quantum reservoir computing. *Phys. Rev. Appl.*, *11*, 034021. https://doi.org/10.1103/PhysRevApplied.11.034021

Nakajima, K., Hauser, H., Li, T., & Pfeifer, R. (2015). Information processing via physical soft body. *Scientific Reports*, *5*(1), 10487.

Nakajima, M., Inoue, K., Tanaka, K., Kuniyoshi, Y., Hashimoto, T., & Nakajima, K. (2022). Physical deep learning with biologically inspired training method: Gradient-free approach for physical hardware. *Nature Communications*, *13*(1), 7847.

Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information: 10th anniversary edition*. Cambridge University Press. https://doi.org/10.1017/CBO9780511976667

Nishijima, S., Otsuka, Y., Ohoyama, H., Kajimoto, K., Araki, K., & Matsumoto, T. (2018). Resonant tunneling via a ru–dye complex using a nanoparticle bridge junction. *Nanotechnology*, *29*(24), 245205.

Pascanu, R., Gulcehre, C., Cho, K., & Bengio, Y. (2014). How to construct deep recurrent neural networks.

Piccinini, G., & Maley, C. (2021). Computation in Physical Systems. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Summer 2021). Metaphysics Research Lab, Stanford University.

Qiao, J., Li, F., Han, H., & Li, W. (2017). Growing echo-state network with multiple subreservoirs. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(2), 391–404. https://doi.org/10.1109/TNNLS.2016.2514275

Sakurai, R., Nishida, M., Sakurai, H., Wakao, Y., Akashi, N., Kuniyoshi, Y., Minami, Y., & Nakajima, K. (2020). Emulating a sensor using soft material dynamics: A reservoir computing approach to pneumatic artificial muscle. *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, 710–717.

Sangwan, V. K., & Hersam, M. C. (2020). Neuromorphic nanoelectronic materials. *Nature Nanotechnology*, *15*(7), 517–528. https://doi.org/10.1038/s41565-020-0647-z

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Shah, S., Wee, J., Song, T., Ceze, L., Strauss, K., Chen, Y.-J., & Reif, J. (2020). Using strand displacing polymerase to program chemical reaction networks. *Journal of the American Chemical Society*, *142*(21), 9587–9593. https://doi.org/10.1021/jacs.0c02240 doi: 10.1021/jacs.0c02240

Sillin, H. O., Aguilera, R., Shieh, H.-H., Avizienis, A. V., Aono, M., Stieg, A. Z., & Gimzewski, J. K. (2013). A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing. *Nanotechnology*, *24*(38), 384004. https://doi.org/10.1088/0957-4484/24/38/384004

Stieg, A. Z., Avizienis, A. V., Sillin, H. O., Martin-Olmos, C., Aono, M., & Gimzewski, J. K. (2012). Emergent criticality in complex turing b-type atomic switch networks. *Advanced Materials*, *24*(2), 286–293. https://doi.org/https://doi.org/10.1002/adma.201103053

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., & Hirose, A. (2019). Recent advances in physical reservoir computing: A review. *Neural Networks*, *115*, 100–123. https://doi.org/https://doi.org/10.1016/j.neunet.2019.03.005

Tanaka, H., Akai-Kasaya, M., TermehYousefi, A., Hong, L., Fu, L., Tamukoh, H., Tanaka, D., Asai, T., & Ogawa, T. (2018). A molecular neuromorphic network device consisting of single-walled carbon nanotubes complexed with polyoxometalate. *Nature Communications*, *9*(1), 2693. https://doi.org/10.1038/s41467-018-04886-2

Tomayko, J. E. (1985). Helmut hoelzer's fully electronic analog computer. *Annals of the History of Computing*, *7*(3), 227–240. https://doi.org/10.1109/MAHC.1985.10025

Torrejon, J., Riou, M., Araujo, F. A., Tsunegi, S., Khalsa, G., Querlioz, D., Bortolotti, P., Cros, V., Yakushiji, K., Fukushima, A., Kubota, H., Yuasa, S., Stiles, M. D., & Grollier, J. (2017). Neuromorphic computing with nanoscale spintronic oscillators. *Nature*, *547*(7664), 428–431.

Toyoizumi, T., & Abbott, L. F. (2011). Beyond the edge of chaos: Amplification and temporal integration by recurrent networks in the chaotic regime. *Phys. Rev. E*, *84*, 051908. https://doi.org/10.1103/PhysRevE.84.051908

Tsunegi, S., Taniguchi, T., Miwa, S., Nakajima, K., Yakushiji, K., Fukushima, A., Yuasa, S., & Kubota, H. (2018). Evaluation of memory capacity of spin torque oscillator for recurrent neural networks. *Japanese Journal of Applied Physics*, *57*(12), 120307.

Tsunegi, S., Taniguchi, T., Nakajima, K., Miwa, S., Yakushiji, K., Fukushima, A., Yuasa, S., & Kubota, H. (2019). Physical reservoir computing based on spin torque oscillator with forced synchronization. *Applied Physics Letters*, *114*(16), 164101. https://doi.org/10.1063/1.5081797

Uchida, A., McAllister, R., & Roy, R. (2004a). Consistency of nonlinear system response to complex drive signals. *Phys. Rev. Lett.*, *93*, 244102. https://doi.org/10.1103/PhysRevLett.93.244102

Uchida, A., McAllister, R., & Roy, R. (2004b). Consistency of nonlinear system response to complex drive signals. *Phys. Rev. Lett.*, *93*, 244102. https://doi.org/10.1103/PhysRevLett.93.244102

van Daalen, M., Jeavons, P., & Shawe-Taylor, J. (1993). A stochastic neural architecture that exploits dynamically reconfigurable fpgas. *[1993] Proceedings IEEE Workshop on FPGAs for Custom Computing Machines*, 202–211. https://doi.org/10.1109/FPGA.1993.279462

Verstraeten, D., Schrauwen, B., D'Haene, M., & Stroobandt, D. (2007). An experimental unification of reservoir computing methods [Echo State Networks and Liquid State Machines]. *Neural Networks*, *20*(3), 391–403. https://doi.org/https://doi.org/10.1016/j.neunet.2007.04.003

Vlachas, P., Pathak, J., Hunt, B., Sapsis, T., Girvan, M., Ott, E., & Koumoutsakos, P. (2020a). Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, *126*, 191–217. https://doi.org/https://doi.org/10.1016/j.neunet.2020.02.016

Vlachas, P., Pathak, J., Hunt, B., Sapsis, T., Girvan, M., Ott, E., & Koumoutsakos, P. (2020b). Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, *126*, 191–217.

Wang, S.-S., & Yang, G.-Y. (2015). Recent advances in polyoxometalate-catalyzed reactions. *Chemical Reviews*, *115*(11), 4893–4962.

Yamanaka, Y., Yaguchi, T., Nakajima, K., & Hauser, H. (2018). Mass-spring damper array as a mechanical medium for computation. In V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, & I. Maglogiannis (Eds.), *Artificial neural networks and machine learning – icann 2018* (pp. 781–794). Springer International Publishing.

Yildiz, I. B., Jaeger, H., & Kiebel, S. J. (2012). Re-visiting the echo state property. *Neural Networks*, *35*, 1–9.

Zhakatayev, A., Kim, K., Choi, K., & Lee, J. (2018). An efficient and accurate stochastic number generator using even-distribution coding. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *37*(12), 3056–3066. https://doi.org/10.1109/TCAD.2018.2789732

Zhang, Y., Qu, P., Ji, Y., Zhang, W., Gao, G., Wang, G., Song, S., Li, G., Chen, W., Zheng, W., Chen, F., Pei, J., Zhao, R., Zhao, M., & Shi, L. (2020). A system hierarchy for brain-inspired computing. *Nature*, *586*(7829), 378–384. https://doi.org/10.1038/s41586-020-2782-y

LIST OF PUBLICATIONS

PEER-REVIEWED JOURNAL PAPERS

1. **S. Kan**, K. Nakajima, T. Asai, M. Akai-Kasaya, Physical Implementation of Reservoir Computing through Electrochemical Reaction, *Advanced Science*, 9, 2104076, 2022. .

2. **S. Kan**, K. Nakajima, Y. Takeshima, T. Asai, Y. Kuwahara, M. Akai-Kasaya, Simple Reservoir Computing Capitalizing on the Nonlinear Response of Materials: Theory and Physical Implementations, *Physical Review Applied*, 15(2), 2021. .

3. **S. Kan**, Y.Sasaki, T. Asai, M. Akai-Kasaya, Applying a Molecular Device to Stochastic Computing Operation for a Hardware AI System Design, *Journal of Signal Processing*, 25(6), 2021. .

4. M. Akai-Kasaya, Y. Takeshima, **S. Kan**, K. Nakajima, T. Oya, T. Asai, Performance of reservoir computing in a random network of single-walled carbon nanotubes complexed with polyoxometalate, *Neuromorphic Computing and Engineering*, 2, 014003(1)-014003(10), 2022.

CONFERENCE PAPERS AND ABSTRACTS

1. **S. Kan**, K. Nakajima, T. Asai, M. Akai-Kasaya, A physical system that enables reservoir computing through electrochemical reactions, *In-Materio Neuromorphic Computing*, Nambu Yoichiro Hall, Osaka University, Japan, 2022.

2. **S. Kan**, Y. Sasaki, T. Asai, M. Akai-Kasaya, Applying a molecular device to stochastic computing operation for a hardware AI system design, *RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing*, Online, Japan, 2021.

3. **S. Kan**, Y. Takeshima, K. Nakajima, T. Asai, M. Akai-Kasaya, An Electrochemical Reaction Reservoir Computing realized by Multiple Data Acquisition System, *The 68th JSAP Spring Meeting*, Online, Japan, 2021.

4. **S. Kan**, K. Nakajima, T. Asai, M. Akai-Kasaya, Reservoir computing properties in spiking neural network models with modular topology, *The 9th RIEC International Symposium on Brain Functions and Brain Computer*, Online, Japan, 2020.

5. **S. Kan**, K. Nakajima, T. Asai, M. Akai-Kasaya, Availability of nonlinear response of materials using in the construction of simple reservoir computing, *2020 International Symposium on Nonlinear Theory and Its Applications*, Online, Japan, 2020.

6. **S. Kan**, K. Nakajima, T. Asai, M. Akai-Kasaya, Availability of nonlinear response of materials using in the construction of simple reservoir computing, *IEICE Technical Committee Submission System*, Online, Japan, 2020.

INVITED TALK

1. **S. Kan**, K. Nakajima, T. Asai, M. Akai-Kasaya, Simple reservoir computing capitalizing on the nonlinear response of materials: theory and physical implementations, *Joint Symposium of JSPS-DST Bilateral Research on Charge- and Spin-Blockade in Ultrathin-Layers of Single Molecule Magnets*, Online, Japan, 2021.

2. N. Hagiwara, **S. Kan**, T, Asai, M. Akai-Kasaya, Construction of a neural network using organic materials and ions, *The 29th International Workshop on Active-Matrix Flat-panel Displays and Devices (AM-FPD)*, Ryukoku University Avanti Kyoto Hall, Online, Japan, 2022.